ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/326960941

DECISION SUPPORT TECHNOLOGY FOR MODULAR SYSTEMS, Technical Report, 2nd ed., in Russian, 351 p., Aug. 2018

Technical Report · August 2018
DOI: 10.13140/RG.2.2.21141.01763

CITATIONS
0

1 author:

Mark Sh. Levin
Russian Academy of Sciences

185 PUBLICATIONS **823** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Scheduling View project

All content following this page was uploaded by Mark Sh. Levin on 10 August 2018.

The user has requested enhancement of the downloaded file.

Mark Sh. Levin (Марк Шмуилович Левин)

DECISION SUPPORT TECHNOLOGY FOR MODULAR SYSTEMS

2nd version

ТЕХНОЛОГИЯ ПОДДЕРЖКИ РЕШЕНИЙ ДЛЯ МОДУЛЬНЫХ СИСТЕМ

2-я версия

Moscow, 2013/2018 (Москва - 2013/2018 г.)
© Mark Sh. Levin/Марк Шмуилович Левин, 2018

Preface to the 2nd Version (2018)

This book version contains additional materials:

1. The basic reference list is extended by some recent author's publications:

(i) Levin M.Sh. On combinatorial models of generations of wireless communication systems. J. of Commun. Technol. Electr., 2018, Vol. 63, No. 6, pp. 667–679.

(ii) Levin M.Sh. An example of combinatorial evolution and forecasting of requirements to communication systems. J. of Commun. Technol. Electr., 2017, Vol. 62, No. 12, pp. 1499–1505.

(iii) Levin M.Sh. Combinatorial framework for planning of geological exploration. Electr. prepr., 14 p., Jan. 22, 2018, http://arxiv.org/abs/1801.07229 [cs.AI]

(iv) Levin M.Sh. Towards combinatorial modeling of wireless technology generations. Electr. prepr., 20 p., Aug. 29, 2017, http://arxiv.org/abs/1708.08996 [cs.NI]

(v) Levin M.Sh. Note on evolution and forecasting of requirements: communication example. Electr. prepr., 8 p., May 22, 2016, http://arxiv.org/abs/1705.07558 [cs.NI]

(vi) Levin M.Sh. Towards integrated glance to restructuring in combinatorial optimization. Electr. prepr., 31 p., Dec. 20, 2015, http://arxiv.org/abs/1512.06427 [cs.AI]

(vii) Левин М.Ш. Пример комбинаторной эволюции и прогнозирования требования к коммуникационных системам. "Информационные процессы", 2017, вып. 17, ном. 2, С. 92-100.

(viii) Левин М.Ш. О комбинаторных моделях поколений беспроводных систем связи. "Информационные процессы", 2017, вып. 17, ном. 3, С. 172-187.

2. The text involves corresponding new references to the above-mentioned publications.

3. New section (13.7) on evolution and forecasting of generations of wireless communication systems is included.

Note, the material of the book (version of 2013) was used as the basic one for the preparation of the author Springer book (2015) (this fact was pointed out in the book):

Levin M.Sh. Modular System Design and Evaluation. Springer, 2015.

Moscow, August 2018

Mark Sh. Levin
Inst. for Inform. Transmission Problems (Kharkevich Institute)
Russian Academy of Sciences

http://www.mslevin.iitp.ru/

Preface to the 1st Version (2013)

In recent years, the significance of modularity and modular systems has been increased in various domains (engineering, information technology, organizational systems, social-economical systems). Modularity is useful basis for many life cycle stages of modular systems, for example: design to obtain a set of system decision alternatives, reusing of system modules, maintenance, efficient improvement and/or reconfiguration of systems.

In this material, decision support technology as a set of typical combinatorial frameworks is suggested for life cycle of modular systems with special hierarchical morphological structure: *1.* building of hierarchical system model, *2.* hierarchical modular design, *3.* evaluation of modular system, *4.* detection of system bottlenecks, *5.* system improvement, *6.* multistage design (i.e., design of system trajectory), *7.* combinatorial modeling of system evolution and forecasting. In addition, combinatorial frameworks for aggregation of modular decisions are described.

The suggested system model "hierarchical morphological structure" consists of the following: (i) hierarchical (tree-like) structure of the examined system, (ii) sets of design alternatives (DAs) for each leaf node of the system model (system component), (iii) ordinal estimates of DAs or interval estimates based on multisets, (iv) ordinal estimates of compatibility between DAs for different system components (the ordinal compatibility estimates are used instead of binary estimates in morphological analysis).

The above-mentioned combinatorial frameworks are based on combinatorial optimization problems (knapsack problem, multiple choice problem, assignment/allocation problems, clique problem, spanning tree problems, multicriteria ranking/sorting, graph coloring, etc.) and involve computing algorithms and man-machine (interactive) procedures.

The following objects can be considered: (1) physical system (e.g., building, electronic device), (2) organizational system (e.g., team structure), (3) software, (4) information system or service, (5) plans (in management, in economics, in medicine, etc.), (6) modular educational courses, (7) standards and protocols (e.g., in communication), (8) requirements.

The first part of the material (chapters 2, 3, 4, 5, 6, 7, 8, 9) contains descriptions of combinatorial frameworks above. Many illustrative numerical examples are described as well. The second part of the material (chapters 10, 11, 12, 13, 14, 15, 16, 17, 18) is targeted to descriptions of applications, for example: modular design of packaged software, design of communication systems, design of communication protocols, planning in biomedicine, evolution of standards for multimedia information transmission, design of telemetry system, modular design of management system for smart homes, design of concrete technology, modeling and improvement of building, design of modular educational courses, combinatorial planning of geological exploration. The described applied examples can be used in future research projects, as typical prototype problems, in education.

The material is based on many author's publications (since 1981, including papers and fragments from two books) and was integrated during the author's work in recent nine years in Inst. for Information Transmission Problems of Russian Academy of Sciences. This text corresponds to the author's course "System Design" [22, 345, 350, 353], which was realized (2004...2008) in Faculty of Radio Engineering and Cybernetics at Moscow Inst. of Physics and Technology (State Univ.). It may be used as a textbook.

Moscow, November 2013

Mark Sh. Levin (Inst. for Inform. Transmission Problems)
http://www.mslevin.iitp.ru/

Предисловие к 2-й версии книги

В данной версии книги добавлены следующие материалы:

1. Список литературы расширен за счет некоторых публикаций автора за последние годы:

(i) Levin M.Sh. On combinatorial models of generations of wireless communication systems. J. of Commun. Technol. Electr., 2018, Vol. 63, No. 6, pp. 667–679.

(ii) Levin M.Sh. An example of combinatorial evolution and forecasting of requirements to communication systems. J. of Commun. Technol. Electr., 2017, Vol. 62, No. 12, pp. 1499–1505.

(iii) Levin M.Sh. Combinatorial framework for planning of geological exploration. Electr. prepr., 14 p., Jan. 22, 2018, http://arxiv.org/abs/1801.07229 [cs.AI]

(iv) Levin M.Sh. Towards combinatorial modeling of wireless technology generations. Electr. prepr., 20 p., Aug. 29, 2017, http://arxiv.org/abs/1708.08996 [cs.NI]

(v) Levin M.Sh. Note on evolution and forecasting of requirements: communication example. Electr. prepr., 8 p., May 22, 2016, http://arxiv.org/abs/1705.07558 [cs.NI]

(vi) Levin M.Sh. Towards integrated glance to restructuring in combinatorial optimization. Electr. prepr., 31 p., Dec. 20, 2015, http://arxiv.org/abs/1512.06427 [cs.AI]

(vii) Левин М.Ш. Пример комбинаторной эволюции и прогнозирования требования к коммуникационных системам. “Информационные процессы”, 2017, вып. 17, ном. 2, С. 92-100.

(viii) Левин М.Ш. О комбинаторных моделях поколений беспроводных систем связи. “Информационные процессы”, 2017, вып. 17, ном. 3, С. 172-187.

2. В текст добавлены соответствующие ссылки на указанные публикации.

3. Включен раздел (13.7) по эволюции и прогнозу поколений беспроводных систем связи.

Следует отметить, материал данной книги (версия 2013 г.) был использован как базовый при подготовке книги автора на английском, изданной в издательстве Springer (2015) (этот факт указан в книге):

Levin M.Sh. Modular System Design and Evaluation. Springer, 2015.

Москва, август 2018 г.

Марк Шмуилович Левин
Институт проблем передачи информации им. А.А. Харкевича
Российская академия наук

http://www.mslevin.iitp.ru/

Предисловие к 1-й версии книги

В последние годы в различных областях (информационные технологии, техника, организационные системы, социально-экономические системы) значительно возрастает значимость модульности и модульных систем. С точки зрения жизненного цикла для модульных систем можно рассматривать аналогичные системные этапы, включая ранние стадии проектирования, проектирование, тестирование, транспортирование, эксплуатацию, техническое обслуживание утилизацию, улучшение/модификация. Модульность является удобной основой практически для всех этапов жизненного цикла модульных систем, и в первую очередь для следующих: проектирование с получением большого числа вариантов системных решений, повторное использования системных модулей, эффективное улучшение и/или реконфигурация систем.

В данном материале предлагаются технология поддержки решений в виде типовых комбинаторных схем (frameworks) для поддержки жизненного цикла модульных систем с иерархической морфологической структурой: (1) построение иерархической модели системы, (2) иерархическое модульное проектирование, (3) оценивание модульной системы, (4) выявление системных узких мест, (5) улучшение системы, (6) многостадийное проектирование (т.е., проектирование траектории системы), (7) комбинаторное моделирование эволюции системы и прогнозирование. Дополнительно описаны вспомогательные комбинаторные схемы агрегации модульных решений.

Предложенная автором модель системы "иерархическая морфологическая структура" (расширение системной модели из морфологического анализа) представляет собой следующее: (i) иерархическая (древовидная) структура исследуемой системы, (ii) множества проектных альтернатив (DAs) для каждой висячей вершины системной модели (компонента), (iii) оценки проектных альтернатив на основе порядковых шкал или на основе специальных интервальных шкал в виде мультимножеств, (iv) порядковые совместимости между проектными альтернативами для различных системных компонентов (вместо бинарной совместимости, которая используется в морфологическом анализе).

Указанные комбинаторые схемы основаны на комбинациях задач/моделей комбинаторной оптимизации (задача о рюкзаке, задача блочного рюкзака, задача о назначении и задача размещения, задачи о клике, задачи покрывающих деревьев, многокритериальное ранжирование, раскраска графов и др.) и включают вычислительные алгоритмы и человеко-машинные процедуры для решения комбинаторных задач.

В качестве объекта исследования могут рассматриваться следующие: (1) физическая система (например, здание, электронная аппаратура), (2) организационная система (например, структура бригады исполнителей), (3) комплексы программ, (4) информационная система или сервис(ы), (5) планы (в управлении, в экономике, в медицине и др.), (6) модульные образовательный курс(ы), (7) стандарты и протоколы (например, протоколы связи, стандарты передачи информации), (8) системы требования.

Первая материала (главы 2, 3, 4, 5, 6, 7, 8, 9) содержит описания комбинаторных схем (т.е., построение иерархических моделей, оценивание систем, комбинаторный синтез, выявление узких мест, улучшение систем, агрегирование модульных решений, много-стадийное проектирования с целью построения траектории системы, моделирование комбинаторной эволюции систем и прогнозирование). Здесь имеются реалистичные иллюстративные числовые примеры, облегчающие понимание.

Существенная часть материала (главы 10, 11, 12, 13, 14, 15, 16, 17, 18) посвящена описанию применения предложенных комбинаторных схем для решения реальных прикладных задач, в частности: модульное проектирование компьютерных систем; проектирование и обновление систем и сетей связи; моделирование и проектирование протоколов передачи информации; планирование в биомедицине; построение и моделирование эволюции стан-

дартов передачи мультимедийной информации; проектирование в телеметрии; модульное проектирование систем управления в умном доме; моделирование, анализ и планирование технологии бетона; модульное моделирование и улучшение зданий; проектирование модульных учебных курсов; комбинаторное формирование планов в геологии и экономике.

Рассмотренные прикладные примеры могут использоваться для следующего: (а) для исследований (например, в модифицированных и/или расширенных версиях), (б) как типовые задачи-прототипы, (в) в учебных курсах.

Материал базируется на многочисленных публикациях автора (начиная с 80-х годов) и был интегрирован в последние годы во время работы автора (с 2004 г.) в Институте проблем передачи информации им. А.А. Харкевича Российской академии наук (ИППИ РАН, http://www.iitp.ru/).

Текст соответствуют факультетскому курсу автора "Проектирование систем" [22, 345, 350, 353] (http://www.mslevin.iitp.ru/SYSD.HTM), прочитанному на Факультете радиотехники и кибернетики Московского физико-технического института (государственного университета) в 2004-2008 гг.., и может быть использован, по мнению автора, как учебное пособие.

Москва, ноябрь 2013 г.

Марк Шмуилович Левин (ИППИ РАН)
http://www.mslevin.iitp.ru/

Оглавление

1 Введение **13**
1.1 О системном проектировании . . . 13
1.2 Модульность и иерархические модели . . . 15
1.3 Типовые комбинаторные схемы . . . 17
1.4 Основные системные приложения . . . 19
1.5 Выводы к главе . . . 19

2 Схемы построения иерархических моделей **21**
2.1 Предварительные сведения . . . 21
2.2 Экспертная процедура 'сверху-вниз' . . . 22
2.3 Иерархическая кластеризация . . . 23
2.4 Об использовании онтологии . . . 25
2.5 Покрывающие деревья . . . 25
2.6 Оптимальные организационные иерархии . . . 28
2.7 Много-уровневые структуры . . . 29
2.7.1 Много-уровневой подход . . . 29
2.7.2 Типовые иерархические уровни в системах связи . . . 30
2.7.3 Уровневая k-связная сеть . . . 31
2.7.4 О задаче проектирования иерархической сети . . . 34
2.8 Морфологическая иерархия . . . 35
2.9 Схема для много-уровневой структуры . . . 35
2.10 Выводы к главе . . . 36

3 Схемы оценивания систем **37**
3.1 Предварительные сведения . . . 37
3.2 Рассматриваемые типы шкал . . . 39
3.2.1 Базовые шкалы . . . 40
3.2.2 Системная шкала на основе порядковых оценок компонентов . . . 40
3.2.3 Специальная шкала для системных компонентов . . . 41
3.2.4 Интервальные оценки в виде мультимножеств . . . 42
3.3 Трансформация шкал . . . 44
3.4 Интеграция шкал и качество систем . . . 47
3.5 Числовые примеры . . . 50
3.6 Выводы к главе . . . 52

4 Методы комбинаторного синтеза **53**
4.1 Предварительные сведения . . . 53
4.2 Описание основных подходов . . . 55
4.2.1 Морфологический анализ . . . 55
4.2.2 Метод близости к идеальной точке . . . 56
4.2.3 Морфологический метод с выбором по Парето . . . 56

4.2.4 О применении линейного программирования . . . 57
4.2.5 Задача блочного рюкзака . . . 57
4.2.6 Задачи назначения/размещения . . . 58
4.3 Иерархическое морфологическое проектирование . . . 59
4.4 Комбинаторный синтез с интервальными оценками . . . 62
4.5 Выводы к главе . . . 64

5 Схемы выявления системных узких мест 65
5.1 Предварительные сведения . . . 65
5.2 Традиционные методы управления качеством . . . 67
5.3 Узкие места при морфологическом проектировании . . . 70
5.4 Критические элементы в многоуровневых структурах . . . 73
5.5 Предсказательное выделение системных узких мест . . . 75
5.5.1 Предсказательное выделение компонентов . . . 75
5.5.2 Предсказательное выделение связанных компонентов . . . 76
5.6 Интеграция данных на основе клики над потоками графов . . . 77
5.6.1 Предварительное описание . . . 77
5.6.2 Основная задача . . . 79
5.6.3 Задачи над потоками графов . . . 81
5.6.4 Пример . . . 82
5.6.5 Обсуждение . . . 83
5.7 Выводы к главе . . . 84

6 Схемы улучшения систем 85
6.1 Предварительные сведения . . . 85
6.2 Типовые ситуации улучшения системы . . . 86
6.2.1 Базовые ситуации улучшения систем . . . 86
6.2.2 Основные ситуации улучшения и комбинаторные задачи . . . 88
6.2.3 Дополнительные ситуации улучшения систем . . . 88
6.3 Подходы к оценке и улучшению систем . . . 89
6.3.1 Качество системы и улучшение . . . 89
6.3.2 О реоптимизации . . . 92
6.4 Улучшение по системным компонентам . . . 94
6.4.1 Базовая схема . . . 94
6.4.2 Задача перекраски вершин графа . . . 95
6.5 Модификация структуры системы . . . 97
6.5.1 Модификация дерева посредством склеивания вершин . . . 97
6.5.2 Задача назначения 'горячих' связей . . . 102
6.5.3 Трансформация дерева в дерево Штейнера . . . 104
6.5.4 Задача пополнения графа . . . 106
6.5.5 Трансформация морфологической структуры . . . 106
6.5.6 Трансформация уровневой структуры . . . 107
6.6 Выводы к главе . . . 107

7 Схемы агрегирования модульных решений 109
7.1 Предварительные сведения . . . 109
7.2 Вспомогательные задачи и стратегии агрегирования . . . 111
7.2.1 Базовые вспомогательные задачи . . . 111
7.2.2 Построение “системного ядра” . . . 113
7.2.3 Стратегии агрегирования . . . 113
7.3 Рассматриваемые структуры . . . 115

7.3.1 Множества . . . 115
7.3.2 Ранжировки (слоистые множества) . . . 116
7.3.3 Морфологические множества . . . 116
7.3.4 Деревья и морфологические структуры . . . 118
7.4 Предварительный иллюстративный пример . . . 119
7.5 Метрики и близости . . . 122
7.5.1 Метрики/близости для множеств . . . 122
7.5.2 Близость для последовательностей . . . 123
7.5.3 Близость для ранжировок . . . 123
7.5.4 Близость для деревьев . . . 126
7.5.5 Близость для морфологических структур . . . 128
7.6 Задачи построения медианы, задачи агрегации . . . 130
7.6.1 Множества . . . 130
7.6.2 Ранжировки . . . 131
7.6.3 Деревья . . . 133
7.6.4 Морфологические структуры . . . 133
7.7 Продолжение иллюстративного примера . . . 134
7.8 Выводы к главе . . . 136

8 Схемы много-стадийного проектирования систем 137
8.1 Предварительные сведения . . . 137
8.2 Общие схемы . . . 138
8.3 Иллюстративные примеры . . . 142
8.3.1 Четырех-стадийная траектория start-up группы . . . 142
8.3.2 Древовидная траектория медицинского лечения . . . 144
8.3.3 Упрощенный пример для орграфа . . . 147
8.3.4 О задачах для нескольких областей . . . 148
8.4 Выводы к главе . . . 149

9 Схемы комбинаторной эволюции систем 151
9.1 Предварительные сведения . . . 151
9.2 Комбинаторная эволюция . . . 153
9.3 Операции изменений . . . 154
9.4 Пример для модульного курса . . . 154
9.4.1 Иерархическая морфологическая модель . . . 154
9.4.2 Постановка задачи, проектирование . . . 156
9.4.3 Анализ и улучшение решений . . . 158
9.4.4 Агрегация решений . . . 159
9.5 Выводы к главе . . . 160

10 Приложения в программных системах 161
10.1 Модульный программный пакет . . . 161
10.2 Модульный пользовательский интерфейс . . . 167
10.3 Эволюция системы СППР КОМБИ . . . 173
10.4 Последовательно-параллельные стратегии . . . 174
10.5 Выводы к главе . . . 178

11 Морфологическая схема для задачи размещения 179
11.1 Описание задачи . . . 179
11.2 Составное решение . . . 182
11.3 Выводы к главе . . . 186

12 Приложения при управлении в умном доме 187
12.1 Предварительные сведения . . . 187
12.2 Примеры для системы управления умным домом . . . 189
12.2.1 Иерархическая модель . . . 189
12.2.2 Составные решения . . . 191
12.2.3 Анализ и улучшение узких мест . . . 191
12.3 Пример для системы автоматизации здания . . . 191
12.3.1 Структура системы и проектирование . . . 191
12.3.2 Расширение системы . . . 192
12.3.3 Улучшение частей системы . . . 193
12.4 Выводы к главе . . . 194

13 Приложения в системах связи 195
13.1 Задача 'последняя миля' . . . 196
13.2 Модульное проектирование GSM сети . . . 198
13.3 Подключение пользователей к точкам доступа . . . 200
13.3.1 Предварительные замечания . . . 200
13.3.2 Постановка задачи . . . 201
13.3.3 Схема решения . . . 204
13.3.4 Численный пример . . . 205
13.3.5 Обсуждение . . . 206
13.4 Передача мультимедийной информации . . . 210
13.4.1 Стандарты . . . 210
13.4.2 Изменения стандартов . . . 214
13.4.3 Численные примеры . . . 221
13.4.4 Обсуждение . . . 223
13.5 Модульность в протоколах и стандартах . . . 223
13.5.1 Синтез коммуникационного протокола . . . 225
13.5.2 Композиция стандарта для мультимедийной информации . . . 228
13.5.3 Обсуждение . . . 230
13.6 Примеры улучшения/расширения сетей . . . 230
13.6.1 Улучшение сети . . . 230
13.6.2 Расширение сети . . . 231
13.6.3 Обсуждение . . . 233
13.7 Эволюция систем связи . . . 234
13.7.1 Поколения беспроводных систем связи . . . 234
13.7.2 Иерархическая структура системы связи . . . 236
13.7.3 Пример улучшения (прогнозирования) . . . 237
13.7.4 Обсуждение . . . 239
13.8 Выводы к главе . . . 239

14 Приложения в строительстве 241
14.1 Технология бетона . . . 241
14.2 Оценивание и улучшение здания . . . 244
14.3 Выводы к главе . . . 248

15 Приложения в биомедицине 249
15.1 План лечения . . . 249
15.1.1 Структура плана, альтернативы, оценки . . . 249
15.1.2 Совместимость и составные решения . . . 251
15.1.3 Улучшение, много-стадийный план . . . 253

15.2 Технология иммунологического анализа . . . 253
15.2.1 Структура и альтернативы . . . 253
15.2.2 Составные решения для стадий . . . 257
15.2.3 Структура системы и системные решения . . . 257
15.3 Выводы к главе . . . 259

16 Приложения в телеметрии, сенсорных системах 261
16.1 Телеметрическая система . . . 261
16.2 Беспроводной сенсор . . . 266
16.3 Выводы к главе . . . 271

17 Приложения в менеджменте 273
17.1 Планирование исследования в геологии . . . 273
17.1.1 Исследование месторождений . . . 273
17.1.2 Агрегирование решений . . . 279
17.1.3 Использование интервальных оценок . . . 280
17.2 Стратегии политического маркетинга . . . 281
17.2.1 Предварительные сведения . . . 281
17.2.2 Составная схема решения . . . 282
17.2.3 Пример . . . 282
17.3 Проектирование бригады . . . 286
17.4 Композиция продукта при электронной торговле . . . 287
17.5 Комбинаторное формирование инвестиций . . . 289
17.6 Выводы к главе . . . 291

18 Приложения в преподавании 293
18.1 Предварительные сведения . . . 293
18.2 Учебная среда и проектные задачи . . . 294
18.3 Проектирование учебного курса . . . 295
18.4 Агрегированный курс по проектированию . . . 298
18.5 План студента . . . 306
18.6 Выводы к главе . . . 313

19 Заключение 315

Глава 1

Введение

В данной главе [1] содержится следующее: краткий обзор современных подходов к системному проектированию, краткие сведения об актуальности модульных иерархических систем, краткое описание семи типовых комбинаторных инженерных схем для модульных систем, предложенных автором, список основных приложений.

1.1 О системном проектировании

Среди основных походов к системному проектированию можно указать следующие:

I. Общие подходы:

1.1 иерархическое проектирование систем и декомпозиционные подходы [112, 245, 314, 309];

1.2 модульное проектирование [72, 112, 259, 461];

1.3 проектирование на основе матрицы проектной структуры (design structure matrix) [110, 534];

1.4 проектирование на основе многокритериального принятия решений [65, 112, 164, 313, 314, 434, 510];

1.5 морфологический анализ [64, 83, 278, 486, 598];

1.6 объектно-ориентированное проектирование [96];

1.7 проектирование на основе компонентов [248, 323, 453, 461].

II.Формальные подходы к системному проектированию:

2.1 аксиоматическое проектирование [537];

2.2 формальные методы проектирования [104]:

III. Оптимизация:

3.1 многодисциплинарная оптимизация [49];

3.2 методы смешанного целочисленного нелинейного программирования [191, 224];

3.3 различные методы глобальной оптимизации [192, 193];

3.4 нелинейная многоцелевая оптимизация [426, 533];

3.5 многоцелевая эволюционная оптимизация, включающая генетические алгоритмы [138, 446].

IV. Подходы на основе методов искусственного интеллекта:

4.1 специальные экспертные системы и системы на основе баз знаний [231, 423, 540];

[1] Материал главы частично основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. chapter 1.
(ii) Levin M.Sh. Composite Systems Decisions. Spinger, 2006. chapter 1.
(iii) M.Sh. Levin, Note on combinatorial engineering frameworks for hierarchical modular systems. Electr. prepr., 11 p., Mar. 29, 2013. http://arxiv.org/abs/1304.0030 [math.OC]

4.2 проектирование на основе грамматических описаний систем, включая подходы к проектированию составных систем (composable systems) [208, 514];

4.3 методы проектирования на основе многоагентных подходов [116];

4.4 методы проектирования на основе задач выполнимости (constraint satisfaction problems) [497, 532]).

В России накоплен уникальный опыт по проектированию сложных систем. Можно привести примеры научных направлений в области системного проектирования:

(1) декомпозиционное проектирование сложных систем (акад. РАН П.С. Краснощеков и др.; ВЦ РАН) [14];

(2) теория решения изобретательских задач Г. Альтшуллера (ТРИЗ) [52];

(3) морфологическое проектирование систем (угольных шахт, химических технологий) с использованием многокритериального выбора (акад. РАН С.В. Емельянов, д.т.н. О.А. Сергеев, и др.; ИПУ РАН, ИСА РАН и ИМАШ РАН) [6, 10];

(4) проектирование систем на основе морфологического и многокритериального анализа, близости решений к идеальным композициям (к.т.н., д.п.н. В.Н. Якимец и др.; ИСА РАН) [9];

(5) аппроксимационно-комбинаторный метод к проектированию развивающихся систем (д.т.н. В.Р. Хачатуров, ВЦ РАН) [41];

(6) распределенные системы поддержки процессов проектирования и принятия решений (д.т.н. Э.А. Трахтенгерц, ИПУ РАН) [40];

(7) интерактивные проектные подходы к синтезу структур (д.т.н. Е.И. Артамонов и др., ИПУ РАН) [3];

(8) системная среда САПР для проектирования СБИС (акад. РАН А.Л. Стемпковский, Институт САПР РАН) [39];

(9) метод исследования пространства параметров (д.ф.-м.н. И.М. Соболь, ИПМ РАН и д.т.н. Р.Б. Статников, ИМАШ РАН & Naval Postgraduate School - USA) [38];

(10) декомпозиционные подходы и иерархическая оптимизация (д.ф.-м.н. В.И. Цурков, ВЦ РАН) [42];

(11) моделирование иерархических систем (член.-корр. РАН Ю.С. Попков, ИСА РАН) [37];

(12) математические методы виртуального проектирования технологий (акад. РАН И.М. Макаров, д.т.н. В.З. Рахманкулов, к.ф-м.н. А.А. Ахрем, ИСА РАН) [33];

(13) методы конфигурации и реконфигурации систем на основе искусственного интеллекта (в частности, constraint satisfaction problem, fuzzy sets, concurrent engineering, онтологические подходы) (д.т.н. А.В. Смирнов, СПИИ РАН) [521, 522, 523];

(14) композиционные методы проектирования информационных систем (д.ф.-м.н. Л.А. Калиниченко и др., ИПИ РАН) [11, 315];

(15) методы поискового конструирования [12];

(16) методы синтеза иерархических систем управления (д.т.н. В.Н. Цыгичко и др.; ИСА РАН) [43];

(17) оптимизационные методы проектирования структур организаций (член-корр. РАН, д.т.н. Д.А. Новиков, д.ф.-м.н. А.А Воронин, к.ф.-м.н. С.П. Мишин, к.т.н. М.В. Губко; ИПУ РАН и Волгоградский гос. университет) [5, 227, 433, 568];

(18) методы автоматизированного проектирования систем, включая генетические методы (д.т.н. И.П. Норенков, МГТУ им. Н.Э. Баумана) [35].

Отметим некоторые современные ключевые проблемы построения процессов проектирования [22, 23, 336, 339, 341, 344]:

1. Общая организация процесса проектирования, включая параллелизацию (распараллеливание) процесса, возможность включения в процессы проектирования экспертов из прикладных областей знаний, координацию различных профессиональных групп.

2. Комбинаторная сложность “пространств” проектных решений.

3. Алгоритмическая и вычислительная сложность процессов решения возникающих задач, а также, “эффективность и удобство” включения в технологию решения специалистов и экспертов.

4. Организация контрольных точек в процессе проектирования для оценивания предварительных решений, для перепроектирования/улучшений.

5. Организация проектной информации (“пространств” проектных решений).

6. Реализация дополнительных проектных задач: поиск узких мест, оценивание, перепроектирование/ улучшение и др.

1.2 Модульность и иерархические модели

В последние годы значительно возросла значимость модульных (многокомпонентных) систем [72, 136, 157, 209, 259, 282, 336, 344, 358, 371, 561, 572].

Модульность систем является базой для ускоренной разработки систем, получения большого числа вариантов системных решений, повторного использования системных модулей, эффективного улучшения и/или реконфигурации систем [209, 226, 237, 259, 336, 344, 557, 558, 561, 562]. С другой стороны, иерархические подходы являются очень эффективным средством для моделирования, анализа и проектирования различных систем.

В данной работе предложены типовые комбинаторные инженерные схемы (как композиции взаимосвязанных комбинаторых задач и соответствующих им алгоритмов и процедур решения) для модульных систем с иерархической моделью (морфологического типа) (Рис. 1.1).

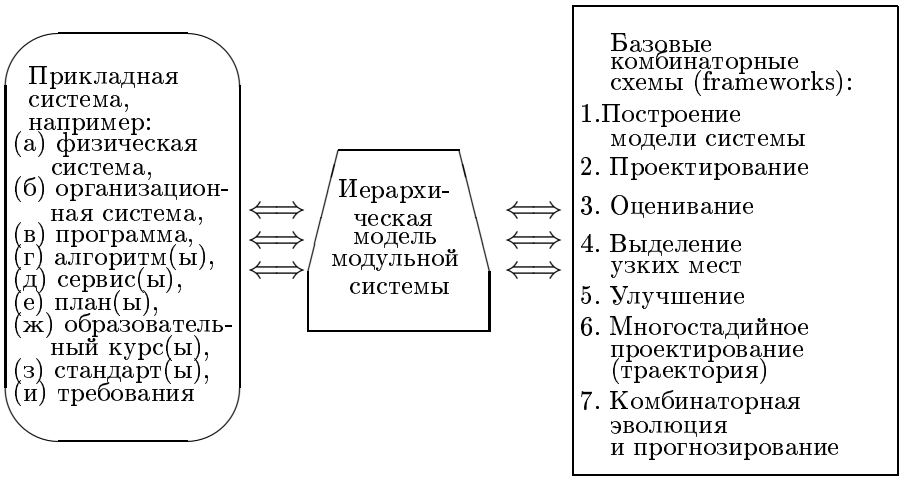


Рис. 1.1. Система - иерархическая модель - комбинаторная схема

Следует отметить особую роль иерархий. Иерархические модели играют центральную роль в системных науках, технике, информационных технологиях [207, 239, 246, 294, 303, 439]. Можно указать некоторые основные типы иерархий [207, 239, 303, 365]: (1) различные типы деревьев (Рис. 1.2, Рис. 1.3, Рис. 1.4) [200, 207, 303]; (2) органическая иерархия (т.е., с связями между вершинами-детьми, Рис. 1.5) [140]; (3) иерархия (как дерево с дополнительными ребрами, Рис. 1.6) [328]; (4) “морфологическая иерархия” (иерархичесая морфологичекая модель) [336, 344, 357, 358] (Рис. 1.7); (5) многоуровневая структура (Рис. 1.8) [44, 84, 418, 443, 458, 545, 546]; (6) специальная двух-уровневая модель: сеть верхнего уровня - морфологические структуры для каждой вершины сети (Рис. 1.9).

Можно также указать ряд специальных приложений иерархий: (1) иерархические схемы данных и знаний [102, 303, 326, 328, 484]; (2) информационные концептуальные пространства [127]; (3) организационные иерархии [73, 227, 433, 568]); (4) многоуровневые

сложные системы [425]; (5) эволюционные деревья [54, 432]; (6) онтологии [146, 256, 454, 563]; (7) специальные иерархические структуры программ (statecharts) [246]; (8) деревья решений [46, 212, 218, 219]); (9) иерархии критериев в принятии решений, например, в методе аналитических иерархий [496]; (10) иерархические сети доступа [216].

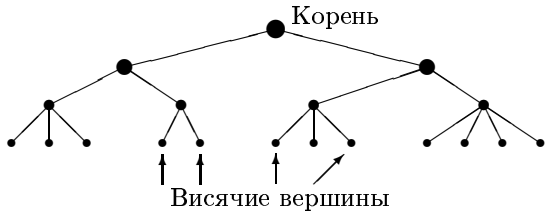


Рис. 1.2. Дерево

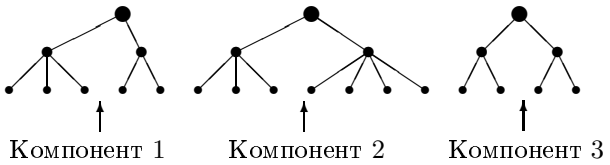


Рис. 1.3. Лес

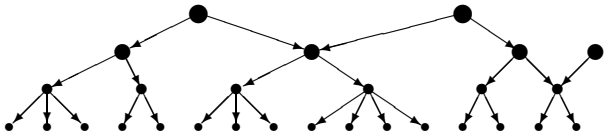


Рис. 1.4. Поли-дерево

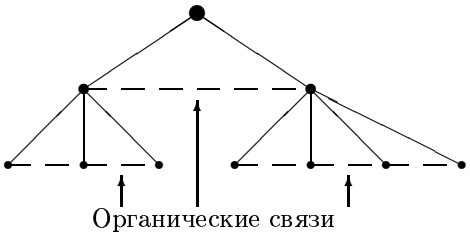


Рис. 1.5. Органическая иерархия

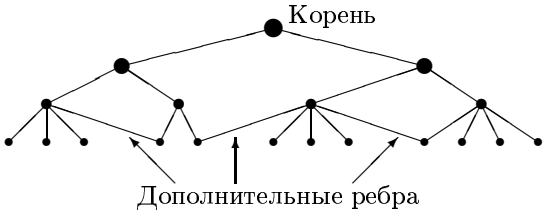


Рис. 1.6. Иерархия

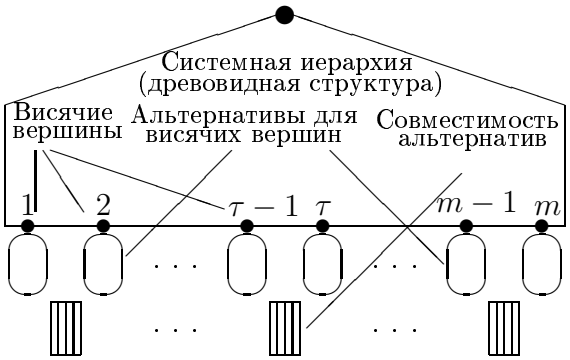


Рис. 1.7. Системная морфология

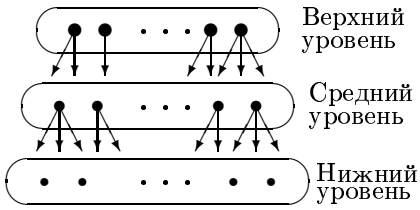


Рис. 1.8. Многоуровневая структура

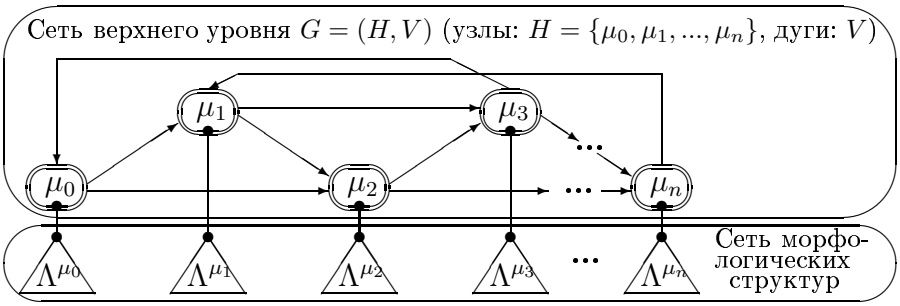


Рис. 1.9. Двух-уровневая модель: сеть с морфологиями

В данной работе исследование основывается на использовании, в качестве системной модели, специальной морфологической структуры (Рис. 1.7), предложенной автором в конце 80-х - начале 90-х годов [16, 17, 329, 330, 332, 336, 344, 365]. Дополнительно используются древовидные структуры, многоуровневые структуры и специальная новая двухуровневая структура, представленная на Рис. 1.9 (глава 8).

1.3 Типовые комбинаторные схемы

Рассматриваемые типовые комбинаторные схемы (frameworks) представляют собой комбинации задач/моделей для поддержки различных стадий жизненного цикла модульных систем (Рис. 1.10).

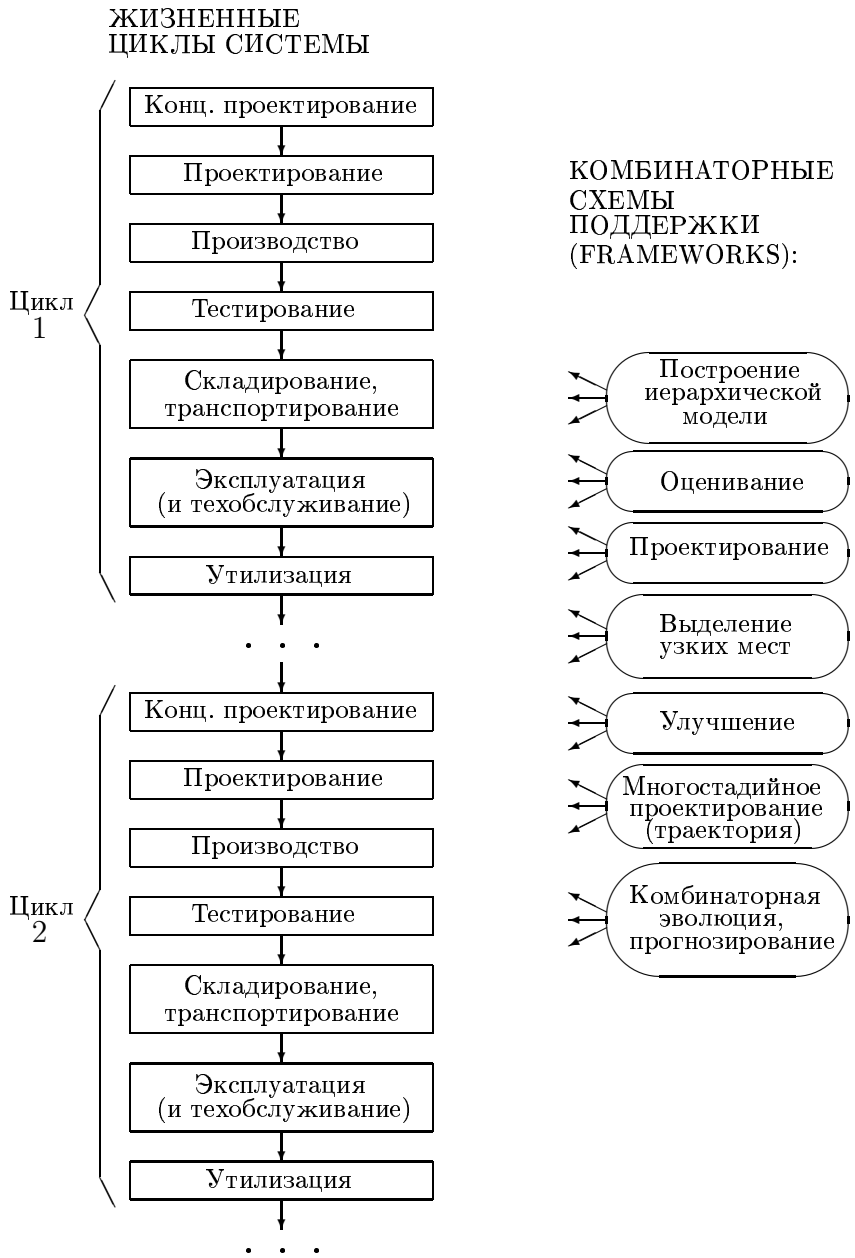


Рис. 1.10. Типовые комбинаторные схемы поддержки

Список комбинаторных схем включает следующее [344, 346]:

1. Построение иерархической модели системы (T_1).

2. Иерархическое модульное проектирование (T_2).

3. Оценивание модульной системы (T_3).
4. Выявление системных узких мест (T_4).
5. Улучшение системы (T_5).
6. Многостадийное проектирование (т.е., проектирование траектории системы) (T_6).
7. Комбинаторное моделирование эволюции системы и прогнозирование (T_7).

Дополнительно следует также указать важную вспомогательную комбинаторную схему агрегации модульных решений.

Рассматриваемые комбинаторые схемы могут быть применены к следующим системных объектам: системы, планы, требования к системам, стандарты (e.g., [336, 344, 357]). Общая схема исследуемой области представлена на Рис. 1.11.

Рис. 1.11. Общая схема исследуемого направления

Комбинаторные схемы базируются на задачах комбинаторной оптимизации и можно рассматривать следующую четырехуровневую архитектуру [355]:

1. Базовые задачи комбинаторной оптмизации (задача о рюкзаке, задача блочного рюкзака, кластеризация, ранжирование, задача о назначании/размещении, покрывающие деревья, раскраска графов, задачи маршрутизации, клика и др.).

2. Составные комбинаторные задачи, включающие несколько базовых задач, в частности, многокритериальные версии указанных базовых задач.

3. Типовые инженерные комбинаторные схемы (frameworks), указанные выше (т.е., проектирование системы - комбинаторный синтез, построение модели, оценивание, улучшение, выявление узких мест, моделирование комбинаторной эволюции).

4. Системные приложения.

Следует заметить, что в приложениях типовые комбинаторные схемы часто используются совместно в рамках общей прикладной проектной схемы, например:

Проектирование иерархической модели прикладной системы, проектирование системы, оценивание системных решений, выявление узких мест в полученных системных решениях, улучшение системных решений.

1.4 Основные системные приложения

Таблица 1.1 содержит список основных опубликованных системных приложений, при этом указаны комбинации примененных комбинаторных схем.

Таблица 1.1. Приложения и комбинаторные схемы

Приложение	Примененные схемы							Ссылки на публикации автора
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	
1.Электронная торговля	⋆	⋆	⋆	⋆	⋆	⋆		[348, 359]
2.Веб-система	⋆	⋆	⋆	⋆	⋆	⋆		[354, 357, 391]
3.Стратегия многокритериального ранжирования	⋆	⋆						[336, 364]
4.СППР КОМБИ	⋆						⋆	[329, 336]
5.Модульный программный комплекс	⋆	⋆	⋆	⋆	⋆			[343, 344]
6.Ноутбук	⋆	⋆			⋆			[357]
7.Региональная сеть связи	⋆		⋆					[400]
8.GSM сеть связи	⋆	⋆						[358, 388]
9.Телеметрическая система	⋆	⋆	⋆	⋆	⋆			[361, 387]
10.Система безопасности	⋆	⋆	⋆	⋆	⋆			[357, 392]
11.Радио сенсор	⋆	⋆	⋆	⋆	⋆			[357, 394, 402]
12.Тестирование	⋆	⋆						[386]
13.Протокол связи	⋆	⋆	⋆	⋆	⋆	⋆	⋆	[370, 393]
14.Человеко-машинный интерфейс	⋆	⋆						[330, 336, 341]
15.Здание	⋆	⋆	⋆	⋆				[344, 385]
16.Управление в умном доме	⋆	⋆	⋆	⋆	⋆			[369, 398]
17.Инвестиции	⋆	⋆						[336, 357]
18.План маркетинга	⋆							[23, 32]
19.Вибрационный конвейер	⋆	⋆						[336, 381]
20.Геологическое планирование	⋆	⋆						[30, 336]
21.Технология бетона	⋆	⋆						[344, 382]
22.Планирование лечения	⋆	⋆						[344, 383]
23.Учебные курсы	⋆	⋆	⋆	⋆	⋆	⋆	⋆	[336, 344, 357, 366]
24.Стандарты для мультимедиа	⋆	⋆	⋆	⋆	⋆	⋆	⋆	[368, 370, 390]
25.Иммунологический анализ	⋆	⋆						[344, 384]

1.5 Выводы к главе

В данной вводной главе приведено краткое описание и обоснование схемы исследования:

Приложение $\Longleftrightarrow$ *Морфологическая модель* $\Longleftrightarrow$ *Комбинаторные схемы*

В качестве иерархической морфологической модели системы используется, в основном, предложенная автором расширенная модель из морфологического анализа: (i) иерархическая (древовидная) структура системы, (ii) множества проектных альтернатив для каждой висячей вершины системной модели, (iii) порядковые оценки проектных альтератив DAs (или оценки на основе специальных интервальных шкал в виде мультимножеств), (iv) порядковые совместимости между проектными альтернативами для различных системных компонентов.

Предложенные комбинаторые схемы основаны на комбинациях задач/моделей комбинаторной оптимизации (задача о рюкзаке, задача блочного рюкзака, задача о назначении/размещении, задачи о клике, задачи покрывающих деревьев, многокритериальное ранжирование, раскраска графов и др.) и включают вычислительные алгоритмы и человеко-машинные процедуры для решения комбинаторных задач.

В качестве прикладного объекта исследования могут рассматриваться следующие: (1) физическая система (например, здание, электронная аппаратура), (2) организационная система (например, структура бригады исполнителей), (3) комплексы программ, (4) информационная система или сервис(ы), (5) планы (в управлении, в экономике, в медицине и др.), (6) модульные образовательный курс(ы), (7) стандарты и протоколы (протоколы связи, стандарты передачи информации), (8) системы требования.

В последующих главах представлены описания семи типовых комбинаторных схем и дополнительной комбинаторной схемы (агрегация модульных решений), а, также, примеры применения указанных схем в различных приложениях (компьютерные системы, системы связи, телеметрия, организационные системы, строительство, биомедицина, экономика, геология, образование и др.).

Глава 2

Схемы построения иерархических моделей

В данной главе [1] содержится описание основных подходов к построению иерархических моделей систем (т.е., деревья, иерархии, уровневые структуры).

2.1 Предварительные сведения

Задачи построения иерархических моделей являются важными во многих областях (информационные системы, моделирование знаний и др.). Для построения иерархических структур применяются различные методы [44, 46, 146, 180, 207, 212, 218, 219, 227, 238, 239, 277, 303, 317, 413, 418, 432, 436, 458, 568, 575, 586]. В данной главе описано следующее:

I. Подходы к построению иерархий как моделей систем (включая экспертные процедуры, иерархическая кластеризация и покрывающие деревья).

II. Схемы трансформации иерархий, например: преобразование исходного дерева в дерево Штейнера, реструктуризация комбинаторных задач, добавление в иерархические структуры "горячих связей" (hotlink assignment).

На Рис. 2.1 представлена схема базового проектного процесса для построения иерархической структуры.

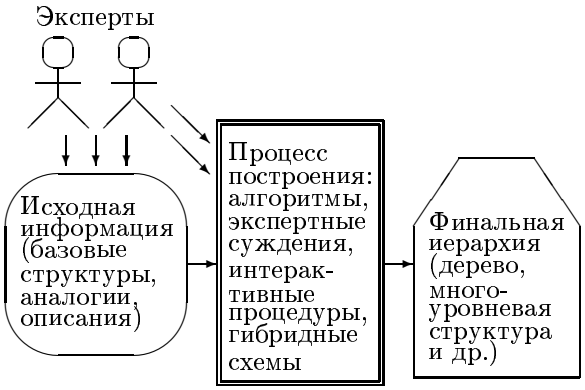


Рис. 2.1. Схема построения иерархии

Можно выделить следующие основные типы задач построения иерархий систем:

Задача 2.1. Проектирование на основе экспертных суждений (экспертных процедур):

[1] Материал данной главы основан на публикации: M.Sh. Levin, Towards design of hierarchy (research survey). Electr. prepr., 36 p., Dec. 8, 2012. Http://arxiv.org/abs/1212.1735 [math.OC].

Вход: описание системы.

Выход: иерархия как результат разбиения системы на части (дерево, иерархия, уровневая структура).

Задача 2.2. Базовое проектирование:

Вход: множество элементов системы, параметры элементов и их взаимосвязей.

Выход: множество кластеров или покрывающая структура над множеством исходных элементов системы (дерево, иерархия, уровневая структура).

Задача 2.3. Покрытие:

Вход: исходный граф/сеть как модель системы (т.е., множество элементов системы и ребра).

Выход: покрывающие иерархические структуры (дерево, иерархия, уровневая структура).

Задача 2.4. Перепроектирование (модификация, трансформация, улучшение):

Вход: исходная иерархическая структура (дерево, иерархия, уровневая структура).

Выход: новая иерархическая структура с некоторыми требуемыми свойствами (дерево, иерархия, уровневая структура).

Задача 2.5. Реструктуризация (специальный случай модификации):

Вход: (i) исходная иерархическая структура (дерево, иерархия, уровневая структура). (ii) целевая иерархическая структура (дерево, иерархия, уровневая структура).

Выход: новая иерархическая структура (дерево, иерархия, уровневая структура) с учетом следующего: (а) "дешевая" трансформация исходной структуры, (б) "маленькая" близость между новой структурой и целевой структурой.

В основном, указанные задачи кратко описаны с помощью типовой комбинаторной схемы решения:

(i) инженерное описание,

(ii) формулировка задачи,

(iii) некоторые возможные версии задачи,

(iv) базовая схема решения,

(v) перспективные версии задачи (например, многокритериальная постановка, учет неопределенности).

2.2 Экспертная процедура 'сверху-вниз'

В основном, экспертные процедуры построения иерархий систем базируются на знаниях специалистов в конкретных областях (domain experts) и типовых "технологических" схемах (например, жизненный цикл продукции: проектирование, производство, тестирование, техническое обслуживание, использование, утилизация). Такие процедуры включают следующие фазы:

1. разбиение системы на подсистемы;

2. разбиение каждой подсистемы на ее части;

3. разбиение каждой части на ее компоненты; и т.д.

Целесообразно указать базовые алгоритмические правила для разбиения систем на части: (а) разбиение по физическим частым системы, (б) разбиение по системным функциям, (в) разбиение по стадиям обработки информации. Два прикладных примера иллюстрируют указанный тип экспертных процедур:

(1) план лечения астмы (Рис. 2.2) [344, 383],

(2) двух-этажное здание (Рис. 2.3) [344, 385].

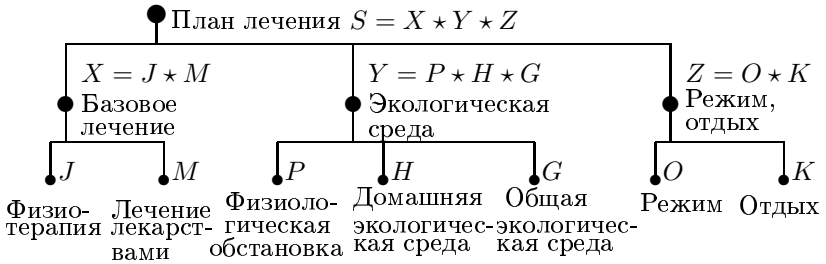


Рис. 2.2. Иерархическая модель плана лечения астмы [344, 383]

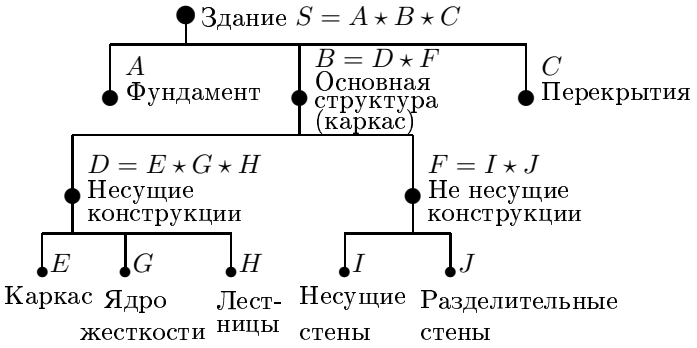


Рис. 2.3. Структура здания [344, 385]

2.3 Иерархическая кластеризация

Имеется множество n элементов (например, альтернатив, элементов системы):

$$A = \{A_1, ..., A_i, ..., A_n\}$$

и соответствующая векторная оценка по m параметрам $(T_1, ..., T_j, ..., T_m)$ для каждого элемента A_i: $z_i = (z_{i,1}, ..., z_{i,j}, ..., z_{i,m})$. Базовый агломеративный алгоритм (полиномиальный) имеет вид (процесс интеграции пар элементов 'снизу-вверх') [221, 524]:

Стадия 1. Вычисление матрицы расстояний для пар элементов $\forall(A(i_1), A(i_2)),\ A(i_1) \in A,\ A(i_2) \in A,\ i_1 \neq i_2$ (в простейшем случае: Эвклидово расстояние):

$$d_{i_1 i_2} = \sqrt{\sum_{j=1}^{m}(z_{i_1,j} - z_{i_2,j})^2}.$$

Стадия 2. Выделение пары элементов с наименьшим расстоянием и интеграция соответствующих элементов данной пары в новый интегрированный элемент.

Стадия 3. Окончание процесса или пере-вычисление матрицы расстояний пар элементов и переход к *стадии 2.*

В результате работы данного алгоритма получается древовидная структура (на каждом шаге проводится интеграция одной пары элементов). Некий базовый подход к процедуре интеграции элементов (получение средних значений параметров для нового интегрированного элемента) имеет вид: $(J_{i_1,i_2} = A_{i_1} \& A_{i_2})$: $\forall j \quad z_{J_{i_1,i_2},j} = \frac{z_{i_1,j} + z_{i_2,j}}{2}$. Очевидно, что процесс интегрирования элементов (расчет новых значения параметров) можно основывать на других функциях, например: max, min . Следует заметить, что интеграция нескольких элементов может проводиться аналогично. Пример иерархической кластеризации приведен в Таблице 2.1 и на Рис. 2.4. Рассматриваются 8 элементов (восемь человек анализируются на основе их особенностей: интересы и склонности).

Таблица 2.1. Критерии, оценки

Человек	K_1	K_2	K_3	K_4
A_1	0	5	2	3
A_2	5	2	3	3
A_3	4	3	1	2
A_4	4	3	4	2
A_5	3	5	3	5
A_6	1	5	2	5
A_7	3	3	5	5
A_8	3	3	4	4

Использованные критерии соответствуют следующим параметрам (порядковая шкала $[0, 5]$): (i) склонность к математическому и логическому мышлению (K_1), (ii) интерес к музыке (K_2), (iii) интерес к спорту (K_3), (iv) интерес к путешествиям (K_4). Результат применения базового алгоритма иерархической кластеризации приведен на Рис. 2.4.

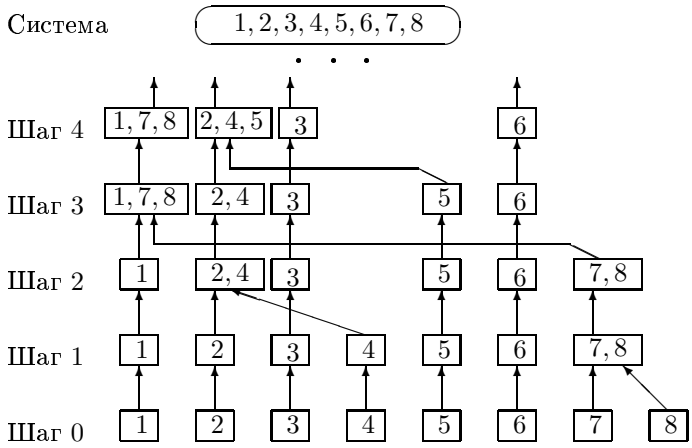


Рис. 2.4. Пример иерархической кластеризации

В последние годы предложены различные модификации агломеративного алгоритма (например, [186, 542]). Можно рассмотреть некоторые модификации на основе работы [347].

Во-первых, можно использовать порядковую шкалу при оценивании близости между элементами.

Во-вторых, можно использовать векторную оценку близости элементов (в частности, на основе Парето-подхода [470]).

В-третьих, процесс кластеризации может быть организован как последовательное выделение клик [207].

Дополнительно, возможно представление процесса кластеризации в виде оптимизационной задачи (включая многокритериальные постановки). При этом, в качестве целевых функций может использоваться следующее: (1) расстояние между элементами внутри кластера, (2) расстояние между элементами различных кластеров. Иногда используются другие целевые функции: число кластеров или близость к заданному интервалу числа кластеров в получаемом решении, число элементов в кластерах. Другие варианты целевых функций, используемых при кластеризации, приведены в [258, 280]. Следует отметить, что перспективным является объединение алгоритмов иерархической кластеризации и интерактивных экспертных процедур.

2.4 Об использовании онтологии

Последние десятилетия широко используется подход в представлению информации на основе онтологий в различных прикладных областях [146, 206, 225, 256, 409, 444, 454, 563]: системы на основе знаний, системное проектирование, библиотечные системы, химия, биомедицинская информация, концептуальное моделирование, семантические Веб-системы. Онтологии представляют собой артифакты are artifacts в виде структур (логических, лингвистических, таксономических) для описания областей или "пространств" задач [454, 563]. Фактически, при этом используются иерархические (многоуровневые) структуры.

В рамках исследований в области онтологий выделяют ряд основных проблем [146, 256, 454, 455, 456, 563]: (а) проектирование, (б) сравнение, (в) интеграция или слияние, (г) выравнивание (alignment).

Общая схема при построении онтологий была предложена в [454]. В работе [204] процесс выявления онтологий представляется как следующая последовательность действий:

Стадия 1. Выполнение предварительной работы на основе доступных документов.

Стадия 2. Создание первой версии онтологии (как структуры).

Стадия 3. Генерация концептов и отношений (например, на основе текстовых документов).

Стадия 4. Гармонизация полученной онтологии посредством анализа других онтологий и концептуальных описаний.

Стадия 5. Улучшение и подтверждение правильности/корректности.

В качестве основных источников при построении онтологий обычно указываются следующие: концептуальные схемы (классификации, тезаурусы, списки номенклатур), источники из Интернета, документы на естественных языках, справочники, языки моделирования (например, UML, сети Петри).

2.5 Покрывающие деревья

Задачи о покрывающих деревьях могут использоваться в качестве построения иерархических моделей, когда имеется предварительная сеть над системными элементами. Многие годы задачи о покрывающих деревьях используются в приложениях (например, проектирование сетей, проектирование коммуникационных протоколов, проектирование сверхбольших интегральных схем) и интенсивно исследуются в комбинаторной оптимизации [8, 147, 173, 207, 217, 261, 303, 583, 584]. Здесь представлена краткая информация о трех задачах о покрывающих деревьях: (а) минимальное покрывающее дерево, (б) минимальное дерево Штейнера, (в) задача о покрытии деревом с максимальным числом висячих вершин. Таблица 2.2 содержит список основных типов задач покрытия деревом и соответствующие литературные источники. Рис. 2.5 иллюстрирует две задачи о покрытии деревом.

Задача минимального покрывающего дерева (или минимального взвешенного покрывающего дерева) является базовой [147, 171, 202, 207, 476]. Пусть $G = (A, E)$ будет связным графом (A - множество вершин, E - множество ребер/дуг) с неотрицательными весами ребер/дуг.

Покрывающее дерева данного графа $T = (A, E')$ $(E' \subseteq E)$ представляет собой подграф, который является деревом и связывает все вершины исходного графа. Общий вес (стоимость) покрывающего дерева $c(T)$ является суммой весов всех ребер/дуг (т.е., E'). Минимальное покрывающее дерево (или минимальное взвешенное покрывающее дерево)

T^* представляет собой покрывающее дерево с общим весом меньше или равным весу любого другого покрывающего дерева $c(T^*) = \min_{\{T\}} c(T)$. Для данной задачи имеются полиномиальные алгоритмы [147, 202, 207, 476]. Среди основных таких алгоритмов можно указать следующие [147, 207]: (а) алгоритм Прима (Prim's algorithm), (б) алгоритм Краскала (Kruskal's algorithm), (в) алгоритм Борувки (Boruvka's algorithm).

Таблица 2.2. Список основных типов задач в области покрывающих деревьев

Задачи о покрытии	Источники
1.Покрывающее дерево	
1.1.Минимальное покрывающее дерево	[147, 202, 207, 476]
1.2.Покрывающее дерево с минимальным диаметром	[214]
1.3.Минимальный покрывающий лес	[207, 477]
1.4.Минимальное покрывающее мульти-дерево	[213, 266, 549]
1.5.Многокритериальное покрывающее дерево	[61, 128, 162, 242]
2.Дерево Штейнера	
2.1.Дерево Штейнера с единичными весами	[447]
2.2.Дерево Штейнера на узкие места	[68, 172, 173]
2.3.Дерево Штейнера с минимальным числом вершин Штейнера	[125, 172, 173, 405]
2.4.Терминальная задача дерева Штейнера	[168, 196, 406]
2.5.Дерева Штейнера с весами вершин	[229, 300, 507, 597]
2.6.'Prize-collecting' дерево Штейнера	[235, 261, 408, 559]
2.7."Живучая" сеть Штейнера	[173]
2.8.Дерево Штейнера с раскраской	[173]
2.9.Задача расписания с деревом Штейнера	[173]
2.10.Дерево Штейнера с ограничениями	[149, 490]
2.11.Дерево Штейнера с ограничениями для пунктов ретрансляции	[149, 570]
2.12.Дерево Штейнера с полезностью	[148]
2.13.Обобщенная задача дерева Штейнера	[45, 63, 170, 171, 580]
2.14.Обобщенная задача дерева Штейнера для звезды	[297]
2.15.Стохастическая задача дерева Штейнера	[232, 233]
2.16.Динамическое дерево Штейнера	[262]
2.17.Дерево Штейнера в режиме on-line	[50, 63, 580]
2.18.Групповая задаче дерева Штейнера	[124, 174]
2.19.Лес Штейнера	[185]
2.20.Многокритериальное дерево Штейнера	[399, 401, 573]
2.21.Многокритериальное дерево Штейнера с стоимостью вершин Штейнера	[401]

Рис. 2.5. Две задачи о покрытии деревом: иллюстрация [355]

В более сложных случаях используется "покрывающий лес": некоторый граф (не обязательно связный) имеет "минимальный покрывающий лес", который объединяет минимальные покрывающие деревья для связных компонентов графа [207, 477].

В последние годы исследуются задачи многокритериальных покрывающих деревьев [61, 128, 162, 242]. В этом случае каждому ребру (или каждой дуге) графа соответствует

векторный вес и целевая функция оптимизационной задачи рассматривается как сумма соответствующих компонентов векторных весов: $\overline{c}(T)$. При этом осуществляется поиск Парето-эффективных (по $\overline{c}(T)$) решений. Эта задача является NP-трудной (даже в двухкритериальном случае) [61, 128]). Для данной задачи используются следующие алгоритмы [61, 128]: (i) многокритериальный алгоритм Прима, (ii) многокритериальный алгоритм Краскала, (iii) генетические алгоритмы, (iv) алгоритмы многокритериальной эволюционной оптимизации (EMO), (v) эвристики, (vi) методы на основе знаний.

В задаче дерева Штейнера [45, 63, 170, 171, 173, 207, 217, 261, 300, 507, 580, 583], покрывающее дерево может включать дополнительные вершины (вершины Штейнера). Здесь оптимальное решение T_s представляет собой дерево (дерево Штейнера). Общий вес дерева Штейнера может быть меньше, чем в случае задачи минимального покрывающего дерева. Эта задача является NP-трудной.

Описания многих типов моделей задачи дерева Штейнера содержится в [217]. Среди основных подходов к решению задач дерева Штейнера можно указать следующие: (1) точные алгоритмы (переборные алгоритмы типа метода ветвей-и-границ) [134, 408] и алгоритмы на основе динамического программирования [197]; (2) различные эвристики [149, 569, 571], включая следующие: (i) быстрые (жадные) алгоритмы [124, 149]; (ii) аппроксимирующие адгоритмы [45, 173, 229, 233, 235, 300, 597]; (iii) генетические алгоритмы [287]; (iv) методы на основе искусственного интеллекта [281]; (v) методы локального поиска [117].

В многокритериальных задачах дерева Штейнера [399, 401, 573]) рассматривается векторный вес для каждого ребра (дуги) и целевая функция может иметь вид вектора сумм по каждому компоненты весов-векторов для ребер (дуг): $\overline{c}(T_s)$. В результате, производиться поиск Парето-эффективных решений (по $\overline{c}(T_s)$) [399, 401]. Здесь можно указать следующие подходы: (i) известные эвристические подходы [573]; (ii) специальные многоуровневые макро-эвристики: (а) эвристика на основе разбиения-синтеза [336], (б) эвристики на основе покрывающих деревьев [399], (в) составная многостадийная схема [401].

Задача построения покрывающего дерева с максимальным числом висячих вершин рассмотрена в [51, 207, 301] (Рис. 2.6). Покрывающий граф содержит следующие типы вершин: (а) корень, (б) внутренние вершины, такие вершины могут рассматриваться как виртуальная “шина” (“bus”) в сети, (в) висячие вершины. Таким образом, задача состоит в максимизации числа висячих вершин или или в минимизации числа внутренних вершин. Эта задача является NP-трудной [207].

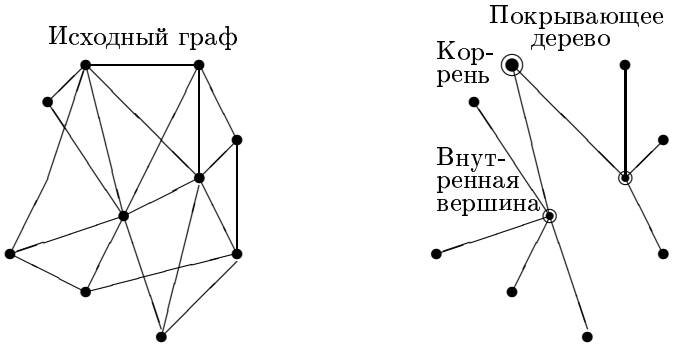


Рис. 2.6. Максимизация числа висячих вершин

Эта задача используется в проектировании и тестировании сетей (деревья радиовещания с минимизацией энергетических затрат в радио сетях) [404, 552], задачи размещения в электрических сетях [536]. При этом применяются следующие алгоритмы: (1) точные алгоритмы (метод ветвей-и-границ) [187, 199], (2) локальная оптимизация [410], (3) жадные алгоритмы [411], (4) аппроксимационные алгоритмы [526], (5) различные новые типы

эвристик (например, Bee Colony algorithm) [169, 520]. Можно указать некоторые перспективные обобщения рассматриваемой задачи, например: (а) многокритериальное описание висячих вершин, (б) учет неопределенности параметров, (в) учет динамики (т.е., режима реального времени).

С точки зрения точного решения, указанная задача эквивалентна задаче связного доминирующего множества (т.е., построение покрывающего подграфа с минимальным числом внутренних вершин) ("connected dominating set") [93, 119, 131, 207]. Обзор задач этого типа содержится в [93]. Задача связного доминирующего множества играет центральную роль в сенсорных радио сетях, мобильных сетях типа (MANET), в тестировании сетей [93, 158, 404, 552]. Эта задача используется в коммуникационных протоколах [93]: (i) координация доступа на среднем уровне (media access) (ii) маршрутизация типа unicast, multicast, broadcast, (iii) маршрутизация на основе задачи размещения, (iv) консервация энергии, (v) управление топологией, (vi) выявление ресурсов в сетях типа MANET. Для той задачи применяются следующие типы алгоритмов: (а) апроксимационные [131, 228], (б) эвристики [115, 501]. Аналогично, представляются перспективными следующие обобщения данной задачи: (а) многокритериальное описание вершин, (б) учет неопределенности параметров, (в) учет динамики (т.е., режима реального времени).

2.6 Оптимальные организационные иерархии

Организационные иерархии исследуются многие годы [1, 2, 73, 161, 227, 417, 433, 547, 568, 575]). Обычно рассматриваются следующие типы организационных структур в государственных организациях и в фирмах (производство, продажи и др.): бюрократические структуры, функциональные структуры, структуры в рамках производства определенного продукта/изделия на фирме (product structure), матричные структуры (интеграция функциональных структур и структур по продуктам). В основном, используются следующие иерархические уровни организаций: (а) огранизация, (б) филиалы, (в) отделы, (г) рабочие группы, (д) индивидуалы. На Рис. 2.7 представлен пример иерархической структуры для университета.

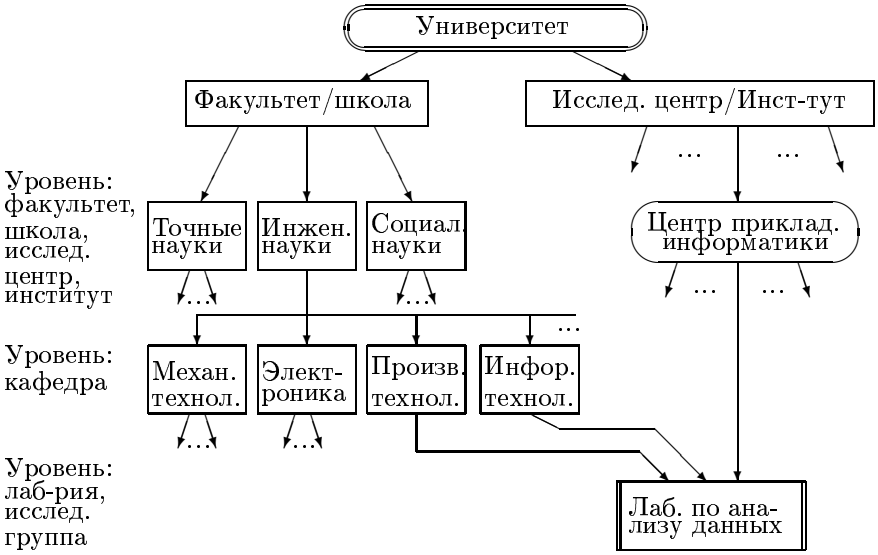


Рис. 2.7. Упрощенный пример структуры университета

В последние годы интерес к построению организационных иерархий повысился (например, [227, 433, 568]). Здесь можно рассматривать подходы на основе покрывающих

структур или "прямое" проектирование. Два базовых типа задач "прямого" проектирования могут рассматриваться так [328, 349]:

1. построить иерархию, обладающую наилучшим значением (значениями) их свойств(а);

2. построить иерархию, являющуюся наиболее "близкой" к некой "идеальной" структуре (или множеству "идеальных" структур).

Пусть $\{H_i = (A_i, E_i), i = \overline{1,n}\}$ будет множеством исходных структур (иерархий). Пусть $H^g(A_g, E_g)$ $(H^g \in \{G_i\})$ означает некую целевую ("идеальную") структуру, пусть $\rho(H_{i_1}, H_{i_2})$ означает близость (или расстояние) между двумя структурами (иерархиями) H_{i_1} и H_{i_2} $(\forall H_{i_1}, G_{i_2} \in \{H_i\})$. Дополнительно, целесообразно рассмотреть свойства структуры (например, степень вершины, связность вершины) $H_{i_1} \in \{H_i\}$:

$$\overline{\psi(H_{i_1})} = (\psi_1(H_{i_1}), ..., \psi_\mu(H_{i_1}), ..., \psi_m(H_{i_1})).$$

Здесь ищется иерархия $H^* \in \{H_i\}$. Таким образом, задача первого указанного типа (максимизация свойства) имеет вид:

$$\max_{H^* \in \{H_i\}} \ \psi_\mu(H^*) \quad \forall \ \mu = \overline{1,m}.$$

Эта задача похожа на известную задача "graph augmentation", т.е., модификация исходного графа для получения графа с заданными свойствами [182, 296].

Во втором указанном типе задачи целевая функция имеет вид: $\min_{H^* \in \{H_i\}} \ \rho(H^*, H^g)$.

Очевидно, что могут быть рассмотрены многокритериальные формулировки задачи. Дополнительно, можно рассмотреть интегрированную версию задачи, например:

$$\min_{H^* \in \{H_i\}} \ \rho(H^*, H^g) \qquad s.t. \quad \psi_\mu(H^*) \geq d^\mu \ \ \forall \ \mu = \overline{1,m},$$

где $(d^1, ..., d^\mu, ..., d^m)$ - векторное ограничение для свойств. Указанные оптимизационные модели обычно являются сложными целочисленными задачами (или задачами смешанного целочисленного пограммирования). Для их решения используются различные методы решения (переборные алгоритмы, эвристики, методы искусственного интеллекта).

Кроме того, имеет следующий оптимизационный подход к проектированию "оптимальных" организационных иерархий [227, 433, 568]. Данный подход основан на использовании некого функционала P от иерархической структуры $G \in \Omega$ (Ω - множество иерахических структур в виде направленных ациклических "уровневых" графов; $P : \ \Omega \longrightarrow \ [0, +\infty)$) и задача имеет вид: $\arg\min_{G \in \Omega} P(G)$. В основном структура G предполагается деревом, а функционал P - выпуклым [227, 433, 568]. Приложения данного подхода включают технологические (производственные) процессы, логистические системы.

2.7 Много-уровневые структуры

2.7.1 Много-уровневой подход

Много-уровневый подход является базовым для моделирования сложных систем [355, 414, 415, 425, 545, 546]. Фактически, данный подход направлен на уменьшение сложности систем, т.е., система разбивается на части (уровни) и соответственно разбиваются на части системные задачи. Следует указать основные методологические шаги. Во-первых, в многоуровневых системах выделяют главные свойства: (i) устойчивость (стабильность), (ii) управляемость, (iii) адаптивность, (iv) самоорганизация. [425]. Во-вторых, для компьютерных систем предложено описание на основе семи уровней [546]: (1) уровень электроники,

(2) уровень микропрограммирования, (3) уровень операционной системы, (4) уровень языка Assembler, (5) уровень алгоритмических языков, (6) уровень прикладных систем (например, DBMS, DSS), (7) уровень приложений. В-третьих, для коммуникационных систем предложены семь уровней передачи данных (OSI model) [545, 594]:

(1) физический уровень (сигналы и двоичная передача данных, медиа-информация),

(2) уровень связи данных (data link), физическая адресация,

(3) сетевой уровень (определение путей, логическая адресация),

(4) транспортный уровень (связи пользователь-пользователь, надежность и управления потоками)

(5) уровень сессий (вязь между головными узлами - hosts, управление сессиями между приложениями),

(6) уровень представления (представление данных, шифрование и дешифрование, преобразование машинно-зависимых данных в машинно-независимые данные),

(7) уровень приложений.

Много-уровневый подход может быть применен к многим прикладным системам/областям. Например, четырех-уровневая структура была предложена для моделирования области "комбинаторные оптимизационные задачи" [355]:

(i) уровень базовых задач комбинаторной оптимизации,

(ii) уровень многокритериальных комбинаторных задач,

(iii) уровень типовых составных комбинаторных схем,

(iv) уровень типовых приложений.

Много-уровневые системные иерархии иерархии включают следующее:

(а) иерархические уровни;

(б) множество элементов каждого иерархического уровня, описание этих элементов (т.е., параметры);

(в) взаимосвязи на множестве элементов каждого иерархического уровня;

(г) связи между элементами соседних иерархических уровней.

Таким образом, можно рассмотреть следующую схему проектирования много-уровневой иерархии:

1. Генерация много-уровневой структуры, например: (i) разбиение исходного набора элементов/узлов на части, соответствующие уровням, (ii) описание элементов уровней (узлов, дуг) (например, трафик), (iii) построение структуры для каждой части (уровня), например: путь, несколько путей, дерево, кольцо, полный граф (клика) или их комбинации.

2. Определение (поиск) связей между элементами соседних уровней.

2.7.2 Типовые иерархические уровни в системах связи

Во-первых, представляется важным представить типовую иерархию сети связи [153, 210, 317, 443]. Традиционная иерархия сетей включает следующие уровни:

(а) международные сети (между странами, между континентами) GAN;

(б) общегородские сети (metropolitan network MN);

(в) широко-масштабная сеть (wide area network WAN);

(г) локальные сети (local area network LAN).

IBM Red Book содержит полезное разбиение сетей о числу узлов [443]: (i) сети связи большого размера ($>$ 500 узлов); (ii) сети связи среднего размера ($<$ 500 nodes); (iii) сети связи малого размера ($<$ 80 nodes). С инженерной точки зрения можно рассматривать следующие иерархические уровни:

1. Базовая сеть (backbone network).

2. Глобальная сеть как множество взаимосвязанных сегментов сетей, включая следующее: (а) дополнительные центры, (б) перекрестные связи, (в) "мосты".

3. Сеть доступа/сеть сегментов (кластеров): (например, двух-связная топология).

4. Распределительная сеть, очень простая топология (например, шина, звезда, дерево, кольцо).

Упрощенная иерарархия сети может быть представлена как следующая (Рис. 2.8):

1. Верхний уровень: узлы и связи для связи кластеров: *1.1.* базовые узлы кластеров (сегменты сети), *1.2.* центры коммуникации, *1.3.* перекрестные связи (связь между центрами), 1.4. связи между соседними кластерами.

2. Средний уровень: базовые кластеры (сегменты связи/сети доступа, двух-связные топологические модули).

3. Нижний уровень: распределительные сети (например, шина, дерево, кольцо).

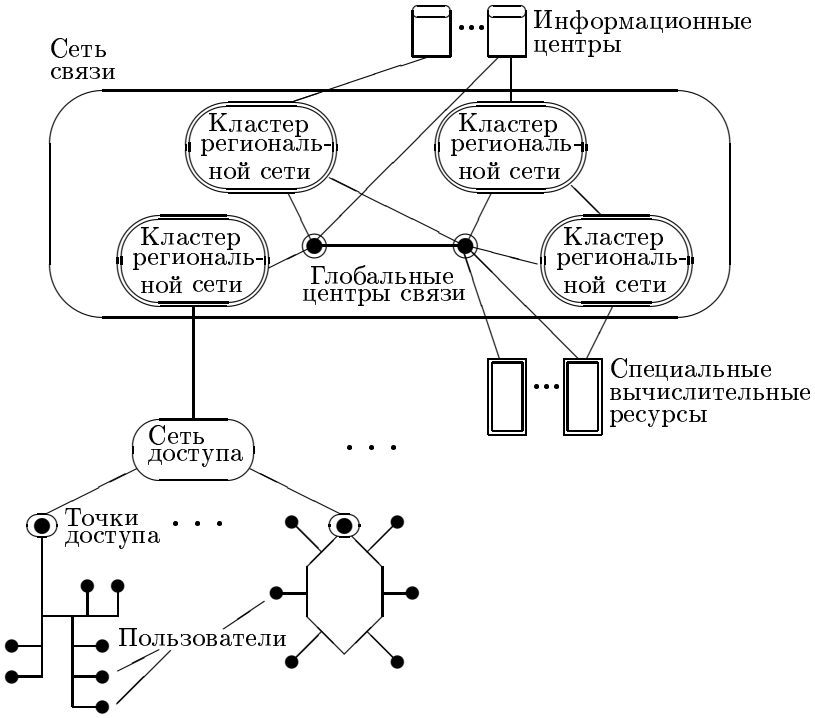


Рис. 2.8. Иллюстрация много-уровневой сети связи

2.7.3 Уровневая k-связная сеть

Специальная версия 2-связной сети была предложена в [84]. Обобщеная k-связная сеть такого типа описана в [355]. Пусть $G = (A, E)$ будет исходная сеть (граф), где A - множество вершин (узлов), E - множество ребер и $|A| \geq k \times (k+1) + k$.

Рассматриваемый тип k-связной сети имеет вид: (а) k "центров", где каждый "центр" представляет собой $(k+1)$-вершинную клику ("центры" не имеют пересечений), (б) множество узлов $M \subseteq A$ ($|M| \geq k$) включает все узлы, которые не относятся к "центрам"; имеется связь (ребро) между $\forall \alpha \in M$ и каждым "центр", т.е., k ребер (одно ребро для связи "центром"). В результате получается трех-уровневая сеть (Рис. 2.9): (i) уровень конечных пользователей, (ii) уровень специальных "центров" узлов связи: выбранные и/или специальные дополнительные узлы (эти узлы полкзны для размещения ключевого оборудования связи); (iii) "центры" связи (такие "центры" могут соответствовать коммуникационным провайдерам или коммуникационным операторам или их филиалам).

На Рис. 2.10 представлен пример четырех-связной структуры.

Доказательство k-связности для предложенного типа сети основано на рассмотрении 4-х специальных случаев. Пусть имеются два узла $a, b \in A$. Указанные случаи имеют вид:

Случай 1: a и b относятся к одному "центру" (т.е., $(k+1)$-клике). Узлы a и b имеют связи: (i) прямая связь (a, b), (ii) $(k-1)$ двух-реберная связь посредством другого узла этого "центра". Таким образом, узлы a и b имеют k связей (без пересечения).

Случай 2: a относится к "центру" A и b относится к "центру" B (т.е., к другому). Узлы a и b имеют следующие связи: двух-реберная связь посредством другого узла $\alpha \in M$ (т.е., k-связей этого типа). Таким образом, узлы a и b имеют k связей (без пересечений).

Случай 3: a относится к некому "центру" и $b \in B$. Узлы a и b имеют следующие связи: (i) прямая связь (a, b), (ii) $(k-1)$ трех-реберных связей: посредством некоторого узла каждого другого "центра" и посредством другого узла из M ($(k-1)$ различных пути). Таким образом, узлы a и b имеют k связей (без пересечений).

Случай 4: $a, b \in M$. Узлы a и b имеют k различных связей: узел a, некий "центр", узел b.

Рис. 2.9. Трех-уровневая модель сети

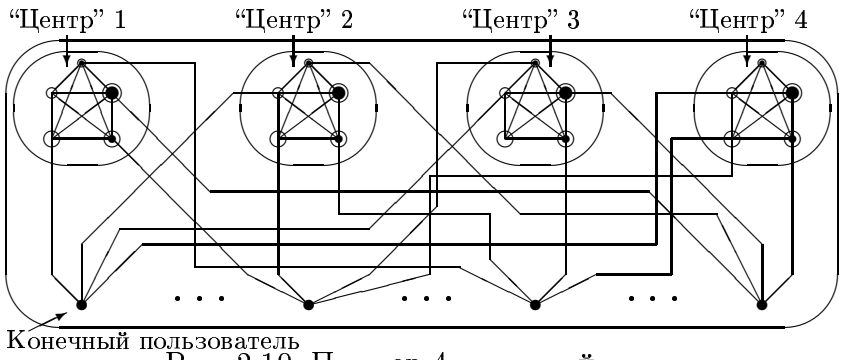


Рис. 2.10. Пример 4-х связной сети

Таким образом, схема построения ("снизу-вверх") k-связной структуры имеет вид:

Стадия 1. Выбор $k \times (k+1)$ вершин для k "центров" (при этом используется многокритериальное ранжирование).

Стадия 2. Кластеризация выбранных для получения k кластеров как "центров" (каждый кластер включает $(k+1)$ вершин, эти кластеры не имеют пересечений).

Стадия 3. Определение связей с конечными пользователями: связь каждой вершины (которая не относится к "центру") с каждым "центром" (т.е., с только одним узлом каждого "центра"). Здесь может быть использована задача блочного рюкзака или ее модификации.

Приведем иллюстративный пример двух-связной сети. На Рис. 2.11 представлены исходные узлы.

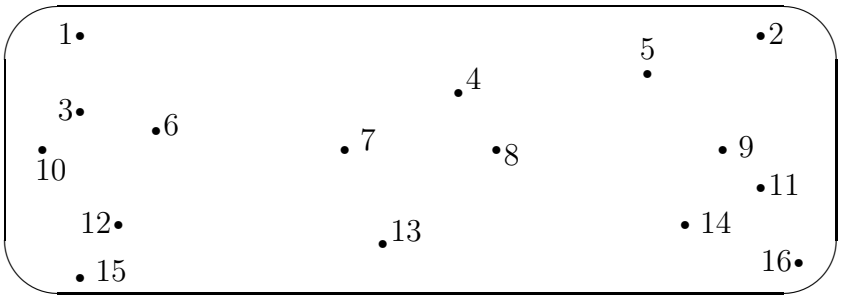


Рис. 2.11. Исходные узлы сети

Для “центров” выбираются следующие узлы: $\{3, 5, 9, 10, 12, 14\}$. Другие узлы соответствуют пользователям.

Версия результирующей двух-связной сети (*версия 1*) представлена на Рис. 2.12:

Версия 1. “Центры” являются локально размещенными (т.е., региональные “центры”): “центр” 1: узлы 3, 10, и 12; “центр” 2: узлы 5, 9, и 14.

С другой стороны, целесообразно рассмотреть другую ситуацию с “упрощением” связей для конечных пользователей (например, минимизация расстояний связи). В этом случае, необходимо использовать следующее алгоритмическое правило на *стадии 2* указанной схемы решения:

Узлы каждого “центра” должны ’покрывать’ всю сеть (т.е., для каждого конечного пользователя должен близко находиться хотя бы один узел “центра”)

Вторая схема решения для построения k-связной структуры имеет вид:

Стадия 1. Кластеризация исходных узлов для получения k кластеров.

Стадия 2. Построение k “центров”: выбор вершины в каждом полученном кластере как элемента каждого (некий представитель “центра”), связь вершин в каждом “центре”.

Стадия 3. Построение связей для каждого конечного пользователя: связь каждой вершины (который не относится к “центру”) с каждым “центром” (т.е., с только одним узлом в каждом “центре”).

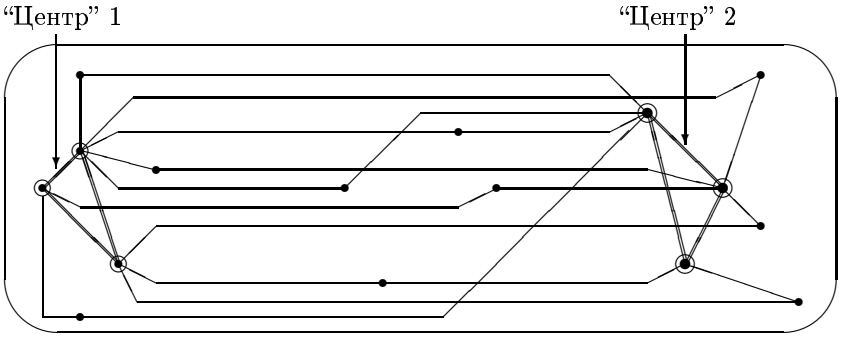


Рис. 2.12. Двух-связная сеть: локально размещенные “центры”

Пример такого типа результирующей двух-связной сети представлен на Рис. 2.13:

Версия 2. “Центры” распределены по сети: “центр” 1: узлы 3, 9, и 12; “центр” 2: узлы 5, 10, и 14.

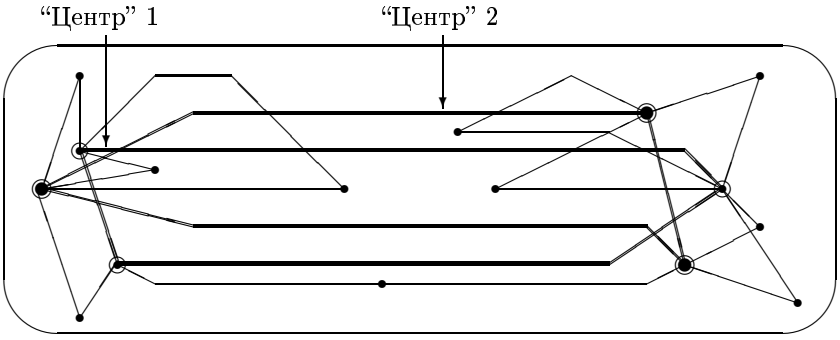


Рис. 2.13. Двух-связная сеть: “центры ” распределены по сети

2.7.4 О задаче проектирования иерархической сети

Базовая задача проектирования иерархической двух-уровневой сети заключается в нахождении покрывающей сети минимальной стоимости, включающей две части: (i) основная путь (или несколько путей, дерево, кольцо), (ii) вторичные деревья [153, 457, 458, 479]. Таким образом, исходная сеть разбивается на две части:

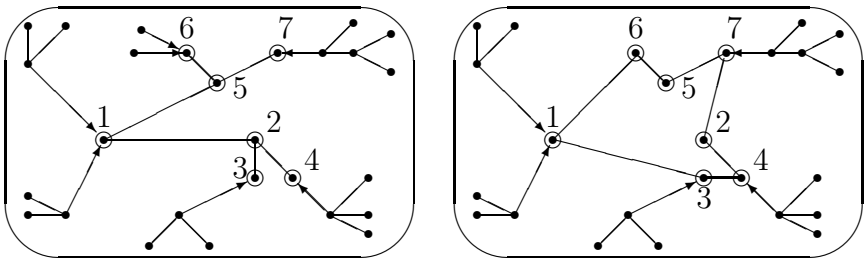
1. Часть верхнего уровня (первичные узлы, основной путь): некий путь (или несколько путей, дерево, кольцо), включающее первичные дуги, которые связанные с некоторыми узлами сети (т.к., первичными узлами).

2. Часть нижнего уровня (вторичные узлы, вторичные деревья): эта часть включает одно или более деревьев, чьи дуги (вторичные) являются мене “дорогими”, чем дуги первичных дуг.

Здесь каждая дуга имеет стоимость (d_{ij}, $\forall i, j \in A$, A - множество узлов). Общая стоимость выбранных дуг в покрывающей структуре используется как минимизируемая целевая функция. Задача формулируется как модель комбинаторной оптимизации [153], эта задача является NP-трудной [71]. Для решения задачи предлагались различные подходы, например: (а) точные переборные методы (метод ветвей и границ, методы на основе динамического программирования), (б) эвристики (включая Лагранжеву релаксацию), (в) эволюционные алгоритмы. Часто на предварительном этапе строится минимальной покрывающее дерево, в котором висячие узлы используются как вторичные. Далее представлены два иллюстративных примера:

(i) верхняя часть представляет собой дерево над первичными узлами $\{1, 2, 3, 4, 5, 6, 7\}$ (Рис. 2.14а),

(ii) верхняя часть представляет собой кольцо (цикл) над первичными узлами $\{1, 2, 3, 4, 5, 6, 7\}$ (двухсвязный случай) (Рис. 2.14б).



(а) верхний уровень: дерево (б) верхний уровень: кольцо

Рис. 2.14. Пример двух-уровневой сети

Приложения рассмотренной задачи включают следующие: системы связи, компьютерные сети, транспортные сети, распределенные энергетические системы. Следует отметить, что также рассматривается задача построения много-уровневых сетей [71, 135].

2.8 Морфологическая иерархия

Морфологическая иерархия обобщает модель системы из морфологического анализа [598] следующим образом [336, 344, 357, 358, 360]: (i) иерархическая (древовидная) структура исследуемой системы, (ii) проектные альтернативы для каждой висячей вершины системной модели (компонента), (iii) оценки проектных альтератив DAs на основе порядковых шкал [336, 344, 349] или специальных интервальных шкал в виде мультимножеств [360]; (iv) порядковые совместимости между проектными альтернативами для различных системных компонентов (вместо бинарной совместимости, которая используется в морфологическом анализе).

Для построения морфологической иерархии можно использовать схему:

Стадия 1. Построение иерархии (например, на основе одного их описанных выше методов).

Стадия 2. Генерация DAs для каждой висячей вершины системной модели (экспертные процедуры и/или информационные базы).

Стадия 3. генерация шкал для оценивания DAs и оценивание DAs (обычно на предварительном этапе используется многокритериальное описание).

Стадия 4. Генерация порядолвой шкалы для оценивания совместимости между DAs и оценивание этой совместимости.

2.9 Схема для много-уровневой структуры

Общая схема проектирования много-уровневой топологии имеет вид:

Стадия 1. Разбиние исходного множества узлов на подмножества, соответствуюшие уровням.

Стадия 2. Построение топологии для каждого уровня.

Стадия 3. Проектирование связей между узлами соседних уровней.

На Рис. 2.15 иллюстрируется схема проектирования: *1.* разбиение исходного множества узлов на два множества, соответствующие двум уровням; *2.* построение топологии для каждого уровня: (2.1) построение топологии (здесь: дерево) для “центральной” части (верхний уровень), (2.2) кластеризация узлов на нижнем уровне, (2.3) выбор/назначение “главных” узлов (head) в каждом кластере, (2.4) построение топологии (например, дерево, звезда, цикл, клика) для каждого кластера на нижнем уровне; *3.* построение связей между узлами нижнем уровне (т.е., “главных” узлов кластеров) и узлами на верхнем уровне.

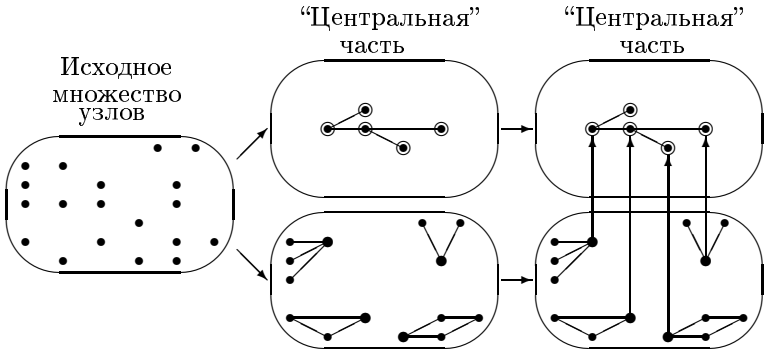


Рис. 2.15. Проектной схема для двух-уровневой сети

В результате получаются три уровня:

уровень 1: узлы “центральной” части с соответствующей топологией (здесь: дерево),

уровень 2: “главные” узлы кластеров на нижнем уровне,

уровень 3: другие узлы кластеров и соответсвующие топологии для каждого кластера (на нижнем уровне).

2.10 Выводы к главе

В главе описан обобщенный взгляд автора на проектирование системных иерархий. Такие задачи представляются часто ключевыми в системном анализе, в системном проектировании. В предложенном материале многие проблемы представлены на концептуальном уровне в виде упрощенных схем (инженерные описания, постановки задач, базовые схемы решения, перспективные версии задач). Очевидно, что многие упомянутые вопросы требуют дополнительных исследований.

Глава 3

Схемы оценивания систем

В данной главе [1] приведено краткое описание подходов к оцениванию иерархических модульных систем.

3.1 Предварительные сведения

Материал данной главы посвящен основным подходам к оцениванию иерархических модульных систем, включая следующие вопросы:

(i) основные шкалы, используемые для оценивания компоненты модульных систем и систем в целом (количественные порядковые шкалы, многокритериальные описания или векторые оценки, два типа шкал типа частичного порядка)

(ii) базовые задачи трансформации шкал в виде отображения 1:

исходная шкала $\Rightarrow$ *результирующая шкала*,

(iii) базовые типы методов для интеграции шкал в виде отображения 2:

исходные шкалы $\Rightarrow$ *результирующая интегрированная шкала.*

Предполагается, что указанные отображения являются монотонными (или анти- монотонными). Следует отметить, что метод *data envelopment analysis* здесь не рассматривается [553]. Оценивание составных (модульных) систем основывается на оценивании системных компонентов (включая их совместимость) и интеграции получаемых локальных оценок в общую оценку системы [336, 338, 344, 360]. В качестве основной задачи оценивания используется задача интеграция оценок системных компонентов (оценки совместимости системных компонентов могут рассматриваться как дополнительные оценки системных компонентов). В целом, описывается подход к оцениванию на основе комбинаций указанных двух задач преобразования шкал: трансформация шкалы и интеграция шкал. Далее укажем следующее:

(а) составная (модульная система) (двух-уровневая) $S = S_1 \star ... \star S_i \star ... \star S_m$ (где S_1, ... S_i, ... S_m обозначают системные компоненты/части) (Рис. 3.1) [336, 344, 357, 358, 360],

(б) локальные области (шкалы, множества оценок) для оценивания качества (т.е., эффективность, совершенство, "полезность") системных компонентов $\{S_1, ..., S_i, ..., S_m\}$ (и/или их проектных альтернатив DAs: $\{X_{i,1}, ..., X_{i,q_i} |\ i = \overline{1,m}\}$) и общая область (шкала, множество оценок) для оценивания целой системы S (Рис. 3.2) [336, 338, 344, 358, 360].

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. chapters 3, 4.
(ii) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapter 3.
(iii) Levin M.Sh. System synthesis with morphological clique problem: Fusion of subsystem evaluation decisions. Inform. Fusion, Vol. 2, No. 3, 2001, pp. 225-237.
(iv) M.Sh. Levin, Note on evaluation of hierarchical modular systems. Electr. prepr., 15 p., May 21, 2013. http://arxiv.org/abs/1305.4917 [cs.AI]

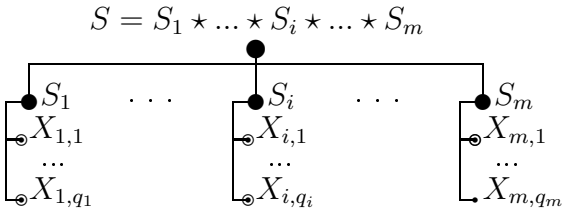


Рис. 3.1. Пример модульной системы

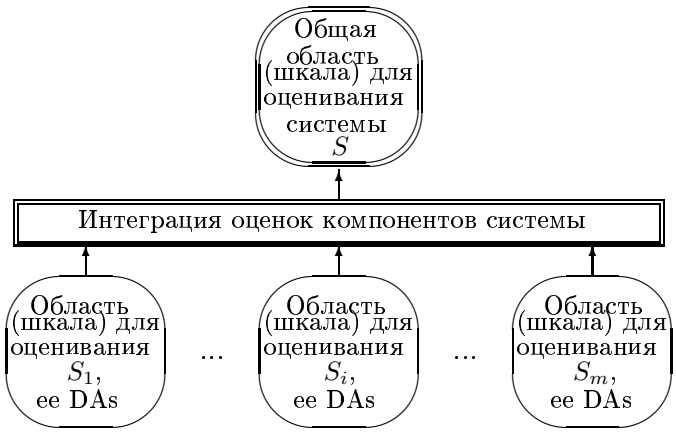


Рис. 3.2. Оценивание двух-уровневой системы

Во-первых, можно указать следующие базовые методы оценивания/измерения, которые могут быть использованы как базовые “блоки” в схемах измерения систем: (1) экспертные суждения (на основе специалистов в проблемной области);

(2) процедуры измерения: (a) технические измерения/испытания, (b) обработка накопленных статистических данных, (c) экспертное оценивание, (d) обработка данных из баз данных, (e) гибридные процедуры;

(3) имитационное моделирование;

Во-вторых, две базовые ситуации системного оценивания могут рассматриваться: [336, 338, 344]:

Ситуация 1. Оценивание целой системы для получения общей оценки системы (например, экспертные процедуры, тестирование системы, обработка статистических данных, накопление и обработка данных их баз данных, специальные технические процедуры измерения, гибридные процедуры).

Ситуация 2. Двух-стадийная схема:

2.1. Оценивание компонентов системы.

2.2. Интеграция оценок компонентов системы для получения общей оценки системы (эта стадия может выполняться итеративно несколько раз для многоуровневой системы).

Пример трех-уровневой системы представлен на Рис. 3.3. Здесь следующие процедуры оценивания рассматриваются:

(1) оценивание DAs для висячих вершин системной модели (т.е., компонентов системы.) (например, количественная шкала, порядковая шкала, многокритериальное описание, шкала в виде частично-упорядоченного множества);

(2) интеграция полученных оценок DAs для получения интегрированных оценок для составных системных частей (т.е., системные части на более высоком иерархическом уровне);

(3) интеграция полученных оценок для частей системы для получения интегрированной общей оценки системы (или ее версий).

Иерархическая схема оценивания приведена на Рис. 3.4.

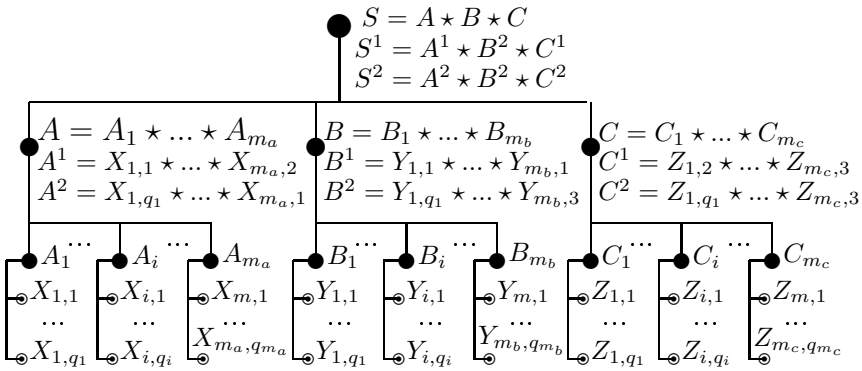


Рис. 3.3. Трех-уровневая составная система

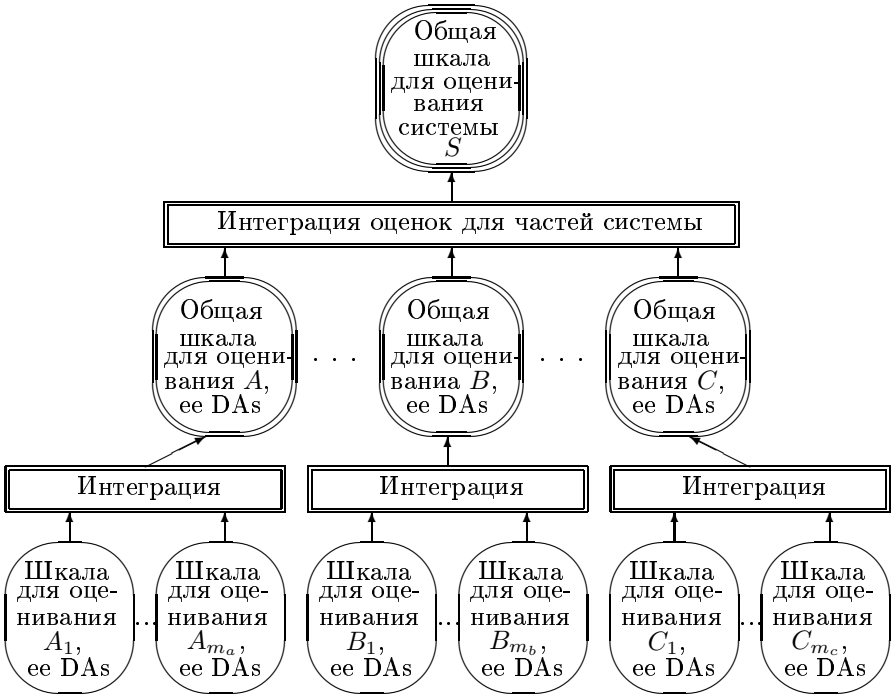


Рис. 3.4. Схема оценивания трех-уровневой системы

3.2 Рассматриваемые типы шкал

В Таблице 3.1 представлен перечень типов используемых шкалы оценивания (для частей системы, для системы в целом): количественная шкала, порядковая шкала, многокритериальное описание, шкалы на основе частичного порядка [190, 290, 336, 338, 344, 360, 429, 492, 533, 595].

Таблица 3.1. Типы используемых шкал оценивания систем

Типы шкал	Источники
1.Количественная шкала	[190, 290, 533]
2.Порядковая шкала	[106, 321, 336, 429, 492, 595]
3.Многокритериальное описание (векторная оценка на основе количественных и/или порядковых оценок)	[290, 429, 470, 492, 533]
4.Шкала частичного порядка на основе поряковых оценок	[336, 338, 344, 358],
5.Шкала частичного порядка на основе оценок в виде мультимножеств	[360]

3.2.1 Базовые шкалы

Приведем иллюстрации для указанных базовых типов используемых шкал.

Во-первых, на Рис. 3.5 представлены иллюстрации для количественной шкалы, порядковой шкалы и многокритериального описания (т.е., пространство векторных оценок):

(а) количественная шкала, например, интервал (β, α), α соответствует наилучшей точке, β соответствует наихудщей точке (Рис. 3.5а);

(б) качественная (порядковая) шкала: $[1, 2, ..., \kappa]$, 1 соответствует наилучшей точке, т.е., точка i доминирует точку $i+1$ (Рис. 3.5б);

(в) многокритериальные описания (т.е., векторые оценки) (Рис. 3.5в).

Следует отметить, что бинарное отношение доминирования для точек шкал в случаях (а) и (б) является очевидным. В случае (в), отношение доминировнаия иллюстрируется на Рис. 3.5: $\alpha_2 \succ \beta_2$, $\alpha_2 \succ \beta_3$, $\alpha_2 \succ \beta_4$. При использовании доминирования по Парето [429, 470], приведенные отношения доминирования расширяются следующим: $\alpha_2 \succ^P \beta_1$. Здесь можно рассматривать следующие упорядоченные уровни качества (как некую специальную порядковую шкалу D, иллюстрация приведена на Рис. 3.5в):

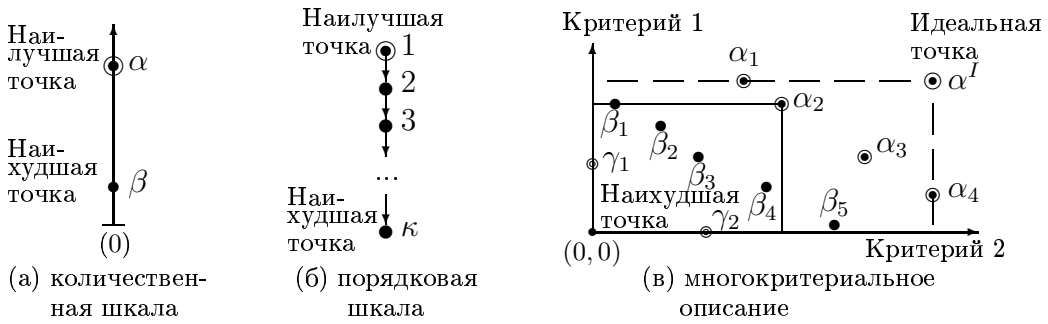
(i) идеальная точка (наилучшая точка) α^I,

(ii) уровень Парето-эффективных точек (точки: $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$),

(iii) точки, которые близки к Парето-эффективным точкам (например, их можно преобразовать в Парето-эффективные точки за счет небольшого улучшения): $\{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$,

(iv) следующий уровень качества (например, между точками предыдущего уровня и наихудшей точкой): $\{\gamma_1, \gamma_2\}$),

(v) наихудшая точка.



(а) количественная шкала (б) порядковая шкала (в) многокритериальное описание

Рис. 3.5. Шкалы: количественная, порядковая, векторы

3.2.2 Системная шкала на основе порядковых оценок компонентов

Описания шкал на основе частично-упорядоченных множеств для оценки качества составных (модульных) систем было предложено в рамках метода ИММП ([330, 332, 336, 338, 344, 358], глава 4). В базовой версии этого метода оценки базируются на порядковых оценках DAs и их совместимости. Здесь можно рассмотреть два случая:

(1) шкала для качества системы на основе порядковых оценок системных компонентов (оценка системного компонента по шкале: $\iota = \overline{1, l}$; 1 соответствует наилучшему значению);

(2) шкала для качества системы с порядковых оценок системных компонентов и порядковых оценок совместимости между системными компонентами (оценка совместимости системных компонентов по шкале: $w = \overline{1, \nu}$; ν соответствует наилучшему уровню).

Для системы, состоящей из m частей/компонентов, дискретная область ("пространство", частичных порядок, "решетка") основана на векторе: $N(S) = (w(S); n(S))$, где

$w(S)$ равен минимуму парной совместимости между DAs (соответствующим разным системным компонентам), (т.е., $\forall\ P_{j_1}$ и P_{j_2}, $1 \leq j_1 \neq j_2 \leq m$) in S, $n(S) = (\eta_1, ..., \eta_r, ..., \eta_k)$, где η_r равен числу DAs на уровне качества r в S $(\sum_{r=1}^{k} n_r = m)$.

Пример для трех-компонентной системы $S = X \star Y \star Z$ (Рис. 3.1) следующий имеет вид. Используются порядковые шкалы: (а) порядковая шкала для элементов (приоритеты): $[1, 2, 3]$, (б) порядковая шкала для совместимости: $[0, 1, 2, 3, 4]$. Для данного примера Рис. 3.6а представляет шкалу качества для системных компонентов и Рис. 3.6б представляет интегрированную шкалу с учетом оценок совместимости (каждый "треугольник" соответствует шкале на Рис. 3.6а).

Как было указано выше, можно использовать некие обобщенные уровни качества системы (шкала D на Рис. 3.5б):

1. Идеальная точка $N(S^I)$ (S^I - идеальное решения для системы).

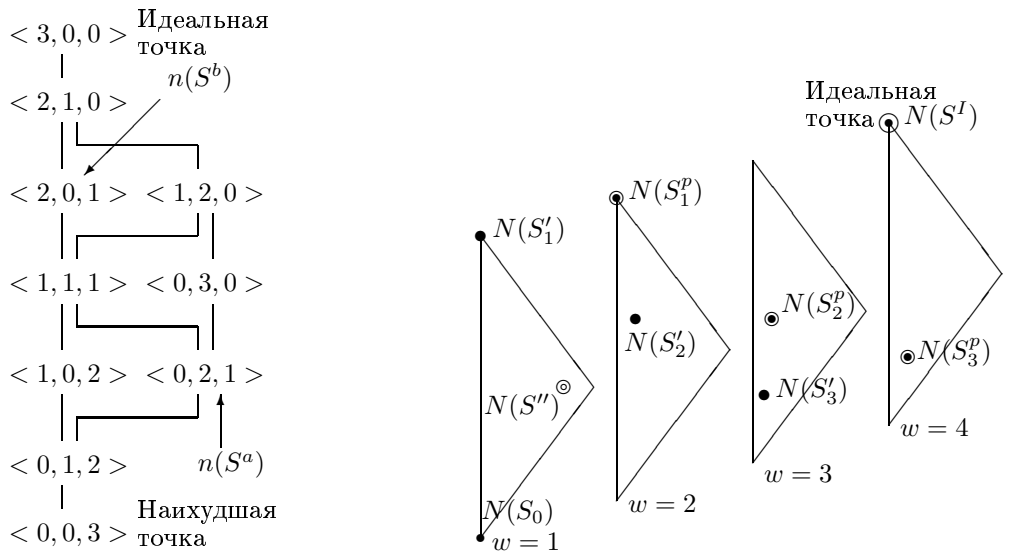
2. Уровень Парето-эффективных решений: $\{S_1^p, S_2^p, S_3^p\}$; составные оценки имеют вид: $N(S_1^p) = (2; 3, 0, 0)$, $N(S_2^p) = (3; 1, 1, 1)$, $N(S_3^p) = (4; 0, 2, 1)$.

3. Следующий уровень качества (т.е., область, которая является "соседней" к уровню Парето-эффективных решений): $\{S_1', S_2', S_3'\}$; составные оценки имеют вид: $N(S_1') = (1; 3, 0, 0)$, $N(S_2') = (2; 1, 1, 1)$, $N(S_3') = (3; 0, 2, 1)$.

4. Следующий уровень качества: S''; составная оценка имеет вид: $N(S'') = (1; 0, 3, 0)$.

5. Наихудшая точка: S_0; составная оценка имеет вид: $N(S_0) = (1; 0, 0, 3)$.

Отметим, что оценка совместимости между DAs также может быть использована на основе шкалы в виде частично-упорядоченного множества (как $n(S)$) [338, 344]. В этом случае общая шкала для оценки качества системы будет имеет аналогичный виде (немного сложнее).



(а) шкала на основе оценок элементов $n(S)$

(б) шкала на основе оценок элементов и оценок совместимости $N(S)$

Рис. 3.6. Шкала для системы на основе порядковых оценок

3.2.3 Специальная шкала для системных компонентов

Некоторые варианты обобщения задачи морфологической клики (случаи шкал оценки DAs в виде некоторых простых частичных порядков) и соответствующие обобщенные варианты дискретных "пространств" качества (решеток качества) для составных решений рассмотрены в [338, 344]. Рис. 3.7 иллюстрирует рассмотренные шкалы. Пример решетки качества приведен на Рис. 3.8.

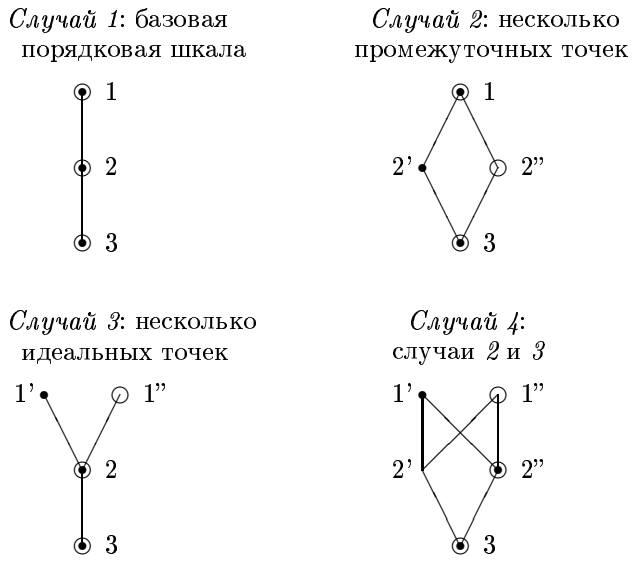


Рис. 3.7. Варианты простых шкал [338, 344]

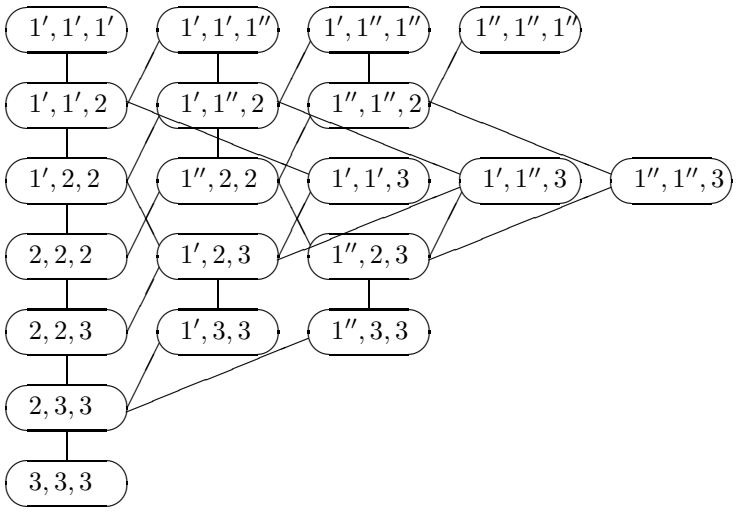


Рис. 3.8. Расширенная решетка качества: *случай 3* [338, 344]

3.2.4 Интервальные оценки в виде мультимножеств

Шкала оценивания системы в (частично-упорядоченное множество) на основе интервальных оценок в виде мультимножеств для компонентов системы была предложена в [360]. Здесь также можно рассмотреть два случая:

(i) использование только оценок системных компонентов,

(ii) использование оценок системных компонентов и оценок совместимости между компонентами.

Рис. 3.9 иллюстрирует такую шкалу для интервальных оценок в виде мультимножеств, соответствующих задаче оценивания системных компонентов $P^{3,4}$ (оценивание по порядковой шкале $[1, 3]$ с использованием четырех элементов; оценки $(2, 0, 2)$, $(3, 0, 1)$, $(1, 0, 3)$ - не используются) [360]. Очевидно, что здесь также можно применить обобщенную системную шкалу типа D.

Некий пример для четырех-компонентной системы приведен на Рис. 3.10.

Предполагается, что для оценивания DAs используются интервальные оценки в виде мультимножеств (шкала на Рис. 3.9). Для оценивания системы из четырех частей необходимо учесть совместимость между DAs (для совместимости используется шкала:

$[0, 1, 2, 3]$). Получается интегрированная шкала для системы, изображенная на Рис. 3.11 (предполагается интегральная оценка для системы типа медианы [360]).

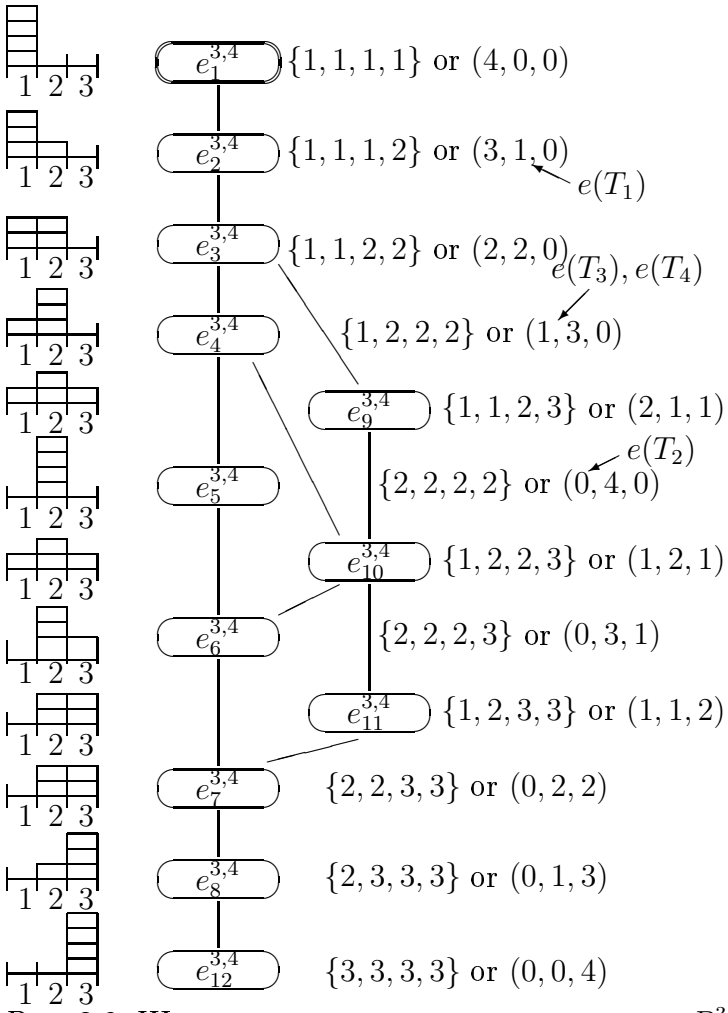


Рис. 3.9. Шкала для системы на основе задачи $P^{3,4}$

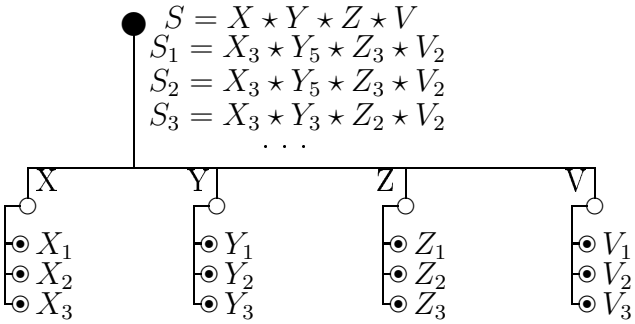


Рис. 3.10. Пример четырех-компонентной системы

Иллюстрация обобщенной порядковой шкалы для системы типа D имеет вид (для Рис. 3.11):

1. Идеальное решение S^I, составная оценка: $e(S^I) = (3; 4, 0, 0)$.

2. Уровень Парето-эффективных решений: $\{S_1^p, S_2^p, S_3^p\}$; составные оценки имеют вид: $e(S_1^p) = (1; 3, 1, 0)$, $e(S_2^p) = (2; 2, 2, 0)$, $e(S_3^p) = (3; 0, 2, 2)$.

3. Следующий уровень качества системы (точки, соседние с Парето-эффективными): $\{S_1', S_2'\}$; составные оценки имеют вид: $e(S_1') = (1; 1, 3, 0)$, $e(S_2') = (2; 0, 1, 3)$.

4. Следующий уровень качества S''; составная оценка имеет вид: $e(S'') = (1; 0, 1, 3)$.

5. Наихудшее решение S_0; составная оценки имеет вид: $e(S_0) = (1; 0, 0, 3)$.

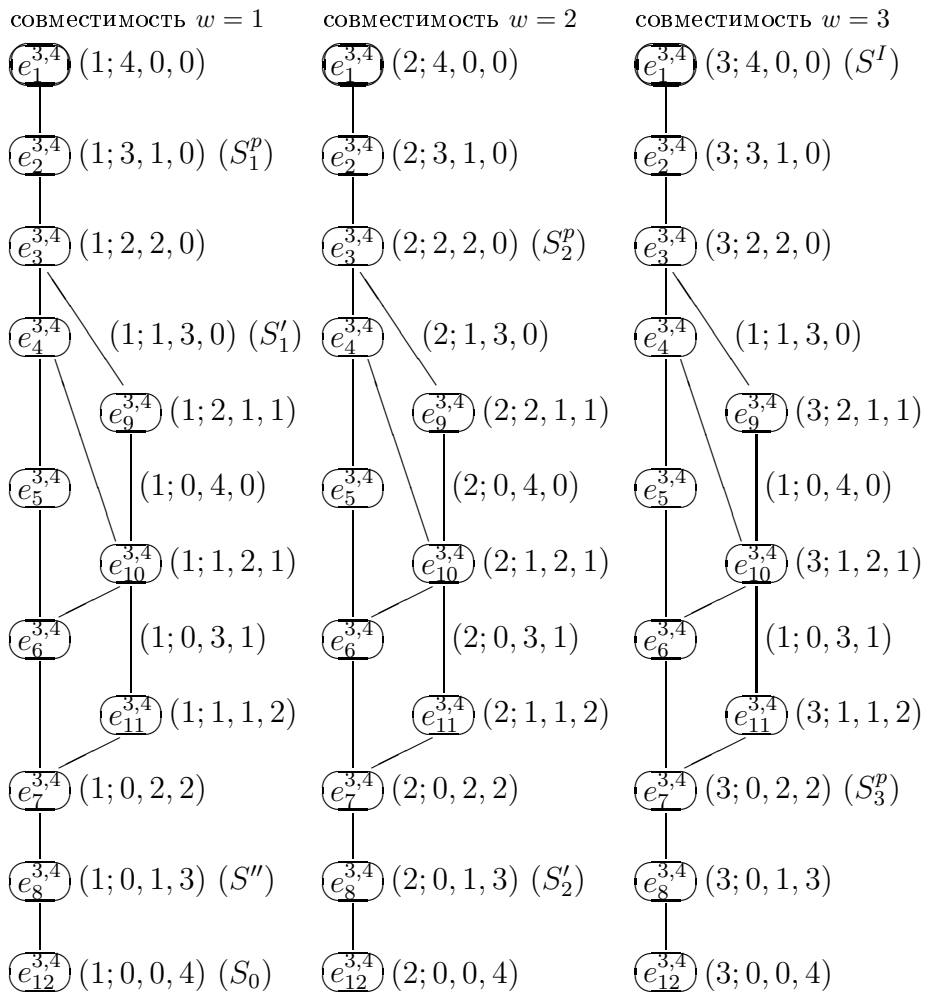


Рис. 3.11. Интегрированная шкала (с совместимостью, $P^{3,4}$)

3.3 Трансформация шкал

В Таблице 3.2 представлены задачи трансформации для базовых типов шкал оценивания (в основном, результирующая порядковая шкала соответствует окончательному решению). Рассматриваются следующие задачи трансформации шкал:

1. *Количественная шкала ⇒ Количественная шкала.*

2. *Количественная шкала ⇒ Порядковая шкала.*

3. *Порядковая шкала ⇒ Порядковая шкала.*

4. *Многокритериальное описание ⇒ Порядковая шкала.* Эта задача соответствует основной задаче в области принятия решений (ordinal decision making, sorting problem) [336, 28, 429, 492, 595].

5. *Шкала в виде частичного порядка ⇒ Порядковая шкала* [336, 344, 360].

6. *Многокритериальное описание ⇒ Количественная шкала.* Эта задача соответствует основной задаче в области принятия решений, когда строится функция полезности [190, 290, 533].

7. *Многокритериальное описание (порядковые шкалы) ⇒ Шкала в виде частичного порядка на основе порядковых оценок.* Здесь часто предполагается использование одинаковых порядковых шкал для всех системных частей/компонентов (или критериев/параметров). Эта задача описана в [336, 344, 349, 358].

8. *Многокритериальное описание (порядковые шкалы) ⇒ Шкала в виде частичного порядка на основе интервальных оценок в виде мультимножеств.* Здесь часто предполага-

ется использование одинаковых порядковых шкал для всех системных частей/компонентов (или критериев/параметров). Эта задача описана в [360].

9. *Шкала в виде частичного порядка на основе интервальных оценок в виде мультимножеств* ⇒ *Шкала в виде частичного порядка на основе интервальных оценок в виде мультимножеств.* Эта задача описана в [360].

10. *Многокритериальное описание* ⇒ *Многокритериальное описание.* (некоторые отображения пространств, например, многокритериальное шкалирование [98]).

Отметим, что указанные типы задач 4, 6, 7, 8 представляют собой интеграцию шкал.

Таблица 3.2. Рассматриваемые задачи трансформации шкал

Исходная шкала	Результирующая шкала				
	Количественная шкала	Порядковая шкала	Многокритериальное описание	Шкала как частичный порядок (порядковые оценки)	Шкала как частичный порядок (интервальные оценки
1.Количественная шкала	⋆	⋆	–	–	–
2.Порядковая шкала	–	⋆	–	–	–
3.Многокритериальное описание (на основе порядковых и/или количественных оценок)	⋆ [190] [290] [533]	⋆ [28, 336] [429, 492] [595]	⋆ [98]	⋆ [336, 344] [349, 358]	⋆ [360]
4.Шкала как частичный порядок (на основе) порядковых оценок)	–	⋆ [336, 344]	–	–	–
5.Шкала как частичный порядок (на основе интервальных оценок в виде мультимножеств)	–	⋆ [360]	–	–	⋆ [360]

Первый тип задачи трансформации шкал (*количественная шкала* ⇒ *количественная шкала*) (Рис. 3.12) может основываться (в простейшем случае) на линейной функции ($y = ax + b$).

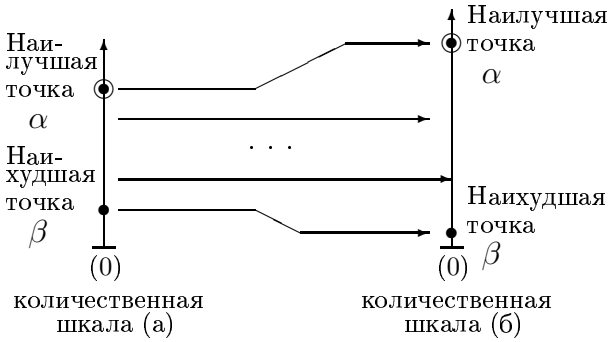


Рис. 3.12. Трансформация количественной шкалы

Второй тип задачи трансформации (*количественная шкала* $\Rightarrow$ *порядковая шкала*) проиллюстрирован на Рис. 3.13. Здесь количественная шкала (или рассматриваемый интервал значений (β, α)) разбивается на множество интервалов, каждый из которых соответствует уровню результирующей порядковой шкалы. Процедура разбиения (т.е., определение границ-порогов) может быть основана на различных подходах (например, вычислительные схемы, экспертные процедуры, использование специально заданных опорных точек) [28, 48, 321, 511].

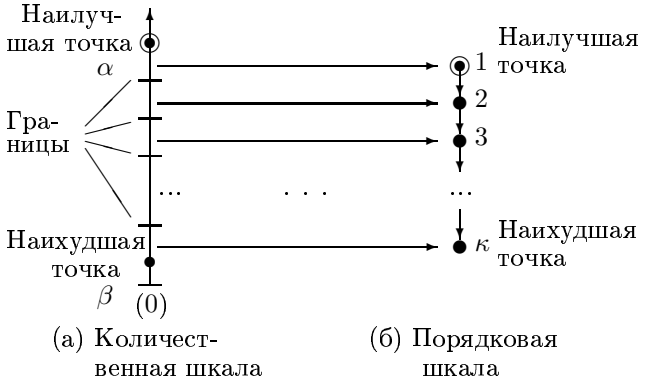


Рис. 3.13. Количественная шкала $\Rightarrow$ порядковая шкала

Два типовых случая для третьего типа задачи трансформации шкал (т.е., отображение) *порядковая шкала* $\Rightarrow$ *порядковая шкала* представлены на Рис. 3.14. Этот тип отображения может быть основан на экспертных суждениях (т.е., на знаниях экспертов к конкретной прикладной области - domain experts).

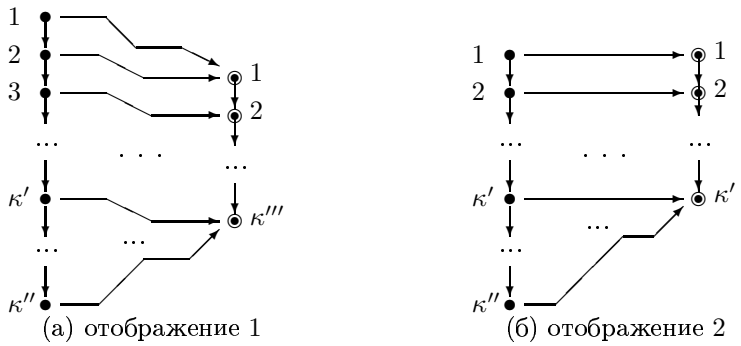


Рис. 3.14. Порядковая шкала $\Rightarrow$ порядковая

Для четвертого типа задачи преобразования шкал (*многокритериальное описание* $\Rightarrow$ *порядковая шкала*), используются следующие подходы: (1) двух-стадийный метод: векторные оценки $\Rightarrow$ значения функции полезности $\Rightarrow$ результирующая оценки на основе порядковой шкалы; (2) последовательное выделений уровней Парето-эффективных решений; (3) последовательное выделение "максимальных" точек; (4) использование разбиения исходного пространства векторных оценок с помощью кривых равного качества (или под-областей равного качества); здесь могут использоваться экспертные процедуры или логические методы (например, [28]) (Рис. 3.15); (5) схемы решения на основе опорных решений (т.е., неких образцов решений); (6) методы порогов несравнимости (ELECTRE, PROMETHEE и др.) (например, [106, 492]); (7) специальные интерактивные процедуры на основе логических методов [28, 48, 511]; (8) использование порядковой шкалы типа D (например, Рис. 3.5в, Рис. 3.6б): (i) идеальное решение; (ii) Парето-эффективные решения; (iii) точки, близкие к Парето-эффективным решениям; (iv) решения, которые находятся "между" точками предыдущего уровня и наихудшим решением; (v) наихудшее решение.

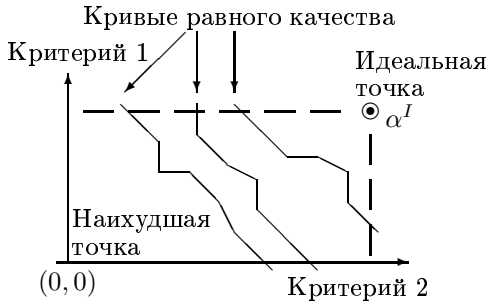


Рис. 3.15. Кривые равного качества

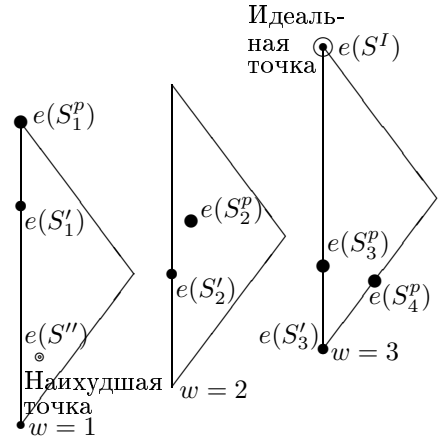


Рис. 3.16. Пример уровней качества

В пятом типе задачи преобразования шкал (*шкала на основе частичного порядка* $\Rightarrow$ *порядковая шкала*), используются методы, аналогичные методам для задачи типа 2, например: последовательное выделение уровней Парето-эффективных решений и т.п.

Для шестого типа задачи преобразования шкал, на Рис. 3.16 представлены уровни качества (порядковая шкала типа D как на Рис. 3.6б): (i) идеальное решение $e(S^I)$, (ii) Парето-эффективные решения (т.е., точки: $\{e(S_1^p), e(S_2^p), e(S_3^p), e(S_4^p)\}$, (iii) точки следующего уровня качества: $\{e(S_1'), e(S_2'), e(S_3')\}$, (iv) точка следующего уровня качества: $e(S'')$, (v) наихудшая точка. Здесь I, $S_i^p (i = \overline{1,4})$, $S_j' (j = \overline{1,3})$, S'' соответствуют версиям системы.

Аналогично для задачи типа 9, общая порядковая шкала качества системы представлена в виде частично-упорядоченного множества на основе интервальных оценок в виде мультимножеств на Рис. 3.11: (i) идеальное решение; (ii) Парето-эффективные решения; (iii) точки следующего уровня качества; (iv) точка следующего уровня качества; (v) наихудшая точка.

3.4 Интеграция шкал и качество систем

Можно перечислить следующие методы интеграции оценок качества компонентов систем и совместимости компонентов систем для получения оценки общего качества системы (т.е., оценивание модульной системы) (Таблица 3.3):

1. *Количественные оценки* $\Rightarrow$ *интегрированная количественная оценка*: (1.1) подход на основе функции полезности (utility function) [190, 290], (1.2) метод аналитических иерархий (AHP) и его модификации [496], (1.3) методы типа TOPSIS (оценивание качества на основе близости к идеальному решению) [319, 516], (1.4) схемы на основе анализа и использования опорных решений, (1.5) гибридные методы.

2. *Количественные оценки и порядковые оценки* $\Rightarrow$ *интегрированная порядковая оценка* (sorting problems) [336, 344, 595]: (2.1) использование порядковой шкалы D [336, 344], (2.2) последовательное выделение Парето-эффективных решений (т.е., уровней Парето) [429, 470], (2.3) последовательное выделение 'максимальных' точек, (2.4) методы порогов несравнимости [106, 492], (2.5) схемы на основе анализа и использования опорных решений, (2.6) гибридные методы [28, 364].

3. *Порядковые оценки* $\Rightarrow$ *интегрированная порядковая оценка* (sorting problems) [336, 344, 595]: (3.1) метод интегрированных таблиц [7, 344], (3.2) человеко-машинные процедуры (expert judgment) для построения границ классов (областей одинакового качества системы) в общей области характеристик системы (т.е., порядковая шкала качества системы) (Рис. 3.17) [28, 321], (3.3) человеко-машинные процедуры, реализующие экспертные и/или логические методы, для построения границ классов (областей одинакового качества системы) в общей области характеристик системы (т.е., порядковая шкала качества

системы) (Рис. 3.17) [28, 48, 511], (3.4) схемы на основе анализа и использования опорных решений, (3.5) гибридные методы [28, 364].

4. *Порядковые оценки ⇒ интегрированная оценка на основе шкалы в виде частичного порядка* [336, 344, 358]: (4.1) вычисление интегрированной оценки на основе шкалы в виде частичного порядка, (4.2) использование экспертных суждений для построения интегрированной оценки на основе шкалы в виде частичного порядка.

5. *Оценки на основе шкалы в виде частичного порядка Poset-like estimates ⇒ интегрированная оценка на основе шкалы в виде частичного порядка* [360]: (5.1) интегрированная оценка, (5.2) оценка медианного типа, (5.3) использование экспертных суждений.

6. *Векторные оценки ⇒ интегрированная векторная оценка*: (6.1) объединение исходных многокритериальных областей ('пространств'), (6.2) простая интеграция исходных многокритериальных областей (например, суммирование по компонентам векторных оценок), (6.3) специальные отображения.

Таблица 3.3. Подходы к интеграции оценок компонентов

Методы	Шкалы для системных компонентов	Шкала для качества системы	Тип интеграции	Источники
1.Анализ полезности TOPSIS, AHP	Количественные	Количественная	Функция полезности, TOPSIS, AHP	[190, 290, 319] [496, 516]
2.Интегрирующие таблицы	Порядковая	Порядковая	Иерархические интегрирующие таблицы	[7, 344]
3.Парето-подход	Количественные, порядковые	Порядковая	Выделение уровня Парето	[429, 470]
4.Метод порогов несравнимости (ELECTRE, PROMETHEE)	Количественные, Порядковые	Порядковая	Выделение доминирующих точек	[106, 492]
5.Уровень 'максимальных' ('минимальных') элементов	Количественные, порядковые	Порядковая	Выделение 'максимальных' и/или 'минимальных') элементов	
6.Интерактивная процедура (экспертные суждения)	Порядковые	Порядковая	Границы классов для разбиения векторной области	[321]
7.Интерактивная процедура (экспертные и логические методы)	Порядковые	Порядковая	Разделяющие границы классов для векторных областей	[48, 28, 511]
8.Объединение областей измерения	Векторные оценки	Векторная оценка	Интегрирование областей (медиана, консенсус)	[336, 344, 360]
9.ИММП с порядковыми оценками	Порядковые оценки	Частичный порядок на основе порядковых оценок	Выделение уровня Парето	[336, 344, 349] [358]
10.ИММП с интервальными оценками в виде мультимножеств	Интервальные оценки в виде мультимножеств	Частичный порядок на основе интервальных оценок в виде мультимножеств	(а) Оценка интегрированная или медианная (б) Уровень Парето	[360]

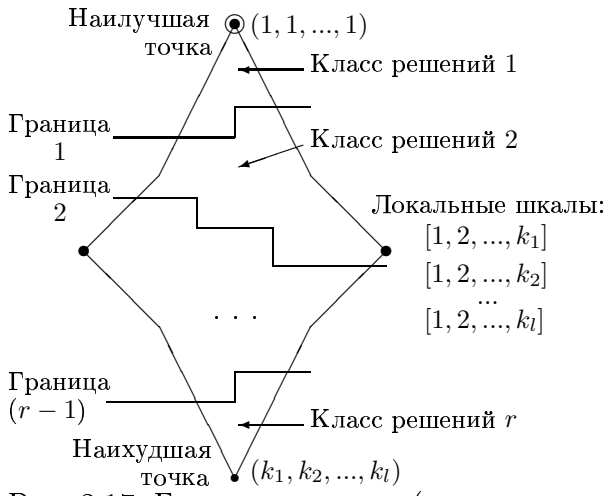


Рис. 3.17. Границы классов (порядковое качество)

С точки зрения специалиста в прикладной области (domain expert), может быть целесообразно проиллюстрировать два метода:

(а) интегрированные таблицы (Рис. 3.18, Рис. 3.19; численные примеры для системы: интегрированные таблицы и иерархический процесс оценивания системы),

(б) метод TOPSIS (Рис. 3.20; иллюстрация для “расширенного” метода в случае использования нескольких идеальных точек.

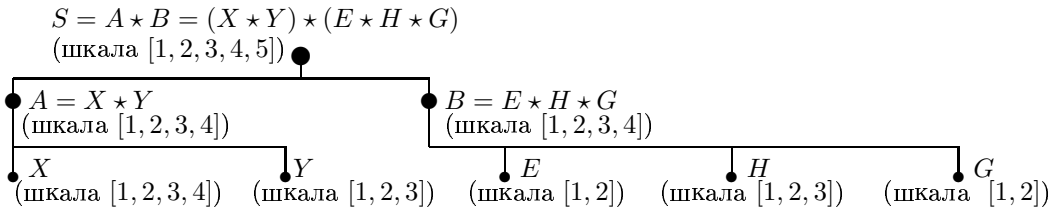


Рис. 3.18. Пример системы, шкал для компонентов

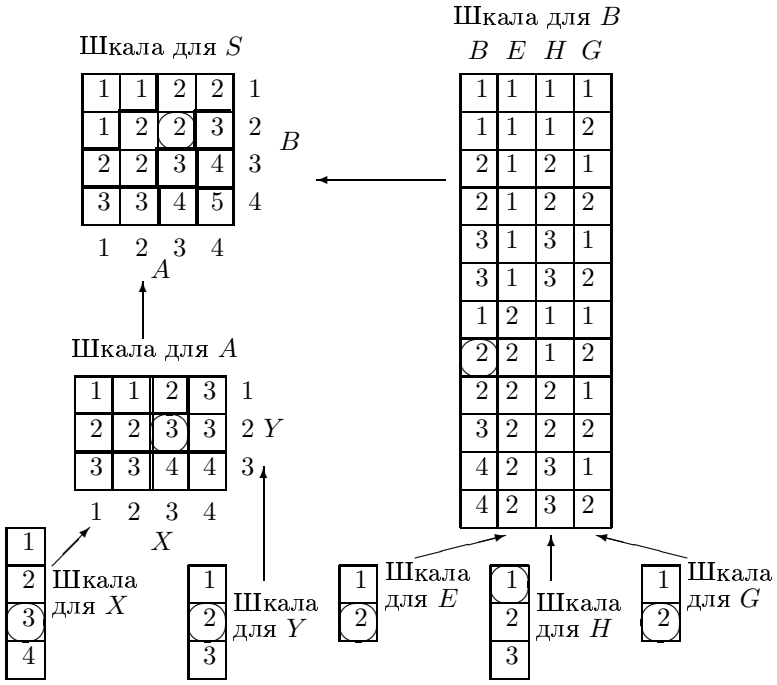


Рис. 3.19. Интеграция шкал на основе таблиц

В базовой версии метода TOPSIS трансформация исходной векторной оценки альтернатив в финальную шкалу качества основывается на упрощении задачи за счет использования близости (обычно - расстояния) от оценки альтернативы до идеальной точки. В

общем случае, можно упорядочивать альтернативы на основе вектора $\rho = (\rho^-, \rho^+)$, где ρ^+ соответствует близости к наилучшей точке (или точкам) и ρ^- соответствует близости к наихудщей точке (или точкам).

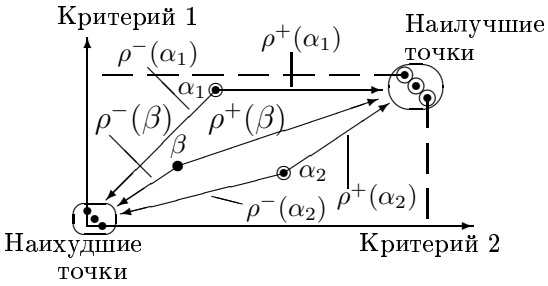


Рис. 3.20. Иллюстрация для метода типа TOPSIS

3.5 Числовые примеры

Здесь представлен пример для четырех-компонентной студенческой бригады (Рис. 3.21) (оценки совместимости компонентов не рассматриваются). В Таблице 3.4 приведены исходные оценки проектных альтернатив (DAs) для компонентов рассматриваемой системы на основе четырех типов шкал:

(i) количественные оценки (шкала $(1, 3)$, 1 соответствует наилучшему уровню);

(ii) векторные (двух-элементные) порядковые оценки (шкала (x, y), $(1, 1)$ соответствует наилучшему уровню, например, x - это оценка "математике", y - это оценка "физике");

(iii) порядковые оценки (шкала $[1, 2, 3]$, 1 соответствует наилучшему уровню);

(iv) интервальные оценки в виде мультимножеств (используется задача оценки типа $P^{3,4}$, Рис. 3.9).

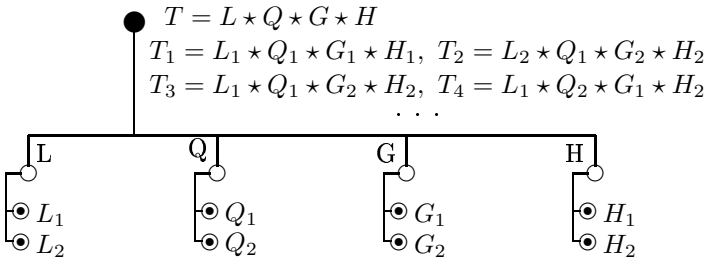


Рис. 3.21. Пример четырех-компонентной бригады

Table 3.4. Исходные данные

DA	Количественные оценки (шкала $(1, 3)$)	Векторные оценки (x, y)	Порядковые оценки (шкала $[1, 2, 3]$)	Интервальные оценки в виде мультимножеств (задача $P^{3,4}$)
L_1	1.5	$(2, 1)$	1	$(3, 1, 0)$
L_2	1.8	$(2, 2)$	2	$(0, 4, 0)$
Q_1	1.1	$(1, 1)$	1	$(4, 0, 0)$
Q_2	2.7	$(2, 3)$	3	$(0, 3, 1)$
G_1	1.2	$(1, 1)$	1	$(3, 1, 0)$
G_2	2.4	$(3, 2)$	2	$(1, 2, 1)$
H_1	1.4	$(1, 2)$	1	$(2, 2, 0)$
H_2	3.1	$(3, 3)$	3	$(0, 2, 2)$

Следующие иллюстративные примеры представлены:

Пример 3.1. Количественные оценки DAs интегрируются на основе простейшей аддитивной функции (функция полезности) (Рис. 3.22): $f(T_1) = 1.5 + 1.1 + 1.2 + 1.4 = 5.2$

(наилучшее решение), $f(T_2) = 1.8 + 1.1 + 2.4 + 3.1 = 8.4$, $f(T_3) = 1.5 + 1.1 + 2.4 + 3.1 = 8.1$, $f(T_4) = 1.5 + 2.7 + 1.2 + 3.1 = 8.5$; соответствующее отношение предпочтения имеет вид: $T_1 \succ T_3 \succ T_2 \succ T_4$.

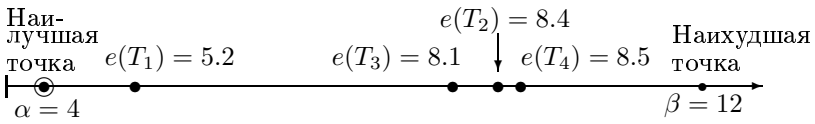


Рис. 3.22. Результирующая количественная шкала

Пример 3.2. Локальные порядковые оценки DAs интегрируются в результирующую оценку для модульного решения (используется метод интегрирующих таблиц, Рис. 3.23):

$\{e(T_1) = 1\}$, $\{e(T_2) = 4\}$, $\{e(T_3) = 3\}$, $\{e(T_4) = 3\}$.

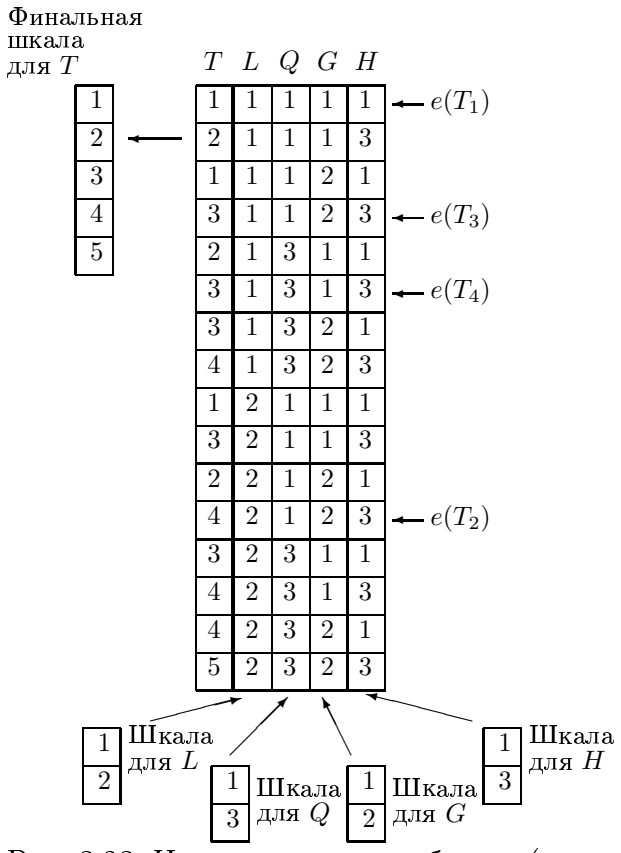


Рис. 3.23. Интегрирующие таблицы (пример 3.2)

Пример 3.3. Векторные (двух-компонентные) оценки интегрируются в результирующую порядковую оценку: (1) суммирование (по каждому компоненту векторных оценок) (т.е., для T_1,T_2,T_3,T_4), (2) выделение Парето-эффективных решений (Рис. 3.24):

(а) векторные оценки: $e(T_1) = (5, 5)$, $e(T_2) = (9, 8)$, $e(T_3) = (9, 7)$, $e(T_4) = (8, 8)$;

(б) доминирование (предпочтения): $T_1 \succ T_2$, $T_1 \succ T_3$, $T_1 \succ T_4$, $T_3 \succ T_2$, $T_4 \succ T_2$.

(в) результирующая порядковая шкала типа D: Парето-эффективное решение (уровень 1): $\{T_1\}$; следующий уровень качества (уровень 2): $\{T_3, T_4\}$; следующий уровень качества (уровень 3): $\{T_2\}$.

Таким образом получаются приоритеты: $r(T_1) = 1$, $r(T_2) = 3$, $r(T_3) = 2$, $r(T_4) = 2$.

Пример 3.4. Порядковые оценки DAs трансформируются в оценку на основе частичного порядка (Рис. 3.25), выделение Парето-эфективных решений:

(а) оценки на основе частичного порядка: $n(T_1) = (4, 0, 0)$, $n(T_2) = (1, 2, 1)$, $n(T_3) = (2, 1, 1)$, $n(T_4) = (2, 1, 1)$;

(б) доминирование (предпочтения): $T_1 \succ T_2$, $T_1 \succ T_3$, $T_1 \succ T_4$, $T_3 \succ T_2$, $T_4 \succ T_2$;

(в) результирующая порядковая шкала типа D: Парето-эффективное решение (уровень 1): $\{T_1\}$, следующий уровень качества (уровень 2): $\{T_3, T_4\}$, следующий уровень качества (уровень 3): $\{T_2\}$.

Таким образом получаются приоритеты: $r(T_1) = 1$, $r(T_2) = 3$, $r(T_3) = 2$, $r(T_4) = 2$.

Пример 3.5. Интервальные оценки в виде мультимножеств для DAs (используется шкала частичного порядка с Рис. 3.9) трансформируются (на основе поиска медиан) в результирующую интервальную оценку в виде мультимножества (используется шкала частичного порядка с Рис. 3.9), выделение Парето-эффективных решений:

(а) интервальные оценки в виде мультимножеств: $n(T_1) = (3, 1, 0)$, $n(T_2) = (0, 4, 0)$, $n(T_3) = (1, 3, 0)$, $n(T_4) = (1, 3, 0)$;

(б) доминирование (предпочтения): $T_1 \succ T_2$, $T_1 \succ T_3$, $T_1 \succ T_4$, $T_3 \succ T_2$, $T_4 \succ T_2$;

(в) результирующая порядковая шкала типа D: Парето-эффективное решение (уровень 1): $\{T_1\}$, следующий уровень качества (уровень 2): $\{T_3, T_4\}$, следующий уровень качества (уровень 3): $\{T_2\}$.

Таким образом получаются приоритеты: $r(T_1) = 1$, $r(T_2) = 3$, $r(T_3) = 2$, $r(T_4) = 2$.

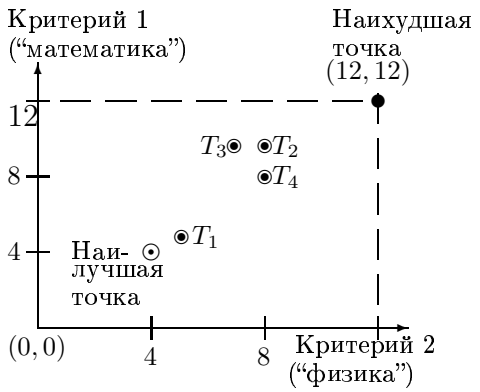


Рис. 3.24. Векторная 'область'

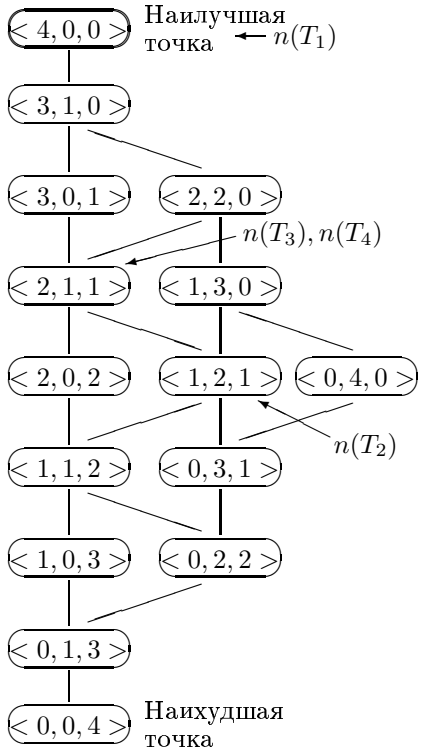


Рис. 3.25. Порядок $n(T) = (\eta_1, \eta_2, \eta_3)$

3.6 Выводы к главе

Данная глава была посвящена описанию основных методов оценивания модульных систем. Можно указать несколько актуальных в будущем направления исследований в рассмотренной области:

1. исследование много-стадийных процедур трансформации шкал;

2. рассмотрение различных прикладных задач;

3. использование опорных решений;

4. учет неопределенности;

5. анализ связей между задачами оценивания и базовыми задачами теории принятия решений.

Глава 4

Методы комбинаторного синтеза

В данной главе [1] приведено описание задачи комбинаторного синтеза модульной системы (как выбор вариантов для частей системы) и краткое описание основных методов комбинаторного синтеза: (а) морфологический анализ; (б) методы на основе многокритериального анализа (метод близости к идеальной точке, метод выделения Парето-эффективных решений); (в) задача блочного рюкзака, включая версию на основе интервальных оценкок в виде мультимножеств (разработан автором); (г) задача о назначениях; (д) иерархическое морфологические проектирование (разработан автором) на основе порядковых оценок и на основе интервальных оценкок в виде мультимножеств.

4.1 Предварительные сведения

Морфологический анализ (МА) был предложен Ф. Цвики (F. Zwicky) в 1943 году для проектирования космических устройств [598]. Этот метод проектирования представляет собой известное мощное средство для синтеза модульных систем в различных областях [64, 278, 336, 344, 486, 487, 598]. МА основан на приеме "разделяй и властвуй" (*divide and conquer*). Метод основан на иерархической структуре проектируемой системы. Разделение системы на части осуществляется на основе следующих подходов: (а) по системным компонентам/частям, (б) по системным функциям. Здесь системная иерархия на основе компонентов (подсистем, частей) используется как базовая. Такой подход соответствует модульным системам в различных областях (техника, информационные технологии, управление) [72, 259, 316, 336, 344, 467, 561]. Многие годы применение МА в проектировании систем было существенно ограничено вследствие необходимости исследования огромного комбинаторного проектных множества комбинаций. С другой стороны, современные компьютерные системы позволяют решать сложные комбинаторные задачи и иерархические модели систем позволяют использовать схемы решения на основе разбиения/декомпозиции систем. Это приводит к расширению области использования методов проектирования на основе МА. Современные тренды в области использования и модификации МА могут быть

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Springer, 1998. chapter 3.
(ii) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapter 3.
(iii) Levin M.Sh. Towards morphological system design. *7th IEEE Int. Conf. on Industrial Informatics 'INDIN 2009'*, UK, 95-100, 2009.
(iv) Levin M.Sh. Morphological methods for design of modular systems (a survey). Electr. prepr., 20 p., Jan. 9, 2012. Http://arxiv.org/abs/1201.1712 [cs.SE]
(v) Levin M.Sh. Multiset estimates and combinatorial synthesis. Electr. prepr., 30 p., May 9, 2012. http://arxiv.org/abs/1205.2046 [cs.SY]
(vi) Levin M.Sh. System synthesis with morphological clique problem: fusion of subsystem evaluation decisions. Inform. Fusion, 2001, Vol. 2, No. 3, pp. 225–237.

основаны на следующем: (1) иерархическое моделирование систем, (2) оптимизация, (3) многокритериальное принятие решений, (4) учет неопределенности.

В качестве основной проектной задачи (т.е., построения конфигурации модульной системы) рассматривается выбор альтернатив для частей системы (Рис. 4.1). Предполагается, что составная (модульная) система состоит из m системных частей:

$$\{P(1), ..., P(i), ..., P(m)\}.$$

Для каждой части (т.е., $\forall i,\ i = \overline{1, m}$) имеются альтернативы (проектные варианты DAs) $\{X_1^i, X_2^i, ..., X_{q_i}^i\}$, где q_i - число альтернатив для части i. Модель проектирования имеет вид:

Выбрать альтернативу для каждой части системы с учетом локальных и глобальных целей/предпочтений и ограничений.

Рис. 4.1. Конфигурация системы (выбор)

Некоторые другие задачи конфигурации модульных систем описаны в [349].

Список рассматриваемых проектных подходов к комбинаторному синтезу включает следующее:

(1) базовая версия морфологического анализа (по Ф. Цвики) (МА) (т.е., поиск всех допустимых по совместимости комбинаций элементов: один элемент для каждой части системы, [83, 97, 278, 486, 598];

(2) модификация морфологического анализа в виде поиска допустимых (по совместимости) комбинаций элементов и выбора комбинации, которая наиболее “близка” к комбинации из лучших элементов для каждой части системф (т.е., “идеальной” комбинации) [9, 64];

(3) модификация морфологического анализа посредством редукции к задаче линейного программирования (МА&линейное программирование) [13];

(4) модификация морфологического анализа посредством редукции к задаче блочного рюкзака (MCP) [351, 397, 530];

(5) модификация морфологического анализа посредством редукции к задаче квадратичного программирования (QAP) [120, 336, 349];

(6) многокритериальная модификация морфологического анализа с использованием выделения лучших результирующих комбинаций элементов как Парето-эффективных решений (МА&Парето метод): (а) поиск всех допустимых (по совместимости элементов) комбинаций (один представитель для каждой части системы), (б) оценивание полученных комбинаций элементов по набору критериев и выбор Парето-эффективных решений [6, 10];

(7) иерархическое морфологическое многокритериальное проектирование (ИММП) [336, 344, 358];

(8) версия ИММП с использованием интервальных оценок в виде мультимножеств [360].

Характеристики основных методов представлены в Таблице 4.1.

Таблица 4.1. Краткие характеристики методов

Метод	Шкала для DAs	Шкала для IC	Качество решения	Источники
1.MA	Нет	$\{0,1\}$	Допустимость	[278, 486, 598]
2.Близость к идеальной точке	Нет	$\{0,1\}$	Близость к идеальной точке	[9, 64]
3.MA&линейное программирование	Количественная	$\{0,1\}$	Аддитивная функция	[13]
4.Задача блочного рюкзака	Количественная	Нет	Аддитивная функция	[397, 530]
5.QAP	Количественная	Количественная	Аддитивная функция	[120, 349]
6.MA&Парето подход	Нет	$\{0,1\}$	Многокритериальная	[6, 10]
7.ИММП	Порядковая	Порядковая	Многокритериальная	[336, 344]
8.ИММП с интервальными оценками в виде мультимножеств	Интервальная в виде мультимножеств	Порядковая	Многокритериальная	[360]

4.2 Описание основных подходов

4.2.1 Морфологический анализ

Метод MA включает следущие стадии:

Стадия 1. Построение структуры системы как множество компонентов (частей) системы.

Стадия 2. Генерация проектных альтернатив DAs (вариантов) для каждой части системы (множество DAs для части системы (компонента) называется морфологическим классом).

Стадия 3. Оценивание совместимости в виде Булевых оценок для каждой пары DAs (одна альтернатива для одного морфологического класса, другая альтернатива для другого морфологического класса). Оценка совместимости 1 соответствует совместимости проектных альтернатив, оценки 0 соответствует несовместимости альтернатив.

Стадия 4. Генерация всех допустимых композиций (комбинаций) (т.е., одна альтернатива для каждого компонента системы) с учетом совместимости всех пар альтернатив.

Этот метод является переборным. Рис. 4.2 иллюстрирует MA (двоичные оценки совместимости содержатся в Таблице 4.2).

В примере указаны следующие морфологические классы (а) морфологический класс 1: $\{X_1^1, X_2^1, X_3^1, X_4^1, X_5^1\}$, (б) морфологический класс i: $\{X_1^i, X_2^i, X_3^i, X_4^i, X_5^i\}$, (в) морфологический класс m: $\{X_1^m, X_2^m, X_3^m\}$. Приводится упрощенный случай для трех частей системы. Результирующее решение (допустимое) в виде композиции проектных альтернатив имеет вид: $S_1 = X_2^1 \star ... \star X_3^i \star ... \star X_1^m$.

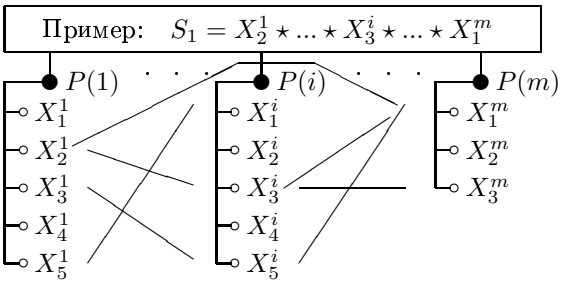


Рис. 4.2. Иллюстрация для МА

Таблица 4.2. Оценки совместимости

	X_1^i	X_2^i	X_3^i	X_4^i	X_5^i	X_1^m	X_2^m	X_3^m
X_1^1	0	0	0	0	0	0	0	0
X_2^1	0	0	1	0	0	1	0	0
X_3^1	0	0	0	0	1	0	0	0
X_4^1	0	0	0	0	0	0	0	0
X_5^1	1	0	0	0	0	0	0	0
X_1^i						0	0	0
X_2^i						0	0	0
X_3^i						1	0	1
X_4^i						0	0	0
X_5^i						1	0	0

4.2.2 Метод близости к идеальной точке

Метод близости к идеальной точке основан на анализе "близости" получающихся допустимых решений к некому идеальному решению (идеальной точке) [64, 9]. На Рис. 4.3 представлен иллюстрирующий пример для данного метода (двоичные оценки совместимости содержатся в Таблице 4.3).

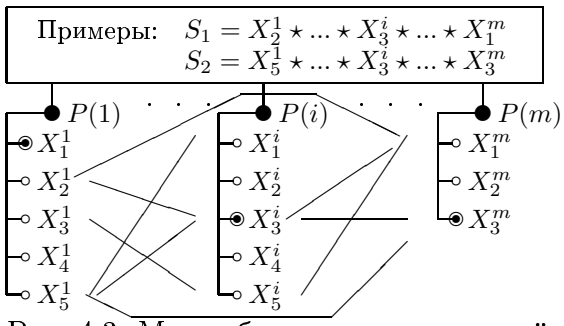


Рис. 4.3. Метод близости к идеальной точке

Таблица 4.3. Оценки совместимости

	X_1^i	X_2^i	X_3^i	X_4^i	X_5^i	X_1^m	X_2^m	X_3^m
X_1^1	0	0	0	0	0	0	0	0
X_2^1	0	0	1	0	0	1	0	0
X_3^1	0	0	0	0	1	0	0	0
X_4^1	0	0	0	0	0	0	0	0
X_5^1	1	0	1	0	0	0	0	1
X_1^i						0	0	0
X_2^i						0	0	0
X_3^i						1	0	1
X_4^i						0	0	0
X_5^i						1	0	0

В рамках данного метода в каждом морфологическом классе определяется лучшая проектная альтернатива (например, на основе вычислений, на основе экспертных суждений) и их комбинация считается идеальным решением (идеальной точкой). В примере (Рис. 4.3) в качестве лучших альтернатив в морфологических классах используются следующие: X_1^1, X_3^i, and X_3^m. Идеальное решение имеет вид: $S_o = X_1^1 \star ... \star X_3^i \star ... \star X_3^m$. Но это решение является недопустимым по совместимости Можно рассмотреть допустимые решения: $S_1 = X_2^1 \star ... \star X_3^i \star ... \star X_1^m$ и $S_2 = X_5^1 \star ... \star X_3^i \star ... \star X_3^m$.

Пусть $\rho(S', S'')$ - близость (по элементам) для двух составных проектных решений $S', S'' \in \{S\}$. Тогда целесообразно искать следующее решение $S^* \in \{S^a\} \subseteq \{S\}$ ($\{S^a\}$ - множество допустимых решений): $S^* = Arg\ \min_{S \in \{S^a\}} \rho(S, S_o)$. В иллюстративном примере решение $S_2 = X_5^1 \star ... \star X_3^i \star ... \star X_3^m$ является более близким к идеальному решению S_o (т.е., $\rho(S_2, S_o) \preceq \rho(S_1, S_o)$). В общем случае можно исследовать различные версии близости [9, 64].

4.2.3 Морфологический метод с выбором по Парето

Интегрированный подход на основе МА и многокритериального принятия решений был предложен в виде следующей схемы [10, 6]:

Стадия 1. Использование МА для построения допустимых композиций элементов (допустимых по совместимости элементов решений).

Стадия 2. Генерация критериев для оценивания полученных допустимых решений.

Стадия 3. Оценивание полученных решения по критериям и выбор Парето-эффективных решений.

Данный поход иллюстрирован на основе примера на Рис. 4.4 (двоичные оценки совместимости содержатся в Таблице 4.4). Следующие допустимые решения построены: $S_1 = X_2^1 \star ... \star X_3^i \star ... \star X_1^m$, $S_2 = X_5^1 \star ... \star X_3^i \star ... \star X_3^m$, $S_3 = X_5^1 \star ... \star X_5^i \star ... \star X_3^m$. Эти решения оцениваются по критериям для выделения Парето-эффективных решений.

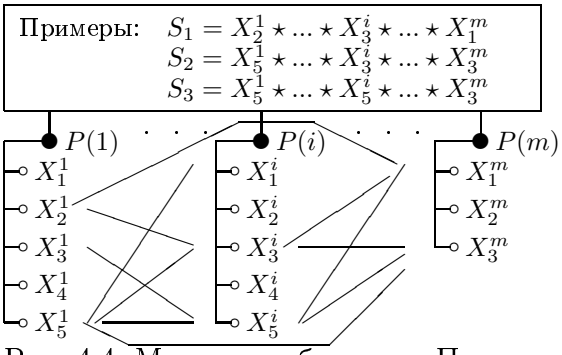


Рис. 4.4. Метод с выбором по Парето

Таблица 4.4. Оценки совместимости

	X_1^i	X_2^i	X_3^i	X_4^i	X_5^i	X_1^m	X_2^m	X_3^m
X_1^1	0	0	0	0	0	0	0	0
X_2^1	0	0	1	0	0	1	0	0
X_3^1	0	0	0	0	1	0	0	0
X_4^1	0	0	0	0	0	0	0	0
X_5^1	1	0	1	0	1	0	0	1
X_1^i						0	0	0
X_2^i						0	0	0
X_3^i						1	0	1
X_4^i						0	0	0
X_5^i						1	0	1

4.2.4 О применении линейного программирования

В статье [13] метод МА редуцируется к задаче линейного программирования с Булевыми переменными. Получаемые ограничения имеют вид неравенств с Булевыми переменными и целевая функция имеет вид аддитивной функции. Далее используется эвристическая схема решения (в виде переборного алгоритма).

4.2.5 Задача блочного рюкзака

Базовая версия задачи о рюкзаке имеет вид [207, 291, 419]:

$$\max \sum_{i=1}^{m} c_i x_i \quad s.t. \ \sum_{i=1}^{m} a_i x_i \le b, \ x_i \in \{0,1\}, \ i = \overline{1,m},$$

где $x_i = 1$ при выборе элемента i, c_i - "полезность") элемента i, a_i - вес элемента i (требуемый ресурс). Обычно используются неотрицательные коэффициенты. Задача относится к классу NP-трудных [207, 419] и для ее решения применяются переборные алгоритмы (например, метод ветвей и границ, динамическое программирование), приближенные алгоритмы с ограниченной относительной погрешностью (FPTAS) [291, 419]. В случае задачи блочного рюкзака [291, 419], исходное множество элементов разбивается на группы и производится выбор элементов из каждой группы с учетом общего ограничения по ресурсу. Каждый элемент имеет два индекса: (i,j), где i - номер группы, j - номер элемента в группе. Задача блочного рюкзака имеет вид:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij} x_{ij} \qquad s.t. \ \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{ij} x_{ij} \le b; \ \sum_{j=1}^{q_i} x_{ij} \le 1 \ i = \overline{1,m}, \ x_{ij} \in \{0,1\}.$$

При многокритериальном описании каждого элемента (т.е., (i,j)) имеется векторная "полезность" $\overline{c_{i,j}} = (c_{i,j}^1, ..., c_{i,j}^{\xi}, ..., c_{i,j}^r)$ и многокритериальная задача блочного рюкзака имеет

вид:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{ij}^{\xi} x_{ij}, \ \forall \xi = \overline{1,r} \qquad s.t. \ \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b, \ \ \sum_{j=1}^{q_i} x_{ij} = 1 \ \forall i = \overline{1,m}, \ \ x_{ij} \in \{0,1\}.$$

Очевидно, что для этой задачи целесообразно искать Парето-эффективные решения. Такой подход использовался для проектирования и улучшения ряда прикладных систем (программные комплексы, электроника, коммуникации) [351, 354, 397, 400, 530]. Можно указать следующие схемы решения для приведенной задачи: (i) переборные алгоритмы (например, метод ветвей и границ, динамическое программирование), (ii) эвристика на основе предварительного ранжирования элементов для получения их приоритетов и пошаговой упаковки элементов в рюкзак (жадный алгоритм), (iii) многокритериальное ранжирование элементов для получения порядковых приоритетов и испольщование приближенной схемы решения на основе дискретной области качества решения (как будет определено далее для ИММП).

Следует отметить, что в данном подходе совместимость элементов не рассматривается.

4.2.6 Задачи назначения/размещения

Задачи назначения/размещения широко используются в многих прикладных областях [120, 207, 469]. Простейшая задача о назначениях включает неотрицательную матрицу соответствий $\Upsilon = ||c_{i,j}||$ $(i = \overline{1,n}, \ j = \overline{1,n})$, где $c_{i,j}$ - 'полезность' назначения элемента i на позицию j. Задача имеет вид [207]:

Найти назначение $\pi = (\pi(1), ...\pi(i), ..., \pi(n))$ *элементов* i $(i = \overline{1,n})$ *на позиции* $\pi(i)$, *которое соответствует обобщенной эффективности:* $\sum_{i=1}^{n} c_{i,\pi(i)} \to \max$.

Задача квадратичного назначения (QAP) дополнительно учитывает взаимосвязи между элементами различных групп (каждая группа элементов соовтествует определенной позиции) [120, 469]. Пусть неотрицательное значение $d(i, j_1, k, j_2)$ обозначает "полезность" совместимости между элементом j_1 в группе J_i и элементом j_2 в группе J_k. Указанный тип совместимости добавляется к целевой функции. Задача QAP может рассматриваться как версия MA:

$$\max \sum_{i=1}^{m} \sum_{j=1}^{q_i} c_{i,j} x_{i,j} + \sum_{l<k} \sum_{j_1=1}^{q_l} \sum_{j_2=1}^{q_k} d(l, j_1, k, j_2) \ x_{l,j_1} x_{k,j_2}, \ \ l = \overline{1,m}, \ \ k = \overline{1,m};$$

$$s.t. \ \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{i,j} \ x_{i,j} \leq b, \ \ \sum_{j=1}^{q_i} x_{i,j} \leq 1 \ \forall i = \overline{1,m}, \ \ x_{i,j} \in \{0,1\}.$$

Задача QAP относится к классу NP-трудных. Обычно используются следующие методы решения: переборные методы (например, метод ветвей и границ) или эвристики (например, жадные алгоритмы, генетические алгоритмы). В случае многокритериальной задачи о назначении целевая функция трансформируется в векторную целевую функцию, т.е., $c_{i,j} \ \Rightarrow \overline{c_{i,j}} = (c_{i,j}^1, ..., c_{i,j}^{\xi}, ..., c_{i,j}^r)$, и результирующая целевая функция может иметь вид:

$$(\sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j}^1 x_{i,j}, ..., \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j}^{\xi} x_{i,j}, ..., \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j}^r x_{i,j} \ ).$$

Очевидно, что здесь целесообразно искать Парето-эффективные решения. Аналогично, задача QAP может быть трансформирована в многокритериальную задачу QAP.

4.3 Иерархическое морфологическое проектирование

Базовая версия иерархического морфологического многокритериального проектирования ИММП представлена в [330, 333, 336, 344, 349, 358]. Предполагается следующее:

(а) древовидная модель системы;

(б) составная оценки качества системы, которая интегрирует качество компонентов и качество связей между компонентами IC (совместимость);

(в) монотонные критерии для системы и ее компонентов;

(г) качество системных компонентов и IC оцениваются на основе координированных шкал.

Используются следующие обозначения:

(1) проектные альтернативы (DAs) для висячих (терминальных) вершин системной модели;

(2) приоритеты DAs ($\iota = \overline{1,l}$; 1 соответствует наилучшему уровню);

(3) порядковая совместимость для каждой пары DAs ($w = \overline{1,\nu}$; ν соответствует наилучшему уровню).

ИММП включает следующие фазы:

Фаза 1. Построение древовидной модели системы (предварительная фаза).

Фаза 2. генерация DAs для висячих вершин системной модели.

Фаза 3. Иерархический выбор DAs и их композиция составные DAs для соответствующего более высокого иерархического уровня системной иерархии (задача морфологической клики).

Фаза 4. Анализ и улучшение результирующих составных DAs (решений).

Рис. 4.5 и Рис. 4.6 иллюстрируют общую схему ИППМ.

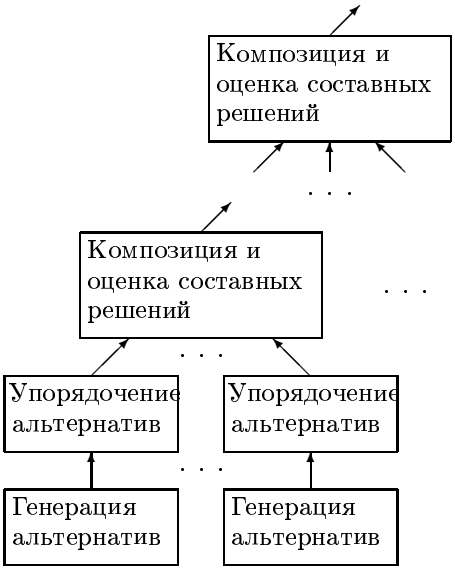


Рис. 4.5. Схема 'снизу-вверх'

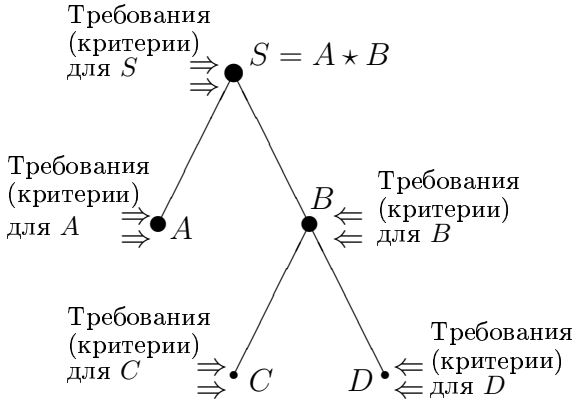


Рис. 4.6. Иерархия требований

Далее подробнее рассматривается задача морфологической клики. Пусть система S состоит из m частей (компонентов): $\{P(1), ..., P(i), ..., P(m)\}$ (Рис. 4.1). Для каждой системной части генерируется множество проектных альтернатив DAs. Задача имеет вид:

Найти составную проектную альтернативу $S = S(1) \star ... \star S(i) \star ... \star S(m)$ *из* DAs *(один представитель* $S(i)$ *для каждого системного компонента* $P(i)$, $i = \overline{1,m}$*) с ненулевой оценкой совместимости между выбранными проектными альтернативами.*

В ИММП используется дискретная область ("пространство") как шкала качества системы на основе на вектора: $N(S) = (w(S); e(S))$, где $w(S)$ - минимум парной совместимости между DAs, которая соответствует различным системным компонентам (т.е., $\forall\ P_{j_1}$ и P_{j_2}, $1 \leq j_1 \neq j_2 \leq m$) в S, $e(S) = (\eta_1, ..., \eta_\iota, ..., \eta_l)$, где η_ι - число DAs ι-го качества в S ($\sum_{\iota=1}^{l} \eta_\iota = m$). Рис. 4.7 иллюстрирует шкалу (частичный порядок, решетку) качества для составного решения S по элементам $e(S)$ (3 части системы и 3 уровня качества элементов DAs $[1, 3]$).

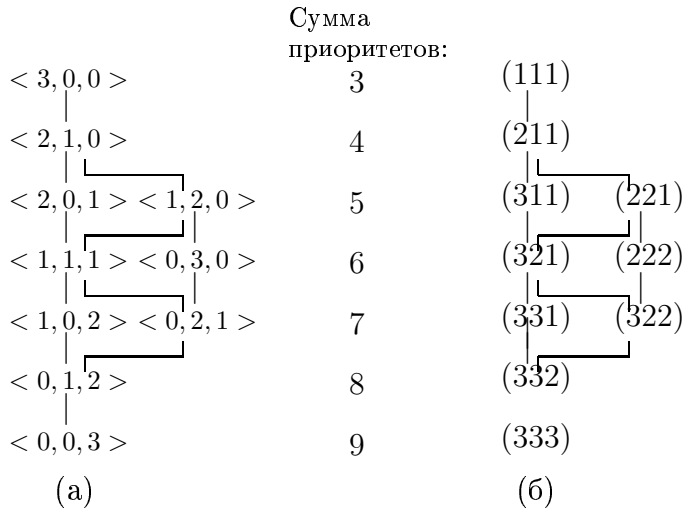


Рис. 4.7. Два представления решетки качества $e(S)$

На Рис. 4.8 представлена решетка качества системного решения S с учетом совместимости по вектору $N(S)$ (аналогично: 3 части системы, 3 уровня качества элементов DAs $[1, 3]$; шкала для совместимости $[1, 3]$).

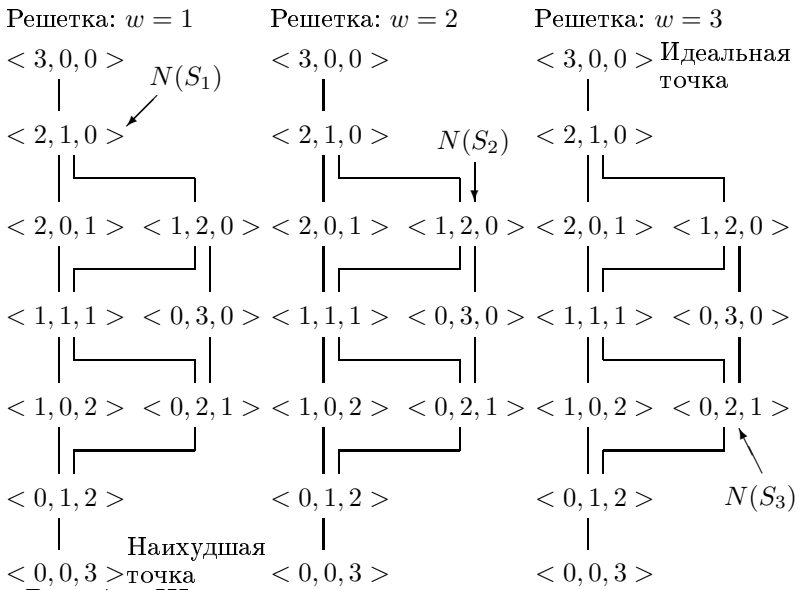


Рис. 4.8. Шкала качества как частичный порядок

При этом производится поиск составных решений, которые недоминируются по вектору $N(S)$ (т.е., Парето-эффективные решения). Таким образом, базовая версия задачи морфологической клики в базовой версии ИММП соответствует следующей оптимизационной модели (две целевые функции, одно ограничение):

$$\max\ e(S), \quad \max\ w(S), \qquad s.t.\ w(S) \geq 1.$$

"Максимизация" $e(S)$ основана на использовании соответствующего частичного порядка.

Эта задача относится к классу NP-трудных (поскольку ее более простая подзадача является NP-трудной [304]).

Очевидна схема решения для указанной задачи включает два этапа (переборный алгоритм):

ШАГ 1. Построить допустимые композиции.
ШАГ 2. Отобрать Парето-эффективные составные DAs (по $N(S)$).

Можно рассматривать эвристические приближенные схемы решения [336]. Например, схема решения может быть направлена на исследование допустимых решений, начиная с идеальной точки и близких к ней решений (с отсевом недопустимых решений, которые на являются морфологическими кликами). Кроме того, автором был программно реализован алгоритм на основе динамического программирования [338]. Дополнительное следует отметить, что эксперт может участвовать на всех стадиях процесса решения.

Для допустимых решений можно ввести некую обобщенную порядковую шкалу в виде уровней качества системы [336]: (i) идеальная точка; (ii) Парето-эффективные точки; (iii) соседняя область по отношению Парето-эффективным точкам (составные решения из этой области-множества могут быть трансформированы в Парето-эффективную точку на основе неких простых улучшающих акций).

Метод ИММП иллюстрируется на Рис. 4.9 и Рис. 4.10 на основе численного примера для системы их трех частей $S = X \star Y \star Z$. Приоритеты DAs представлены на Рис. 4.9 в круглых скобках и указаны на Рис. 4.10. Таблица 4.5 содержит оценки совместимости (эти оценки указаны на Рис. 2.7). В примере, получаются составные Парето-эффективные решения (Рис. 4.8, Рис. 4.9, Рис. 4.10, Рис. 4.11):

$S_1 = X_2 \star Y_2 \star Z_2, \;\; N(S_1) = (1; 2, 1, 0);$
$S_2 = X_1 \star Y_2 \star Z_2, \;\; N(S_2) = (2; 1, 2, 0);$
$S_3 = X_1 \star Y_1 \star Z_3, \;\; N(S_3) = (3; 0, 2, 1).$

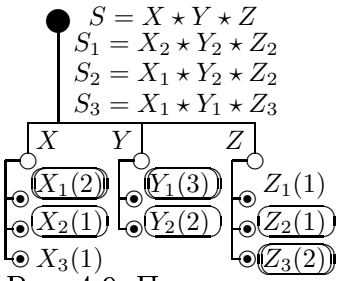


Рис. 4.9. Пример композиции

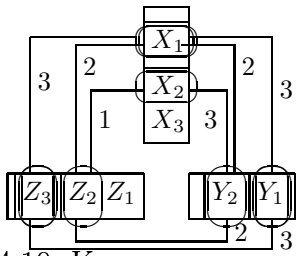


Рис. 4.10. Концентрическое представление

Таблица 4.5. Совместимость

	Y_1	Y_2	Z_1	Z_2	Z_3
X_1	3	2	0	2	3
X_2	0	3	0	1	0
X_3	0	0	0	0	1
Y_1			0	0	3
Y_2			0	2	0

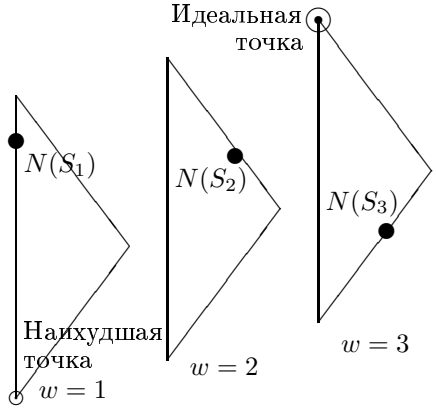


Рис. 4.11. Шкала качества системы

4.4 Комбинаторный синтез с интервальными оценками

Мультимножества исследуются в ряде работ, например: [36, 585]. Интервальные оценки в виде мультимножеств были предложены автором в [360]. Данный подход заключается в назначении элементов $(1, 2, 3, ...)$ на уровни порядковой шкалы $[1, 2, ..., l]$. В результате получается оценки в виде мультимножества, где базовое множество включает уровни порядковой шкалы: $\Omega = \{1, 2, ..., l\}$ (уровни шкалы являются линейно упорядоченными: $1 \succ 2 \succ 3 \succ ...$) и задача оценивания (для каждого объекта, альтернативы) заключается в выборе мультимножества над множесвом Ω с учетом двух условий:

1. мощность полученного мультимножества равна заданному числе элементов $\eta = 1, 2, 3, ...$ (рассматриваются мультимножества мощности η);

2. "конфигурация" мультимножества имеет вид: отобранные элементы Ω покрывают интервал на шкале $[1, l]$ (т.е.,. "интервальная оценка в виде мультимножества").

Таким образом, оценка e для альтернативы A имеет вид: $e(A) = (\eta_1, ..., \eta_\iota, ..., \eta_l)$, где η_ι - число элементов на уровне ι ($\iota = \overline{1, l}$), или $e(A) = \{\overbrace{1, ..., 1}^{\eta_1}, \overbrace{2, ...2}^{\eta_2}, \overbrace{3, ..., 3}^{\eta_3}, ..., \overbrace{l, ..., l}^{\eta_l}\}$. Число мультимножеств мощности η с элементами из исходного конечного множества мощности l, называется "коэфициент мультимножества" [303, 585]: $\mu^{l,\eta} = \frac{l(l+1)(l+2)...(l+\eta-1)}{\eta!}$. Это число соответствует числе возможных оценок (без условия 2). В случае выполнения условия 2 число оценок несколько меньше. Обозначим задачу оценивания на основе введенных интервальных оценок в виде мультимножеств: $P^{l,\eta}$. Шкала в виде частично-упорядоченного множества для задачи оценивания $P^{3,4}$ представлена на Рис. 4.12.

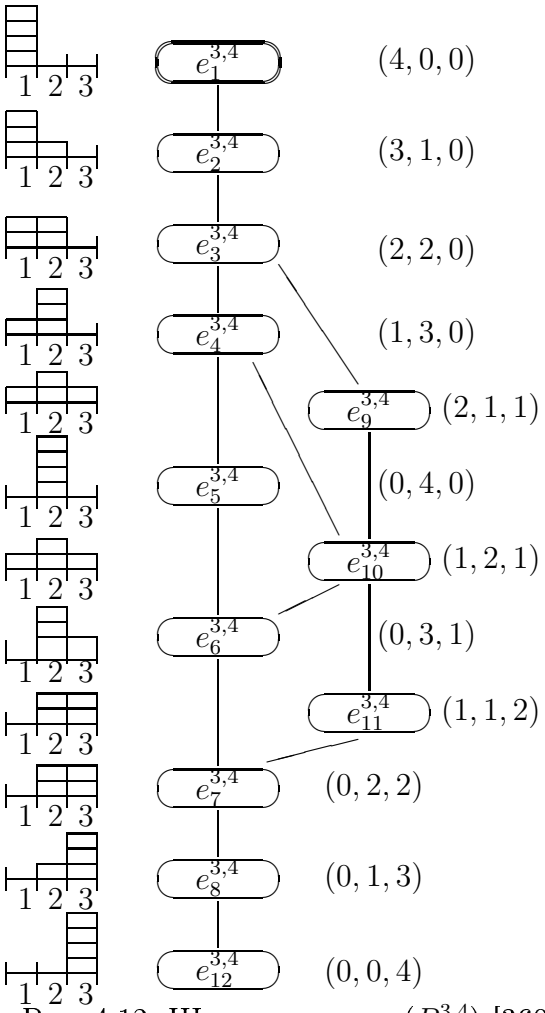


Рис. 4.12. Шкалы, оценки ($P^{3,4}$) [360]

Дополнительно, следующие операции над введенными оценками в виде мультимножеств могут быть рассмотрены [360]: интеграция, близость (в виде вектора), агрегация, выравнивание. В данной работе будут использованы операции интеграции, агрегации и близости.

Интеграция оценок основана на суммирования по компонентам Пусть имеется n оценок: оценка $e^1 = (\eta_1^1, ..., \eta_\iota^1, ..., \eta_l^1)$, ..., оценка $e^\kappa = (\eta_1^\kappa, ..., \eta_\iota^\kappa, ..., \eta_l^\kappa)$, ..., $e^n = (\eta_1^n, ..., \eta_\iota^n, ..., \eta_l^n)$. Тогда интегрированная оценка имеет вид: $e^I = (\eta_1^I, ..., \eta_\iota^I, ..., \eta_l^I)$, где $\eta_\iota^I = \sum_{\kappa=1}^n \eta_\iota^\kappa \ \ \forall \iota = \overline{1,l}$. Фактически, используется операция $\uplus$: $e^I = e^1 \uplus ... \uplus e^\kappa \uplus ... \uplus e^n$.

Теперь рассмотрим близость оценок для двух адьтернатив A_1, A_2 с соответствующими оценками $e(A_1), e(A_2)$. Близость определяется так: $\delta(e(A_1), e(A_2)) = (\delta^-(A_1, A_2), \delta^+(A_1, A_2))$, где компоненты вектора имеют вид (i) δ^- - число одношаговых изменений элемента $\iota + 1$ в элемент ι ($\iota = \overline{1, l-1}$) (это соответствует "улучшению"); (ii) δ^+ - число одношаговых изменений элемента ι в элемент $\iota + 1$ ($\iota = \overline{1, l-1}$) (это соответствует "ухудшению"). Предполагается следующее: $|\delta(e(A_1), e(A_2))| = |\delta^-(A_1, A_2)| + |\delta^+(A_1, A_2)|$.

Теперь можно рассмотреть построение оценки-медианы (операция агрегации). Пусть имеется исходное множество оценок $E = \{e_1, ..., e_\kappa, ..., e_n\}$, пусть D - множество всех возможных оценок ($E \subseteq D$). Тогда оценка-медиана имеет вид (два случая: "обобщенная медиана" M^g и "медиана множества" M^s):

$M^g = \arg\min_{M \in D} \ \sum_{\kappa=1}^n \ |\delta(M, e_\kappa)|; \quad M^s = \arg\min_{M \in E} \ \sum_{\kappa=1}^n \ |\delta(M, e_\kappa)|$.

При морфологическом проектировании (метод ИММП) с интервальными оценками в виде мультимножеств интервальные оценки используются для оценивания проектных альтернатив DAs. Такой же вид оценки используется для результирующей системы $S = S(1) \star ... \star S(i) \star ... \star S(m)$: агрегированная оценка ("обобщенная медиана") на основе соответствующих оценок компонентов, входяших в решение. Таким образом, $N(S) = (w(S); e(S))$, где $e(S)$ - "обобщенная медиана" на основе оценок компонентов решения. Модифицированная задача (на основе использования оценок в виде мультимножеств) имеет вид:

$$\max \ e(S) = M^g = \arg\min_{M \in D} \ \sum_{i=1}^m \ |\delta(M, e(S_i))|, \quad \max \ w(S),$$

$$s.t. \quad w(S) \geq 1.$$

Кроме того, можно использовать интервальные оценки и в задаче блочного рюкзака. Базовая задача блочного рюкзака имеет вид (раздел 4.2.5):

$$\max \sum_{i=1}^m \sum_{j=1}^{q_i} c_{ij} x_{ij} \qquad s.t. \ \ \sum_{i=1}^m \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b; \ \ \sum_{j=1}^{q_i} x_{ij} \leq 1 \ i = \overline{1,m}, \ \ x_{ij} \in \{0,1\}.$$

При использовании оценок в виде мультимножеств для оценивания "полезности" элементов $e_i, i \in \{1, ..., i, ..., m\}$ (вместо c_i), следующая агрегированная оценка в виде мультимножества может рассматривается в виде целевой функции ("максимизация"): (а) агрегированная оценка в виде мультимножества как "обобщенная медиана", (б) агрегированная оценка в виде мультимножества как "медиана множества", (в) интегрированная оценка в виде мультимножества.

Рассмотрим следующий вариант блочной задачи о рюкзаке:

(1) оценки в виде мультимножеств для "полезности" элементов $e_{i,j}$, $i \in \{1, ..., i, ..., m\}, j = \overline{1, q_i}$ (вместо c_{ij}),

(2) агрегированная оценка как "обобщенная медиана" (или "медиана множества") используется как целевая функция ("максимизация").

Пусть имеется исходное множество элементов:

$\{(1,1), (1,2), ..., (1,q_1), ..., (i,1), (i,2), ..., (i,q_i), ..., (m,1), (m,2), ..., (m,q_m)\}$.

Булева переменная $x_{i,j}$ соответствует выбору элемента (i,j). Решение представляет собой подмножество исходного множества элементов: $S = \{(i,j)|x_{i,j} = 1\}$. Задача имеет вид:

$$\max\ e(S) = \max\ M = \ \arg\min_{M \in D} \sum_{(i,j) \in S = \{(i,j)|x_{i,j}=1\}} |\delta(M, e_{i,j})|,$$

$$s.t.\ \ \sum_{i=1}^{m} \sum_{j=1}^{q_i} a_{ij} x_{i,j} \leq b; \ \ \sum_{j=1}^{q_i} x_{ij} = 1; \ \ x_{ij} \in \{0,1\}.$$

Очевидно, эта задача похожа на рассмотренную выше постановку, но без учета совместимости отобранных элементов.

4.5 Выводы к главе

В данной главе приведены краткие описания основных подходов к комбинаторному синтезу. Значительное внимание уделено методам на основе морфологического анализа, включая разработанное автором иерархическое морфологическое многокритериальное проектирование (базовая версия и версия с интервальными оценками в виде мультимножеств). Таблица 4.6 содержит сравнение упрощенных качественных характеристик для указанных основных методов (на основе мнения автора).

Таблица 4.6. Качественное сравнение методов

Метод	Вычислительная сложность	Учет совместимости компонентов	Удобство для выбора решений	Удобство для экспертов
1.МА	Высокая	Да, бинарная	Трудно	Трудно
2.МА & Близость к идеальной точке	Высокая	Да, бинарная	Просто	Хорошо
3.МА & выбор по Парето	Высокая	Да, бинарная	Средне, анализ Парето-эффективых решений	Хорошо
4.Задача блочного рюкзака	Низкая/средняя	Нет	Просто	Средне
5.QAP	Высокая	Да	Просто	Средне
6.ИММП	Низкая/средняя	Да, порядковая	Просто	Хорошо
7.ИММП с интервальными оценками в виде мультимножеств	Средняя	Да, порядковая	Средне	Хорошо

Глава 5

Схемы выявления системных узких мест

В данной главе [1] приведено описание подходов к выделению узких мест в иерархических модульных систем.

5.1 Предварительные сведения

В данной главе предполагается следующее (Рис. 5.1) [373]:

1. Исследуемая модульная система имеет модель в виде морфологической структуры [336, 344, 358, 365] или сети.

2. В качестве узких мест системы (system bottlenecks) рассматриваются следующие системные элементы: (i) компоненты системы (или system component fault), (ii) взаимосвязи между системными компонентами (совместимость), (iii) группа системных компонентов (или group of system faults), (iv) группа взаимосвязанных системных компонентов (или composite system faults). Таким образом, системные узкие места рассматриваются как следующее: (а) системный компонент или часть системы низкого качества, (б) сбой системного компонента/части системы (system component fault), (в) композиция указанных объектов.

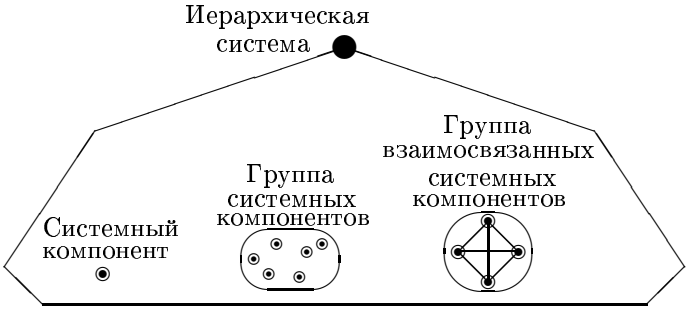


Рис. 5.1. Система и системные узкие места

Рассматриваемые подходы к выделению системных узких мест приведены в Таблице 5.1:

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. chapter 3.
(ii) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapter 4.
(iii) M.Sh. Levin, Towards Detection of Bottlenecks in Modular Systems. Electr. prepr., 12 p., June 1, 2013. http://arxiv.org/abs/1306.0128 [cs.AI]
(iv) M.Sh. Levin, Towards clique-based fusion of graph streams in multi-function system testing. Informatica (Lith.), 23(3), 2012, pp. 391-404.

I. Базовые подходы к управлению качеством: (1.1.) Парето-диаграмма [265], (1.2.) многокритериальный анализ в виде выделения Парето эффективных точек и/или многокритериальное ранжирование [18, 344].

II. Выделение системных частей низкого качества: (2.1.) выделение критических компонентов/модулей, (2.2.) выделение критической взаимосвязи между компонентами (совместимость между компонентами), (2.3.) анализ структуры системы и выявление ситуации, когда структура системы должна быть улучшена.

III. Выбор группы связанных системных компонентов как составного системного сбоя (например, посредством ИММП [336, 344, 358, 360], посредством интеграции информации на основе клики [362, 386]).

IV. Выделение критических объектов в сетях: (4.1) выделение критических компонентов в сетях, (4.2) выявление связи между узлами низкого качества, (4.3) выявление сетевой топологии низкого качества, например: (а) определение базовых внутренних узлов сети на основе задачи покрывающего дерева с максимальным числом висячих вершин [51, 119, 207], (б) определение базовых внутренних узлов сети на основе задачи "connected dominating sets problem" [93, 119, 207], (4.4) определение критических объектов в сети на основе задачи "hierarchical network design problem" [71, 153, 479].

V. Предсказательное выявление системных узких мест: (5.1.) предсказательное выявление системных компонентов на основе прогноза их параметров, (5.2.) предсказательное выявление критических компонентов сети (узлы низкого качества, связи узлов низкого качества, топология сети низкого качества), (5.3.) предсказательное выявление группы связанных системных компонентов: интеграция данных на основе клики над потоками графов [362].

Таблица 5.1. Подходы к выделению системных узких мест

Рассматриваемые объекты	Базовые методы/модели выделения	Предсказательное выделение (прогноз/динамика)
1.Системный компонент, сбой системного компонента	(а) Парето-диаграмма [265] (б) многокритериальный анализ/ранжирование (sorting) [18, 344]	(а) Парето-диаграмма (прогноз параметров системных компонентов) (б) многокритериальный анализ/ранжирование (sorting) на основе прогноза параметров
2.Взаимосвязи системных компонентов	(а) Парето-диаграмма [265] (б) многокритериальный анализ/ранжирование (sorting)	(а) Парето-диаграмма на основе прогноза параметров системных компонентов (б) многокритериальный анализ/ранжирование (sorting) на основе прогноза параметров
3.Группа системных компонентов или составной сбой	Многокритериальный анализ (sorting) [18, 364, 492, 595]	Многокритериальный анализ (sorting) на основе прогноза параметров сети [18, 364, 492, 595]
4.Критические узлы в сетях	(а) покрывающее дерево с максимальным числом терминалов [51, 119, 207] (б) connected dominating set [93, 119, 207] (в) иерархическое проектирование сети [71, 153, 479] (г) связь узлов низкого качества	(а) покрывающее дерево с максимумом терминалов на основе прогноза (б) connected dominating set (на основе прогноза) (в) иерархическое проектирование сети (на основе прогноза) (г) связь узлов низкого качества (на основе прогноза сети)
5.Группа связанных системных компонентов	(а) ИММП [336, 344, 358, 360] (б) интеграция данных на основе клики [362, 386]	(а) ИММП на основе прогноза системы [336, 344, 358, 360] (б) интеграция данных на основе клики для потоков графов [362]

5.2 Традиционные методы управления качеством

Можно указать два традиционных метода выявления узких мест в рамках управления качеством: (i) метод Парето диаграммы, (ii) многокритериальный анализ (ранжирование). Первый метод является основным в рамках японской системы управлением качества и заключается в анализе надежности компонентов исследуемой системы (или частоты сбоев/отказов) [265]:

Шаг 1. Определение исходного множества системных компонентов/частей для анализа.

Шаг 2. Оценивание надежности системных компонентов (или частоты сбоев/отказов).

Шаг 3. Выбор ненадежных системных комопнентов в качестве узких мест на основе порогов в Парето диаграмме.

Схема многокритериального анализа основана на анализе вектора параметров системных компонентов [18, 344]:

Шаг 1. Определение исходного множества системных компонентов/частей для анализа.

Шаг 2. Оценивание системных компонентов оп критериям.

Шаг 3. Многокритериальное ранжирование системных компонентов (например, выделение Парето-эффективных элементов, использование методов порогов несравнимости, анализ на основе методы функции полезности) для выбора наиболее важных компонентов с точки зрения общей надежности системы.

Эти два метода иллюстрируются на примере нагнетателя для газо-перекачивающих агрегатов [18]. Древовидная структура данного устройства имеет вид (Рис. 5.2):

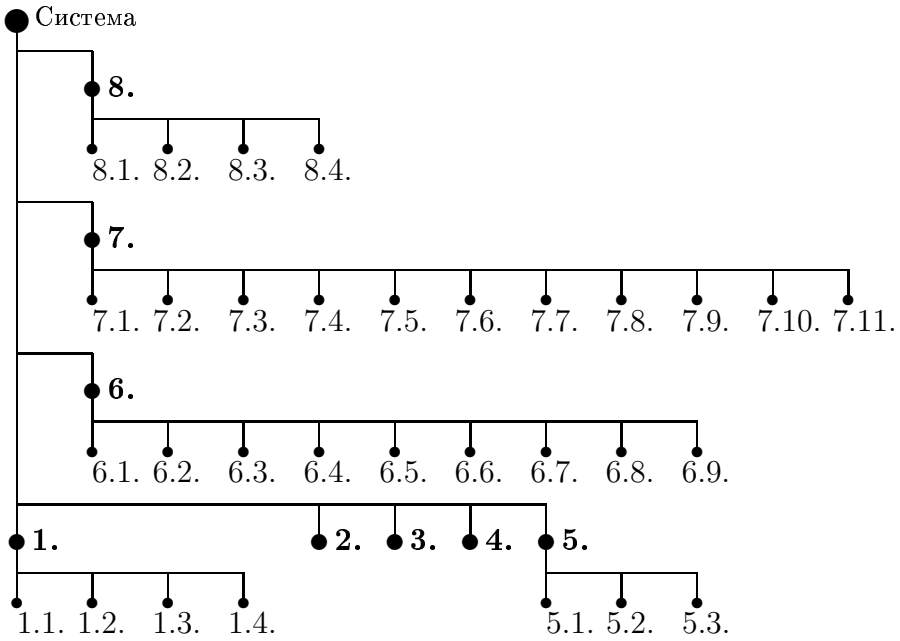


Рис. 5.2. Структура исследуемого устройства

1. Детали: *1.1.* внешний корпус, *1.2.* крышка корпуса, *1.3.* внутренний корпус с закладными элементами, *1.4.* уплотнения корпуса.

2. Поддерживающие блоки.

3. Уплотнения щелевые масляные.

4. Ротор.

5. Узлы соединения нагнетателей с приводом: *5.1.* полумуфта, *5.2.* зубчатая обойма, *5.3.* торционный вал.

6. система смазки, *6.1.* маслобак, *6.2.* масляные фильтры, *6.3.* основной масло-насос, *6.4.* пусковой масло-насос, *6.5.* арматура, *6.6.* клапаны, *6.7.* регулятор температуры, *6.8.* масло-охладители, *6.9.* вентиляторы для масло-охладителей.

7. Система масляных уплотнителей: *7.1.* масло-бак, *7.2.* масло-фильтр, *7.3.* главный насос, *7.4.* пусковой насос, *7.5* регулятор давления, *7.6.* гидро-аккумулятор, *7.7.* дегазатор, *7.8.* масло-отводчик, *7.9.* трубопровод, *7.10.* клапаны, *7.11.* уплотнительные резиновые кольца,

8. Упорные подшипники: *8.1.* размерный блок для упорного подшипника, *8.2.* wrapper rings, *8.3.* ограничительные кольца, *8.4.* дистанционные кольца.

Для оценивания системных компонентов используются шесть критериев (локальные, системные): *1.* C_1, частота отказов (в процентах); *2.* C_2, время ремонта в случае отказа компонента; *3.* C_3, стоимость ремонта компонента; *4.* C_4, уровень влияния отказа компонента на другие системные компоненты, шкала $[0, 1, 2]$ (нет влияния: 0, влияние существует: 1, сильное влияние: 2); *5.* C_5, масштаб использования компонента, шкала $[0, 1, 2]$ (компонент применяется только в данном устройстве: 0, компонент применяется также в других устройствах: 1, компонент широко применяется в различных устройствах: 2); *6.* C_6, уровень влияния отказа компонента на общую безопасность системы, шкала $[0, 1]$ (нет влияния: 0, влияние существует: 1).

Таблица 5.2 содержит многокритериальное описание исследуемой системы (т.е., оценки компонентов по указанным шести критериям), предполагается использование статистических данных, обработки информации, экспертных суждений [18].

Таблица 5.2. Оценки системных компонентов

Часть системы/ компонент	Оценки по критериям					
	C_1	C_2	C_3	C_4	C_5	C_6
1.1.	0.25	12.8	18.0	0	0	1
1.2.	0.00	12.8	18.0	0	0	1
1.3.	0.87	12.8	18.0	1	0	0
1.4.	0.25	12.8	18.0	0	0	1
2.	1.53	6.4	5.5	2	0	0
3.	0.30	9.6	6.4	1	0	0
4.	6.80	12.8	18.0	1	0	0
5.1	0.00	4.8	5.9	1	0	0
5.2	0.16	4.8	5.9	1	0	0
5.3	0.94	3.2	3.7	1	0	0
6.1	0.10	1.6	1.2	0	0	0
6.2	0.50	1.6	1.2	1	0	0
6.3	5.60	3.2	4.0	1	0	0
6.4	0.81	3.2	3.1	1	0	0
6.5	0.35	0.8	1.4	1	1	0
6.6	0.35	0.8	1.4	1	1	0
6.7	0.20	0.8	1.4	1	1	0
6.8	1.50	28.8	48.7	1	1	0
6.9	0.70	1.6	2.5	1	1	0
7.1	0.00	1.6	1.2	0	0	0
7.2	0.35	1.6	1.2	1	0	0
7.3	0.00	0.8	1.4	1	0	0
7.4	0.20	3.2	3.1	1	0	0
7.5	1.50	2.4	2.0	1	2	0
7.6	0.00	0.8	1.9	1	0	0
7.7	1.50	1.6	2.9	0	0	0
7.8	1.40	2.4	2.0	1	0	0
7.9	0.70	0.8	1.4	0	0	0
7.10	0.20	0.8	1.4	1	0	0
7.11	70.00	0.8	1.4	2	0	0
8.	0.70	3.2	1.4	2	0	0
8.1	0.20	3.2	1.4	2	0	0
8.2	0.00	3.2	1.4	2	0	0
8.3	0.00	3.2	1.4	2	0	0
8.4	0.00	3.2	1.4	2	0	0

Выбранные на основе Парето диаграммы узкие места (системные компоненты) представлены на Рис. 5.3 (использованы оценки по критерию C_1):

(а) порог 1 (6.8), выбранные системные компоненты: 4, 7.11;

(б) порог 2 (1.5), выбранные системные компоненты: 2, 4, 6.3, 6.8, 7.5, 7.7, 7.11.

Многокритериальное ранжирование (sorting problem) направлено на выделение узких мест как наиболее важных системных компонентов при использовании всех критериев [18, 28, 364, 492, 595]. Здесь для ранжирования применена версия метода типа ELECTRE [28, 364, 492]. При этом использованы веса критериев: 1.0 (C_1), 0.3 (C_2), 0.4 (C_3), 0.5 (C_4), 0.2 (C_5), 3.0 (C_6). Результаты ранжирования представлены на Рис. 7.4:

Уровень 1 (системные узкие места): 2, 4, 6.3, 6.8, 7.11.

Уровень 2: 1.3, 5.3, 6.4, 7.5, 7.8.

Уровень 3: 1.1, 1.4, 3, 5.1, 5.2, 6.5, 6.6, 6.9, 7.4, 7.7, 8.1, 8.

Уровень 4: другие компоненты.

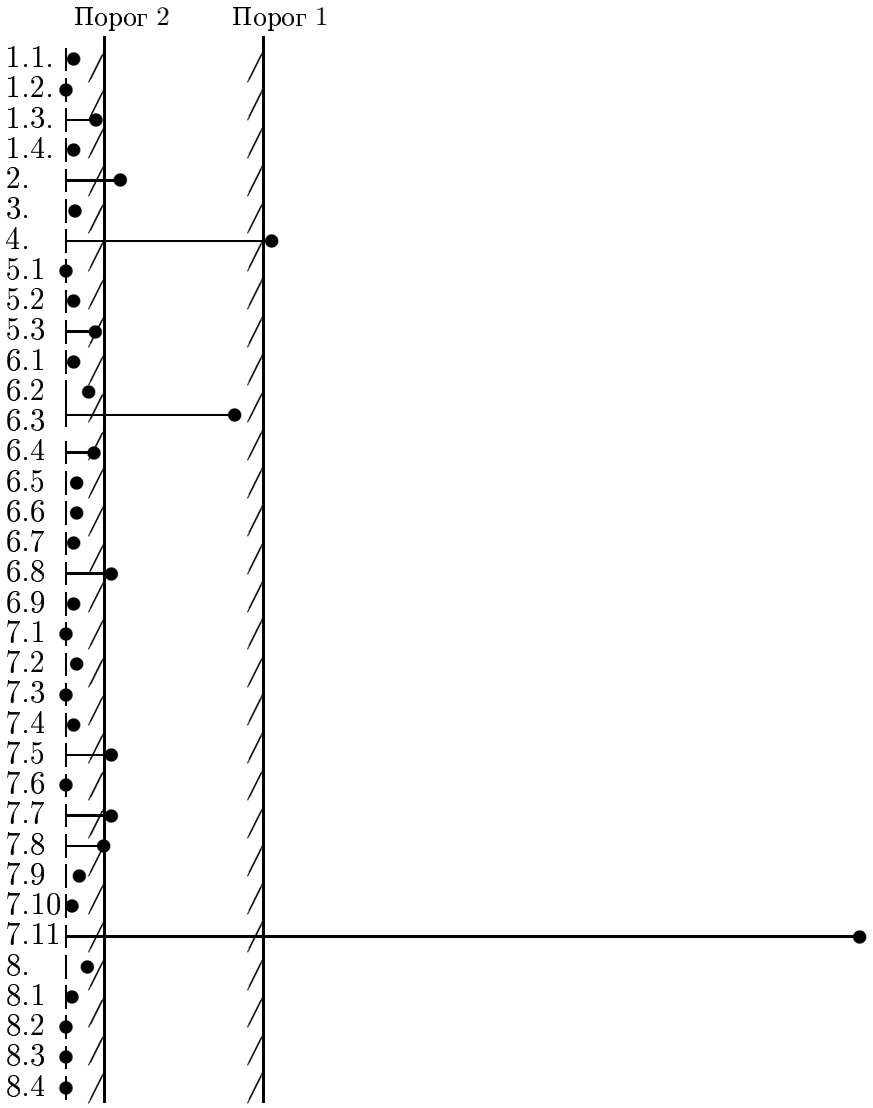


Рис. 5.3. Схема на основе Парето-диаграммы

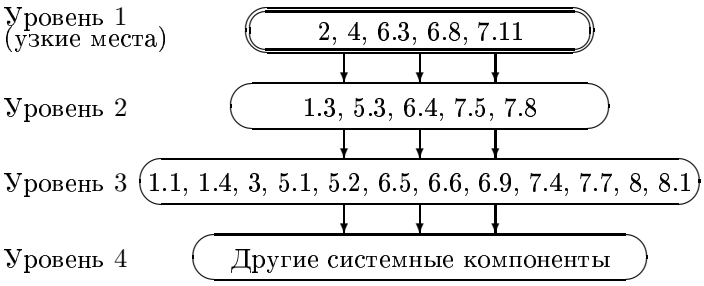


Рис. 5.4. Результаты многокритериального ранжирования

5.3 Узкие места при морфологическом проектировании

В методе ИММП результирующее составное решение формируется из проектных альтернатив (DAs) для системных компонентов с учетом качества их совместимости (IC). В базовой версии ИММП используются следующие порядковые шкалы: (1) порядковая шкала для качества системных комопнентов (приоритеты) ($\iota = \overline{1, l}$; 1 соответствует наилучшему уровню качества); (2) шкала качества системы с учетом порядковых оценок компонентов и порядковых оценок совместимости компонентов ($w = \overline{0, \nu}$; ν соответствует наилучшему уровню).

Для системы из m частей дискретное "пространство" (частичный порядок, решетка) качества системы на основе вектора: $N(S) = (w(S); n(S))$, где w $w(S)$ - минимум парной совместимости между DAs, соответствующими различным системным компонентам, $n(S) = (\eta_1, ..., \eta_r, ..., \eta_k)$, где η_r - число DAs уровня качества r в S $(\sum_{r=1}^{k} n_r = m)$. Оптимизационная задача проектирования имеет вид:

$$max\ N(S), \quad max\ w(S), \quad w(S) \geq 0.$$

Для рассматриваемого проектного метода численный пример (Рис. 5.5) иллюстрирует выделение системных узких мест. Имеется четырех-компонентная система $S = X \star Y \star Z \star H$. Проектные альтернативы для системных компонентов представлены на Рис. 5.5 (порядковые оценки качества DAs в виде приоритетов указаны в круглых скобках, использована шкала $[1, 3]$, 1 соответствует наилучшему качеству). Таблица 5.3 содержит порядковые оценки совместимости (IC) между DAs (шкала $[0, 3]$, 3 соответствует наилучшей совместимости).

На Рис. 5.6а представлена шкала качества системы по компонентам, на Рис. 5.6б представлена интегрированная шкала на основе частичного порядка с учетом совместимости системных компонентов (каждый "треугольник" соответствует шкале на Рис. 5.6а). Рассматриваются два Парето-эффективных системных решения: (i) $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$, $N(S_1) = (1; 2, 1, 1)$; (ii) $S_2 = X_2 \star Y_2 \star Z_2 \star H_2$. $N(S_2) = (2; 0, 1, 3)$.

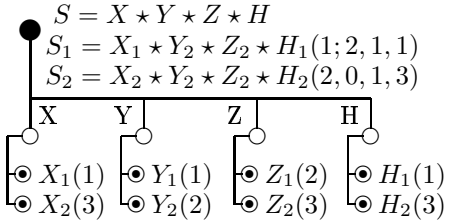


Рис. 5.5. Система из четырех компонентов

Таблица 5.3. Совместимость

	Y_1	Y_2	Z_1	Z_2	H_1	H_2
X_1	3	2	2	2	1	3
X_2	0	3	2	3	1	2
Y_1			2	2	1	2
Y_2			2	3	2	3
Z_1					1	2
Z_2					3	3

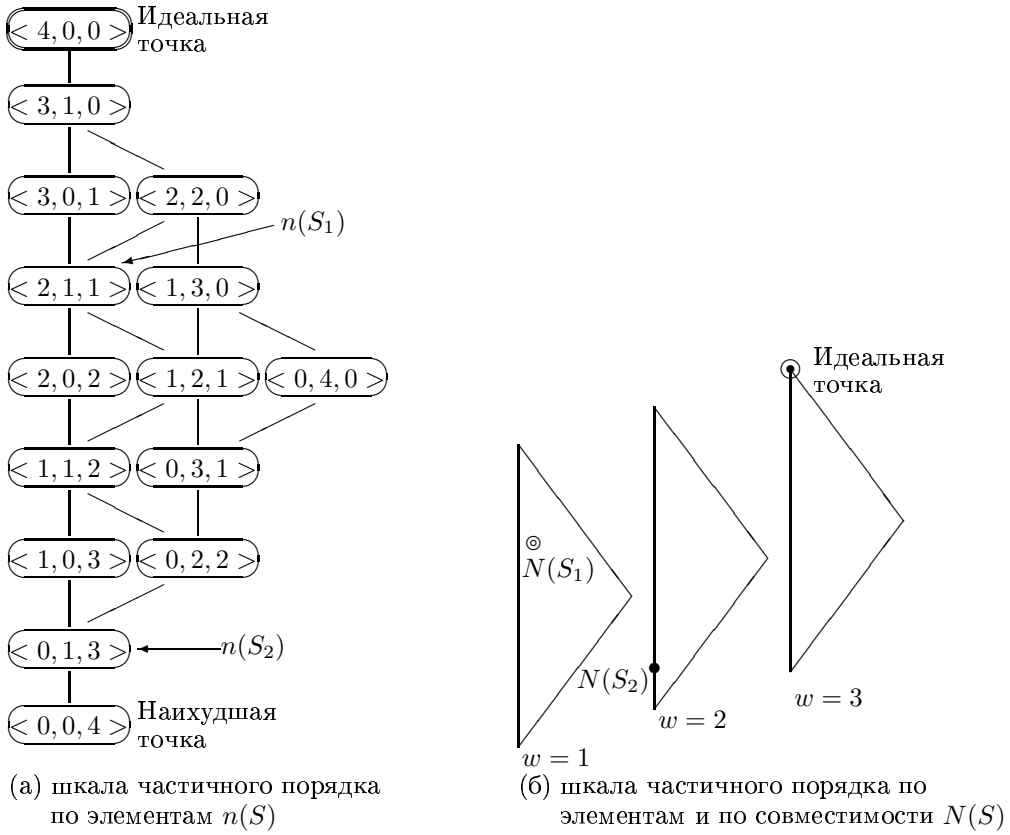
В качестве системных узких мест рассматриваются системные компоненты (DA) или совместимости между парой DAs. Можно использовать такие схемы выделения узких мест:

Схема 1. Многокритериальное ранжирование системных компонентов (DAs).

Схема 2. Многокритериальное ранжирование связей между компонентами.

Схема 3. Совместное многокритериальное ранжирование DAs и связей между ними.

Схема 4. Выявление связанных системных компонентов (т.е., как составные системные отказы): интегрирование данных на основе клики.



(а) шкала частичного порядка по элементам $n(S)$

(б) шкала частичного порядка по элементам и по совместимости $N(S)$

Рис. 5.6. Шкала качества системы S: частичный порядок

На Рис. 5.7 представлено системное решение $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$ (включая оценки DAs и оценки их совместимости).

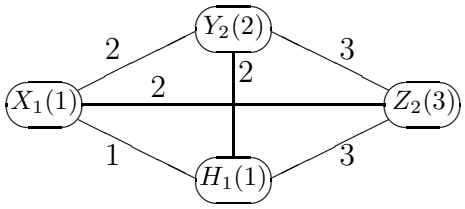


Рис. 5.7. Концентрическое представление S_1

В Таблице 5.4 представлены шесть узких мест (по компонентам, по совместимости компонентов). Очевидно, каждое узкое место может быть оценено по критериям в случае анализа их улучшения (например, возможная полезность для качества системы, стоимость операции улучшения). Далее возможно использование многокритериального ранжирования узких мест (точнее операций их улучшения) на основе оценок по критериям для выбора наиболее перспективного (с точки зрения улучшения) узкого места (или узких мест).

Таблица 5.4. Узкие места, улучшения для S_1

Составные DAs	Узкие места		Операция улучшения w/ι
	DA	IC	
1. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$	Y_1		$2 \Rightarrow 1$
2. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$	Z_2		$2 \Rightarrow 1$
3. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$		(X_1, Y_2)	$2 \Rightarrow 3$
4. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$		(X_1, Z_2)	$2 \Rightarrow 3$
5. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$		(X_1, H_1)	$1 \Rightarrow 3$
6. $S_1 = X_1 \star Y_2 \star Z_2 \star H_1$		(Y_2, H_1)	$2 \Rightarrow 3$

Выделение системных узких мест как группы взаимосвязанных узких мест можно рассматривать как выявление множества системных компонентов низкого качества, которые связаны между собой на высоком уровне совместимости. Такая ситуация соответствует новому типу составных отказов, которые был предложен в [362, 386]. Здесь некоторые "слабые" (т.е., не очень опасные) системные отказы являются "очень сильно" связанными и это может привести к очень серьезным составным системным отказам. Приведенное описание может быть представлено как двух-критериальная оптимизационная задача:

$min\ n(B)$, $max\ w(B)$; B - это подрешение для системного решения S.

На Рис. 5.8 представлено составное решение $S_2 = X_2 \star Y_2 \star Z_2 \star H_1$, $N(S_2) = (2; 0, 1, 3)$ (из примера на Рис. 5.5). Для этого четырех-компонентного решения можно можно рассмотреть три трех-компонентных подсистемы (подрешения):

$B_1 = X_2 \star Y_2 \star Z_2$, $N(B_1) = (2; 0, 1, 2)$;

$B_2 = X_2 \star Z_2 \star H_2$, $N(B_2) = (2; 0, 0, 3)$;

$B_3 = X_2 \star Y_2 \star H_2$, $N(B_3) = (2; 0, 2, 1)$;

$B_4 = Y_2 \star Z_2 \star H_2$, $N(B_4) = (3; 0, 2, 1)$.

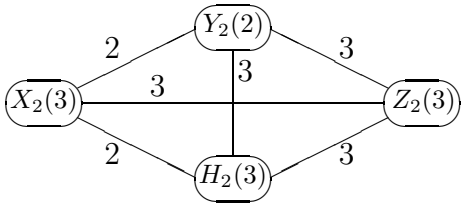
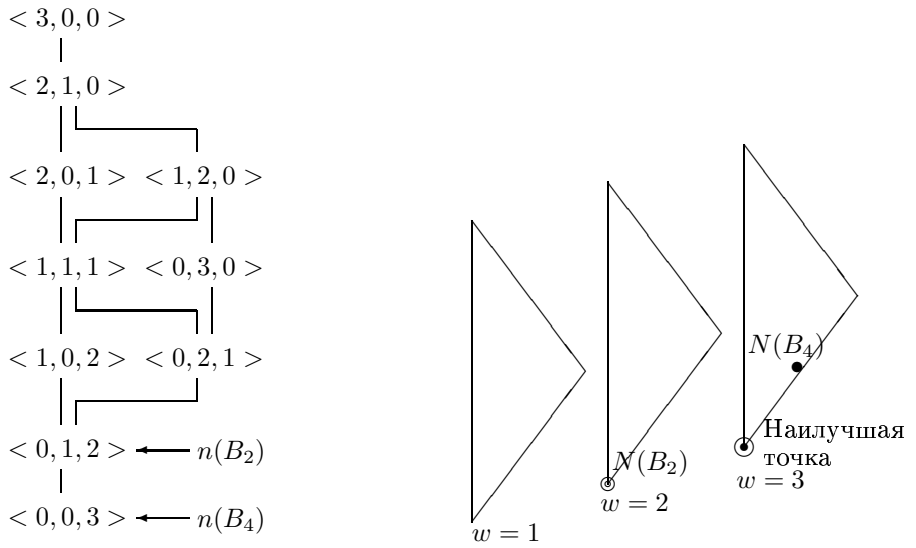


Рис. 5.8. Концентрическое представление S_2

Две Парето-эффективных подсистемы (как составных узких мест) имеют вид (Рис. 5.9): B_2 и B_4.



(а) шкала частичного порядка по элементам $n(B)$ (б) шкала частичного порядка по элементами и по совместимости $N(B)$

Fig. 5.9. Шкала частичного порядка для подсистемы B

5.4 Критические элементы в многоуровневых структурах

С общей точки зрения, представляется важным исследование многоуровневых структур/сетей (Рис. 5.10) [365].

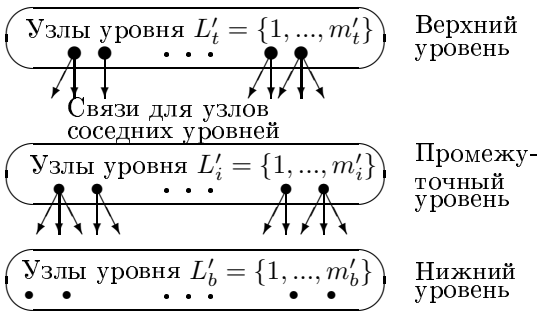


Рис. 5.10. Многоуровневые структуры

В рамках данного направления можно выделить следующие типы задач выделение системных узких мест:

Тип I: для уровня структуры/сети:

(i) выделение критических узлов в сетях (например, задача покрывающего дерева с максимизацией числа висячих вершин, задача connected dominating sets),

(ii) выделение группы критических узлов сети,

(iii) выделение группы взаимосвязанных критических узлов сети,

(iv) выявление топологии сети низкого качества.

Тип II: для соседних уровней: выделение критических связей между узлами соседних уровней.

Тип III: для многих уровней: выделение неправильного (или низкого качества) распределения узлов по уровням структуры/сети.

Рассмотрим некоторые указанные задачи для уровня структуры.

Во-первых, выделение критических узлов (или узла) на уровне структуры/сети может быть основано на методах, которые были представлены в предыдущих секциях (например, метод Парето-диаграммы, многокритериальный арнализ/ранжирование, выделение связанных узлов в виде интеграции данных на основе клики). Во-вторых, могут быть использованы три известные задачи комбинаторной оптимизации. Рис. 5.11 иллюстрирует эти задачи: (а) *задача покрытия с максимизацией числа висячих вершин* , (б) *задача покрытия с минимизацией числа внутренних вершин* , (в) *задача построения иеархической (двух-уровневой) сети* . При этом элементы множества внутренних узлов структуры/сети может рассматриваться как критические узлы (например, для улучшения, для тестирования) или как узкие места.

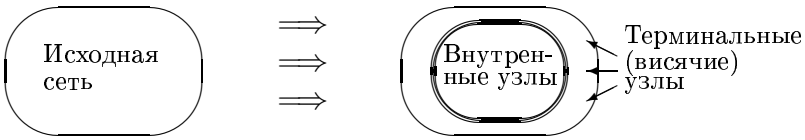


Рис. 5.11. Max висячих узлов (или Min внутренних узлов)

Задача “maximum leaf spanning tree” формулируется так [51, 119, 207]):

Найти покрывающее дерево для исходного графа, в котором число висячих вершин дерева является максимальным

В общем случае покрывающее дерево содержит следующие типы вершин: (а) корень; (б) внутренние узлы, внутренние узлы могут рассматриваться как виртуальная шина “bus” в сетевых технологиях; (в) висячие (терминальные) узлы. Таким образом задача заключается в максимизации числа висячих узлов или макимизации числа внутренных узлов. Эта задача является NP-трудной [207].

С точки зрения построения точных алгоритмов, данная задача является эквивалентной задаче построения точного “connected dominating set” (NP-трудная) [93, 119, 207]:

Найти минимальное множество вершин $D \subseteq A$ *для исходного графа* $G = (A, E)$ *так, чтобы соответствующий* D *подграф* $G' = (D, E')$ $(E' \subseteq E)$ *являлся связанным доминирующим множеством и* D *являлся доминирующим множеством для* G.

Базовая задача проектирования иерархического двух-урвоневой сети имеет вид [71, 153, 479]:

Найти двух-уровневую покрывающую сеть минимальной стоимости, состоящую из двух частей: (i) основной (внутренний) путь (или несколько путей, дерево, кольцо), (ii) вторичные деревья.

Таким образом, исходная сеть разбивается на две части:

(а) основная часть (т.е., часть верхнего иерархического уровня): некий путь (или несколько путей, дерево, кольцо), составленное из первичных дуг сети, которые связаны с набором узлов сети (т.е., первичными узлами сети);

(б) вторичная часть (т.е., вторичные узлы, вторичные деревья): эта часть включает одно или несколько деревьев, чьи дуги (вторичные дуги) являются менее дорогими для использования, чем первичные дуги.

При этом каждая дуги имеет стоимость (d_{ij}, $\forall i, j \in A$, A - множество узлов). Общая стоимость выбранных дуг в покрывающей структуре используется как минимизируемая целевая функция. Эта задача формулируется как задача комбинаторной оптимизации [153]) и является NP-трудной [71].

Очевидно, что подобные задачи могут использоваться для выделений критических дуг в сетях.

В упомянутых выше задачах *типа II* и *типа III*, решение заключается в назначении элементов на "позиции" (т.е., задачи о назначании или размещении). При этом, могут потребоваться новые варианты постановки комбинаторных задач для выявления назначения низкого качества в существующих сетевых решениях. Отметим, что использование метода ИММП для "расширенной" задачи о назначениях было предложено в [336, 349] (глава 11).

Следует отметить, что выделение сетевой топологии низкого качества требует специальных дополнительных исследований. Известная задача "augmentation problem" [182] может рассматриваться как вариант такого исследования.

5.5 Предсказательное выделение системных узких мест

Предсказательное выделение системных узких мест может быть рассмотрено так: (Рис. 5.12):

Шаг 1. Исследование существующих измененний и/или будущих изменений системных параметров и/или структуры системы.

Шаг 2. Анализ эволюции системы (т.е., соответствующей системной траектории, траекторий системных параметров).

Шаг 3. Прогнозирование системных параметров для построения прогноза системы.

Шаг 4. Выделение системных узких мест на основе прогнозных значений системных параметров (т.е., прогноз системы, прогноз системных параметров).

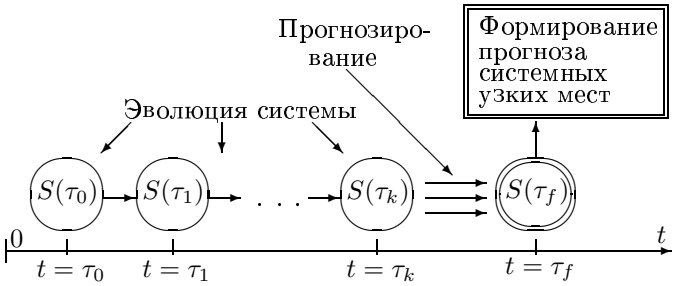


Рис. 5.12. Предсказательное выделение узких мест

В качестве исследуемых объектов (с точки зрения системных узких мест) могут рассматриваться: системные компоненты, группы связанных системных компонентов, структура системы.

Для сетевых систем указанные задачи предсказательного выделения узких мест представляются более сложными.

5.5.1 Предсказательное выделение компонентов

Для предсказательного выделения системных узких мест как системных компонентов можно использовать аналогичные методы (т.е., Парето-диарамму, многокритериальное ранжирование). При этом прогнозы системных параметров являются исходными данными. В рассматриваемом примере (Рис. 5.2, Таблица 5.2, Рис. 5.3, Рис. 5.4), прогноз данных из Таблицы 5.2 должен использоваться.

5.5.2 Предсказательное выделение связанных компонентов

Предсказательное выделение узких мест при иерархическом морфологическом проектировании может рассматриваться как и в предыдущей секции (т.е., анализ эволюции системы, вычисление прогноза системы, выделение узких мест посредством указанных методов). Упрощенный пример иллюстрирует выделение составных узких мест (как подсистем) для четырех-компонентной системы $S = X \star Y \star Z \star H$ с Рис. 5.5. На Рис. 5.13 представлен численный пример для эволюции и прогноза решения $S_2 = X_2 \star Y_2 \star Z_2 \star H_2$ $(N(S_2) = 2; 0, 1, 3)$. Используется следующая временная ось: базовая временная точка $t = \tau_0$, следующая временная точка (эволюция) $t = \tau_1$, прогнозная временная точка $t = \tau_2$ (т.е., $t = \tau_f$).

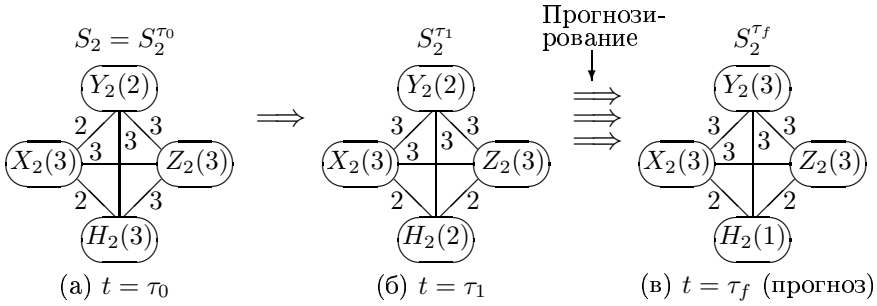


Рис. 5.13. Эволюция решения S_2 и прогноз

Отметим, для базовой временной точки (τ_0), получены две подсистемы (как составные узкие места) (Рис. 5.9): $B_2 = X_2 \star Z_2 \star H_2$, $N(B_2) = (2; 0, 0, 3)$; $B_4 = Y_2 \star Z_2 \star H_2$, $N(B_4) = (3; 0, 2, 1)$. Для следующей временной точки, подсистемы и их оценки имеют вид: (i) $t = \tau_1$: $N(B_1^{\tau_1}) = (3; 0, 1, 2)$, $N(B_2^{\tau_1}) = (2; 0, 1, 2)$, $N(B_3^{\tau_1}) = (2; 0, 2, 1)$, $N(B_4^{\tau_1}) = (2; 0, 2, 1)$; (ii) $t = \tau_f(\tau_2)$: $N(B_1^{\tau_2}) = (3; 0, 0, 3)$, $N(B_2^{\tau_2}) = (3; 1, 1, 1)$, $N(B_3^{\tau_2}) = (2; 1, 0, 2)$, $N(B_4^{\tau_2}) = (2; 1, 1, 1)$. В результате, следующие подсистемы получены как составные узкие места:

(а) $t = \tau_1$ (Fig. 5.14а): $B_1^{\tau_1} = X_2 \star Y_2 \star Z_2$, $N(B_1^{\tau_1}) = (3; 0, 1, 2)$;

(б) $t = \tau_f(\tau_2)$ (Fig. 5.14б): $B_1^{\tau_2} = X_2 \star Y_2 \star Z_2$, $N(B_1^{\tau_2}) = (3; 0, 0, 3)$.

Таким образом, прогнозное узкое место имеет вид: $B_1^{\tau_2} = X_2 \star Y_2 \star Z_2$. На Рис. 5.15 представлена траектория составного узкого места.

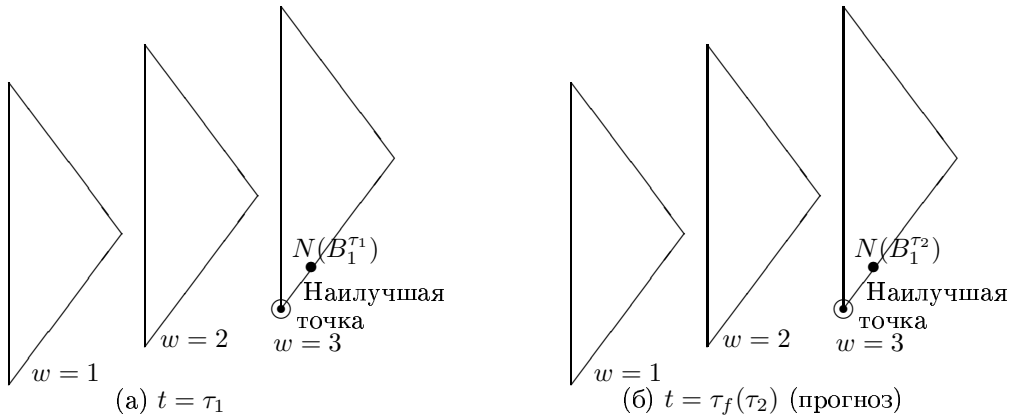


Рис. 5.14. Шкала частичного порядка для подсистемы B

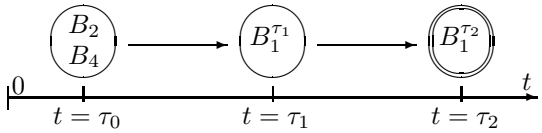


Рис. 5.15. Траектория узкого места

5.6 Интеграция данных на основе клики над потоками графов

5.6.1 Предварительное описание

Сначала рассмотрим упрощенный пример модульной системы, состоящей из следующих компонентов: базовое оборудование s_1, подсистема управления s_2, подсистема безопасности s_3, персонал для обслуживания оборудования при производственном процессе s_4, персонал для технического обслуживания и тестирования s_5, пероснал для удаленного управления s_6. Исследуются три функции системы (Рис. 5.16): производственная функция $f_1 : \{s_1, s_2, s_3, s_4\}$, функция технического обслуживания и тестирования $f_2 : \{s_1, s_2, s_3, s_5\}$, функция удаленного управления $f_3 : \{s_1, s_2, s_6\}$. В случае плохой координации между указанными функциями системы (т.е., их реализацией) может возникнуть следующая ситуация: (i) базовое оборудование s_1 вышло из строя (не работает) (персонал обслуживания оборудования s_4, т.е., посредством функции f_1); (ii) подсистема безопасности s_3 не работает (действия персонала по техническому обслуживанию и тестированию s_5, т.е., посредством функции f_2); (iii) подсистема управления s_2 находится в неисправном состоянии (неправильные действия персонала для удаленного управления s_6, т.е., посредством функции f_3). На Рис. 5.16 изображена указанная ситуация (порядковые оценки вершин графа представлена в кружках; при этом используются следующие оценки для вершин/узлов: 1 - состояние неисправности, 2 - состояние частичной исправности ('OK'), 3 - состояние полной исправности ('OK')).

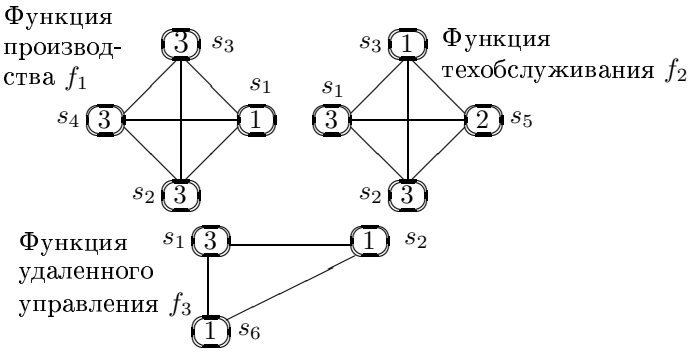


Рис. 5.16. Упрощенный пример

В результате комбинация связанных системных компонентов (как клика) $\{s_1, s_2, s_3\}$ может привести к комбинированному системному отказу.

Описываемый подход базируется на уровневой схеме (Рис. 5.17, [386]).

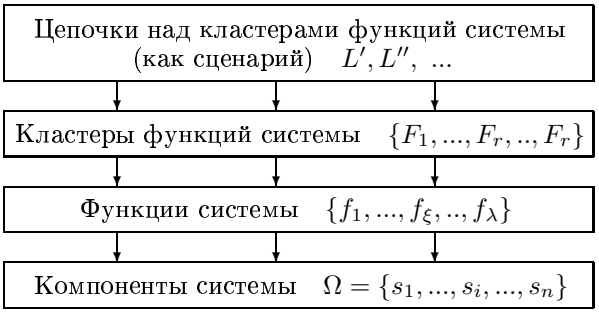


Рис. 5.17. Уровневое тестирование системы

Схема тестирования системы включает следующие стадии:

Стадия 1. Тестирование компонентов системы (например, [284]).

Стадия 2. Анализ системных функций, их взаимосвязей, построение кластеров функций (т.е., функции, которые выполняются совместно, одновременно) [386].

Стадия 3. Построение интегрированного графа над системными компонентами для каждого кластера функций.

Стадия 4. Выявление клики (или квази-клики) в интегрированном графе [462, 472].

Упрощенная схема процесса тестирования приведена на Рис. 5.18 (интегрированный граф соответствует кластеру функций $F' = \{f_1, ..., f_\xi, ..., f_\lambda\}$). Задача выявления клики в интегрированном графе рассматривается как задача интегрирования данных (fusion) или выявления структуры (structure mining) [137, 263]. Таким образом, схема основана на выявлении межграфовых клик (cross-graphs cliques) или квази-клик [207, 272, 386, 462, 472].

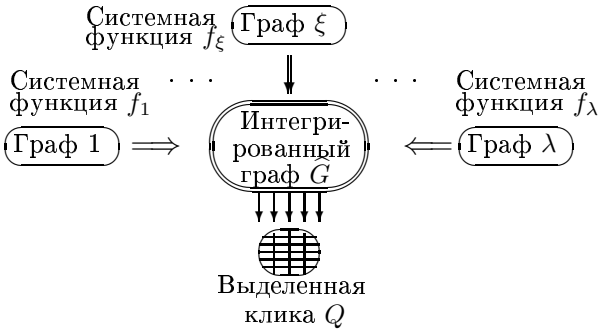


Рис. 5.18. Выявление клики

Рассмотрим следующие базовые множества [386]. Пусть $\Omega = \{s_1, ..., s_i, ..., s_n\}$ будет множество подсистем или основных системных компонентов. Пусть имеется множество системных функций $f = \{f_1, ..., f_\xi, ..., f_\lambda\}$. Для каждой системной функции f_ξ имеется следующее: (i) подмножество системных компонентов, $\Omega(f_\xi) \subseteq \Omega$, которые используются данной системной функцией f_ξ; (ii) некий граф)обычно полный над элементами множества $\Omega(f_\xi)$: $G(f_\xi) = (\Omega(f_\xi), E(f_\xi))$.

Также имеется множество кластеров системных функций $F = \{F_1, ..., F_r, ..., F_p\}$, где $F_r \subseteq f$; кластер каждой системной функции F_r $(r = \overline{1,p})$ представляет собой подмножество множества системных функций F и для каждого кластера системных функций его элементы (и соответствующие функции) выполняются в один и тот же момент времени. Для каждого кластера системных функций F_r можно построить соответствующий интегрированный граф: $G(F_r) = G(\Omega(F_r), E(F_r)) = \cup_{f_j \in F_r} G(f_j)$, где $\Omega(F_r) = \cup_{f_j \in F_r} \Omega(f_j)$ и $E(F_r) = \cup_{f_j \in F_r} E(f_j)$.

Для каждой подсистемы s_i применяется процедура тестирования (тестирование компонента) и такая процедура позволяет получить порядковый результат (оценку), в частности, по шкале: $[1, 2, 3, 4]$ (1 - компонент не исправен, 2 - существенный отказ компонента, 3 - незначительный отказ компонента, 4 - полная работоспособность компонента). Указанная порядковая оценка соответствует состоянию компонента (можно использовать для обозначения цвет). В результате получается граф с порядковыми оценками (весами) вершин (или раскрашенными вершинами). Таким образом, для каждого кластера функций F_r можно исследовать соответствующий интегрированный граф (с порядковыми весами вершин): $\widehat{G}(F_r) = \widehat{G}(S(F_r), E(F_r))$.

Результаты тестирования компонента могут быть различны для различных системных функций. В этом случае в качестве итоговой оценки компонента назначается наихудшая.

Пусть Q_h будет клика над вершинами h (например, Q_4, здесь оценка каждой вершины $\leq l$, $l = 1, 2, 3, 4$), "аппроксимация" по числу вершин или по оценкам оценкам вершин

обозначается посредством “widetilde” (пример: $\widetilde{Q}_4$). Таким образом, в интегрированном графе с порядковыми весами вершин (или цветами) $\widehat{G}(F_r)$ исследуется подструктуры (подграфы) (посредством правила: в структуре каждый “цвет” имеет вид: “ $= 1''$, “ $\leq 2''$, и т.п.). На Рис. 5.19 представлена иллюстрация 4-вершинной структуры (порядковые оценки вершин графа указаны в кружках):

1. Клика, размерность клики равна (или больше чем) числу функций в кластере F_r (Рис. 5.19a): Q_r .

2. Квази-клика по ребрам/взаимосвязям (т.е., ребро - отсутствует) (Рис. 5.19b): Φ_r .

3. Квази-клика по вершинам (в выделенном подграфе не все вершины имеют оценку $\leq l,\ l = 1, 2, ...$) (Рис. 5.19c): Θ_r .

4. Квази-клика по вершинам и ребрам (Рис. 5.19d): $\widetilde{\Theta}_r$ or $\widetilde{\Phi}_r$.

5. Под-клика или клика меньшей размерности (т.е., число вершин в клике v меньше, чем число функций в F_r (Рис. 5.19e): $Q_v,\ v < p$).

6. Квази-клика (структурная аппроксимация): (i) по вершинам, (ii) по ребрам, (iii) по вершинам и ребрам ($\widetilde{Q}$ и т.п.).

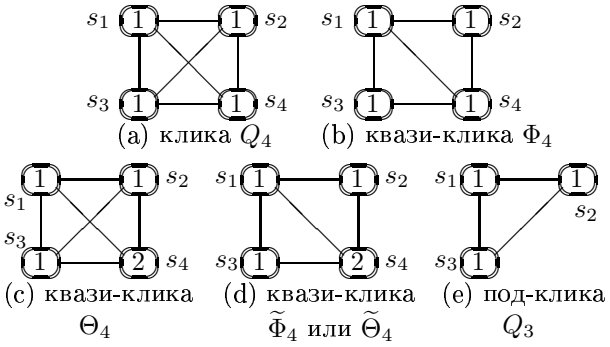


Рис. 5.19. Примеры клики и квази-клики

Представляется важным указать следующее:

I. Ситуация, когда оценка системного компонента равна 1 (т.е., компонент не исправен), является критической и может привести к системному отказу.

II. Ситуация, когда несколько связанных между собой (по времени и/или по работе системы) системных компонента имеют оценки на “среднем уровне” (т.е., существенный отказ компонента или незначительный отказ компонента), может привести к системному отказу в сложной системе [139]. В данном разделе основное внимание уделено такой системной ситуации (Рис. 5.20, оценки вершин: ≤ 3).

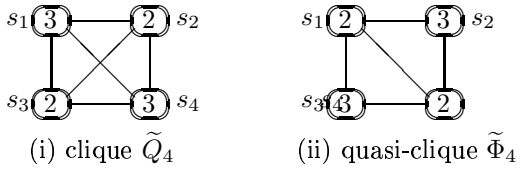


Рис. 5.20. Примеры отказов системы

Очевидно, оценки системных компонентов на уровне 1 и 2 могут также использоваться как базовые.

5.6.2 Основная задача

Здесь рассматривается базовая задача (*задача 1*). Структурная интеграция (structural fusion) оценок качества для системных компонентов по порядковой шкале иллюстрируется на Рис. 5.21, Рис. 5.22, Рис. 5.23. Оценки для системных компонентов представлены

Рис. 5.21 (компоненты s_3, s_5, s_6: 1; компонент s_1: 2; компонент s_7: 3; другие компоненты: 4). Базовая задача (*задача 1*) имеет вид:

Найти для много-функциональной ситуации (в интегрированной графе для кластера функций F_r) клику Q_h (число вершин равно числу функций в кластере F_r или более с уровнем оценок $\leq l$ $(l = 1, 2, ...)$.

Таким образом получается известная задачи клики:

Найти наибольшую клику (полный подграф) Q в ненаправленном графе

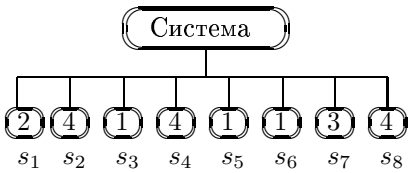


Рис. 5.21. Структура системы и оценки компонентов

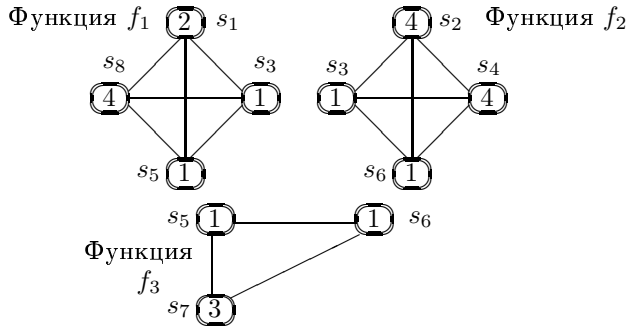


Рис. 5.22. Кластер системных функций

Полученное множество системных компонентов (вершин) можно исследовать как "подмножество критических системных компонентов" или "системный синдром" (аналог "синдрома" в медицине). На Рис. 5.21 и на Рис. 5.22, $F_r = \{f_1, f_2, f_3\}$. Далее, на Рис. 5.23 вершины клики следующие $\{s_3, s_5, s_6\}$. При этом оценки указанных вершин равны 1.

Ясно, что клика Q_λ может отсутствовать. В этом случае важно осуществлять поиск квази-клики $\widetilde{Q}_\lambda$ или клик, которые содержат меньшее число вершин.

Таким образом, другая задача структурной интеграции данных (structural fusion) имеет вид (*задача 1a*):

Найти квази-клику $\widetilde{Q}_\lambda$ для много-функциональной ситуации, т.е., без некоторых связей/ребер и/или с уровнем оценок: $\leq l$, $(1 < l \leq 3)$.

В качестве примеров квази-клик (множества вершин) можно привести следующие (Рис. 5.23): (a) $\{s_1, s_3, s_5, s_6\}$, оценки: ≤ 2; (b) $\{s_1, s_3, s_5, s_6, s_7\}$, оценки ≤ 3. Следует отметить, что часто является возможным (или целесообразным) выделение нескольких клик (или квази-клик).

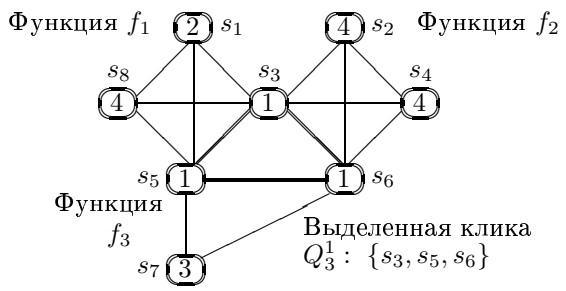


Рис. 5.23. Интегрированный граф для кластера функций

5.6.3 Задачи над потоками графов

На Рис. 5.24. представлена иллюстрация для случая потоков графов.

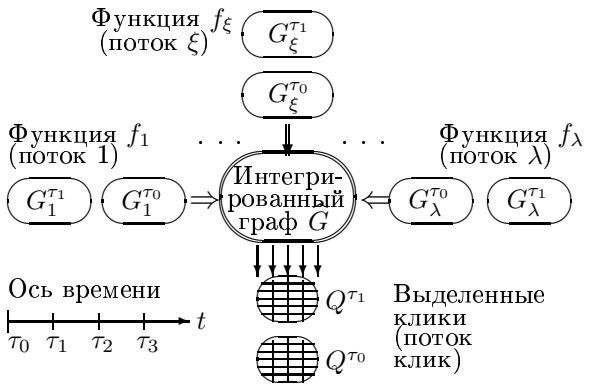


Рис. 5.24. Поток клик над потоками графов

Здесь рассматривается временная ось:

$t = \{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5, ...\}$.

Таким образом, исследуется следующее:

(i) последовательность состояний системных компонентов ($j = \overline{1,k}$):

$s_j(t) = \{s_j^{\tau_0}, s_j^{\tau_1}, s_j^{\tau_2}, s_j^{\tau_3}, s_j^{\tau_4}, s_j^{\tau_5}, ...\}$,

где $s_j(\tau_\eta) \in \{0,1\}$ $(\eta = 0,1,2,3,4,5,...)$, $s_j(\tau_\eta) = 1$ если функция j используется в момоент времени τ_η и $s_j(\tau_\eta) = 0$ - в противном случае;

(ii) последовательность системных функций ($\xi = \overline{1,\lambda}$):

$f_\xi(t) = \{f_\xi^{\tau_0}, f_\xi^{\tau_1}, f_\xi^{\tau_2}, f_\xi^{\tau_3}, f_\xi^{\tau_4}, f_\xi^{\tau_5}, ...\}$,

где $s_j(\tau_\eta) \in \{0,1\}$, $s_j(\tau_\eta) = 1$ если функция j используется в момент времени τ_η и $s_j(\tau_\eta) = 0$ - в противном случае;

(iii) последовательность графов для функций системы ($\xi = \overline{1,\lambda}$):

$G_{f_\xi}(t) = \{G_{f_\xi}^{\tau_0}, G_{f_\xi}^{\tau_1}, G_{f_\xi}^{\tau_2}, G_{f_\xi}^{\tau_3}, G_{f_\xi}^{\tau_4}, G_{f_\xi}^{\tau_5}, ...\}$

(если $s_j^{\tau_\eta} = 0$, соответствующий граф $G_{f_\xi}^{\tau_\eta}$ является пустым);

(iv) последовательность графов для кластера системных функций $(r = \overline{1,p})$:

$G_{F_r}(t) = \{G_{F_r}^{\tau_0}, G_{F_r}^{\tau_1}, G_{F_r}^{\tau_2}, G_{F_r}^{\tau_3}, G_{F_r}^{\tau_4}, G_{F_r}^{\tau_5}, ...\}$

(если $s_j^{\tau_\eta} = 0$, соответствующий граф $G_{F_r}^{\tau_\eta}$ является пустым).

Отметим, что цепь кластеров системных функций (например, $L =< F', F'', F''' >$) рассматривается как сценарий (в общем случае, сценарий может иметь более сложный вид, например, дерево, сеть).

Множество дополнительных задач включает следующие:

Задача 2. Выделение клики, когда число вершин меньше, чем число функций в кластере функций (т.е., под-клики).

Задача 3. Динамические задачи ("цвета" вершин являются функциями времени): *3.1.* существование временного интервала, когда клика существует; *3.2.* существование временного интервала, когда квази-клика существует;

Задача 4. Анализ временных интервалов, когда клика (или квази-клика) существует и мониторинг клики (квази-клики) как некой "критической" структуры (подструктуры). В результате может быть получена некая трасса специальной структуры (например, клика Q): T_Q.

Задача 5. Проектирование действий как составного плана по разрушению критической под-структуры (или под-структур) (т.е., клики или клик, квази-клики или квази-клик).

5.6.4 Пример

В данноем разделе представлен численный иллюстративный пример. В Таблице 5.5. и Таблице 5.6 содержится описание исследуемого множества функций системы и кластеров функций: $\{f_1, f_2, f_3, f_4, f_5\}$ и $\{F_1, F_2, F_3\}$.

Таблица 5.5. Функции системы

Функции системы	Системные компоненты							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
f_1	⋆		⋆		⋆			⋆
f_2		⋆	⋆	⋆		⋆		
f_3					⋆	⋆	⋆	
f_4	⋆				⋆			
f_5					⋆	⋆		⋆

Таблица 5.6. Кластеры функций

Кластеры функций системы	Функции системы				
	f_1	f_2	f_3	f_4	f_5
F_1	⋆	⋆	⋆		
F_2		⋆		⋆	⋆
F_3			⋆		⋆

Рассматриваются следующие временные оси:

$$\{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$$

(i.e., τ_η, $\eta = \overline{0,5}$). Цепочка кластеров функций (некий сценарий) имеет следующий вид:

$$L' =< F_2^{\tau_0}, F_1^{\tau_1}, F_3^{\tau_2}, F_1^{\tau_3}, F_1^{\tau_4}, F_2^{\tau_5} >$$

где *верхний* означает момент времени. На Рис. 5.23 изображен интегрированный граф $\widehat{G}(F_1)$, На Рис. 5.25 представлены два интегрированных графа $\widehat{G}(F_2)$ и $\widehat{G}(F_3)$.

(a) интегрированный граф $\widehat{G}(F_2)$ (b) интегрированный граф $\widehat{G}(F_3)$

Рис. 5.25. Интегрированные графы для кластеров функций

Далее рассматриваются следующие базовые задачи: задача 3, задача 4, задача 5. На Рис. 5.26 представлен потоки состояний (state streams) для системных компонентов (вершин) s_1, s_2, s_3, s_4, s_5, s_5, s_6, s_7, s_8 с учетом указанной выше временной оси. Дополнительно, Рис. 5.26. содержит следующее: (i) интегрированные графы $\{\widehat{G}\}$, (ii) выделенные структуры (здесь: Q_3), (iii) полученные треки (трассы) из выделенных структур (здесь: T_{Q_3}).

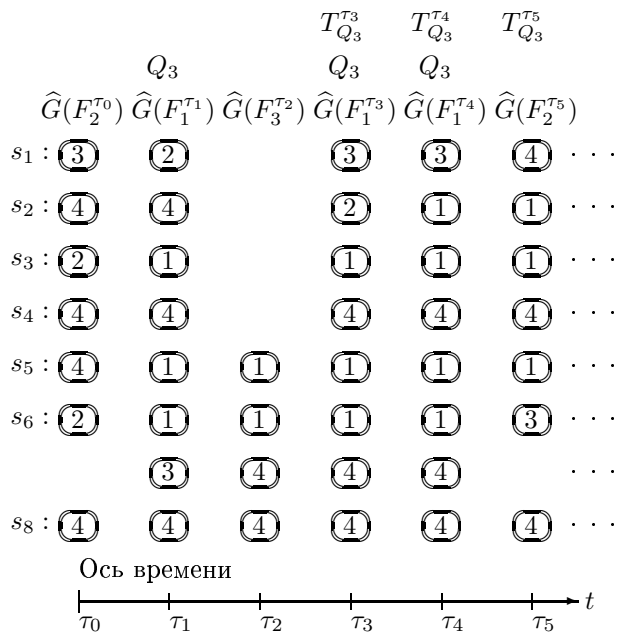


Рис. 5.26. Примеры потоков графов

Теперь можно указать следующие решения:

Задача 3.1. Клика $Q_3 : \{s_3, s_5, s_6\}$ может быть выделена для времени: τ_1, τ_3, τ_4 (оценки вершин равны 1).

Задача 4. Для временного интервала $[\tau_1, \tau_5]$ существует структура с вершинами: $\{s_3, s_5, s_6\}$. С учетом известного инженерного правила "правило захвата трассы: 2 из 3" (т.е., для временного интервала длины 3), клика выделяется 2 раза. В результате представляется целесообразным "инициализировать" (т.е., определить) в момент времени τ_3 клику $Q_3 : \{s_3, s_5, s_6\}$ (оценки вершин равны 1). Следует отметить, что указанное правило используется также как "правило поддержки/сопровождения трассы". После этого возможно поддерживать выделенную структуру как некое событие (т.е., проверять это правило в каждый дискретный момент времени, например: τ_4, τ_5).

Задача 5. Очевидно, что для "разрушения" указанного выше события (т.е., $T_{Q_3}^{\tau_3}$, $T_{Q_3}^{\tau_4}$, $T_{Q_3}^{\tau_5}$) необходимо "разрушить" клику Q_3 в момент времени τ_3 за счет улучшения состояния для s_5 (или s_6) (т.е., улучшение оценки: $1 \to 2$ или $1 \to 3$ или $1 \to 4$).

5.6.5 Обсуждение

Следует отметить важность систем обработки на основе потоков данных (data stream systems) [66, 137, 518, 538]. Такие системы используются во многих приложениях, например: интеграция данных и знаний в информационных системах, интеграция информации в сенсорных системах, анализ и мониторинг надежности промышленных систем.

Среди ключевых параметров таких систем можно указать следующие: (1) число потоков (один, много); (2) типа данных, например: значения (двоичные, порядковые и др.), структуры (графы, отношения предпочтений); (3) размер временного окна (т.е., число последовательных временных моментов, которые анализируются совместно). Упрощенная типология таких систем может выглядеть так [223, 403, 518, 538]:

(a) статический случай для m потоков (streams): (i) интеграция значений, методы обработки: гистограммы, правило "k из m", методы диагностики; (ii) агрегация структур (глава 7), методы обработки (принятие решений, технология знаний, обработка изображений): построение максимальных подструктур (консенсус, медиана), построение минимальных надструктур;

(б) динамический случай (один поток, окно для n временных моментов); методы обработки: правило "k из m", интеграция последовательностей изображений, выделение структур в временных последовательностях графов [75, 137, 518];

(в) комбинированный динамический случай (m потоков и окно для n временных моментов); составные методы обработки (например, динамическое принятие решений на основе процессов Маркова, динамические сети принятия решений) [75, 91, 108, 109, 155].

В случае (в) возможны три стратегии:

стратегия 1: (i) интеграция данных для каждого потока в рамках временного окна (случай б), (ii) "суммирование" результатов для m потоков (случай а);

стратегия 2: (i) "суммирование" данных m потоков для каждого момента времени (случай а), (ii) интеграция результатов в рамках временного окна (случай б);

стратегия 3: комбинированная схема.

Рассмотренная схема реализует *стратегию 2*: (i) интеграция графов в каждый момент времени с выделением клики (или квази-клики), (ii) использование правила "k из m".

5.7 Выводы к главе

Описанные в главе подходы к выделению системных узких мест можно рассматривать как предварительную стадию анализа и нового проектирования (перепроектирования) различных модульных систем. С другой стороны, рассмотренные подходы к выделению узких мест очень близки к процедурам тестирования систем (например, много-функциональное тестирование систем [362, 386]). Представляется важным указать возможные направления исследований в будущем:

1. исследование много-стадийных и комплексных схем выделения системных узких мест;

2. исследовнаие системных узких мест как 'траекторий' системных компонентов;

3. дополнительные исследования выделения узких метс в сетевых системах;

4. учет неопределенности.

Глава 6

Схемы улучшения систем

В данной главе [1] приведено описание подходов к улучшению/расширению иерархических модульных систем. Стратегии улучшения/расширения систем рассматриваются как действия (акции, операции) двух типов: (а) улучшение системных компонентов и/или их взаимосвязей, (б) модификация структуры системы.

6.1 Предварительные сведения

В последние десятилетия актуальность перепроектирования (улучшения, обновления) систем [334, 336, 344, 351, 367, 385, 452].

Традиционные подходы к генерации новых проектных решений основываются на модификации/улучшении существующей системы (продукта) [151]. Часто в качестве базового решения для модификации используется одна существующая система (или продукт) [336, 344, 346, 351, 352, 375, 397, 400] (Рис. 6.1):

Исходная система(ы) $\Rightarrow$ *Процесс улучшения* $\Rightarrow$ *Улучшенная система(ы)* .

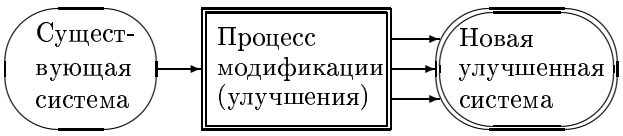


Рис. 6.1. Модификация/улучшение системы

Системные подходы к улучшению могут рассматриваться как следующие:
(а) улучшение системных компонентов и/или их взаимосвязей,
(б) улучшение/расширение структуры системы,
(в) расширение системы в виде проектирования дополнительной части,
(г) комбинированное улучшение/расширение системы.
На Рис. 6.2 представлена иллюстрация общего подхода к расширению системы.

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. chapters 3, 5.
(ii) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapters 5, 8.
(iii) M.Sh. Levin, Towards design of hierarchy (research survey). Electr. prepr., 36 p., Dec. 8, 2012. Http://arxiv.org/abs/1212.1735 [math.OC].
(iv) M.Sh. Levin, Improvement/extension of modular systems as combinatorial reengineering (survey). Electr. prepr., 24 p., Apr. 17, 2013. Http://arxiv.org/abs/1304.4965 [cs.AI].
(v) Levin M.Sh., Aggregation of composite solutions: strategies, models, examples. Electr. prepr., 72 p., Nov. 29, 2011. http://arxiv.org/abs/1111.6983 [cs.SE]

Рис. 6.2. Система и дополнительная часть [351]

На Рис. 6.3 представлены общие действия по улучшению/развитию системы:

(i) улучшение базовой системы,

(ii) расширение на основе проектирования дополнительной части системы,

(iii) координация улучшения базовой системы и проектирования дополнительной части системы.

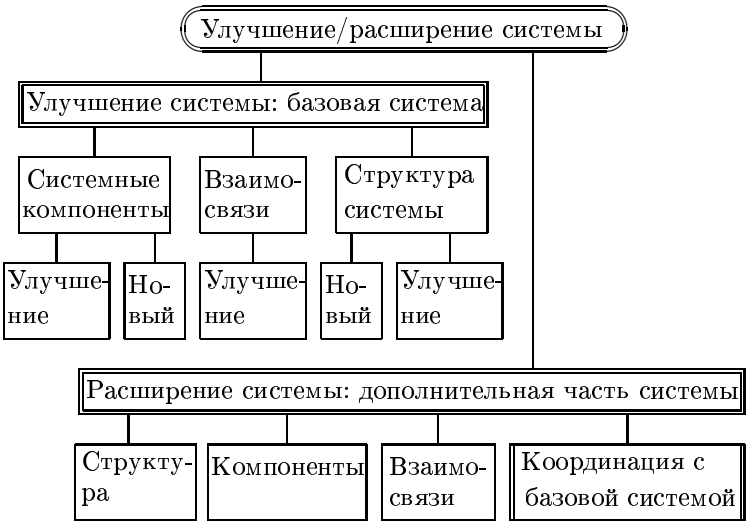


Рис. 6.3. Действия по улучшению системы [351]

Множество основных действий по улучшению/расширению системы включает следующие операции:

(а) улучшение системных компонентов (например, улучшение системного компонента, замена системного компонента);

(б) улучшение связей между системными компонентами (совместимость);

(в) совместное улучшение системных компонентов и их связей;

(г) улучшение структуры системы (замена части системы, добавление части системы, удаление части системы, модификация структуры системы).

Стратегии улучшения/расширения системы основываются на выборе и/или композиции улучшающих действий, указанных выше. Эти стратегии базируются на моделях комбинаторной оптимизации (в частности, многокритериальный выбор, задача о рюкзаке, задача блочного рюкзака, комбинаторный синтез на основе морфологической клики, задача назначения или размещения, перекраска графа, задачи о покрытии, задача выбора новых “горячих” связей - hotlink assignment) [76, 88, 207, 290, 307, 308, 349, 492].

6.2 Типовые ситуации улучшения системы

6.2.1 Базовые ситуации улучшения систем

Основная ситуация улучшения систем представлена на Рис. 6.4 (одна исходная система и одна результирующая-улучшенная система). При этом можно выделить следующие специальные случаи:

Специальный случай 1. Улучшение компонентов системы: (1.1) улучшение системных компонентов, (1.2) улучшение взаимосвязей системных комопнентов (совместимости), (1.3) совместное улучшение системных элементов и их совместимости.

Специальный случай 2. Улучшение в виде модификации структуры системы: (2.1) расширение структуры системы (т.е., добавление части структуры системы), (2.2) модификация структуры системы в виде удаления части структуры системы, (2.3) модификация структуры системы (трансформация, добавление,удаление элементов, добавление/удаление связей элементов), (2.4) совместный случай (т.е., удаление части структуры системы и добавление части структуры системы, модификация структуры системы).

Специальный случай 3. Расширение системы в виде проектирования дополнительной части системы (т.е., дополнительная часть системы и дополнительные элементы системы).

Специальный случай 4. Комбинация указанных выше случаев.

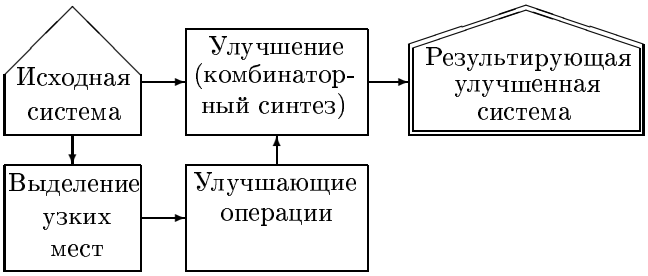


Рис. 6.4. Основное системное улучшение [336, 344, 351]

С общей точки зрения, процессы улучшения/модификации систем основываются на трех типах действий [336, 344, 351]:

(i) улучшение (модификация, обновление, дополнение) компонента системы,

(ii) улучшение совместимости компонентов системы,

(iii) изменение структуры системы.

Можно привести некоторые примеры улучшения модульных систем:

(а) здание [344, 385],

(б) информационная система [336],

(в) человеко-машинная система [344, 341],

(г) протоколы связи и стандарты передачи информации [370, 390],

(д) система управления умными домами [369, 398],

(е) сети связи [351, 397].

Выделение узких мест в системе может рассматриваться как предварительная фаза.

На Рис. 6.5 представлен иллюстративный пример для процесса улучшение системы на основе компонентов системы: $S^a \Rightarrow S^b$ $(X_1 \Rightarrow X_2,\ \ Z_1 \Rightarrow Z_3)$.

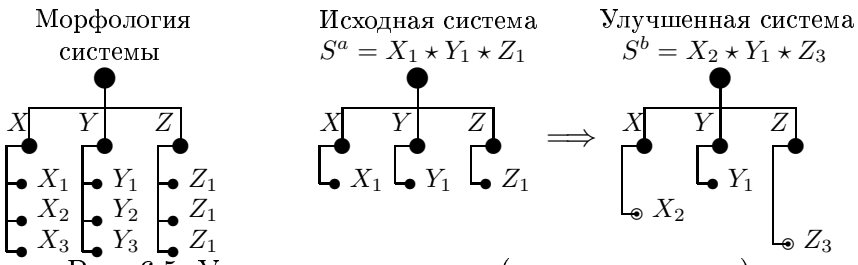


Рис. 6.5. Улучшения системы (по компонентам)

Расширение системы заключается в проектировании дополнительной части системы. Здесь могут быть применены три основные стратегии:

I. Независимое (раздельное) проектирование. Проектирование дополнительной части системы. В результате, новая система включает две части (т.е., исходная часть и дополнительная часть).

II. Обобщенный новое проектирование. Проектирование новой "обобщенной" системы, которая включает исходную часть и дополнительную часть (интегрированное проектирование).

III. Раздельное проектирование с координацией. Проектирование дополнительной части системы, модификация исходной части и координация исходной и дополнительной части системы.

6.2.2 Основные ситуации улучшения и комбинаторные задачи

Таблица 6.1 содержит список основных подходов и соответствующих задач комбинаторной оптимизации для расматриваемых ситуаций улучшения систем (Рис. 6.4).

Таблица 6.1. Ситуации улучшения систем, задачи и модели

Типы улучшения систем	Подходы	Задачи/модели	Источники
Случай 1 (улучшение компонентов системы):			
(1.1) улучшение элементов системы	Замена, перепроектирование	Задачи рюкзачного типа, ИММП, назначение/размещение, перекраска графа	[334, 336, 341] [344, 349, 358] [369, 397]
(1.2) улучшение совместимости элементов	Замена, перепроектирование	Задачи рюкзачного типа, ИММП, назначение/размещение	[336, 344]
(1.3) совместное улучшение элементов системы и их совместимости	Комбинированные подходы	Составные схемы, перекраска графа, ИММП	[336, 344, 349] [358]
Случай 2 (модификация структуры системы):			
(2.1) добавление частей структуры системы	Добавление	Задачи рюкзачного типа, ИММП, добавление горячих связей	[365, 397]
(2.2) удаление частей структуры системы	Удаление	Задачи рюкзачного типа, ИММП	[365]
(2.3) модификация структуры системы	Добавление, удаление, агрегация/интеграция, реструктуризация	Задачи покрытия, задачи рюкзачного типа, ИММП, augmentation problem, реоптимизация	[15, 336, 352, 375] [365, 399, 401]
(2.4) совместный случай	Комбинированные подходы	Составные схемы	[365]
Случай 3 (добавление части системы)	Проектирование	Схемы проектирования, задачи рюкзачного типа, ИММП	[351, 365]
Случай 4 (комбинация предыдущих случаев)	Комбинированные подходы	Схемы из задач	[351, 365]

6.2.3 Дополнительные ситуации улучшения систем

Представляется важным указать дополнительные типовые ситуации улучшения систем: (а) агрегация: построение улучшенной системы как агрегирование нескольких исходных систем (Рис. 6.6, глава 7), (б) проектирование множества улучшенных систем (Рис. 6.7),

(в) совместное агрегирование и проектирование множества улучшенных систем (Рис. 6.8). В случае многокритериальных подходов к проектированию построение множества улучшенных систем может быть основано на получении Парето-эффективных решений, которые могут рассматриваться как множество систем [336, 344, 349, 357].

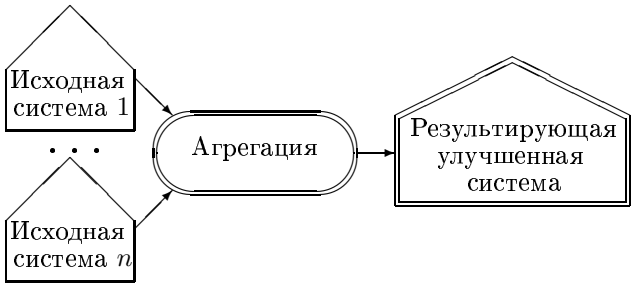


Рис. 6.6. Улучшение системы как агрегация

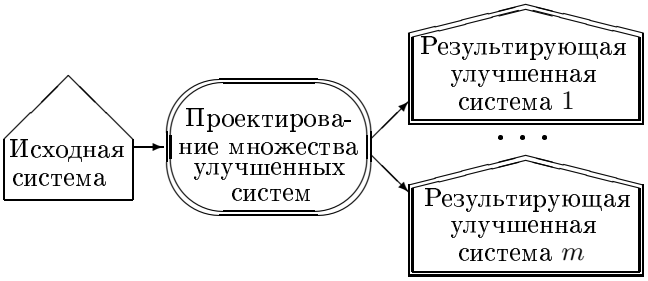


Рис. 6.7. Проектирование множества систем

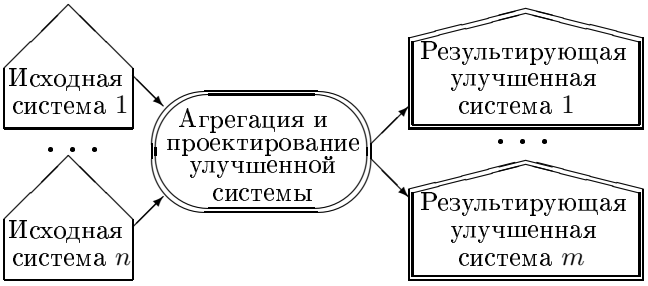


Рис. 6.8. Агрегация/проектирование множества систем

6.3 Подходы к оценке и улучшению систем

6.3.1 Качество системы и улучшение

Процесс улучшения составной (модульной)системы заключается в выборе улучшающих действий (операций) с целью получения повышение общей эффективности (качества, совершенства) системы с учетом общих ограничений на стоимость улучшающих действий. Таким образом, представляется важным рассматривать следующие вопросы:

(i) оценивание эффективности системы (в качестве шкалы для оценивания можно использовать, например, некое “пространство” как шкала, многоразмерное пространство, частичный порядок/решетка)

(ii) выделение узких мест в системе и генерация соответствующих улучшающих действий,

(iii) оценивание улучшающих действий (т.е. их стоимость, полезность), : (i.e., their costs and “profit”),

(iv) комбинаторные задачи улучшения (например, многокритериальный выбор улучшающих действий, задачи рюкзачного типа, комбинаторный синтез в виде выбора и композиции улучшающих действий).

Очевидно, что специальные многошаговые улучшающие процедуры могут также быть использованы.

Можно упомянуть основные подходы к оцениванию общей эффективности (качества, совершенства) систем (глава 3):

1. Функция полезности [190, 290, 533].

2. Многокритериальное описание [290, 470, 492, 533].

3. Шкалы в виде частичного порядка [336, 344, 358].

(3.1) порядковые шкалы для компонентов систем для для совместимости компонентов;

(3.2) порядковые шкалы для компонентов системы, шкала в виде частичного порядка для совместимости компонентов.

4. Шкалы в виде частичного порядка на основе интервальных оценок в виде мультимножеств [360].

(4.1) оценки в виде мультимножеств для компонентов системы и порядковые оценки для совместимости компонентов;

(4.2) оценки в виде мультимножеств для компонентов системы оценки в виде мультимножеств для совместимости компонентов;

Можно привести иллюстрации для указанных методов и их использования для улучшения систем:

Метод 1: шкала для оценивание полезности (Рис. 6.9), исходная система S^a трансформируется в улучшенную систему S^b, эффективность системы обозначена $N(S^a)$ $(N(S^b))$.

Метод 2: многокритериальный случай (два критерия), идеальная точка S^I, четыре Парето-эффективных решения (S_1^p, S_2^p, S_3^p, S_4^p) (Рис. 6.10), три улучшающих процесса:

(i) исходная система S' трансформируется в Парето-эффективное решение S_2^p,

(ii) исходная система S' трансформируется в идеальное решение S^I.

(iii) Парето-эффективное решение S_3^p трансформируется в идеальное решение S^I.

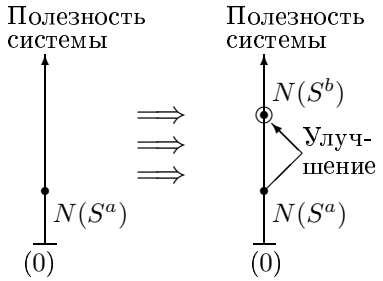


Рис. 6.9. Шкала полезности

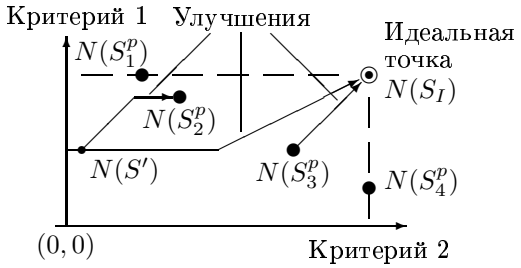


Рис. 6.10. Многокритериальное описание

Метод 3: шкалы качества в виде частичного порядка (или решеток) для оценки качества систем (случай 3.1) [336, 338, 344, 358]. Здесь используется ИММП.

В примере рассматривается трех-компонентная система $S = X \star Y \star Z$. Используются шкалы: (i) порядковая шкала для элементов (приоритеты) $[1, 2, 3]$, (ii) порядковая шкала для совместимости $[1, 2, 3]$. Для данного случая на Рис. 6.11 представлена шкала в виде решетки качества для оценки качества для компонентов и на Рис. 6.12 представлена интегрированная шкала в виде частичного порядка с учетом совместимости компонентов (каждый "треугольник" соответствует частичному порядку с Рис. 6.11). Это *случай 3.1.*

Рис. 6.11 и Рис. 6.12 иллюстрируют улучшающие процессы:

Улучшение A: шкала частичного порядка для качества элементов системы (Рис. 6.11), исходная система S^a $(n(S^a) = (0, 2, 1))$ трансформируется в улучшенную систему (по компонентам) S^b $(n(S^b) = (2, 0, 1))$.

Улучшение B: интегрированная шкала в виде частичного порядка для элементов системы и их совместимости (порядковая шкала используется для оценок совместимости), идеальная точка S^I, три Парето-эффективных решения (S_1^p, S_2^p, S_3^p) (Рис. 6.12), три процесса улучшения:

(а) исходная система S' трансформируется в Парето-эффективное решение S_1^p,

(б) исходная система S' трансформируется в идеальное решение S^I,

(в) Парето-эффективное решение S_3^p трансформируется в идеальное решение S^I.

Используется следующая шкала для качества системы:

(i) идеальная точка;

(ii) Парето-эффективные точки;

(iii) точки, которые являются соседними с Парето-эффективными DAs (например, составное решение этого множество может быть трансформировано в Парето-эффективную точку на основе простых улучшающих операций в виде модификации только одного элемента).

Следует отметить, что совместимость компонентов вектора $N(S)$ может рассматриваться также на основе шкалы частичного порядка (решетки) (как $n(S)$) [338, 344]. В данном случае дискретное пространство качества системы имеет аналогичный вид (решетка).

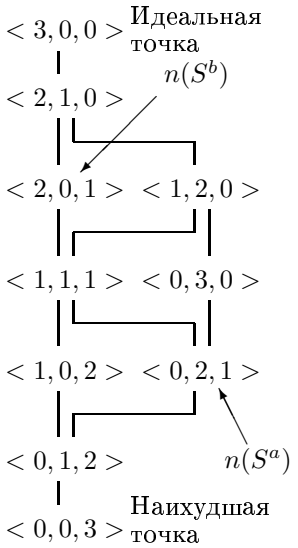


Рис. 6.11. Качество по $n(S)$

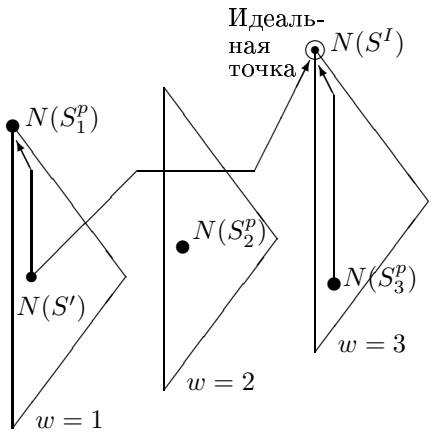


Рис. 6.12. Качество по $N(S)$

Метод 4: Аналогичная область на основе частичного порядка для оценивания (интервальные оценки в виде мультимножеств) DAs и/или их совместимости была предложена в [360] (глава 4).

Такой случай оценивания системы может быть использован в предыдущем методе 3. С другой стороны, этот подход к оцениванию системы является крайне полезным для улучшения по компонентам и для расширения системы. При этом, общая оценка системы рассматривается на основе следующих подходов к агрегации интервальных оценко компонентов системы [360]: (а) интергированная интервальная оценка в виде мультимножеств, (б) медианная интервальная оценка в виде мультимножеств.

На Рис. 6.13 представлен пример такого типа трансформации системы (реконфигурации): (а) замена компонента: $X_1 \Longrightarrow X_2$, (б) удаление компонента: Z_1, (в) добавление двух-компонентной части $U \star V$. На Рис. 6.13 интервальные оценки в виде мультимножеств для компонентов системы и для качества системы изображены в круглых скобках. Медианная интервальная оценка в виде мультимножеств используется для оценивания для качества системы.

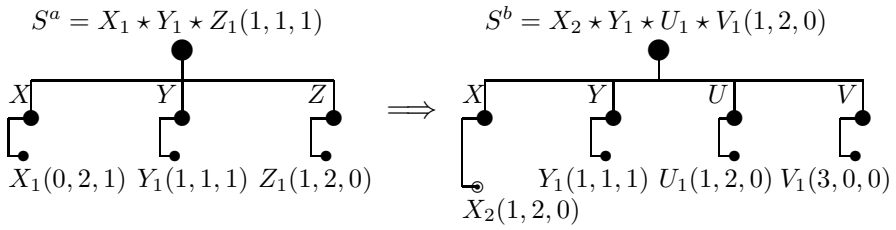


Рис. 6.13. Улучшения системы (по компонентам, по расширению)

6.3.2 О реоптимизации

Последние несколько лет "реоптимизация" стала исследоваться как специальный класс задач комбинаторной оптимизации для ряда известных задач (Таблица 6.2). Формулировка задачи реоптимизации имеет следующий вид:

Дано: (i) пример комбинаторной задачи на графе и соответствующее оптимальное решение, (ii) некие "малые" изменения (perturbations) some "small" perturbations (т.е., модификации) этого примера (например, добавление вершины, удаление вершины).

Вопрос: Возможно вычислить новое хорошее (оптимальное или почти-оптимальное) решение с ограничением на "размер" модификации ?

Таблица 6.2. Исследования реоптимизации

Задача комбинаторной оптимизации	Источники
1.Задача минимального покрывающего дерева	[99]
2.Задачи коммивояжера	[59, 62]
3.Задачи минимального дерева Штейнера	[88, 181]
4.Задачи о покрытии	[89]
5.Задача построения кратчайшей общей надпоследовательности	[90]

Обзор вопросов сложности для задач реоптимизации содержится в [94]. В основном, эти задачи относятся к классу NP-трудных задач и для них предлагаются различные приближенные алгоритмы.

Другой подход к модификации в области задач комбинаторной оптимизации в виде "реструктуризации" был предложен в [352, 375]. Данный подход соответствует многим прикладным задачам перепроектирования в существующих модульных системах. Процесс реструктуризации иллюстрируется на Рис. 6.14 [352, 375].

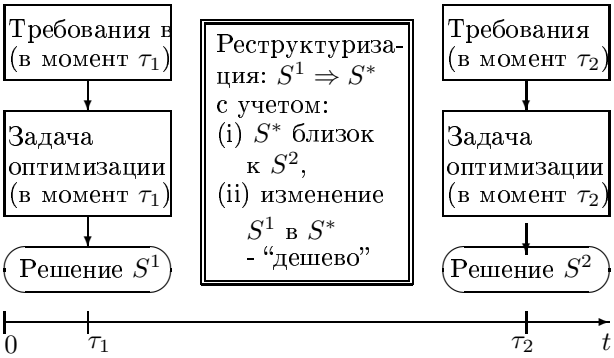


Рис. 6.14. Процесс реструктуризации [352, 375]

Здесь модификация основывается на следующем: включение/удаление элементов (элементов, узлов/вершин, дуг) и изменения структуры. Рассматриваются две характеристики процесса реструктуризации: (i) стоимость реструктуризации решения исходной задачи

(т.е., стоимость отобранных модификаций), (ii) близость полученного реструктуризованного решения к целевому решению.

Рассматриваемый тип задач соответствует задачам перепроектирования (улучшения, обновления) модульных систем и такая ситуация может быть встречена в комплексным программных системах, алгоритмических системах, сетях связи, компьютерных сетях, информационных системах, производственных системах, строительных конструкциях [351, 352, 375].

Соответствующая оптимизационная задача решается для двух моментов времени: τ_1 и τ_2 для получения решений S^1 и S^2. Исследуемая задача реструктуризации заключается в "дешевой" трансформации решения S^1 в решение to S^*, которое является очень близким к S^2. В работе [352, 375], данный подход к реструктуризации описан и иллюстрирован для следующих задач комбинаторной оптимизации: задача о рюкзаке, задача блочного рюкзака, задача о назначении, задачи покрывающих деревьев.

На Рис. 6.15 представлена задача реструктуризации [352, 375].

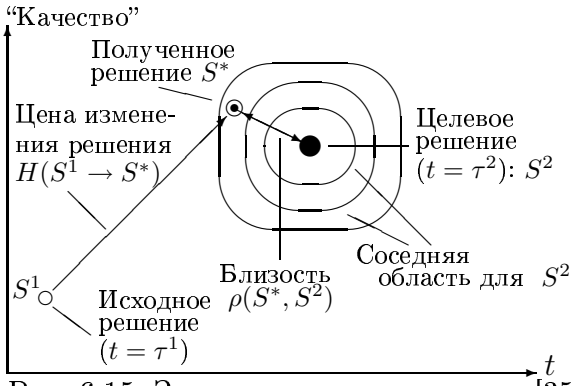


Рис. 6.15. Задача реструктуризации [352, 375]

Пусть P - задача комбинаторной оптимизации с решение в виде структуры S (т.е., подмножество, граф), Ω - исходные данные (элементы, параметры элементов и др.), $f(P)$ - целевая функция(ии). Таким образом, $S(\Omega)$ - решение для исходных данных Ω, $f(S(\Omega))$ - соответствующая целевая функция. Обозначим исходные данные на исходной стадии Ω^1, $f(S(\Omega^1))$ - соответствующая целевая функция, Ω^2 - исходные данные на следующей стадии, $f(S(\Omega^2))$ - соответствующая целевая функция.

Следующие решения могут быть рассмотрены: (а) $S^1 = S(\Omega^1)$ с $f(S(\Omega^1))$ и (б) $S^2 = S(\Omega^2)$ с $f(S(\Omega^2))$. Можно также исследовать цену трансформации одного решения в другое: $H(S^\alpha \to S^\beta)$. Пусть $\rho(S^\alpha, S^\beta)$ - близость (похожесть) между решениями S^α и S^β, например: $\rho(S^\alpha, S^\beta) = |f(S^\alpha) - f(S^\beta)|$. Следует отметить, что функция $f(S)$ является векторной функцией. В результате получается следующая задача реструктуризации (базовая версия):

Найти решение S^* *с учетом следующего* :
(i) $H(S^1 \to S^*) \to \min$, (ii) $\rho(S^*, S^2) \to \min$ (*или ограничение*).

Таким образом, базовая оптимизационная модель может рассматриваться так:

$$\min \rho(S^*, S^2) \quad s.t.\ H(S^1 \to S^*) \leq \widehat{h},$$

где $\widehat{h}$ ограничение для цены изменения решения.

Функция близости $\rho(S^*, S^2)$ может быть рассмотрена как векторная функция (аналогично, для цены изменения решения). Эта задача может быть расширена на случай многокритериальной постановки задачи реструктуризации (в этом случае проводится поиск Парето-эффективных решений).

6.4 Улучшение по системным компонентам

6.4.1 Базовая схема

Улучшение систем на основе компонентов основано на улучшении и/или замене системных компонентов. Базовая схема для указанного типа улучшения имеет вид:

Стадия 1. Выделение множества системных узких мест.

Стадия 2. Генерация улучшающих действий (т.е., улучшение DA, улучшение связей между DAs) и их оценивание.

Стадия 3. Формулировка задачи улучшения системы как задачи комбинаторной оптимизации в виде выбора/комбинации улучшающих действий (операций) (модель: задача блочного рюкзака или метод ИММП для учета связей между улучшающими операциями).

Стадия 4. Решение задачи улучшения системы.

Пример комбинаторного синтеза для составной пяти-компонентной системы представлен на Рис. 6.16. Здесь используется ИММП. Порядковое качество DAs представлено на Рис. 6.16 в круглых скобках (шкала $[1, 2, 3, 4]$). Таблица 6.3 содержит порядковые оценки совместимости (шкала $[0, 3]$).

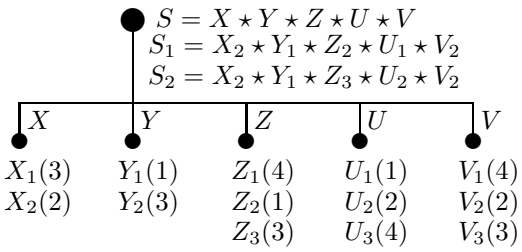


Рис. 6.16. Пример системы

Имеются два результирующих составных Парето-эффективных DAs (Рис. 6.17):

(а) $S_1 = X_2 \star Y_1 \star Z_2 \star U_1 \star V_2$, $N(S_1) = (1; 3, 2, 0, 0)$,

(б) $S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_2$, $N(S_2) = (3; 2, 2, 1, 0)$.

Таблица 6.4 содержит узкие места и улучшающие действия. Далее можно рассмотреть комбинаторный синтез для проектирования плана системного улучшения на основе указанных улучшающих действий (задача блочного рюкзака или ИММП).

Рассматривается процедура улучшения на основе задачи блочного рюкзака для S_2. Это предполагает совместимые улучшающие действия. Таблица 6.5 содержит улучшающие действия и их иллюстративные порядковые оценки.

Соответствующая задача блочного рюкзака имеет вид:

$$\max \sum_{i=1}^{4} \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \ \sum_{i=1}^{4} \sum_{j=1}^{q_i} a_{ij} x_{ij} \le b, \ \ \sum_{j=1}^{q_i} x_{ij} = 1 \ \ \forall i = \overline{1,4}, \ \ x_{ij} \in \{0, 1\}.$$

Очевидно, $q_1 = 2$, $q_2 = 3$, $q_3 = 2$, $q_4 = 2$.

Таблица 6.3. Совместимость

	Y_1	Y_2	Z_1	Z_2	Z_3	U_1	U_2	U_3	V_1	V_2	V_3
X_1	3	3	2	3	2	3	3	0	3	3	2
X_2	3	3	3	3	3	3	3	2	3	3	1
Y_1			3	3	3	3	3	2	3	3	1
Y_2			3	3	3	3	2	1	3	2	2
Z_1						3	1	0	3	1	1
Z_2						3	0	2	3	3	1
Z_3						2	3	0	3	3	1
U_1									3	1	0
U_2									2	3	1
U_3									1	3	2

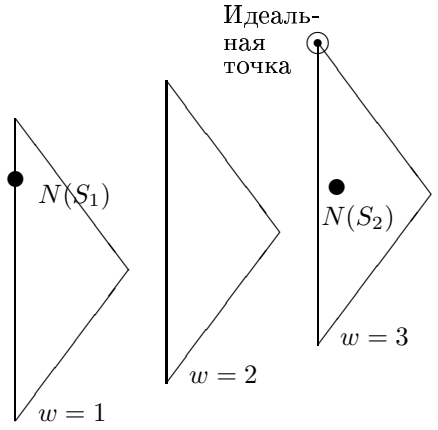


Рис. 6.17. Шкала качества

Таблица 6.4. Узкие места, улучшающие действия

Составные DAs	Узкие места		Действия w/r
	DAs	IC	
$S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_4$	X_2		$2 \Rightarrow 1$
$S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_4$	Z_3		$3 \Rightarrow 2$
$S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_4$	Z_3		$3 \Rightarrow 1$
$S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_4$	U_2		$2 \Rightarrow 1$
$S_2 = X_2 \star Y_1 \star Z_3 \star U_2 \star V_2$	V_2		$2 \Rightarrow 1$
$S_1 = X_2 \star Y_1 \star Z_2 \star U_1 \star V_2$		(U_1, V_2)	$1 \Rightarrow 3$

Таблица 6.5. Улучшающие альтернативы

	DA	Улучшающие действия	Бинарная переменная	Стоимость a_{ij}	Польза c_{ij}
1	X_2	Нет	x_{11}	0	0
2	X_2	$2 \Rightarrow 1$	x_{12}	3	4
3	Z_3	Нет	x_{21}	0	0
4	Z_3	$3 \Rightarrow 2$	x_{22}	1	2
5	Z_3	$3 \Rightarrow 1$	x_{23}	5	4
6	U_2	Нет	x_{31}	0	0
7	U_2	$2 \Rightarrow 1$	x_{32}	4	4
8	V_2	Нет	x_{41}	0	0
9	V_2	$2 \Rightarrow 1$	x_{42}	2	4

Примеры результирующих улучшающих решений, полученных на основе простого жадного алгоритма (упорядочение элементов по невозрастанию c_i/a_i), имеют вид:

(1) $b^1 = 1$: $(x_{11} = 1,\ x_{22} = 1,\ x_{31} = 1,\ x_{41} = 1)$, значение целевой функции $c^{b_1} = 2$, $N(S_2^{b_1}) = (3; 2, 3, 0, 0)$;

(2) $b^2 = 2$: $(x_{11} = 1,\ x_{21} = 1,\ x_{31} = 1,\ x_{42} = 1)$, $c^{b_1} = 4$, $N(S_2^{b_1}) = (3; 3, 1, 1, 0)$;

(3) $b^3 = 7$: $(x_{11} = 1,\ x_{23} = 1,\ x_{31} = 1,\ x_{42} = 1)$, $c^{b_3} = 8$, $N(S_2^{b_3}) = (3; 3, 2, 0, 0)$;

(4) $b^4 = 14$: $(x_{12} = 1,\ x_{23} = 1,\ x_{32} = 1,\ x_{42} = 1)$, $c^{b_4} = 16$, $N(S_2^{b_4}) = (3; 5, 0, 0, 0)$.

6.4.2 Задача перекраски вершин графа

Упрощенная иллюстративная версия улучшения системы по компонентам представляется как задача перекраски вершин графа [349]. Последние годы задачи перекраски вершин графа интенсивно исследуются [67, 76, 236, 437, 438]. Базовая формулировка задачи имеет следующий вид. Имеется следующее: граф $G = (A, E)$ ($|A| = n$), набор (множество) цветов $X = \{x_1, ..., x_k\}$, исходная *конфигурация цветов* для графа G:

$C^o(G) = \{C^o(a_1), ..., C^o(a_i), ..., C^o(a_n)\}$, $C^o(a_i) \in X \ \ \forall a_i \in A \ \ (i = \overline{1, n})$.

Пусть $d_{a_i}(x_{\delta_1}, x_{\delta_2})$ $(\delta_1 = \overline{1,k},\ \delta_1 = \overline{1,k})$ - неотрицательная матрица стоимости перекраски для каждой вершины (т.е., матрица индивидуальной стоимости перекраски $\forall a_i \in A$). Для графа G, целевая *конфигурация цветов* $C^g(G)$ или множество целевых *конфигураций цветов* $\{C^g(G)\}$ также используются. Для каждых двух *комбинаций цветов* ($C^1(G)$ и $C^2(G)$) можно рассмотреть интегрированную стоимость (стоимость перекраски), например, как аддитивную функцию

$$D(C^1(G), C^2(G)) = \sum_{i=1}^{n} d_{a_i}(C^1(a_i), C^2(a_i)).$$

Дополнительно, необходимо рассмотреть близость двух *конфигураций цветов* :

$$\rho(C^{\beta_1}(G), C^{\beta_2}(G)).$$

В результате можно исследовать следующую цепочку трансформации:

$$C^o(G = (A, E)) \Rightarrow C^*(G = (A, E)) \Rightarrow C^g(G = (A, E)),$$

где $C^*(G = (A, E))$ - результирующая *конфигурация цветов* .

В общем случае, целесообразно рассматривать две целевые функции: (i) стоимость трансформации $C^o(G) \Rightarrow C^*(G)$, (ii) качество близости между $C^*(G)$ и $C^g(G)$ (или $\{C^g(G)\}$). Оценивание указанных стоимости и близости может основываться на различных подходах (например, аддитивные функции, минимизация, функции в виде 'min/max', векторная функция).

Теперь можно описать базовую задачу перекраски:

Найти новую конфигурацию цветов $C^*(G)$ для данного графа $G = (A, E)$ *чтобы минимизировать близость результирующей конфигурации цветов графа* $C^*(G(A, E))$ *к целевой конфигурации цветов графа* $C^g(G(A, E))$ *с учетом ограниченной стоимости перекраски* $(\leq \overline{D})$:

$$\min_{\{C(G=(A,E))\}} \rho(C^*(G), C^g(G)))$$

$$s.t.\ \ D(C^o(G), C^*(G)) \leq \overline{D},\ C^*(a_i) \neq C^*(a_j)\ \forall (a_i, a_j) \in E,\ i \neq j.$$

Рис. 6.18 иллюстрирует задачу перекраски графа (его вершин):

$$C^o(G) = (P_2 \star Q_3 \star U_3 \star V_2 \star W_1)\ \Rightarrow\ C^*(G) = (P_1 \star Q_2 \star U_2 \star V_1 \star W_3)\ .$$

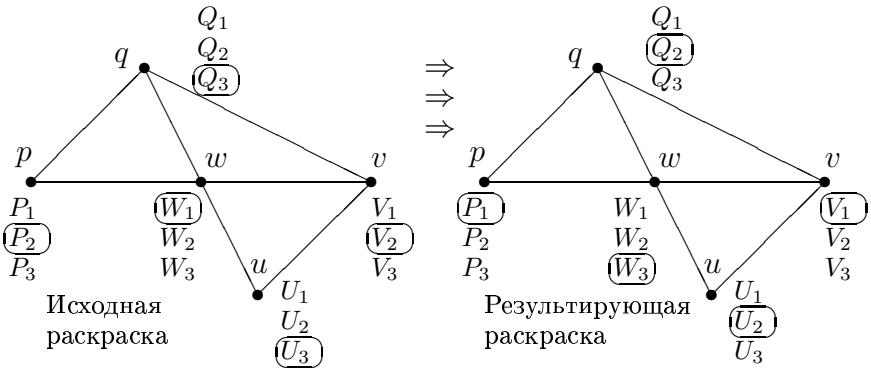


Рис. 6.18. Пример перекраски вершин [349]

6.5 Модификация структуры системы

Модификация структур систем является критической комплексной комбинаторной задачей:

Исходная структура системы $\Longrightarrow$ *Модификация* $\Longrightarrow$ *Финальная структура системы*

Таблица 6.6 содержит список базовых задач модификации структур.

Таблица 6.6. Модификация структур

Задача	Исходная структура	Финальная структура	Источник
1.Модификация древовидных структур:			
1.1.Задача 'горячих' связей (hotlink assignment problem)	Древовидная структура	Древовидная структура с дополнительной связью из корня	[101, 166, 198] [267, 307, 365]
1.2.Модификация дерева: 'склеивание' взвешенных вершин	Взвешенное дерево	Взвешенное дерево с агрегированными вершинами	[15, 16, 336, 365]
1.3.Трансформация дерева в дерево Штейнера	Дерево	Дерево Штейнера	[365, 399, 401]
2.Трансформация морфологической структуры	Морфологическая структура	Морфологическая структура с требуемыми свойствами	[336, 344, 365]
3.Трансформация уровневых структур	Уровневая структура	Уровневая структура с требуемыми свойствами	
4.Задача пополнения графа (graph augmentation problem)	Связный граф	Связный граф с требуемыми свойствами (т.е., с улучшенной связностью)	[182, 296]

6.5.1 Модификация дерева посредством склеивания вершин

Данный раздел [2] содержит краткое описание трансформации дерева (с весами вершин и весами ребер/дуг) посредством интеграции (конденсации) некоторых соседних вершин су учетом ограничений на общий вес максимального хвоста дерева (т.е., длина от корня дерева до висячей вершины). Данная задача была впервые сформулирована как проектирование оверлейной структуры для модульной программной системы [15, 16]. Интегрирование (склеивание) программных модулей требует дополнительной памяти, но это позволяет уменьшить время (т.е., частоту) загрузки некоторых соответствующих модулей. Также могут рассматриваться другие приложения этой модели, например: построение иерархической структуры данных, иерархическая структура сайта в Интернете, задача вызова (call problem). На Рис. 6.19 и Рис. 6.20 содержится иллюстрация на примере проектирования оверлейной структуры на основе интеграции модулей, когда различные программы или модули данных могут использовать одни и те же части ОЗУ (RAM). Для данной комбинаторной оптимизационной задачи (обобщение задачи блочного рюкзака на древовидной

[2]From (with amendments):
(i) M.Sh. Levin, An extremal problem of organization of data. *Eng. Cybern.*, 19(5), 1981, 87-95.
(ii) M.Sh. Levin, *Combinatorial Engineering of Decomposable Systems*, Springer, 1998, Chapter 2.

структуре и при специальных ограничениях) был предложен новый тип приближенного полиномиального алгоритма с ограниченной относительной погрешностью (FPTAS) [15, 16].

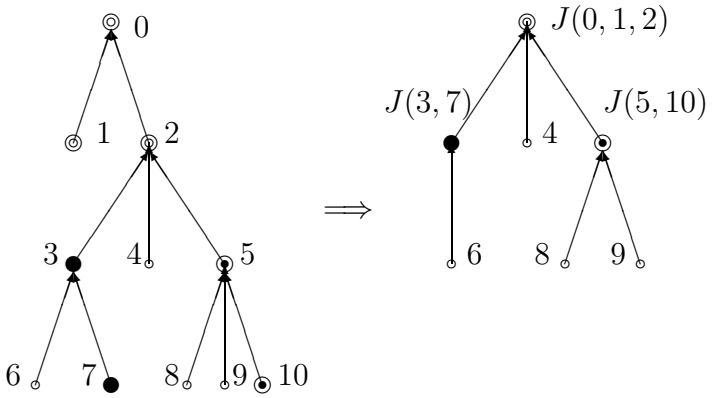


Рис. 6.19. Преобразование программы (оверлейная структура)

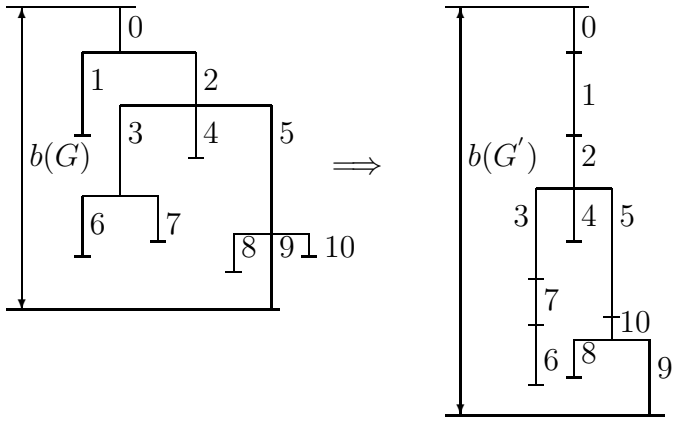


Рис. 6.20. Использование памяти (ОЗУ) [15, 16, 336]

Пусть $G = (A, \Upsilon)$ - ориентированное дерево, где A - множество вершин (модули программ или данных) ($|A| = n$). Пусть Υ - многозначное отображение A в A. Дуги в G ориентированы от корня $a_o \in A$ к висячим вершинам. Каждая вершина $a \in A$ имеет положительных вес (требуемый объем оперативной памяти ОЗУ - RAM) $\beta(a) > 0$. Каждая дуга $(a^{'}, a^{''})$ ($a^{'}, a^{''} \in A$ и $a^{''} \in \Upsilon a^{'}$) имеет вес (т.е., исходную частоту загрузки в оперативную память) $w(a^{'}, a^{''}) > 0$. Этот вес дуги соответствует частоте вызова/обращения (и загрузки) из модуля $a^{'}$ к модулю $a^{''}$.

Пусть $\pi(a^1, a^l) =< a^1, ..., a^i, ..., a^l >$ будет путь ($a^{j+1} \in \Upsilon a^j, j = 1, ..., l-1$). Для каждого пути вводится вес $\lambda(\pi(a^1, a^l) = \sum_{i=1}^{l} \lambda(a^i)$. Весом графа G будет значение

$$\lambda(G) = max_{a^{''} \in A^o}\{\lambda(\pi(a_o, a^{''}))\},$$

где $A^o = \{a \in A \mid \Upsilon a = \emptyset\}$ - множество висячих вершин. Пусть $G_a = (A_a, \Upsilon)$ обозначает поддерево с корнем $a \in A$, и A_a содержит вершину a и все другие вершины, которые могут достигаться из a. Граф $(A_a \setminus a, \Upsilon)$ будет называться *хвост* вершины a, и значение $\lambda^-(a) = \lambda(G_a) - \lambda(a)$ будет называться весом *хвоста* вершины a. Очевидно, что

$$\lambda(a) = max_{a' \in \Upsilon a}\{\lambda(G_{a'})\}.$$

Далее используются вес $w(a)$ и бинарная переменная $x(a)$ $\forall a \in A \setminus a_o$ (1 соответсвтует ситуации, когда дуга, направленная к a, конденсируется - сжимается). Теперь можно определить трансформацию графа G на основе интеграции вершин $a^{'}$ и $a^{''}$:

(а) вершина $a^{'}$ изменяется на $J(a^{'}, a^{''})$ с следующими свойствами:
$\lambda(J(a^{'}, a^{''})) = \lambda(a^{'}) + \lambda(a^{''})$ $\quad \Upsilon J(a^{'}, a^{''}) = (\Upsilon a^{'} \bigcup \Upsilon a^{''}) \setminus a^{''}$;
(б) вершина $a^{''}$ и дуги, которые идут из этой вершины, удаляются.

Для графа G рассматривается бинарный вектор $\kappa(a)$, который включает все $x(a)$ $\forall a \in A \setminus a_o$. Таким образом, исследуются веса вершины a и ее *хвоста* как функции вектора κ: $\lambda(a, \kappa), \lambda^{-}(a, \kappa)$. Теперь можно рассмотреть задачу (тип 1):

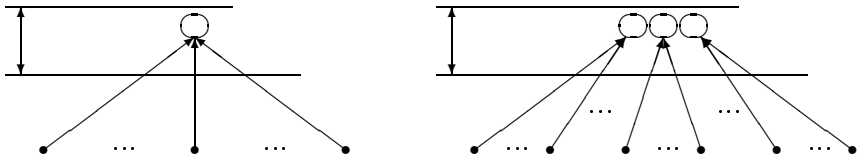
$$\max\ W(\kappa) = \sum_{a \in A \setminus a_o} x(a)w(a) \quad s.t.\ \lambda(a_o, \kappa) + \lambda^{-}(a_o, \kappa) \leq b,$$

где b - положительная константа (т.е., объем доступной оперативной памяти - ОЗУ). Эта формулировка задачи соответствует примеру на Рис. 6.19 и Рис. 6.20.

Дополнительно, рассматривается аналогичная задача (тип 2) с другими ограничениями:

$$\lambda(a_o, \kappa) \leq b^{-},\ \lambda^{-}(a_o, \kappa) \leq b^{+},\ b^{-} + b^{+} = b.$$

Следует отметить, иллюстрации для данного класса рассматриваемых комбинаторных моделей представлены на Рис. 6.21 (базовые задача о рюкзаке и задача блочного рюкзака) и на Рис. 6.22, соответствие задачи и иллюстрации представлено в Таблице 6.7.



(i) задача о рюкзаке (1) (ii) задача блочного рюкзака (2)

Рис. 6.21. Задачи о рюкзаке и блочного рюкзака [15, 16, 336]

Теперь можно рассмотреть некоторые простые случаи задачи (тип 1 и тип 2). Пусть $\Upsilon a_o = \{a_1, ..., a_i, ..., a_m\}$ (и u_i) соответствует некоторой дуге (a_o, a_i) $(w(u_i) = w_i)$. Тогда соответствующая задача (задача 1, эквивалент задачи о рюкзаке, Рис. 6.21i) имеет вид:

$$\max \sum_{i=1}^{m} x_i w_i \quad s.t.\ \lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) \leq b, \quad x_i \in \{0, 1\}.$$

Целевая функция в других простых случаях (1.1 - Рис. 6.22a, 1.2 - Рис. 6.22b, 1.3 - Рис. 6.22c, 1.4 - Рис. 6.22d), которые основаны на задаче о рюкзаке (задача 1, Рис. 6.21i), является аналогичной; далее только ограничения будут указываться для этих задач.

Задача 1.1 (Рис. 6.22a) включает следующее ограничение:

$$\lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) + \max_{1 \leq i \leq m} ((1 - x_i)\lambda(a_i)) \leq b.$$

Эта задача соответствует формированию загрузочного “программного ядра” во многих программных пакетах.

Задача 1.2 (Рис. 6.22b) типа 2 имеет вид:

$$\lambda(a_o) + \sum_{i=1}^{m} x_i \lambda(a_i) \leq b^{-}, \quad \max_{1 \leq i \leq m} ((1 - x_i \lambda(a_i)) \leq b^{+}.$$

Задача 1.3 (Рис. 6.22c) имеет вид:

$$\lambda(a_o + \sum_{i=1}^{m} x_i \lambda^{-}(a_i) + \max_{1 \leq i \leq m} ((1 - x_i \lambda^{-}(a_i)) + \lambda^{+}(a_i)) \leq b.$$

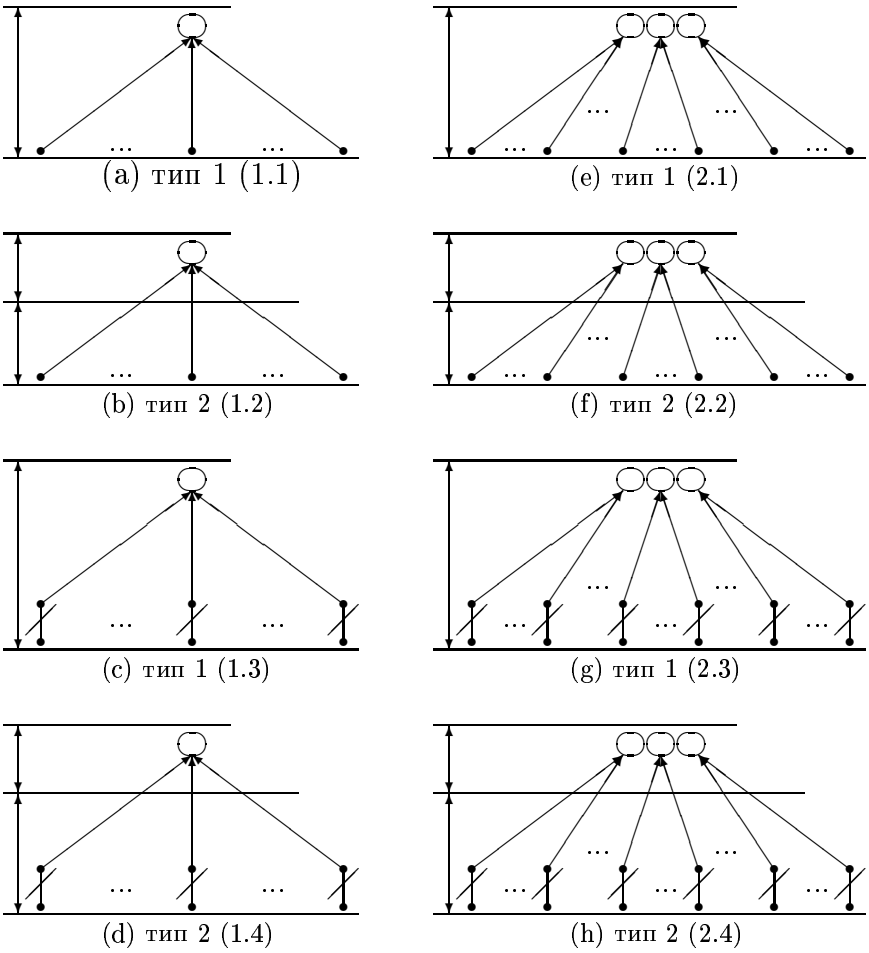


Рис. 6.22. Простейшая и вспомогательные задачи [15, 16, 336]

Таблица 6.7. Задачи рюкзачного типа - иллюстрации

Задача	Иллюстрации
1.Базовая задача о рюкзаке (*задача 1*)	Рис. 6.21i
2.*Задача 1.1*	Рис. 6.22a
3.*Задача 1.2*	Рис. 6.22b
4.*Задача 1.3*	Рис. 6.22c
5.*Задача 1.4*	Рис. 6.22d
6.Задача блочного рюкзака (*задача 2*)	Рис. 6.21ii
7.*Задача 2.1*	Рис. 6.22e
8.*Задача 2.2*	Рис. 6.22f
9.*Задача 2.3*	Рис. 6.22g
10.*Задача 2.4*	Рис. 6.22h

Представляется важным указать следующие свойства этой задачи:

(а) a_i ($\forall a_i \in \Upsilon a_o$) имеет вес $\lambda^-(a_i)$;

(б) a_i ($\forall a_i \in \Upsilon a_o$) имеет только одного сына с весом $\lambda^+(a_i)$, и значение соответствует весу *хвоста*;

(в) только конденсация (сжатие) следующих дуг является допустимым (a_o, a_i) $(i = \overline{1, m})$.

В результате, последовательность простых задач на основе задачи о рюкзаке может быть рассмотрена (Таблица 6.7): 1 (базовая задача о рюкзаке, Рис. 6.21i), 1.1 (аналог задачи о рюкзаке, Рис. 6.22a), 1.2 (Рис. 6.22b), 1.3 (Рис. 6.22c), 1.4 (Рис. 6.22d).

Аналогично, можно указать последовательность вспомогательных задач на основе задачи блочного рюкзака (Таблица 6.7): 2 (базовая задача блочного рюкзака, Рис. 6.21ii),

2.1 (аналог задачи блочного рюкзака, Рис. 6.22e), 2.2 (Рис. 6.22f), 2.3 (Рис. 6.22g), 2.4 (Рис. 6.22h).

В данном случае, задача блочного рюкзака или задача 2 (Рис. 6.21ii) имеет вид:

$$\max\ W(\{x_{ij}\}) = \sum_{i=1}^{m}\sum_{j=1}^{q_i} w(a_{ij})x_{ij}$$

$$s.t.\ \ \lambda(a_o)\sum_{i=1}^{m}\sum_{j=1}^{q_i} x_{ij}\lambda(a_{ij}) \le b,\ \sum_{j=1}^{q_i} x_{ij} = 1,\ i=\overline{1,m};\ x_{ij}\in\{0,1\}.$$

В вспомогательных задачах используются следующее множество Булевых векторов:

$$X = \{\kappa = (x_{ij}^1; x_{ij}^2) | x_{ij}^1, x_{ij}^2 \in \{0,1\};\ j=\overline{1,q_i};\ i=\overline{1,m}\}$$

Дополнительно, следующее ограничение должно быть учтено для всех вспомогательных задач:

$$\sum_{j=1}^{q_i} x_{ij}^1 = 1,\ \forall i;\ x_{ij}^2 \le x_{ij}^1,\ \forall i,j.$$

Также используется следующая модифицированная целевая функция:

$$W(X) = \sum_{i=1}^{m}\sum_{j=1}^{q_i}(x_{ij}^1 w^-(a_{ij}) + x_{ij}^2 w(a_{ij})).$$

Теперь можно рассмотреть вспомогательную задачу, например задачу 2.4, которая соответствует типу 2 (Рис. 6.22h):

$$\lambda(a_o) + \sum_{i=1}^{m}\sum_{j=1}^{q_i} x_{ij}^2\lambda^-(a_{ij}) \le b^-,\ \ \max_{i,j}((1-x_{ij}^2)\lambda^-(a_{ij}) + \lambda^+(a_{ij})) \le b^+.$$

Для последовательности простых задач, можно использовать приближенные алгоритмы, которые основываются на ϵ-приближенном алгоритме ($\epsilon \in [0,1]$) для задачи о рюкзаке [291, 419, 499, 500]. В указанных алгоритмах оценки числа вычислительных операций подобна оценке для задачи о рюкзаке [291, 419] и равна $O(\frac{m^2}{\epsilon})$ [15, 16, 336]. Эти алгоритмы используют упорядочение элементов из множества Υa по невозрастанию $\lambda(a_i)$ или $(\lambda^-(a_i) + \lambda^+(a_i))$.

Подход к решению вспомогательных задач основан на подобном приближенном подходе к задаче блочного рюкзака с следующими оценками числа вычислительных операций и требуемым объемом оперативной памяти соответственно [291, 419]:

$$O(\frac{m}{\epsilon}\sum_{i=1}^{m} q_i),\ O(\frac{m^2}{\epsilon}\max_{1\le i\le m}\{q_i\}).$$

К сожалению, алгоритм с аналогичной оценкой сложности для задачи 2.3 не был построен (Рис. 6.22g). Поэтому (ϵ,δ)-приближенный алгоритм с следующими оценками (число операций, требуемый объем памяти) был предложен [15, 16, 336]:

$$O(\frac{m}{\epsilon\delta}\sum_{i=1}^{m} q_i),\ \ O(\frac{m^2}{\epsilon}\max_{1\le i\le m}\{q_i\}),$$

где δ - относительная погрешность по ограничению.

В случае, когда G - k-уровневое дерево, алгоритм основан на каскадной схеме 'СНИЗУ-ВВЕРХ' (Рис. 6.23) [15, 16, 336]:

Шаг 1. Задача 1.2.

. . .

Шаг j $(j = \overline{2, k-2})$. Задача 2.4.

. . .

Шаг $(k-1)$. Задача 2.3.

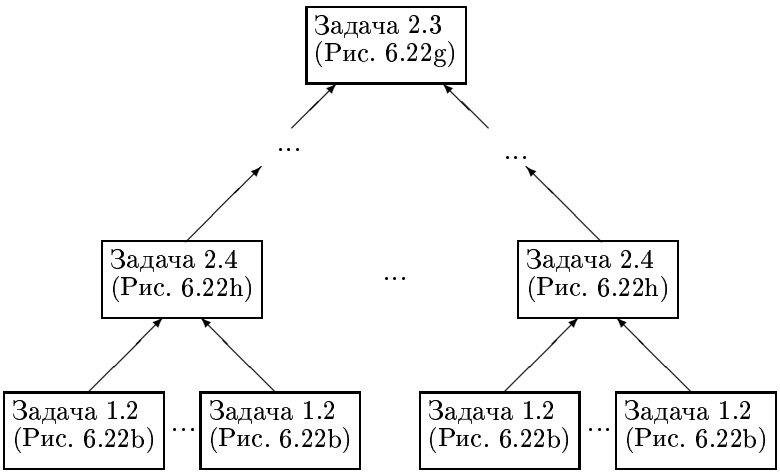


Рис 6.23. Схема 'СНИЗУ-ВВЕРХ' для древовидного графа [15, 16, 336]

Оценки сложности алгоритмов имеют вид (т.е., число операций, объем требуемой памяти):

$$O(\frac{n^2\eta^5(a_o)}{\epsilon\delta^4}),\ O(\frac{m^2\eta^4}{\epsilon\delta^4}),$$

где $m(a) = |\Upsilon(a)|$, $m = \max_{a\in A} m(a)$, $\eta(a) =| A_a \setminus \{a' \in A_a \mid \Upsilon a' = \emptyset\} |$.

В случае 3-уровневого дерева, оценки имеют вид: $O(\frac{n^2\eta^4(a_o)}{\epsilon\delta^3})$.

Перспективные обобщения рассмотренной задачи могут включать следующее:

(а) многокритериальное описание элементов,

(б) более сложная структура (например, параллельно-последовательный граф),

(в) неопределенность,

(г) динамика.

6.5.2 Задача назначения 'горячих' связей

В последние годы была предложена концепция 'горячих' связей для уменьшения сложности доступа в информационных структурах (например, в web-директориях) посредством использования ограниченного числа дополнительных связей ('горячих' связей) в структурах данных [101, 165, 475]. В общем случае, задача назначения 'горячих' связей может рассматриваться как задача обновления (upgrade) сети [198]:

Найти дополнительную дугу(и) для исходного графа как короткий путь к данным с целью уменьшения среднего ожидаемого пути.

Обычно, данная задача рассматривается для деревьев. Пусть $T = (A, E)$ обозначает направленное дерево с максимальной степенью d, корнем $r_0 \in A$ (например, элементы A соответствуют Web сайтам, элементы E соответствуют связям - hyperlinks). Вес узла

равен частоте (вероятности) обращения (доступа, поиска) к нему. Для упрощения предполагается, что требуемая информация содержится в висячих вершинах (узлах). Длина поиска для узла $v \in A$ равна числу дуг в пути из корня r_0 в v.

Пусть $T_u = (A_u, E_u)$ будет поддерево T ($A_u \subseteq A, E_u \subseteq E$), корень расположен в узле $u \in A$ (здесь, u не является сыном r_0). Таким образом, дополнительная прямая связь ("hotlink") имеет вид: (r_0, u). При этом пути ко всем висячим узлам в T_u будут меньшей длины.

Рис. 6.24, Рис. 6.25, Рис. 6.26, и Рис. 6.27 иллюстрируют упрощенные версии задачи горячих связей. На Рис. 6.27 представлено использование внутренних узлов как дополнительных корней (r_1).

Базовая задача назначения горячих связей заключается в назначении k дополнительных связей (из корня) с целью минимизации общего числа шагов для посещения требуемых информационных узлов. С другой стороны, необходимо найти множество k узлов ($U = \{u\}$) в дереве T. В последние годы такие задачи горячих связей интенсивно исследуются (например, построение алгоритмов, вопросы сложности, приближения) [101, 154, 166, 165, 198, 267, 308, 318, 421, 475]. Некоторые версии данного типа задач указаны в Таблице 6.8. В основном, эти задачи относятся к классу NP-трудных задач [267]. Для таких задача предложены приближенные алгоритмы (включая FPTAS) [308, 421].

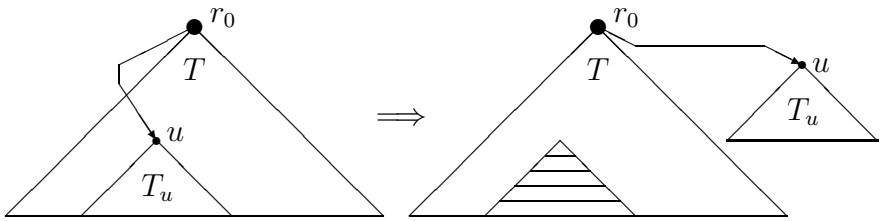


Рис. 6.24. Задача горячей связи (одна связь)

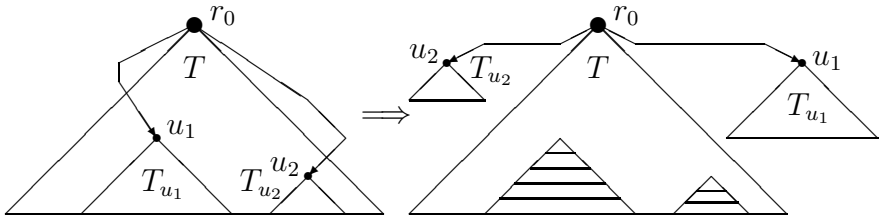


Рис. 6.25. Задача горячей связи (две связь)

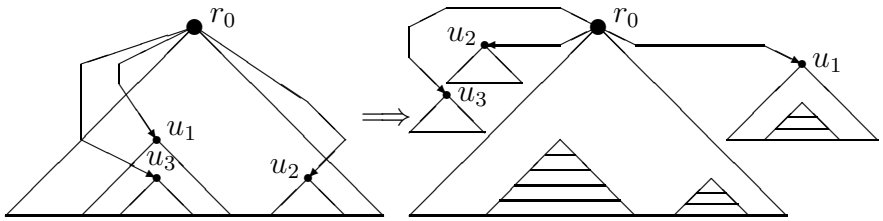


Рис. 6.26. Задача горячей связи (три связи)

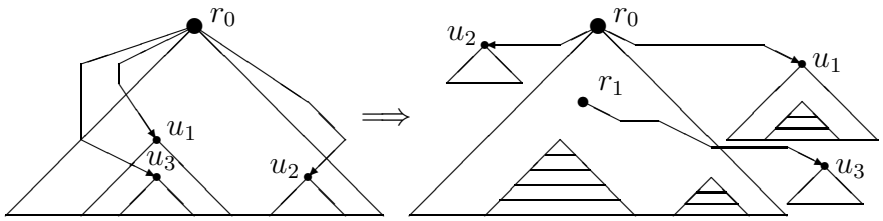


Рис. 6.27. Задача горячей связи (три связи, внутренний корень)

При многокритериальном описании горячих связей и/или выбранных узлов/ поддеревьев (т.е., узлов как u), многокритериальные задачи о рюкзаке или многокритериальные обобщенные задачи назначения могут использоваться [395, 397, 412, 471]. Возможно рассматривать некоторые обобщения задачи горячих связей, например: (i) учетом неопределенности, (ii) задача включения дерева ("hot-tree assignment" problem).

Таблица 6.8. Задачи горячих связей

Задача	Источники
1.Базовая задача горячей связи	[101, 475]
2.Задача с одной связью	[165]
3.Горячая связь только для висячих узлов	[267]
4.Задача с горячими многими связями	[165, 198]
5.Динамическая задача горячих связей	[166]

С общей точки зрения задача горячей связи может рассматриваться как специальный случай задачи пополнения графа ("graph augmentation problem") [182, 296].

6.5.3 Трансформация дерева в дерево Штейнера

Здесь трансформация дерева $T = (A, E)$ в дерево Штейнера $S = (A', E')$ рассматривается как дополнение точек Штейнера в исходное дерево (или предварительно построенное покрывающее дерево) с учетом следующего: "стоимость" (требуемые ресурсы) для каждой точки Штейнера, "полезность" каждой точки Штейнера, общее ограничение по ресурсу (т.е., общая "стоимость" выбранных точек Штейнера) [365, 401]. Рассматривается простейший случай, когда точки Штейнера используются только для "треугольников". Очевидно, что модели с векторными "стоимостью" и "полезностью" могут быть также рассмотрены. Схема решения имеет вид:

Стадия 1. Определение (экспертное оценивание, кластеризация) m регионов (кластеры, группы соседних узлов) в исходном дереве T для возможных дополнительных точек Штейнера.

Стадия 2. Генерация возможных точек Штейнера (кандидатов) и их параметров (т.е., стоимость добавления точек, "полезность").

Стадия 3. Формулирование задачи блочного рюкзака для выбора наилучших дополнительных точек Штейнера с учетом ограничений по ресурсам:

$$\max \sum_{i=1}^{m}\sum_{i=1}^{q_i} c_{ij}x_{ij} \quad s.t. \quad \sum_{i=1}^{m}\sum_{j=1}^{q_i} a_{ij}x_{ij} \le b, \quad \sum_{j=1}^{q_i} x_{ij} = 1, \quad x_{ij} \in \{0,1\};$$

где i - индекс региона ($i = \overline{1,m}$), q_i - число версий для добавления точек Штейнера в регионе $i = \overline{1,m}$, j - индекс версии для добавления точек Штейнера в регионе ($j = \overline{1,q_i}$), x_{ij} - бинарная переменная, равная 1, если выбирается версия j в регионе i, b - общее ограничение по требуемым ресурсам (т.е., общая "стоимость").

Стадия 4. Решение задачи блочного рюкзака для получения результирующего дерева Штейнера S.

Численный пример иллюстрирует данную схему. Исходное дерево имеет вид (Рис. 6.28): $T = (A, E)$, $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$. Определены четыре региона (Рис. 6.28): регион 1: $\{1, 2, 3, 4\}$; регион 2: $\{4, 6, 7\}$; регион 3: $\{4, 5, 6, 9, 11\}$; регион 4: $\{7, 8, 10\}$. Рассматриваются следующие точки Штейнера (Рис. 6.29): регион 1: s_{11}, s_{12}; регион 2: s_{21}; регион 3:

s_{31}, s_{32}; регион 4: s_{41}. Таблица 6.9 содержит исходные данные для задачи блочного рюкзака: бинарные переменные и соответствующие параметры (требуемый ресурс "стоимость", возможная "полезность"). Задача имеет вид:

$$\max \sum_{i=1}^{4}\sum_{i=1}^{q_i} c_{ij}x_{ij} \quad s.t. \quad \sum_{i=1}^{4}\sum_{j=1}^{q_i} a_{ij}x_{ij} \le b, \quad \sum_{j=1}^{q_i} x_{ij} = 1, \quad x_{ij} \in \{0,1\}.$$

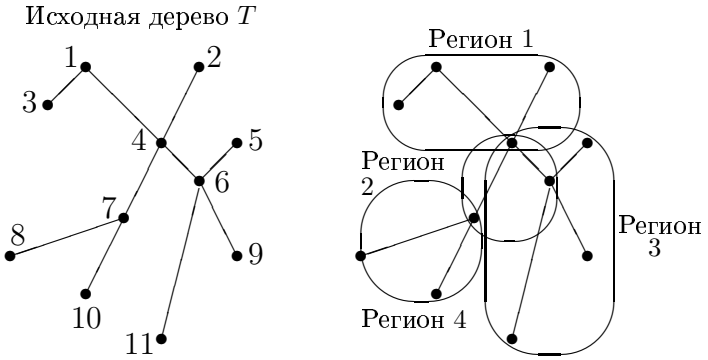


Рис. 6.28. Исходное дерево и регионы (кластеры)

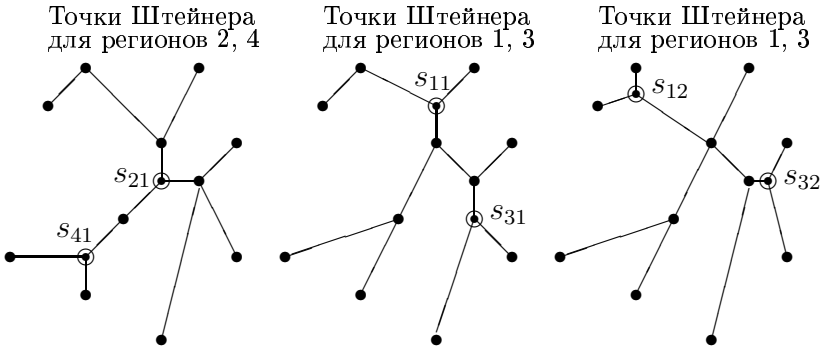


Рис. 6.29. Дополнительные точки Штейнера для регионов

Таблица 6.9. Данные для задачи блочного рюкзака

Регион	Бинарная переменная	Точка Штейнера	"Стоимость" c_{ij}	"Полезность" a_{ij}
Регион 1	x_{11}	Нет	0.0	0.0
	x_{12}	s_{11}	3.1	1.5
	x_{13}	s_{12}	1.2	1.4
Регион 2	x_{21}	Нет	0.0	0.0
	x_{22}	s_{21}	2.0	1.3
Регион 3	x_{31}	Нет	0.0	0.0
	x_{32}	s_{31}	2.4	1.4
	x_{33}	s_{32}	1.8	1.3
Регион 4	x_{41}	Нет	0.0	0.0
	x_{42}	s_{41}	1.5	1.2

На основе применения простых жадных алгоритмов были получены следующие решения (в виде множества дополнительных точек Штейнера) (Рис. 6.30):

(1) $b_1 = 2.9$: $\overline{x}_{b_1}$: $x_{12} = 1$, $x_{21} = 1$, $x_{32} = 1$, $x_{41} = 1$, точки Штейнера $Z_{b_1} = \{s_{11}, s_{31}\}$, общая (аддитивная) "полезность" $\overline{c} = 5.5$;

(2) $b_2 = 4.2$: $\overline{x}_{b_2}$: $x_{12} = 1$, $x_{22} = 1$, $x_{32} = 1$, $x_{42} = 0$; точки Штейнера $Z_{b_2} = \{s_{11}, s_{21}, s_{31}\}$, общая (аддитивная) "полезность" $\overline{c} = 7.5$;

(3) $b_3 = 5.4$: $\overline{x}_{b_3}$: $x_{12} = 1$, $x_{22} = 1$, $x_{32} = 1$, $x_{42} = 1$, точки Штейнера $Z_{b_3} = \{s_{11}, s_{21}, s_{31}, s_{41}\}$, общая (аддитивная) "полезность" $\overline{c} = 9.0$.

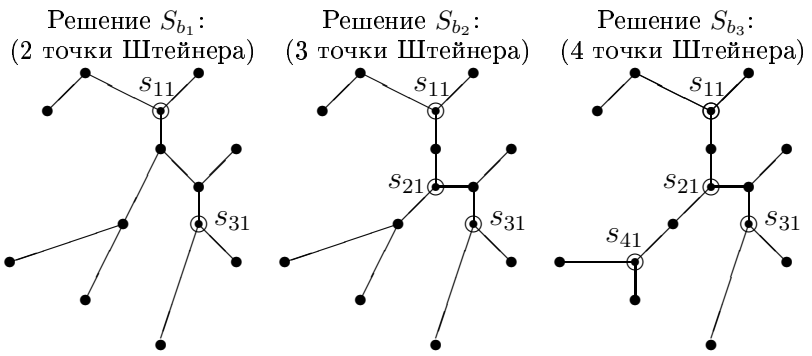


Рис. 6.30. Решения

На основе предложенного подхода могут быть исследованы более сложные задачи с учетом следующего: (i) несколько дополнительных точек Штейнера в одном регионе, (ii) совместимость дополнительных точек в соседних регионах.

6.5.4 Задача пополнения графа

Цель известной задачи пополнения графа ("graph augmentation problem") заключается в модификации исходного графа (сети), например, посредством ребер, так, чтобы получаемый новый граф удовлетворял некоторым требованиям (в частности, улучшение связности) [182, 296].

6.5.5 Трансформация морфологической структуры

Структура морфологической системы содержит следующие части [336, 344, 357, 358, 365]: (1) иерархию системы (обычно дерево), (2) множество висячих узлов (т.е., системных компонентов), (3) множество алтернатив для каждого висячего узла, (4) оценки альтернатив (в частности, порядковые оценки в виде приоритетов, интервальные оценки), (5) оценки совместимости для пар альтерантив. Пример структуры морфологической системы представлен на Рис. 6.16 и в Таблице 6.3. На Рис. 6.31 представлена трансформация структуры морфологическоу системы: $S' \Rightarrow S''$. При этом преобразуется следующее:

(а) иерархия системы (дерево) $T' \Rightarrow T''$,

(б) базовые системные компоненты (висячие узлы) $L' = \{1, ..., i, ..., m'\} \Rightarrow L'' = \{1, ..., i, ..., m''\}$.

Численный пример трансформации структуры системы (как реконфигурация системы) приведен на Рис. 6.13.

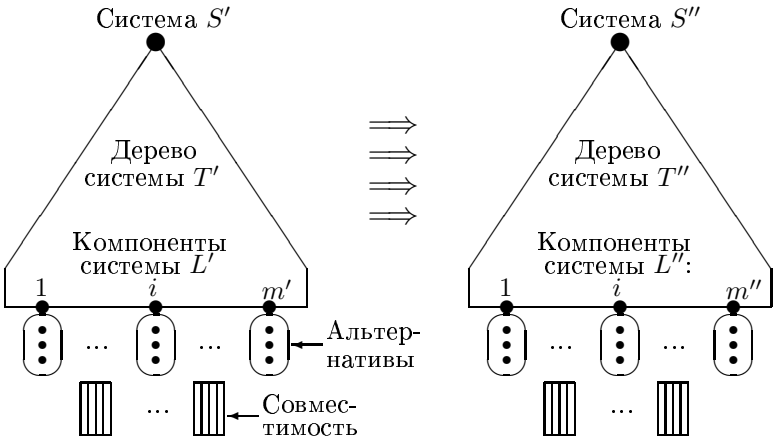


Рис. 6.31. Трансформация морфологической структуры

6.5.6 Трансформация уровневой структуры

Многоуровневая структура системы рассматривается как композиция следующих частей: (а) множество уровней, (б) для каждого уровня: множество узлов, топология над узлами, (в) связи узлов на соседних уровнях.

Процесс модификации уровневых структур включает следующее:

1. Модификация уровневой структуры: (1.1) добавление уровня, (1.2) удаление уровня.

2. Модификация уровня: (2.1) добавление узлов уровня, (2.2) удаление узлов уровня, (2.3) модификация уровневой топология (сеть над узлами уровня).

3. Модификация связей между уровнями (т.е., связь мезду узлами соседних уровней): переназначение.

На Рис. 6.32 представлен процесс модификации многоуровневой системы $S' \Rightarrow S''$.

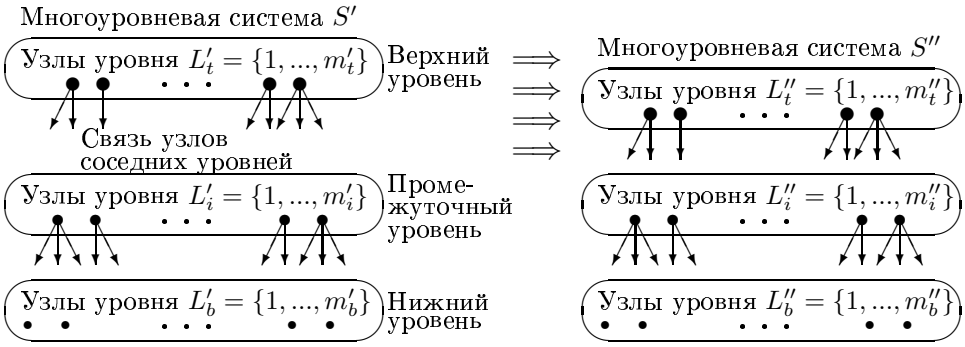


Рис. 6.32. Трансформация многоуровневой структуры

6.6 Выводы к главе

В данной главе представлены основные схемы для улучшения/расширения модульных систем. В качестве возможных исследовательских направлений в будущем можно указать следующие:

1. исследование различных реальных приложений;

2. исследование многостадийных стратегий для улучшения/расширения модульных систем;

3. учет неопределенности.

Глава 7

Схемы агрегирования модульных решений

В данной главе [1] приведено описание комбинаторных схем агрегации модульных решений, включая агрегацию ранжировок, деревьев, морфологических структур.

7.1 Предварительные сведения

В данной главе генерация новых модульных проектных решения рассматривается как агрегация некоторого множества исходных модульных решений (Рис. 7.1):

Найти агрегированное составное (модульное) решение S^{agg} на основе множества исходных составных (модульных) решений $\{S_1, ..., S_\tau, ..., S_m\}$, т.е.,

$\{S_1, ..., S_\tau, ..., S_m\} \Longrightarrow S^{agg}$.

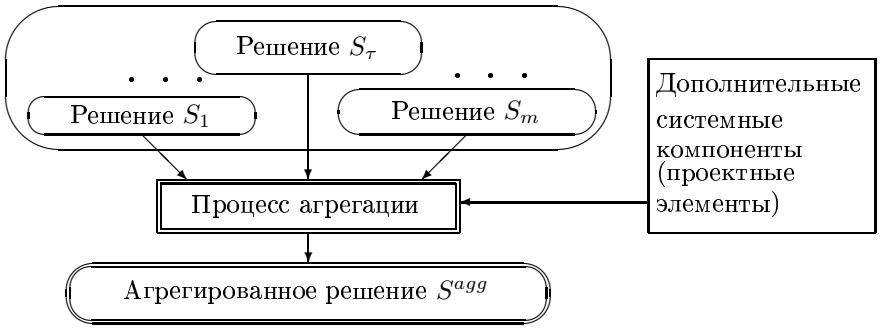


Рис. 7.1. Общая схема процесса агрегации

Следует отметить, что процесс агрегации должен быть основан на анализе приложения (инженерного или управленческого) (Рис. 7.2).

[1] Материал главы основан на публикации: Levin M.Sh., Aggregation of composite solutions: strategies, models, examples. Electr. prepr., 72 p., Nov. 29, 2011. http://arxiv.org/abs/1111.6983 [cs.SE]

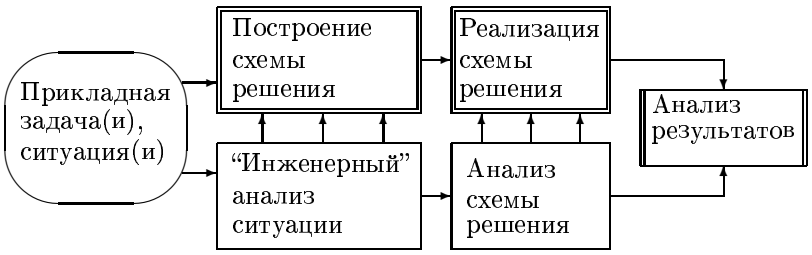


Рис. 7.2. "Инженерный" (экспертный) подход к решению

Можно рассматривать два базовых случая агрегации решений:

Случай 1. Использование элементов m исходных решений.

Случай 2. Использование элементов m исходных решений и дополнительных проектных элементов из расширенного проектного "пространства".

В *случае 1* можно рассматривать две стратегии решения. Во-первых, очевидная стратегию заключается в анализе m исходных решений и построении подструктуры ("системное ядро" K, далее данная "базовая часть системы" в виде подструктуры или медианы (consensus) исходных решений может быть расширена или модифицирована. Для построения "системного ядра" K: можно использовать несколько методов: (а) подстуктура исходных решений, (б) медиана (consensus/agreement) исходных решений, (в) расширенная подструктура, (г) расширенная медиана, (д) выбор наилучшего системного элемента для каждой заданной части системы. При этом, на каждой стадии процесса решения могут быть использованы экспертные процедуры.

Во-вторых, можно спроектировать надструктуру Ω для m исходных решений (например, комбинирование исходных решений, "покрытие" исходных решений). Далее построенная надструктура может быть "сжата" посредством удаления мене важных элементов.

В *случае 2*, можно рассмотреть следующий подход:

Построить "общую" область ("пространство") проектных альтернатив (т.е., элементов решений) и спроектировать новое решение (включая использование новых дополнительных проектных элементов).

Указанная расширенная область решений может быть получена на основе следующих подходов: (а) "покрытие" m исходных решений с помощью новой "области" проектных альтернатив, (б) исследование принципиально новой "области" проектных альтернатив.

В данной главе исследуются следующие базовые структуры: (i) множества [303], (ii) ранжирования (линейно упорядоченнные множества или множества, разбитые на линейно упордоченные подмножества - слои) [144, 145, 336], (iii) морфологическое множества [64, 278, 336, 344, 346, 598], (iv) деревья [207, 246, 303].

Системную модель в виде обобщенной структуры (Λ) можно представить так: $\Lambda = \langle \mathbf{T},\mathbf{P},\mathbf{D},\mathbf{R},\mathbf{I} \rangle$, где используются следующие части [336, 344, 346]: (i) древовидная модель системы $\mathbf{T}$, (ii) множество висячих вершин как базовых частей/компонентов системы $\mathbf{P}$, (iii) множество проектных альтернатив DAs для каждой висячей вершины $\mathbf{D}$, (iv) ранжирование DAs rankings (i.e., ordinal priorities) (т.е., порядковые приоритеты) $\mathbf{R}$, (v) оценки совместимости (порядковые) между DAs $\mathbf{I}$. Рис. 7.3 иллюстрирует обобщенную структуру модульной системы (модульного решения) (Λ). Также рассматриваются следующие структуры:

(а) морфология: $\Phi = \langle \mathbf{P},\mathbf{D} \rangle$;

(б) морфологическое множество: $\overline{\Phi} = \langle \mathbf{P},\mathbf{D},\mathbf{R},\mathbf{I} \rangle$;

(в) морфологическая структура (дерево): $\widetilde{\Phi} = \langle \mathbf{T},\mathbf{P},\mathbf{D},\mathbf{R} \rangle$;

(г) морфологическая структура с совместимостью (т.е., с совместимостью DAs): $\widehat{\Phi} = \langle \mathbf{T},\mathbf{P},\mathbf{D},\mathbf{R},\mathbf{I} \rangle$.

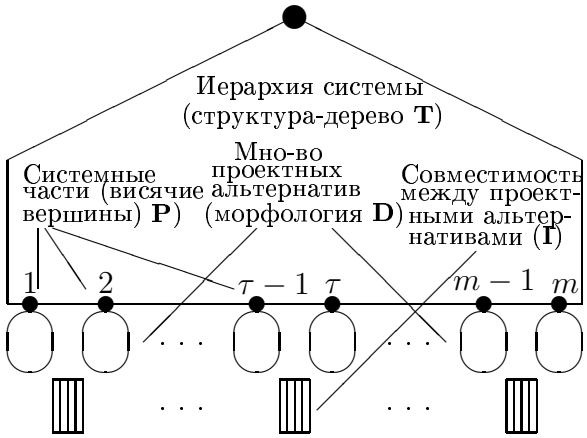


Рис. 7.3. Архитектура модульных систем (Λ)

В Таблице 7.1 кратко представлены основные задачи построения базовых подструктур/надструктур.

Теперь можно описать две базовые системные задачи: (а) выделение/построение "системного ядра", (б) построение расширенной "области" проектных альтернатив. Четыре стратегии решения могут быть использованы:

(1) стратегия расширения: построение "системного ядра" на основе множества исходных решений и его расширение за счет добавления неких дополнительных элементов;

(2) сжимающая стратегия: построение надструктуры для множества исходных решений и удаление из нее некоторых элементов;

(3) комбинированная стратегия (расширение, удаление, операции замены элементов);

(4) стратегия проектирования: построение расширенной "области" проектных элементов и проектирование нового решения.

Указанные четыре стратегии основываются на комбинаторных моделях как вспомогательных (многокритериальное ранжирование/выбор, рюкзак, блочный рюкзак, комбинаторный морфологический синтез).

Таблица 7.1. Базовые структуры и библиографические ссылки

Исходные структуры	Целевая структура	Ссылки
1.Множества	Подмножество, надмножество	[53, 271]
2.Цепочки/ последовательности	Общая подпоследовательность, общая надпоследовательность, медиана, консенсус	[55, 60, 81, 85, 92, 176, 205], [234, 253, 273, 276, 286, 416, 440], [482, 451, 519, 527, 548, 554]
3.Ранжировки	(i)Медиана/консенсус (ii)Размытое ранжирование	[80, 82, 143, 144, 145, 179, 252, 292, 293] [251, 298, 335, 336]
4.Деревья	(i)Консенсус (agreement) как дерево	[53, 54, 86, 184, 189, 269, 478, 531]
	(ii)Консенсус как лес (iii)Наддерево	[123, 241, 489, 581, 582] [86, 167, 255, 509]
5.Графы	Общий подграф, общий надграф	[188, 275, 342, 344]

7.2 Вспомогательные задачи и стратегии агрегирования

7.2.1 Базовые вспомогательные задачи

В случае двух исходных решений в виде множеств элементов A_1 и A_2, Рис. 7.4 иллюстрирует подструктуру $\widetilde{S}_{A_1A_2} \subseteq (A_1 \& A_2)$ и надструктуру $\overline{S}_{A_1A_2} \supseteq (A_1 \bigcup A_2)$ (использована

диаграмма Венна).

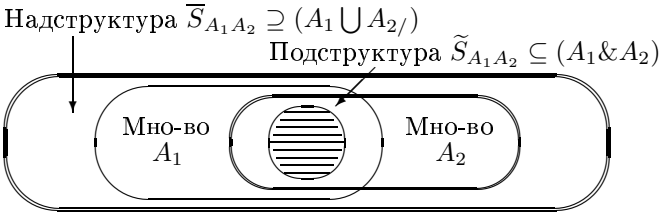


Рис. 7.4. Подструктура и надструктура

Пусть имеется множество исходных решений (структур) $S = \{S_1, ..., S_i, ..., S_n\}$. Пусть функция близости или метрики для решений имеет вид: $\rho(S_{i_1}, S_{i_2})$, $i_1, i_2 \in \{1, 2, ..., n\}$.

Рассматриваются следующие вспомогательные задачи:

Задача 1. Найти максимальную подструктуру:

$$\widetilde{S} = \arg \max_{\{S'\}} (|S'|), \quad \forall S' \in \bigcap_{i=1}^{n} S_i.$$

Задача 2. Найти минимальную надструктуру:

$$\overline{S} = \arg \min_{\{S''\}} (|S''|), \quad \forall S'' \in \bigcup_{i=1}^{n} S_i.$$

Теперь можно рассмотреть определение медианы для исходных множеств S [276, 451, 527]:

(а) “обобщенная медиана” (“generalized median”) M^g определется так:

$$M^g = \arg \ min_{M \in D} \ (\sum_{i=1}^{n} \rho(M, D_i)),$$

где D ($D \supseteq S$) является множеством структур специального вида (поиск такой медианы обычно представляет собой NP-трудную задачу);

(б) упрощенный случай медианы (некая аппроксимация) в виде “медианы множества” (“set median”) M^s над множествами S:

$$M^s = \arg \ min_{M \in S} \ (\sum_{i=1}^{n} \rho(M, S_i)).$$

Здесь ищется один представитель из множества $S = \{S_1, ..., S_i, ..., S_n\}$. Следует заметить, вычисление близости $\rho(M, S_i)$ представляет собой часто также NP-трудную задачу.

Теперь можно привести задачу 3, задачу 4, задачу 5:

Задача 3. Найти “медиану множеств” M^s.

Задача 4. Найти (“обобщенную медиану”) M^g.

Задача 5. Найти расширенную структуру в виде медианы (консенсуса) посредством добавления (или коррекции/редактирования) к базовой структуре в виде медианы/консенсуса некоторых элементов с учетом ресурсных ограничений.

Здесь, некоторые элементы добавляются к медиане (задача 3 или задача 4) с учетом “полезности” и “требуемых ресурсов” для используемых добавлений.

Указанные задачи рассматриваются для следующих типов структур: (i) множества, (ii) *множество морфологий*, (iii) деревья, (iv) деревья с *множеством морфологий*. В случае, когда рассматривается векторная близость/метрика $\rho(S_{i_1}, S_{i_2})$, осуществляется поиск Парето-эффективных решений в задаче 3, задаче 4, задаче 5.

7.2.2 Построение "системного ядра"

Следующие методы можно использовать для построения "системного ядра" K:

(i) построение подструктуры для исходных решений,

(ii) построение медианной структуры (consensus/agreement) для исходных решений,

(iii) построение расширенной подструктуры или расширенной медианной структуры,

(iv) двух-стадийная схема: (а) задание множества базовых системных частей (как подмножество всех частей/компонентов системы), (б) выбор наилучшего системного элемента для каждой заданной базовой системной части.

В некоторых случаях, подрешение (т.е., "системное ядро" как подструктура) является очень маленьким подмножеством. Тогда имеет смысл использовать специальную (более "мягкую") процедуру выбора элементов для "системного ядра". Пусть $S = \{S_1, ..., S_i, ..., S_n\}$ будет множеством исходных решений (структур). В рамках рассматриваемой процедуры, элемент e будет включаться (добавляться) в "системное ядро", если $\eta_i \geq \alpha$ (например, $\alpha \geq 0.5$) где η_e равно числу исходных элементов, которые включают в свой состав e. Использование указанного правила приводит к расширению базового метода построения системной подструктуры.

7.2.3 Стратегии агрегирования

Стратегия расширения имеет следующий вид (Рис. 7.5):

Тип I. Стратегия расширения:

Фаза 1.1. Анализ прикладной задачи, исходных решений, ресурсов, элементов решений.

Фаза 1.2. Выделение базового подрешения или "системного ядра" K на основе m исходных решений (например, подмножество элементов, подструктура, медиана).

Фаза 1.3. Формирование множества дополнительных элементов решения, которые не были использованы ранее.

Фаза 1.4. Выбор наиболее важных элементов из множества дополнительных элементов с учетом следующего: (i) "полезность" выбираемых элементов, (ii) общее ограничение(я) по требуемым ресурсам, (iii) совместимость между выбранными элементами и элементами базового "системного ядра".

Фаза 1.5. Анализ полученного агрегированного решения(или решений).

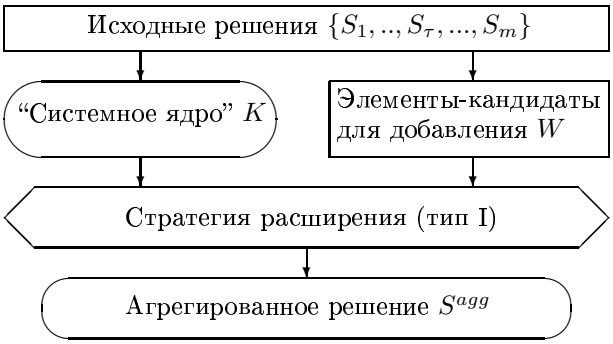


Рис. 7.5. Схема стратегии расширения

Вторая стратегия (стратегия даления/сжатия) основана на предварительном объединении всех элементов m исходных решений и удалении не-важных элементов с учетом удовлетворения некоторых ограничений по ресурсам (Рис. 7.6):

Тип II. Стратегия сжатия (удаления):

Фаза 2.1. Анализ прикладной задачи, исходных решений, элементов решений.

Фаза 2.2. Объединение всех элементов из m исходных решений чтобы сформировать базовое надрешение (например, надструктура, множество всех элементов всех исходных решений).

Фаза 2.3. Формирование множества элементов-кандидатов для удаления.

Фаза 2.4. Выбор не очень важных элементов из множества кандидатов с учетом следующего: (i) интегрированная "полезность" сжатого решения, (ii) общие ограничения по ресурсу(сам), (iii) совместимость между выбранными элементами в сжатом решении.

Фаза 2.5. Анализ полученного агрегированного решения (или решений).

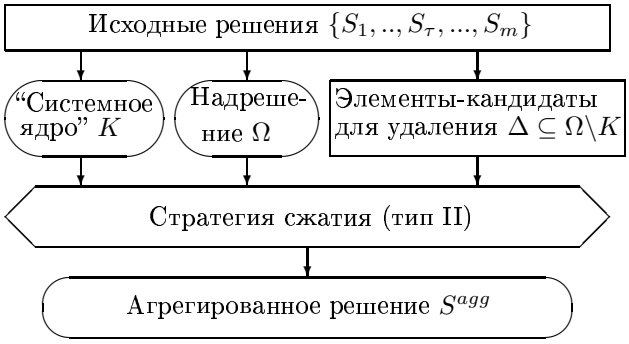


Рис. 7.6. Схема стратегии сжатия

Третья стратегия включает комбинации операций добавления, удаления и замены элементов (Рис. 7.7):

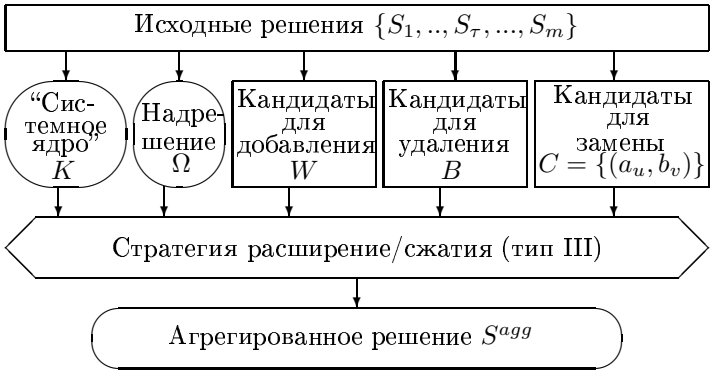


Рис. 7.7. Схема стратегии расширения/сжатия

Тип III. Комбинированная стратегия:

Фаза 3.1. Анализ прикладной задачи, исходных решений, элементов решений.

Фаза 3.2. Выделение на основе m исходных решений некой базы в виде подрешения или "системного ядра" K (например, подмножество элементов, подструкутра, медиана).

Фаза 3.3. Формирование следующего:

(а) множество дополнительных элементов решения как кандидатов для добавления W, которые не были включены в упомянутую базу ($|W \bigcap K| = 0$),

(б) подмножество элементов $B \subseteq K$ как кандидатов для удаления,

(в) множество пар элементов $C = \{(a_u, b_v)|(a_u \in K)\&(b_v \overline{\in} K)\}$, далее множество операций коррекции рассматривается как замена элемента a_u на элемент b_v.

Фаза 3.4. Выбор наиболее важных операций (включая добавление элементов, удаление элементов, замену элементов) с учетом следующего: (i) "полезность" операций, (ii) общие ограничения по требуемым ресурсам, (iii) совместимость между выбранными элементами в результирующем решении.

Фаза 3.5. Анализ полученного агрегированного решения (или решений).

Четвертая стратегия направлена на использование дополнительных элементов из "расширенного" проектного "пространства" (области) элементов. Здесь, результирующее агрегированное решение может включать не только элементы из исходных решений, но и новые элементы. Четвертая стратегия включает следующее:

Тип IV. Стратегия расширенного проектного "пространства" (новое проектирование):

Фаза 4.1. Анализ прикладной задачи, исходных решений, элементов решений.

Фаза 4.2. Расширение объединения множества всех элементов m исходных решений, чтобы сформировать расширенное проектное "пространство".

Фаза 4.3. Новое проектирование составного решения на основе полученного расширенного проектного "пространства" с учетом следующего: (i) "полезность" проектируемого решения, (ii) общие ограничения по требуемым ресурсам, (iii) совместимость между выбранными элементами в результирующем решении.

Фаза 4.4. Анализ полученного агрегированного решения (или решений).

Рис. 7.8 иллюстрирует четвертую стратегию: имеются три исходных решения S_1, S_2, and S_3, результирующее агрегированное решение S^{agg} может включать элементы из исходных решений S_1, S_2 и элементы из расширенного проектного "пространства" (области).

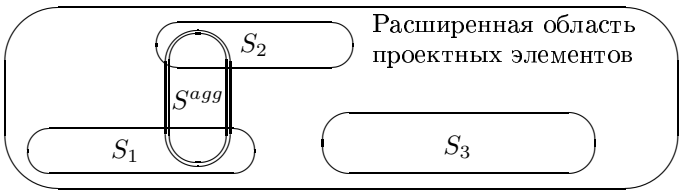


Рис. 7.8. Иллюстрация нового проектирования

7.3 Рассматриваемые структуры

Рассматриваются следующие базовые виды множеств: множества, списки, деревья [167, 303, 541]. Дополнительно, внимание уделено специальному виду составных структур: *морфологическим структурам.*

7.3.1 Множества

Рис. 7.9 содержит пример для двух исходных множеств A и B. Рис. 7.10 содержит примеры подмножества $\widetilde{S}_{AB}$ и надмножества $\overline{S}_{AB}$.

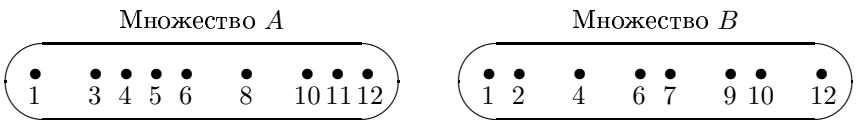


Рис. 7.9. Пример: два исходных множества

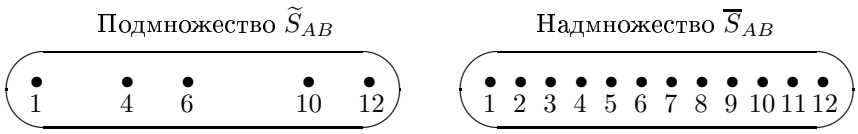


Рис. 7.10. Примеры: подмножество и надмножество

Пусть $\{A_1, ..., A_i, ..., A_m\}$ являются исходными множествами. Тогда подмножество (подструктура) имеет вид: $\widetilde{S}_{\{A_i\}} \subseteq \bigcap_{i=1}^{m} A_i$. Надмножество (надструктура) имеет вид: $\overline{S}_{\{A_i\}} \supseteq \bigcup_{i=1}^{m} A_i$.

7.3.2 Ранжировки (слоистые множества)

Ранжировка рассматривается как слоистое множество (т.е., множество, элементы которого разбиты на линейно упорядоченные подмножества-слои). Пусть $A = \{1, ..., i, ..., n\}$ будет множество исходных элементов. Ранжировка (частичный порядок или разбиение множества) A) рассматривается как линейно упорядоченные подмножества A (Рис. 7.11): $A = \bigcup_{k=1}^{m} A(k)$, $|A(k_1) \cap A(k_2)| = 0$ если $k_1 \neq k_2$, $i_2 \preceq i_1$ $\forall i_1 \in A(k_1)$, $\forall i_2 \in H(k_2)$, $k_1 \leq k_2$. Помножество $A(k)$ будет называться слоем k, каждый элемент $i \in A$ получает приоритет r_i, который равен номеру соответствующего слоя. Описанное разбиение множества A будет называться ранжирование (ранжировка), стратификация, слоистое множество [163, 335, 336, 344, 28, 492, 595]. Очевидно, что линейный порядок элементов множества A является также ранжировкой. Многие годы задачи ранжирования (sorting) интенсивно исследовались [107, 163, 122, 292, 293, 344, 28, 492, 595, 596].

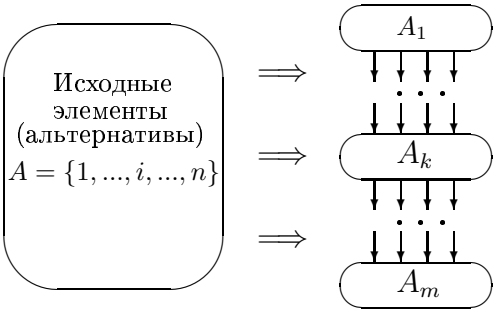


Рис. 7.11. Схема ранжирования

7.3.3 Морфологические множества

Морфологическое множество - это структура, состоящая из следующего:

1. конечное множество целых чисел $N = \{1, ..., \tau, ..., m\}$ (каждое целое τ соответствует части системы);

2. множество элементов (альтернатив) для каждой части системы τ:

$A_\tau = \{A_{\tau 1}, ..., A_{\tau\xi}, ..., A_{\tau q_\tau}\}$, где $A_{\tau\xi}$ - это проектная альтернатива;

3. отношение предпочтения над множеством элементов A_τ (или оценки по заданному множеству критериев или результирующие порядковые приоритеты для каждой альтернативы $p(A_{\tau\xi})$);

4. взвешенные (порядковая шкала) бинарные отношения *совместимости* для каждой пары системных частей $(\alpha, \beta) \in N$ над элементами множеств альтернатив A_α, A_β: R_{A_α, A_β}.

Рассматривается система, состоящая из m частей (Рис. 7.12) [336, 344, 349, 358]. Таким образом, *морфология системы* S_A (или морфология Φ) определяется так: $A = \{A_1, ..., A_\tau, ..., A_m\}$ (Рис. 7.13).

Далее приведены примеры для двух морфологий: морфология A^1 (Рис. 7.14), морфология A^2 (Рис. 7.15), под-морфология $\widetilde{S}_{A^1,A^2}$ (Рис. 7.16), над-морфология $\overline{S}_{A^1,A^2}$ (Рис. 7.17).

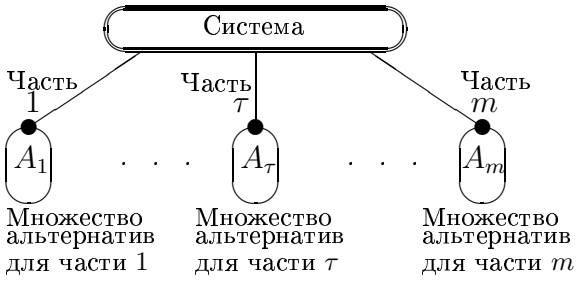


Рис. 7.12. Составная система

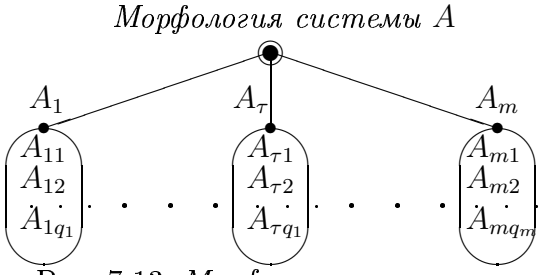


Рис. 7.13. *Морфология системы*

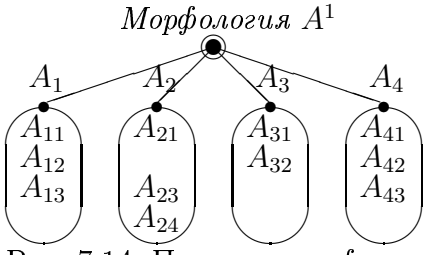


Рис. 7.14. Пример: *морфология* A^1

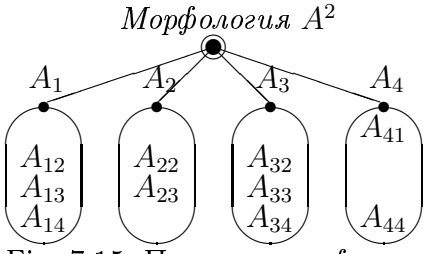


Fig. 7.15. Пример: *морфология* A^2

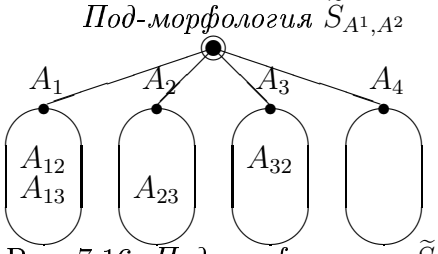


Рис. 7.16. *Под-морфология* $\widetilde{S}_{A^1,A^2}$

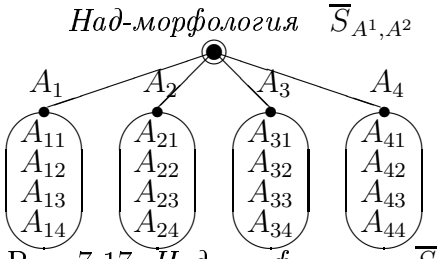


Рис. 7.17. *Над-морфология* $\overline{S}_{A^1,A^2}$

Кроме того, следует рассмотреть оценки совместимости. Агрегирование оценок (по элементам) может быть основано на специальных операциях (например, выбор минимального значения). Таблица 7.2 содержит иллюстративный пример оценок совместимости и их агрегирования.

Таблица 7.2. Агрегация оценок совместимости для A^1, A^2: узлы A_1, A_2

Морфология A^1

	A_{21}	A_{23}	A_{24}
A_{11}	0	1	2
A_{12}	3	2	1
A_{13}	2	3	1

Морфология A^2

	A_{22}	A_{23}
A_{12}	2	3
A_{13}	1	2
A_{14}	2	2

Над-морфология $\overline{S}_{A_1,A_2}$

	A_{21}	A_{22}	A_{23}	A_{24}
A_{11}	0	–	1	2
A_{12}	3	2	2	1
A_{13}	2	1	2	1
A_{24}	–	2	2	–

7.3.4 Деревья и морфологические структуры

На Рис. 7.18 представлены примеры двух деревьев T_1 и T_2. На Рис. 7.19 представлены примеры соответствующих над-дерева $\overline{T}$ и под-дерева $\widetilde{T}$.

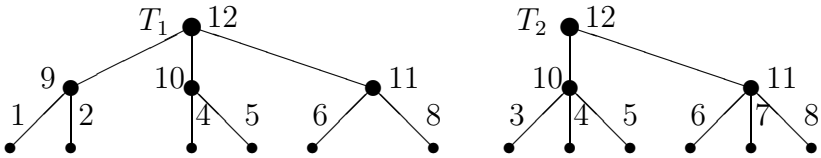


Рис. 7.18. Два исходных дерева

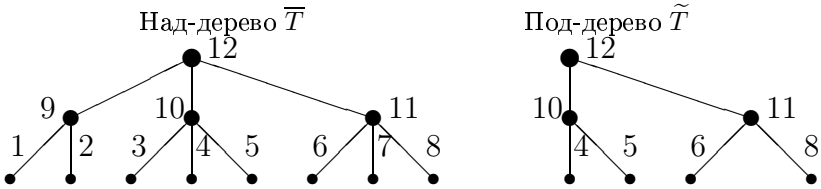


Рис. 7.19. Над-дерево и под-дерево

На Рис. 7.20 представлены примеры двух деревьев с морфологиями Θ_1 и Θ_2. На Рис. 7.21 представлены соответствующие над-дерево и под-дерево с морфологиями $\overline{\Theta}$ и $\widetilde{\Theta}$. Очевидно, что таблицы оценок совместимости и их агрегирование могут быть также рассмотрены.

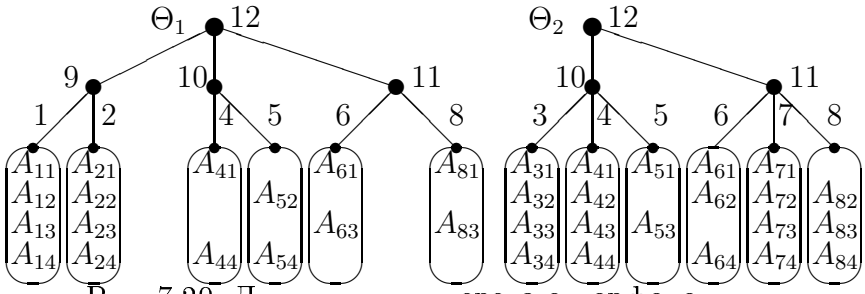


Рис. 7.20. Два исходных дерева с морфологиями

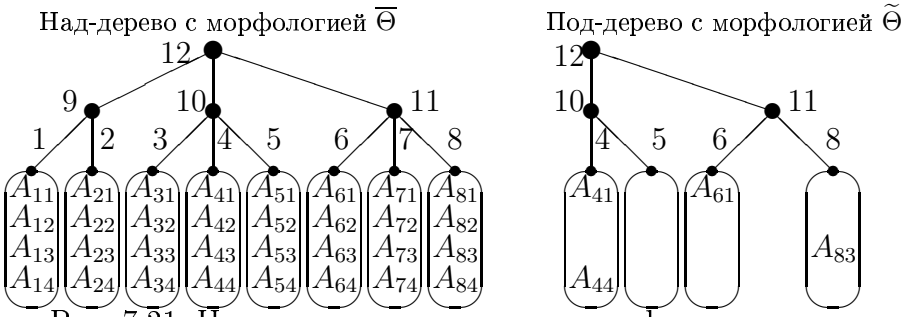


Рис. 7.21. Над-дерево и под-дерево с морфологиями

7.4 Предварительный иллюстративный пример

Структура компьютера имеет вид (Рис. 7.22):

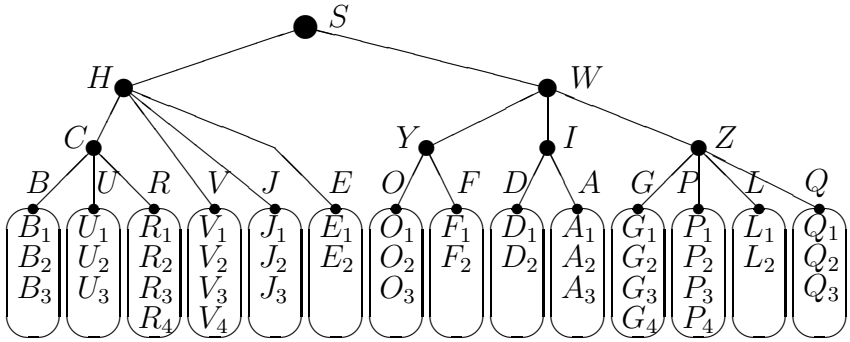


Рис. 7.22. Упрощенная структура ноутбука

0. Ноутбук S.
1. Электроника H:
1.1. Базовые вычисления C:
1.1.1. Материнская плата B: B_1 (P67A - C43(B3) ATX Intel),
B_2 (MSI 870A-G54 ATX AMD), B_3 (ASRoot P67 EXTREME 4(B3) ATX Intel);
1.1.2. Процессор (CPU) U: U_1 (Intel Pentium dual-core processor T 2330),
U_2 (Celeron dual-core processor 2330), U_3 (Intel core 2 T 7200);
1.1.3. Оперативная память (RAM) R: R_1 (1 GB DDR A-DATA),
R_2 (2 GB DDR2 KINGSTON), R_3 (2 GB DDR3 A-DATA),
R_4 (2 GB DDR3 HYPER X KINGSTON);
1.2. Диск (Hard drive) V: V_1 (100 GB HDD), V_2 (120 GB HDD), V_3 (160 GB HDD),
V_4 (200 GB HDD);
1.3. Видео карты (Video/graphic cards) J: J_1 (NVIDIA GeForce CTS 300M),
J_2 (GT 400M Series), J_3 (ATI Radion HD 5000 M Series);
1.4. Модемы E: E_1 (Внутренние модем и антенна), E_2 (Нет).
2. Программы W:
2.1. Операционная система и безопасность Y:
2.1.1. OS O: O_1 (Windows XP), O_2 (Windows Vista); O_3 (Linux).
2.1.2. Программы безопасности F: F_1 (Norton AntiVirus), F_2 (AntiVirus Kaspersky).
2.2. Обработка информации и интернет I:
2.2.1. Поддержка данных и обработка D: D_1 (Microsoft Office), D_2 (Нет).
2.2.2. Доступ к Интернет A: A_1 (Microsoft Internet Explorer), A_2 (Mozilla);
A_3 (Microsoft Internet Explorer & Mozilla).
2.3. Профессиональные программы Z:
2.3.1. Обработка информации G: G_1 (Matlab), G_2 (LabView); G_3 (MatCad), G_4 (Нет);

2.3.2. Специальные среды для разработки программ P: P_1 (C++), P_2 (JAVA); P_3 (Delphi), P_4 (Нет);
2.3.3. Специальные редакторы L: L_1 (LaTex), L_2 (None);
2.3.4. Игры Q: Q_1 (Tetris), Q_2 (Solitaire), Q_3 (Chess).

Структуры четырех исходных вариантов компьютера S_1, S_2, S_3, S_4 представлены на Рис. 7.23, Рис. 7.24, Рис. 7.25, Рис. 7.26. Здесь изменяются только висячие вершины модели системы. are not considered. Совместимость между DAs не рассматривается. Общая схема агрегации представлена на Рис. 7.27 (включая два альтернативных метода построения "системного ядра").

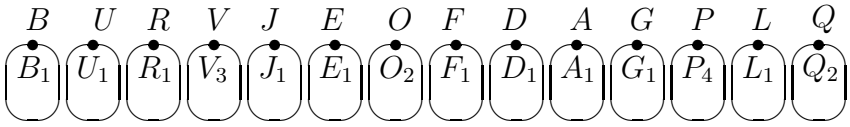


Рис. 7.23. Структура ноутбука S_1

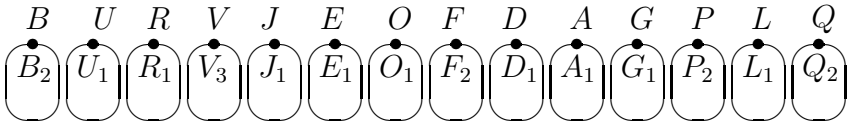


Рис. 7.24. Структура ноутбука S_2

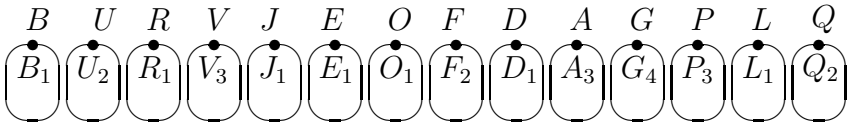


Рис. 7.25. Структура ноутбука S_3

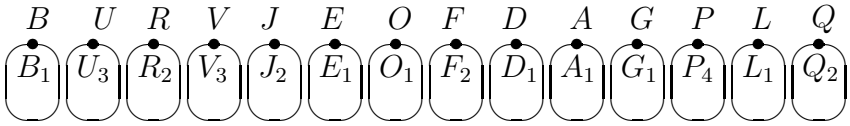


Рис. 7.26. Структура ноутбука S_4

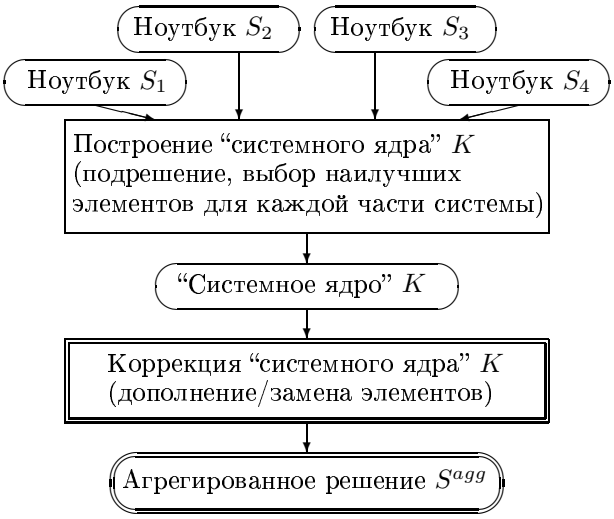


Рис. 7.27. Схема процесса агрегации

Подструктура для рассматриваемых вариантов $\widetilde{S}$ представлена на Рис. 7.28, надструктура $\overline{S}$ представлена на Рис. 7.29. Теперь следует рассмотреть "системное ядро" K как расширение подструктуры $\widetilde{S}$ (Рис. 7.30).

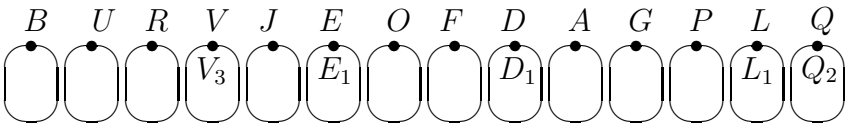


Рис. 7.28. Подструктура решений $\widetilde{S}$

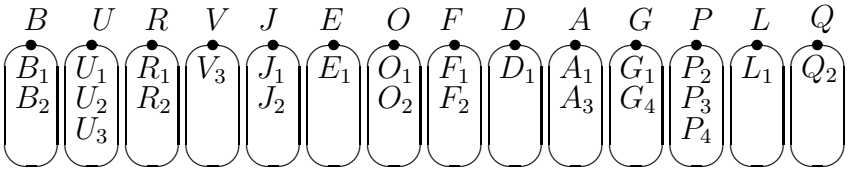


Рис. 7.29. Надструктура решений $\overline{S}$

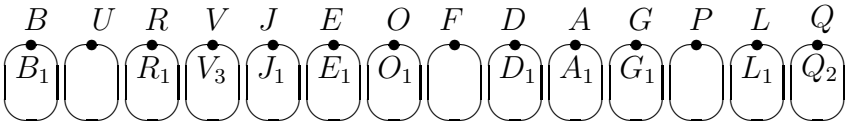


Рис. 7.30. “Системное ядро” K

Теперь комбинированная стратегия агрегации в виде модификации “системного ядра” K может быть применена. Следующее множество операций модификации (как кандидатов) можно рассмотреть:

1. операции добавления: *1.1.* добавление для U: U_1 или U_2 или U_3, *1.2.* добавление для F: F_1 или F_2, *1.3.* добавление для P: P_2 или P_3 или P_4;

2. операции корректировки: *2.1.* замена $B_1 \Rightarrow B_3$, *2.2.* замена $V_3 \Rightarrow V_4$, *2.3.* замена $A_1 \Rightarrow A_3$.

Очевидно, представляется целесообразным оценить указанные операции модификации (например, “стоимость”, “полезность”) и рассмотреть оптимизационную модель. Далее будет приведена соответствуюшая оптимизационная модель рюкзачного вида. Некий пример результирующего решения S^1 представлен на Рис. 7.31 (модификация “системного ядра” K).

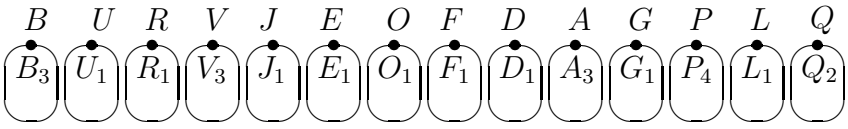


Рис. 7.31. Решение S^1 (модификация “системного ядра” K)

С другой стороны, построение “системного ядра” можно рассматривать как многокритериальный выбор и/или экспертное обоснование. Например, можно рассмотреть следующую базовую структуру “системного ядра”: B, U, R, V, O, F, D, G. Для каждой указанной части системы можно рассмотреть процедуру выбора наилучшего элемента (с учетом элементов исходных решений или дополнительных элементов). Например, можно получить следующее “системное ядро”:

$$K^* = B_1 \star U_2 \star R_2 \star V_3 \star E_2 \star O_2 \star F_2 \star D_2 \star G_2.$$

Процесс корректирования (модификации) данного “системного ядра” может быть основан на следующих операциях:

1. добавление: A_1, P_1, L_1;

2. удаление: E_2;

3. замена: $B_1 \Rightarrow B_3$, $U_2 \Rightarrow U_1$, $O_2 \Rightarrow O_1$.

Результирующее решение S^2 представлено на Рис. 7.32.

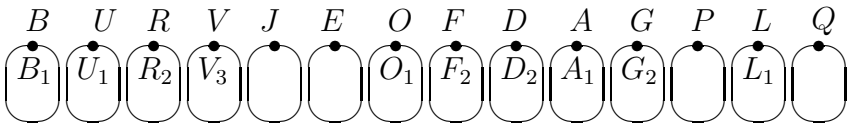


Рис. 7.32. Решение S^2

7.5 Метрики и близости

В данном разделе рассматриваются меры близости между “структурными” оъектами (множества, ранжировки, деревья, графы) A_1 и A_2. Обычно мера расстояния определяется как функция $d(A_1, A_2)$, удовлетворяющая ряду свойств метрики:

1. $d(A_1, A_2) \geq 0$ (неотрицательность),
2. $d(A_1, A_1) = 0$ (идентичность),
3. $d(A_1, A_2) = 0 \Leftrightarrow A \cong B$ (уникальность),
4. $d(A_1, A_2) = d(A_2, A_1)$ (симметрия),
5. $d(A_1, A_2) + d(A_2, A_3) \geq d(A_1, A_3)$ (неравенство треугольника).

Если данная функция удовлетворяет условию $d(A_1, A_2) \leq 1$, то говорят, что данная метрика является нормализованной.

Во многих приложениях приведенные условия являются достаточно “сильными” и рассматриваются более "слабые"свойства. В результате функция $d(A_1, A_2)$ соответствует более “слабой” ситуации, например: (i) квази-метрика, (ii) близость (без свойства 5, например, близость для ранжировок в [335, 336]).

Существует три известных подхода к определению похожести/близости объектов или структур:

(1) традиционные метрики/расстояния (в виде указанной функции) [95, 177, 220, 222, 431];

(2) трансформация минимальной стоимости некого объекта в другой (расстояние редактирования) [87, 113, 244, 327, 420, 543, 567, 592];

(3) максимальная общая подструктура (с учетом согласования данной структуры с исходными объектами) [53, 54, 113, 114, 342, 344, 485, 576].

7.5.1 Метрики/близости для множеств

Рассмотрим множество элементов $A = \{1, ..., i, ..., n\}$. Пусть имеется два подмножества $A^1 \subseteq A$ и $A^2 \subseteq A$. Метрика по элементам (т.е., расстояние) имеет вид (с учетом предположения $|A^1 \bigcup A^2| \neq 0$):

$$\rho_e(A^1, A^2) = 1 - \frac{|A^1 \bigcap A^2|}{|A^1 \bigcup A^2|}.$$

Теперь рассмотрим веса элементов: $w_i \in (0, 1]$ $(i \in A)$. Тогда расстояние (по весам элементов) имеет вид (с учетом предположения $\sum_{i \in (A^1 \bigcup A^2)} w_i \neq 0$):

$$\rho_w(A^1, A^2) = 1 - \frac{\sum_{i \in (A^1 \bigcap A^2)} w_i}{\sum_{i \in (A^1 \bigcup A^2)} w_i}.$$

Рассмотрим пример (Таблица 7.3), который включает:

(а) исходное множество $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$,

(б) два подмножества: $A^1 = \{1, 2, 4, 5\}$ и $A^2 = \{1, 2, 3, 4, 5, 6, 7\}$.

Следующие веса элементов используются: 0.5, 0.6, 0.4, 1.0, 0.7, 0.2, 0.1, 1.0. Получаются следующие расстояния: $\rho_e(A^1, A^2) = 1 - \frac{3}{7} = 0.571$ и $\rho_w(A^1, A^2) = 1 - \frac{1.8}{3.5} = 0.496$.

Таблица 7.3. Числовой пример

i	w_i	A	A^1	A^2	$A^1 \bigcup A^2$	$A^1 \bigcap A^2$
1	0.5	$\star$	$\star$	$\star$	$\star$	$\star$
2	0.6	$\star$	$\star$	$\star$	$\star$	$\star$
3	0.4	$\star$		$\star$	$\star$	
4	1.0	$\star$	$\star$		$\star$	
5	0.7	$\star$	$\star$	$\star$	$\star$	$\star$
6	0.2	$\star$			$\star$	
7	0.1	$\star$		$\star$	$\star$	
8	1.0	$\star$				

В случае использования векторных весов $\overline{w}_i = (w_i^1, ..., w_i^\xi, ..., w_i^r)$, следующая упрощенная векторная близость может рассматриваться:

$$\rho_{\overline{w}}(A^1, A^2) = (\rho_{\overline{w}}^1(A^1, A^2), ..., \rho_{\overline{w}}^\xi(A^1, A^2), ..., \rho_{\overline{w}}^r(A^1, A^2)) =$$

$$(1 - \frac{\sum_{i \in (A^1 \bigcap A^2)} w_i^1}{\sum_{i \in (A^1 \bigcup A^2)} w_i^1}, ..., 1 - \frac{\sum_{i \in (A^1 \bigcap A^2)} w_i^\xi}{\sum_{i \in (A^1 \bigcup A^2)} w_i^\xi}, ..., 1 - \frac{\sum_{i \in (A^1 \bigcap A^2)} w_i^r}{\sum_{i \in (A^1 \bigcup A^2)} w_i^r}).$$

7.5.2 Близость для последовательностей

В основном используются три вида близости для цепочек/пследовательностей (как указывалось в общем случае):

(1) общая часть исходных последовательностей как подпоследовательности, надпоследовательности [58, 57, 234, 273, 274, 554],

(2) последовательность-медиана [305, 451],

(3) расстояние редактирования [57, 244, 327, 488, 574].

Здесь важными задачами являются следующие: (а) анализ алгоритмической сложности [554]; (б) построение полиномиальных алгоритмов [244, 274]; (в) построение приближенных алгоритмов [273].

7.5.3 Близость для ранжировок

Несколько видов расстояний (метрик) или мер близости используются для ранжировок:

1. расстояние Кендалла-Тау (Kendall-Tau) [295];

2. расстояния для частичных ранжировок [74, 183];

3. векторая близость [335, 336].

Далее расстояние Кендалла-Тау и векторная близость для ранжировок кратко описаны (описание основано на [336]). Пусть $\|g_{ij}\|, (i, j \in A)$ будет матрица смежности для графа G:

$$g_{ij} = \begin{cases} 1, & \text{если } i \succ j, \\ 0, & \text{если } i \sim j, \\ -1, & \text{если } i \prec j. \end{cases}$$

Расстояние Кендалла-Тау для графов G^1 и G^2 имеет вид:

$$\rho_K(G^1, G^2) = \sum_{i<j} \mid g_{ij}^1 - g_{ij}^2 \mid,$$

где g_{ij}^1, g_{ij}^2 являются элементами матрицы смежности графов G^1 и G^2 соответственно.

Векторная близость основана на следующих определениях [335, 336]. Пусть $\Psi(S)$ обозначает множество всех слоистых структур A.

Определение 7.1. Будем называть ошибкой первого порядка $\forall i \in A$, и ошибкой второго порядка $\forall (i,j) \in \{A \star A | i \neq j \ \forall S, Q \in \Psi(S)\}$ следующее:

$$\delta_i^\pi(S,Q) = \pi_i(S) - \pi_i(Q),$$

$$\delta_{ij}^\pi(S,Q) = \pi_i(S) - \pi_j(S) - (\pi_i(Q) - \pi_j(Q)),$$

где $\pi_i(S) = l \ \forall i \in A(l)$ в S. Таким образом, для оценки несоответствия (несогласованности) между структурами $S, Q \in \Psi(S)$ относительно элементу i и паре элементов (i,j) получается целочисленная шкала с следующими уровнями:

(i) $-(m-1) \leq r \leq m-1$ для $\delta_i^\pi(S,Q)$,

(ii) $-2(m-1) \leq r \leq 2(m-1)$ для $\delta_{i,j}^\pi(S,Q)$.

Определение 7.2. Пусть

$$x(S,Q) = (x^{-(m-1)}, ..., x^{-1}, x^1, ..., x^{m-1}),$$

$$y(S,Q) = (y^{-2(m-1)}, ..., y^{-1}, y^1, ..., y^{2(m-1)}),$$

будут векторы ошибок (близости) $\forall S, Q \in \Psi(S)$ относительно компонентов i (1-й порядок) и пар компонентов (i,j) (2-й порядок). Компоненты указанных векторов определяются следующим образом:

$$x^r = |\{i \in A | \delta_i^\pi(S,Q) = r\}|/n,$$

$$y^r = 2|\{(i,j) \in \{A * A | i \neq j\} | \delta_{ij}^\pi(S,Q) = r\}|/(n(n-1)).$$

Обозначим множество аргументов для компонентов векторов x и y следующим образом: $\Omega = \{-k, ..., k\}$, множество отрицательных как Ω^-, множество положительных как Ω^+. Дополнительно будут использоваться векторы типа x с агрегированными компонентами (аналогично для y):

$$x^{k_1,k_2} = \sum_{r=k_1}^{k_2} x^r, x^{\leq -k} = \sum_{r=-(m-1)}^{-k} x^r, x^{\geq k} = \sum_{r=k}^{m+1} x^r, k > 0, x^{|r|} = x^r + x^{-r}.$$

Определение 7.3. Пусть $|x(S,Q)| = \sum_{r \in \Omega} x^r$, $|y(S,Q)| = \sum_{r \in \Omega} y^r$ обозначает модули векторов.

Вектор x будет использоваться как базовый.

Определение 7.4. Векторы называются сокращенными, если:

(1) часть терминальных компонентов удаляется, например:

$$x(S,Q) = (x^{-k_1}, x^{-(k_1-1)}, ..., x^{-1}, x^1, ..., x^{k_2-1}, x^{k_2}),$$

и одно или более следующих условий удовлетворяются $k_1 < m-1$, $k_2 < m-1$;

(2) агрегированные компоненты используются:

$$x(S,Q) = (x^{\leq k_1}, ..., x^{k_a-1}, x^{k_a,k_b}, x^{k_b+1}, ..., x^{\geq k_2}),$$

$$x(S,Q) = (x^{|1|}, ..., x^{|r|}, ..., x^{|k|}).$$

Определение 7.5. Вектор x (y) будет называться:

(a) двух-сторонним, если $|\Omega^+| \neq 0$ и $|\Omega^-| = 0$;

(b) одно-сторонним, если $|\Omega^+| = 0$ или $|\Omega^-| = 0$;

(c) симметричным, если $-r \in \Omega^-$ существует $\forall r \in \Omega^+$, и наоборот;

(d) модульным, если он определен в соответствии с определением 7.4 (7.3).

Таким образом, пара линейных порядков на компонентах векторов x и y (определение 7.2) получены: *компонент* $1(-1) \prec ... \prec$ *компонент* $k(-k)$.

Определение 7.6. $x_1(S,Q) \succeq x_2(S,Q)$, $\Omega(x_1) = \Omega(x_2)$, $\forall S, Q \in \Psi(S)$, если некое уменьшение "слабых" компонент x_1 по сравнению с x_2 компенсируется соответствующим увеличением ее "сильных" компонент ($r, p \in \Omega^+$ или $-r, -p \in \Omega^-$):

$$\sum_{r \geq u}^{r} x_1^r - \sum_{r \geq u}^{r} x_2^r \geq 0, \forall u \in \Omega^+ (\forall -u \in \Omega^-, -r \leq -u).$$

Приведем иллюстративный пример:
(а) исходное множество $A = \{1,2,3,4,5,6,7,8,9\}$,
(б) ранжировка
$S^1 = \{A_1^1 = \{2,4\}, A_2^1 = \{9\}, A_3^1 = \{1,3,7\}, A_4^1 = \{5,6,8\}\}$,
(в) ранжировка $S^2 = \{A_1^2 = \{7,9\}, A_2^2 = \{1,3\}, A_3^2 = \{2,5,8\}, A_4^2 = \{4,6\}\}$.
Соответствующие матрицы смежности имеют вид:

$$|g_{ij}(S^1)| = \begin{pmatrix} . & -1 & 0 & -1 & 1 & 1 & 0 & 1 & -1 \\ 1 & . & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & . & -1 & 1 & 1 & 0 & 1 & -1 \\ 1 & 0 & 1 & . & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & . & 0 & -1 & 0 & -1 \\ -1 & -1 & -1 & -1 & 0 & . & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 1 & 1 & . & 1 & -1 \\ -1 & -1 & -1 & -1 & 0 & 0 & -1 & . & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & . \end{pmatrix}$$

$$|g_{ij}(S^2)| = \begin{pmatrix} . & 1 & 0 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & . & -1 & 1 & 0 & 1 & -1 & 0 & -1 \\ 0 & 1 & . & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & . & -1 & 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & 1 & . & 1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & . & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & . & 1 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 & -1 & . & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & . \end{pmatrix}$$

Несложно посчитать расстояние Кендалла-Тау для S^1 и S^2:

$$\rho_K(S^1, S^2) = 31.$$

Векторные близости позволяют более "рельефное" описание различия между структурами S^1 и S^2:

$$\pi(S^1) = (\pi_1(S^1), ..., \pi_i(S^1, ..., \pi_4(S^1) = (3,1,3,1,4,4,3,4,2),$$

$$\pi(S^2) = (\pi_1(S^2), ..., \pi_i(S^2, ..., \pi_4(S^2) = (2,3,2,4,3,4,1,3,1),$$

$$\delta_i^\pi(S^1, S^2) = (1,-2,1,-3,1,0,2,1,1),$$

$$\delta_{ij}^{\pi}(S^1,S^2)=\begin{pmatrix} . & 3 & 0 & 4 & 0 & 1 & -1 & 0 & 0 \\ -3 & . & -3 & -1 & -3 & -2 & -4 & -3 & -3 \\ 0 & 3 & . & -4 & 0 & 1 & 1 & 0 & 0 \\ -4 & 1 & 4 & . & -4 & -3 & -5 & -4 & -4 \\ 0 & 3 & 0 & 4 & . & 1 & -1 & 0 & 0 \\ -1 & -2 & -1 & 3 & -1 & . & -1 & -1 & -1 \\ 1 & 4 & -1 & 5 & 1 & 1 & . & 1 & 1 \\ 0 & 3 & 0 & 4 & 0 & 1 & -1 & . & 0 \\ 0 & 3 & 0 & 4 & 0 & 1 & -1 & 0 & . \end{pmatrix}$$

$$x(S^1,S^2)=(x^{-3},x^{-2},x^{-1},x^1,x^2,x^3)=(1,1,0,5,1,0),$$

$$y(S^1,S^2)=(y^{-6},y^{-5},y^{-4},y^{-3},y^{-2},y^{-1},y^1,y^2,y^3,y^4,y^5,y^6)=$$

$$(0,1,5,5,1,6,6,0,1,1,0,0).$$

Отметим, коэффициенты $\frac{1}{n}$ и $\frac{2}{n(n-1)}$ не использовались в x и y.

7.5.4 Близость для деревьев

Как и в общем случае, для оценки близости (похожести, различия) деревьев применяются три подхода [53, 54, 141, 175, 184, 255, 508, 544, 564, 565]:

(i) метрики (расстояния) [141, 175, 544, 565];

(ii) расстояния редактирования для деревьев [126, 502, 543, 555, 589, 590, 591];

(iii) общее поддерево, медианное дерево, консенсус [53, 54, 184, 189, 255, 269, 509, 531, 577].

Следует отметить, что многие эффективные (полиномиальные) алгоритмы предложены для построения метрик/близостей для деревьев, поскольку деревья являются достаточно “хорошо” упорядоченными структурами [178, 555, 556, 564].

Далее будет использована упрощенная версия двух-компонентной близости для корневых деревьев с маркированными вершинами (labelled trees). Рассмотрим два таких дерева: $T'=(A',E')$ и $T''=(A'',E'')$ (корень в обоих деревьях один и тот же), где множества A' и A'' - вершины, множества E' и E'' - дуги. Рассмотрим *параметр доминирования* $\forall(a,b)\in(E'\bigcap E'')$. Имеются следующие три *случая доминирования* для вершин a и b: (i)) $a\to b$ (a доминирует b, т.е., $a\succ b$), (ii) $b\to a$ (b доминирует a, т.е., $a\prec b$), (iii) a и b являются независимыми. Тогда *параметр доминирования* имеет вид:

$$d_{(a,b)\in(E'\bigcap E'')}=\begin{cases} d^1, & \text{если } a\to b, \\ d^2, & \text{если } b\to a, \\ d^3, & \text{если } a,b \text{ - независимы.} \end{cases}$$

В результате, *параметр изменения* $\forall(a,b)\in(E'\bigcap E'')$ можно определить так:

$$p_{(a,b)\in(E'\bigcap E'')}=\begin{cases} 0, & \text{если } d_{(a,b)\in(E'\bigcap E'')} \text{ - не изменен,} \\ 1, & \text{если } d_{(a,b)\in(E'\bigcap E'')} \text{ - изменен.} \end{cases}$$

Окончательно, рассматриваемая близость для двух деревьев имеет вид:

$$\overline{\rho}(T',T'')=(\rho(A',A''),\rho(E',E'')),$$

где

$$\rho_e(A',A'')=1-\frac{|A'\bigcap A''|}{|A'\bigcup A''|},\quad \rho(E',E'')=\frac{\sum_{(a,b)\in(E'\bigcap E'')}p_{(a,b)\in(E'\bigcap E'')}}{|(E'\bigcap E'')|}.$$

Очевидно, что выполняются следующие свойства:

(1) $0 \le \rho(E', E'') \le 1$, (2) $\rho(E, E) = 0$.

Отметим, что описанный подход может быть использован и для графов. Приведем иллюстративный пример:

Пример 7.1: Деревья T' и T'' представлены на Рис. 7.33.
Общая близость имеет вид: $\overline{\rho}(T', T'') = (0, 1)$.

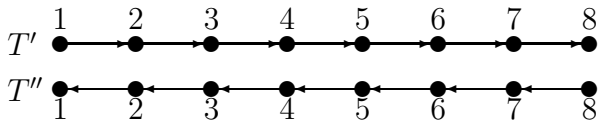


Рис. 7.33. Деревья для примера 7.1

Пример 7.2: Два дерева T' и T'' представлены на Рис. 7.34. Здесь $A' = A''$ и $\rho(A', A'') = 0$. Таблица 7.4 и Таблица 7.5 содержат соответствующие *факторы доминирования* для T' и T'' (изменения указаны 'овалами'). Общая близость имеет следующий вид: $\overline{\rho}(T', T'') = (0, \frac{6}{28})$.

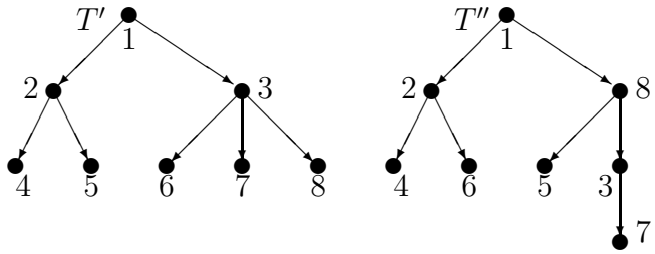


Рис. 7.34. Деревья для примера 7.2

Таблица 7.4. *Фактор доминирования* (T')

	1	2	3	4	5	6	7	8
1	$\star$	d^1	d^1	d^1	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^1	d^1	d^3	d^3	d^3
3	$-$	$-$	$\star$	d^3	d^3	d^1	d^1	d^1
4	$-$	$-$	$-$	$\star$	d^3	d^3	d^3	d^3
5	$-$	$-$	$-$	$-$	$\star$	d^3	d^3	d^3
6	$-$	$-$	$-$	$-$	$-$	$\star$	d^3	d^3
7	$-$	$-$	$-$	$-$	$-$	$-$	$\star$	d^3
8	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\star$

Таблица 7.5. *Фактор доминирования* (T'')

	1	2	3	4	5	6	7	8
1	$\star$	d^1	d^1	d^1	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^1	d^3	d^1	d^3	d^3
3	$-$	$-$	$\star$	d^3	d^3	d^3	d^1	d^2
4	$-$	$-$	$-$	$\star$	d^3	d^3	d^3	d^3
5	$-$	$-$	$-$	$-$	$\star$	d^3	d^3	d^2
6	$-$	$-$	$-$	$-$	$-$	$\star$	d^3	d^3
7	$-$	$-$	$-$	$-$	$-$	$-$	$\star$	d^2
8	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$\star$

Пример 7.3: Деревья T' и T'' представлены на Рис. 7.35:
$A' = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $A'' = \{1, 2, 3, 6, 8, 9, 10\}$, $A' \bigcap A'' = \{1, 2, 3, 6, 8\}$.

Близость для множеств вершин имеет вид: $\rho(A', A'') = \frac{1}{2}$.

Таблица 7.6 и Таблица 7.7 содержат соответствующие contain corresponding *факторы доминирования* для $A' \bigcap A''$ в T' и в T'' (изменения указаны 'овалами'). Общая близость имеет вид: $\overline{\rho}(T', T'') = (\frac{1}{2}, \frac{1}{5})$.

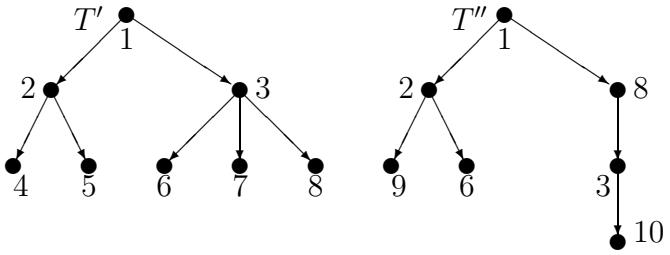


Рис. 7.35. Деревья для примера 7.3

Таблица 7.6. *Фактор доминирования* (T')

	1	2	3	6	8
1	$\star$	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^3	d^3
3	$-$	$-$	$\star$	d^1	d^1
6	$-$	$-$	$-$	$\star$	d^3
8	$-$	$-$	$-$	$-$	$\star$

Таблица 7.7. *Фактор доминирования* (T'')

	1	2	3	6	8
1	$\star$	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^3	d^3
3	$-$	$-$	$\star$	d^1	d^1
6	$-$	$-$	$-$	$\star$	d^3
8	$-$	$-$	$-$	$-$	$\star$

7.5.5 Близость для морфологических структур

Морфологические структуры являются составными, поэтому представляется целесообразным использовать для них векторную близость. В общем случае, структура (Λ) состоит их следующих частей: (i) древовидная модель системы $\mathbf{T}$, (ii) множество висячих вешин $\mathbf{P}$, (iii) множество DAs для каждой висячей вершины $\mathbf{D}$, (iv) ранжировки DAs (т.е., порядковые приоритеты) $\mathbf{R}$, (v) оценки совместимости между DAs $\mathbf{I}$. Векторная близость для двух морфологических структур $\Lambda^\alpha, \Lambda^\beta$ может рассматриваться так:

$\overline{\rho}\ (\Lambda^\alpha, \Lambda^\beta) = (\rho^t(\mathbf{T}^\alpha,\mathbf{T}^\beta),\ \rho^t(\mathbf{P}^\alpha,\mathbf{P}^\beta),\ \rho^t(\mathbf{D}^\alpha,\mathbf{D}^\beta),\ \rho^t(\mathbf{R}^\alpha,\mathbf{R}^\beta),\ \rho^t(\mathbf{I}^\alpha,\mathbf{I}^\beta))$.

Приведем упрощенный пример, когда структура (Λ) состоит из двух частей (а) дерево $T = (A, E)$ (т.е., множество вершин и множество дуг), (б) ранжировки для каждой висячей вершины i ($\bigcup_i R_i$). Тогда близость двух морфологических структур имеет вид:

$$\overline{\rho}(\Lambda', \Lambda'') = (\rho(A', A''), \rho(E', E''), \rho_r(\Lambda', \Lambda'')).$$

Здесь предполагается, что DAs для висячих вершин не изменяются и оценки совместимости не рассматриваются.

Пример 7.4: Деревья T' и T'' представлены на Рис. 7.36:
$A' = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $A'' = \{1, 2, 3, 4, 5, 6, 7, 9\}$, $A' \bigcup A'' = \{1, 2, 3, 4, 5, 6, 7\}$.

Близость для множеств вершин имеет вид: $\rho(A', A'') = \frac{2}{9}$. Таблица 7.8 и Таблица 7.9 содержат соответствующие *факторы доминирования* для $A' \bigcup A''$ в T' и в T'' (изменения указаны 'овалами'). Близость для деревьев имеет вид: $\overline{\rho}(T', T'') = (\frac{2}{9}, \frac{2}{21})$.

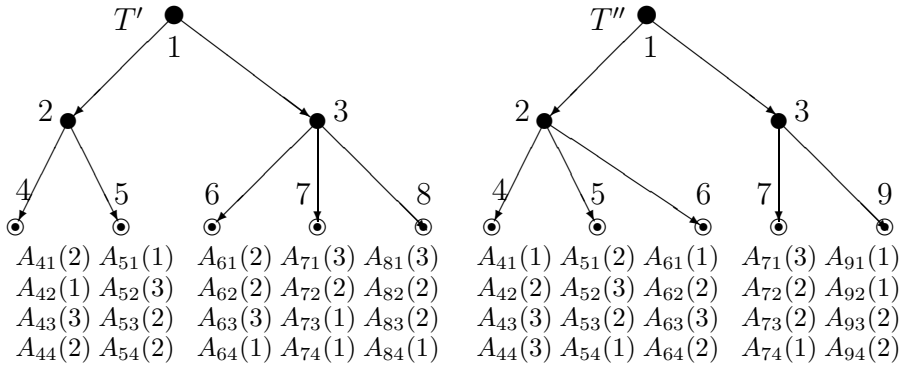


Рис. 7.36. Деревья с морфологией для примера 7.4

Здесь процесс сравнения для ранжировок основан на ранжировках общих висячих вершин: $L(T', T'') = \{4, 5, 6, 7\}$. Таким образом, для каждой указанной вершины можно использовать метрику или близость. В данном случае, используется нормализованное расстояние Кендалла-Тау: вершина 4: $\delta_4(\Lambda', \Lambda'') = \frac{1}{3}$, вершина 5: $\delta_5(\Lambda', \Lambda'') = \frac{1}{2}$, вершина 6: $\delta_6(\Lambda', \Lambda'') = \frac{1}{3}$, вершина 7: $\delta_7(\Lambda', \Lambda'') = \frac{1}{6}$. Результирующую общую близость для ранжировок можно вычислить как среднее значение:

$$\rho_r(\Lambda', \Lambda'') = \frac{\sum_{i \in L(T', T'')} \delta_i(\Lambda', \Lambda'')}{|L(T', T'')|} = 0.33.$$

Окончательно, общая близость имеет вид:

$$\overline{\rho}(\Lambda', \Lambda'') = (\rho(A', A''), \rho(E', E''), \rho_r(\Lambda, \Lambda'')) = (0.22, 0.1, 0.33).$$

Отметим, близость оценок совместимости может быть добавлена к приведенному вектору близости.

Таблица 7.8. *Фактор доминирования* (T')

	1	2	3	4	5	6	7
1	$\star$	d^1	d^1	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^1	d^1	d^2	d^2
3	$-$	$-$	$\star$	d^3	d^3	d^1	d^1
4	$-$	$-$	$-$	$\star$	d^3	d^3	d^3
5	$-$	$-$	$-$	$-$	$\star$	d^3	d^3
6	$-$	$-$	$-$	$-$	$-$	$\star$	d^3
7	$-$	$-$	$-$	$-$	$-$	$-$	$\star$

Таблица 7.9. *Фактор доминирования* (T'')

	1	2	3	4	5	6	7
1	$\star$	d^1	d^1	d^1	d^1	d^1	d^1
2	$-$	$\star$	d^3	d^1	d^1	d^1	d^2
3	$-$	$-$	$\star$	d^3	d^3	d^3	d^1
4	$-$	$-$	$-$	$\star$	d^3	d^3	d^3
5	$-$	$-$	$-$	$-$	$\star$	d^3	d^3
6	$-$	$-$	$-$	$-$	$-$	$\star$	d^3
7	$-$	$-$	$-$	$-$	$-$	$-$	$\star$

7.6 Задачи построения медианы, задачи агрегации

7.6.1 Множества

Рассмотрим случай m множеств: $\{A_1, ..., A_i, ..., A_m\}$. Задача построения расширенной медианы (консенсуса) для указанных m множеств имеет следующий вид. Рассматриваем два параметра $(\forall e \in \bigcup_{i=1}^m A_i)$: "прибыль" или "полезность" $c_e \geq 0$, требуемый ресурс $b_e \geq 0$. Пусть имеется некое базовое множество (например, базовый консенсус) $R_{\{A_i\}} = \widetilde{S}_{\{A_i\}} \subseteq \bigcap_{i=1}^m A_i$ (или $R_{\{A_i\}} = M_{\{A_i\}}$) и $\sum_{e \in R_{AB}} \leq b$, где b - общее ресурсное ограничение. Задача построения "расширенной медианы/консенсуса" имеет вид:

$$\max \sum_{e \in W \subseteq ((\bigcup_{i=1}^m A_i) \backslash R_{\{A_i\}})} c_e \qquad s.t. \sum_{e \in W \subseteq ((\bigcup_{i=1}^m A_i) \backslash R_{\{A_i\}})} b_e \leq b.$$

Данная модель соответствует задаче о рюкзаке.

Теперь можно рассмотреть задачу агрегации для множеств. Сначала рассмотрим задачу агрегации для двух множеств A, B. Метрику обозначим так: $\rho(A, B) \geq 0$, $\rho(A, B) = \rho(B, A)$. В примере используется следующая простая метрика: $\rho(A, B) = |A \bigcap B|/|A \bigcup B|$.

Медианное подмножество (например, модель консенсуса по Кендаллу [295]) имеет вид:

$$M_{AB} = \ arg \min_{\{M\}} (\rho(M, A) + \rho(M, B)).$$

Теперь рассматриваются два параметра $\forall e \in (A \bigcup B)$: "прибыль" или "полезность" $c_e \geq 0$, требуемый ресурс $b_e \geq 0$. Пусть $R_{AB} = \widetilde{S}_{AB}$ (или $R_{AB} = M_{AB}$) используется как базовое множество-"консенсус" и $\sum_{e \in R_{AB}} \leq b$, где b - общее ограничение по ресурсу. Далее рассматривается модель, аналогичная приведенной выше. Задача построения "расширенной медианы/консенсуса" имеет вид:

$$\max \sum_{e \in W \subseteq ((A \bigcup B) \backslash R_{AB})} c_e \qquad s.t. \sum_{e \in W \subseteq ((A \bigcup B) \backslash R_{AB})} b_e \leq b.$$

Данная модель соответствует задаче о рюкзаке.

Теперь можно рассмотреть задачу агрегации в случае многих исходных множеств $\{A_1, ..., A_i, ..., A_m\}$. Аналогично, используются два параметра $\forall e \in \bigcup_{i=1}^m A_i)$: "прибыль" или "полезность" $c_e \geq 0$, требуемый ресурс $b_e \geq 0$. Пусть базовое множество-"консенсус" имеет вид: $R_{\{A_i\}} = \widetilde{S}_{\{A_i\}} \subseteq \bigcap_{i=1}^m A_i$ (или $R_{\{A_i\}} = M_{\{A_i\}}$) и $\sum_{e \in R_{AB}} b_e \leq b$, где b - общее ограничение по ресурсу. Задача имеет вид:

$$\max \sum_{e \in W \subseteq ((\bigcup_{i=1}^m A_i) \backslash R_{\{A_i\}})} c_e \qquad s.t. \sum_{e \in W \subseteq ((\bigcup_{i=1}^m A_i) \backslash R_{\{A_i\}})} b_e \leq b.$$

Эта модель соответствует задаче о рюкзаке.

Следует заметить, что применение других стратегий агрегации может также основывается на моделях рюкзачного типа. Например, в стратегии "сжатия" (удаления) можно использовать надмножество (как объединение исходных множеств), сгенерировать некое множество элементов как кандидатов на удаление, оценить эти элементы по двум параметрам (т.е., "полезность" и требуемый ресурс) и применить модель рюкзачного типа (например, минимизация общей полезности удаляемых элементов при ограничении на общий требуемый ресурс).

7.6.2 Ранжировки

Построение медианы/консенсуса для ранжировок является базовой задачей в области принятия решений [80, 144, 292, 293, 80, 292]. Здесь будут кратко описаны три метода: (i) построение консенсуса на основе задачи о назначении [143, 144, 145, 310]; (ii) эвристический метод [82, 310, 550]; (iii) метод на основе задачи блочного рюкзака [335, 336].

Метод 1. Метод, основанный на использовании расстояния и задачи о назначениях, был исследован в [143, 144, 145] (обычно здесь используется расстояние Кендалла-Тау). Рассмотрим некую версию данного подходы применительно к слоистым структурам Пусть имеется исходное множество элементов (объектов, альтернатив) $A = \{1, ..., i, ..., n\}$. Число уровней (слоев) равно m ($k = \overline{1, m}$). Имеется μ исходных ранжировок множества A: S^1,..., S^λ,..., S^μ. Таким образом, $S^\lambda = \bigcup_{k=1}^{m} A_k^\lambda$. Пусть r_i^λ ($i \in A$) будет приоритет элемента i в S^λ, т.е., номер соответствующего уровня: $r_i^\lambda = k$ if $i \in A_k^\lambda$.

Обозначим результирующую ранжировку (консенсус) так: $S^a = \{A_1^a, ..., A_k^a, ..., A_m^a\}$, и соответствующие приоритеты так: r_i^a $\forall i \in A$. Используются следующие бинарные переменные:

$$x_{ik} = \begin{cases} 1, & \text{если } r_i^a = k \text{ или } i \in A_k^a, \\ 0, & \text{в противном случае.} \end{cases}$$

Задача о назначении для нахождения консенсуса имеет вид (случай слоистых структур):

$$\min \sum_{i=1}^{n} \sum_{k=1}^{m} (\sum_{\lambda=1}^{\mu} |r_i^\lambda - k|) x_{ik} \qquad s.t. \quad \sum_{k=1}^{m} x_{ik} = 1, \ i = \overline{1, n}; \ x_{ik} \in \{0, 1\}.$$

Для базовой задачи о назначении известны полиномиальные алгоритмы [207, 311]. Полученная версия задачи о назначении является более сложной. Здесь для нее используется очевидный жадный алгоритм: выбор ближайшего уровня $\forall i \in A$. Приведем пример (Таблица 7.10):

(а) исходное множество элементов $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$,

(б) три ранжировки (рассматриваются четыре уровня):

$S^1 = \{A_1^1 = \{2, 4\}, A_2^1 = \{9\}, A_3^1 = \{1, 3, 7\}, A_4^1 = \{5, 6, 8\}\}$,

$S^2 = \{A_1^2 = \{2, 3\}, A_2^2 = \{4, 9\}, A_3^2 = \{1, 7\}, A_4^2 = \{5, 6, 8\}\}$,

$S^3 = \{A_1^3 = \{2, 4\}, A_2^3 = \{3, 9\}, A_3^3 = \{1, 5\}, A_4^3 = \{6, 7, 8\}\}$.

Результирующая ранжировка имеет вид (Таблица 7.10):

$$S^a = \{A_1^a = \{2, 4\}, A_2^a = \{3, 9\}, A_3^a = \{1, 7\}, A_4^a = \{5, 6, 8\}\}.$$

Таблица 7.10. Пример для ранжировок

$i \in A$	r_i^1 (S^1)	r_i^2 (S^2)	r_i^3 (S^1)	r_i^a (S^a)	$\overline{r_i^a}$ ($\overline{S^a}$)
1	3	3	3	3	3
2	1	1	1	1	1
3	3	1	2	2	2
4	1	2	1	1	1
5	4	4	3	4	4
6	4	4	4	4	4
7	3	3	4	3	3
8	4	4	4	4	4
9	2	2	2	2	2

Метод 2. Эвристический подход был предложен в [82]. Этот метод достаточно широко используется [550]. Рассмотрим упрощенную версию эвристики для нахождения соответствующего решения $\overline{S^a}$:

$$\overline{r_i^a} = \begin{cases} \lceil (\sum_{\lambda=1}^{\mu} r_i^\lambda)/\mu \rceil, & \text{если } (\sum_{\lambda=1}^{\mu} r_i^\lambda)/\mu - \lceil (\sum_{\lambda=1}^{\mu} r_i^\lambda)/\mu \rceil < 0.5, \\ \lceil (\sum_{\lambda=1}^{\mu} r_i^\lambda)/\mu \rceil + 1, & \text{в противном случае.} \end{cases}$$

Таким образом, решение $\overline{S^a}$ имеет вид (Таблица 7.10):

$$\overline{S^a} = \{\overline{A_1^a} = \{2,4\}, \overline{A_2^a} = \{3,9\}, \overline{A_3^a} = \{1,7\}, \overline{A_4^a} = \{5,6,8\}\}.$$

Метод 3. Здесь рассматривается подход на основе блочной задачи о рюкзаке [335, 336]: исходные ранжировки $\{S^\lambda | \lambda = \overline{1,\mu}\}$ отображаются в "размытую" агрегированную ранжировку S_f^a, т.е., $\{S^\lambda\} \Rightarrow S_f^a,\ S_f^a \in \{S_f\}$, где $\{S_f\}$ - множество всех "размытых" ранжировок. Таким образом, задача имеет вид:

$$\max_{S_f^a \in \{S_f\}} \quad h(S_f^a) \qquad s.t. \quad \eta(S_f^a, S^\lambda) \preceq \eta_o \ \ \forall \lambda = \overline{1,\mu},$$

где h - параметр (качество) результата в виде "усредненной" структуры S_f^a, $\eta(S', S'')$ - векторная близость для двух ранжировок S' и S'', η_o - ограничение на векторную близость (т.е., "опорный" вектор). Рис. 7.37 иллюстрирует данную задачу.

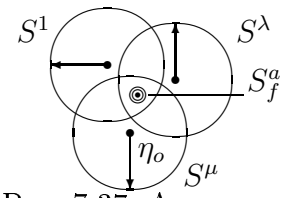


Рис. 7.37. Агрегация [336]

Использованы следующие обозначения: a_{il} - число исходных структур, в которых элемент $i \in A_l$, $l = \overline{1,m}$ (уровни); вектор $\xi_i = (\xi_{i1}, ..., \xi_{il}, ..., \xi_{im})$ определяет "частоту" принадлежности элемента i к уровням $\{A_1, ..., A_l, ..., A_m\}$, где $\xi_{il} = \frac{a_{il}}{\mu}$ (это аналог функции принадлежности элемента i к уровню $l = 1, .., m$). S_f - это множество n интервалов $\{H_i(S_f)\} = \{[d_i^1(S_f), d_i^2(S_f)]\},\ i = \overline{1,n}$. Результирующее "качество" S_f^a основано на следующей функции (похожа на энтропию):

$$\sum_{i=1}^{n} H_i(S_f^a) = \sum_{i=1}^{n} \frac{1}{d_i^2(S_f^a) - d_i^1(S_f^a) + 1} \longrightarrow \max.$$

Далее, используется модульный вектор как близость: $z_o = (z^1, ..., z^k, 0, ..., 0)$. Окончательно, задача имеет вид:

$$\max \sum_{i=1}^{n} H_i(S_f^a) \qquad s.t. \ \ z(S^\lambda, S_f^a) \preceq z_o, \ \forall \lambda = \overline{1,\mu}.$$

Относительно нулевых компонентов $z^{k+1}, ..., z^{k^+}$, можно определить множество допустимых вариантов интервалов $(d_{i\theta} \mid \theta = \overline{1,q_i})$. Задача сводится к следующей модификации задачи блочного рюкзака:

$$\max \sum_{i=1}^{n} \sum_{\theta=1}^{q_i} H_{i\theta}(d_{i\theta}) \kappa_{i\theta}$$

$$s.t. \quad \sum_{r \geq p} \sum_{i=1}^{n} \sum_{\theta=1}^{q_i} b_{i\theta}^r \kappa_{i\theta} \leq \sum_{r \geq p} z^r \ \ p = \overline{1,k}, \qquad \sum_{\theta=1}^{q_i} \kappa_{i\theta} = 1 \ \ i = \overline{1,n}, \ \kappa_{i\theta} \in \{0,1\},$$

где $b_{i\theta}^r$ - сумма компонентов ξ_i, которые отличаются от $d_{i\theta}^1$ $(d_{i\theta}^2)$ по r. Версия описанной схемы агрегирования ранжировок была реализована в СППР КОМБИ [28, 29, 336, 364].

7.6.3 Деревья

При агрегации деревьев, имеется множество исходных деревьев. Можно указать четыре основных подхода: (1) максимальное общее поддерево [53, 189]; (2) медианное дерево (median/agreement tree) [54, 86, 184, 230, 269, 478, 531]; (3) совместимое дерево (compatible tree) [86, 230, 243]; (4) максимальный "медианный" лес (maximum agreement forest) [123, 241, 489, 581, 582]. В основном, указанные задачи относятся к классу NP-трудных задач. [243, 478]. Таким образом, для их решения используются эвристики, приближенные схемы и переборные методы.

Задачу агрегации для множества исходных деревьев $\{T\} = \{T^1, ..., T^i, ..., T^m\}$ можно рассматривать следующим образом (стратегия расширения I):

Стадия 1. Поиск медианного дерева (т.е., "ядра"):

$$T^{agg} = \ arg \ \min_{\{T\}} (\sum_{i=1}^{m} \rho(T, T^i)).$$

Стадия 2. Генерация множества дополнительных элементов (вершины и/или ребра).
Стадия 3. Добавление элементов к T^{agg} (задача рюкзачного типа).

В случае использования векторной близости $\overline{\rho}(T', T'')$, T^{agg} должен искаться на основе поиска Парето-эффективных решений. С другой стороны, можно рассматривать эвристики для построения "ядра", например:

$$K = \bigcup_{i=1}^{m-1} (T^i \bigcap T^{i=1}).$$

7.6.4 Морфологические структуры

Задача агрегации множества исходных структур

$$\{\Lambda\} = \{\Lambda 1, ..., \Lambda^i, ..., \Lambda^m\}$$

может быть рассмотрена так (стратегия дополнения/расширения I):

Стадия 1. Поиск медианного дерева (т.е., "ядра"):

$$\Lambda^{agg} = \ arg \ \min_{\{\Lambda\}} (\sum_{i=1}^{m} \rho(\Lambda, \Lambda^i)).$$

Стадия 2. Генерация множества дополнительных элементов (вершины и/или ребра).
Стадия 3. Добавление элементов к Λ^{agg} (задача рюкзачного типа).

В случае использования векторной близости $\rho(\Lambda', \Lambda'')$, Λ^{agg} должна искаться на основе поиска Парето-эффективных решений. Следует отметить, морфологические структуры (морфологические структуры с совместимостью) являются сложными составными структурами: $\Lambda = \langle \ \mathbf{T}, \mathbf{P}, \mathbf{D}, \mathbf{R}, \mathbf{I} \ \rangle$. Может быть целесообразно использовать следующую эвристическую схему:

Стадия 1. Агрегация множеств системных частей $\{\mathbf{P}\}$.
Стадия 2. Агрегация множеств проектных альтернатив DAs $\{\mathbf{D}\}$.
Стадия 3. Агрегация множеств оценок совместимости $\{\mathbf{I}\}$.
Стадия 4. Агрегация древовидных моделей $\{ \ \mathbf{T} \ \}$.

7.7 Продолжение иллюстративного примера

Рассмотрим заключительную часть иллюстративного примера для ноутбука. Комбинированная стратегия агрегации (стратегия III) используется для двух случаев:

(1) "системное ядро" как расширение подструктуры:

$K' = B_1 \star R_1 \star V_3 \star J_1 \star E_1 \star O_1 \star D_1 \star A_1 \star G_1 \star L_1 \star Q_2$;

(2) "системное ядро" K^*, основанное на многокритериальном выборе наилучших проектных альтернатив для каждой части системы.

Рассмотрим случай 1. Здесь стратегия агрегации как модификация "системного ядра" K' может быть использована. Рассмотрим следующее множество кандидатов для операций модификации:

1. операции добавления: *1.1.* добавление для U: U_1 или U_2 или U_3, *1.2.* добавление для F: F_1 или F_2, *1.3.* добавление для P: P_2 или P_3 или P_4;

2. операции коррекции: *2.1.* замена $B_1 \Rightarrow B_2$, *2.2.* замена $V_3 \Rightarrow V_4$, *2.3.* замена $A_1 \Rightarrow A_3$.

Таблица 7.11 содержит список указанных операций модификации, их порядковые оценки (экспертное оценивание). В результате получается упрощенная задача блочного рюкзака ($c_{ij} = 3 - r_{ij}$, $b = 11.00$):

$$\max \sum_{i=1}^{6} \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \sum_{i=1}^{6} \sum_{j=1}^{q_i} a_{ij} x_{ij} \le b, \sum_{j=1}^{q_i} x_{ij} = 1 \forall i = \overline{1,6}; x_{ij} \in \{0,1\}.$$

Результирующее решение S^{1c} представлено на Рис. 7.38 ($x_{11} = 1$, $x_{12} = 0$, $x_{13} = 0$, $x_{21} = 0$, $x_{22} = 1$, $x_{31} = 0$, $x_{32} = 1$, $x_{33} = 0$, $x_{41} = 0$, $x_{42} = 1$, $x_{51} = 0$, $x_{52} = 1$, $x_{61} = 1$, $x_{62} = 0$). Здесь использовался жадный алгоритм. Оценки совместимости между проектными альтернативами не использовались.

Таблица 7.11. Операции модификации

Операции	Бинарные переменные	Стоимость a_{ij}	Приоритеты r_{ij}
1. Добавление			
1.1. U_1	x_{11}	3	2
1.1. U_2	x_{12}	2	3
1.1. U_3	x_{13}	4	1
1.2. F_1	x_{21}	2	2
1.2. F_2	x_{22}	3	1
1.3. P_2	x_{31}	3	1
1.3. P_3	x_{32}	2	1
1.3. P_4	x_{33}	0	2
2. Замена			
2.1. $B_1 \Rightarrow B_3$	x_{41}	4	1
2.1. Нет	x_{42}	0	2
2.2. $V_3 \Rightarrow V_4$	x_{51}	3	1
2.2. Нет	x_{52}	0	2
2.3. $A_1 \Rightarrow A_3$	x_{61}	2	1
2.3. Нет	x_{62}	0	2

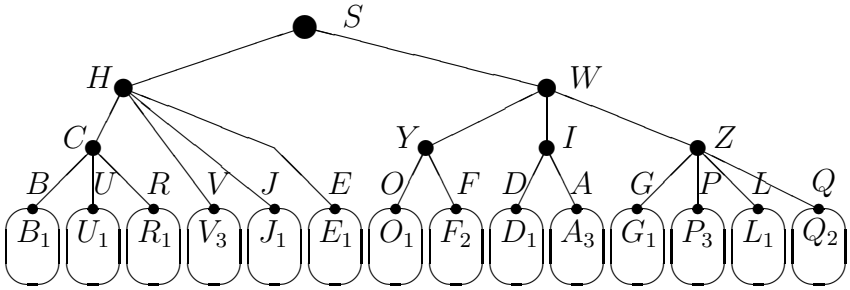


Рис. 7.38. Решение S^{1c} (модификация "системного ядра" K')

Теперь рассмотрим второй случай. Здесь построение “системного ядра” основано на многокритериальном выборе и/или экспертных суждениях. Имеется базовая структура (состав) “системного ядра”: B, U, R, V, O, F, D, G. Для каждой указанной части системы возможно применение процедуры выбора наилучшего системного элемента (проектной альтернативы, рассматриваются элементы исходного решения и дополнительные элементы). Таблица 7.12 содержит проектные альтернативы для компонентов “системного ядра”.

Таблица 7.12. Проектные альтернативы

Проектные альтернативы	Критерии				Приоритеты r_i
	Υ_1	Υ_2	Υ_3	Υ_4	
B_1	3	2	1	2	2
B_2	2	2	1	2	1
B_3	4	1	1	1	2
U_1	3	2	1	2	2
U_2	2	2	1	3	1
U_3	4	1	1	1	2
R_1	1	3	1	3	3
R_2	2	2	2	3	3
R_3	2	2	2	2	1
R_4	3	1	3	1	2
V_1	1	2	1	3	3
V_2	2	1	1	2	2
V_3	3	1	2	1	1
V_4	4	1	2	1	2
E_1	2	1	2	1	1
E_2	0	2	1	2	2
O_1	2	2	1	2	1
O_2	3	1	2	2	2
O_3	1	1	3	1	2
F_1	2	2	1	3	2
F_2	3	1	2	2	1
D_1	1	1	1	1	1
D_2	0	2	2	2	2
G_1	4	1	1	1	1
G_2	4	2	3	2	3
G_3	3	2	3	2	3
G_4	2	3	2	3	2

Использованы следующие критерии (порядковая шкала $[0,4]$): стоимость (Υ_1), полезность (Υ_2), опыт (Υ_3), перспективные особенности (Υ_4). Порядковые оценки основаны на экспертном оценивании, наименьшие оценки являются лучшими. В результате получается следующее “системное ядро” K^*: $K^* = B_2 \star U_2 \star R_3 \star V_3 \star E_1 \star O_1 \star F_2 \star D_1 \star G_1$.

Процесс коррекции системы включает следуюшие операции:

1. добавление: *1.1.* A_1, *1.2.* P_1, *1.3.* L_1;

2. удаление: *2.1.* E_1;

3. замена: *3.1.* $B_2 \Rightarrow B_3$, *3.2.* $U_2 \Rightarrow U_1$, *3.3.* $O_1 \Rightarrow O_3$.

Таблица 7.13 содержит список указанных операций и их оценки. В результате получается задача блочного рюкзака ($c_{ij} = 3 - r_{ij}$, $b = 9.00$):

$$\max \sum_{i=1}^{7}\sum_{j=1}^{2} c_{ij}x_{ij} \qquad s.t. \sum_{i=1}^{7}\sum_{j=1}^{2} a_{ij}x_{ij} \le b,\ \sum_{j=1}^{2} x_{ij} = 1\ \forall i = \overline{1,7};\ x_{ij} \in \{0,1\}.$$

Полученное решение А S^{2c} представлено на Рис. 7.39 ($x_{11} = 1$, $x_{12} = 0$, $x_{21} = 0$, $x_{22} = 1$, $x_{31} = 1$, $x_{32} = 0$, $x_{41} = 1$, $x_{42} = 0$, $x_{51} = 1$, $x_{52} = 0$, $x_{61} = 0$, $x_{62} = 1$, $x_{71} = 1$, $x_{72} = 0$). Был использован жадный алгоритм. Совместимость между проектными альтернативами не рассматривалась.

Таблица 7.13. Операции модификации

Операции	Бинарные переменные	Стоимость a_{ij}	Приоритеты r_{ij}
1. Добавление			
1.1. A_1	x_{11}	1	1
1.1. Нет	x_{12}	0	3
1.2. P_1	x_{21}	3	2
1.2. Нет	x_{22}	0	3
1.3. L_1	x_{31}	1	1
1.3. Нет	x_{32}	0	3
2. Удаление			
2.1. E_1	x_{41}	1	1
2.2. Нет	x_{42}	0	2
3. Замена			
3.1. $B_2 \Rightarrow B_3$	x_{51}	4	1
3.1. Нет	x_{52}	0	3
3.2. $U_2 \Rightarrow U_1$	x_{61}	3	2
3.2. Нет	x_{62}	0	3
3.3. $O_1 \Rightarrow O_3$	x_{71}	1	1
3.3. Нет	x_{72}	0	3

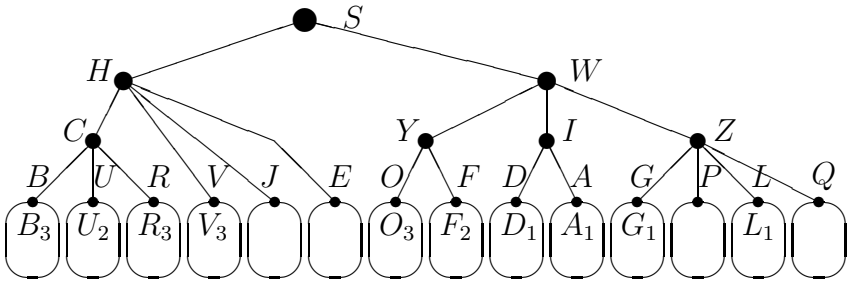


Рис. 7.39. Решение S^{2c}

7.8 Выводы к главе

В данной главе представлено системное описание схем агрегации модульных решений. Предложенные подходы могут оказаться очень полезны при использовании различных методов решения прикладных задач с последующей агрегацией полученных решений.

Можно указать ряд перспективных направлений исследований:

1. исследование и применение различных типов метрик/близостей для модульных решений;

2. анализ более сложных проектных задач, когда модульные решения соответствуют сложным приложениям (например, производственным линиям, цепочкам продуктов);

3. использование различных подходов к учету неопределенности.

Глава 8

Схемы много-стадийного проектирования систем

В данной главе [1] приведено описание комбинаторных схем много-стадийного проектирования (т.е., проектирование траектории системы).

8.1 Предварительные сведения

В главе рассматриваются комбинаторные схемы для много-стадийного проектирования модульных систем (т.е., построение траекторий систем) [374]. При этом предполагается следующее: рассматриваемая модульная система может быть представлена как морфологическая структура: древовидная модель системы, проектные альтернативы (DAs) для висячих вершин модели системы, оценки DAs и их совместимости [336, 344, 358, 365]). Дополнительно, необходимо рассмотреть структуру/сеть верхнего уровня (множество элементов как временных/логических точек/состояний и структуру над этими элементами). В качестве структур для указанного верхнего уровня можно использовать: (а) цепочку, (б) дерево, (в) ациклический граф, (г) общий граф. Таким образом, рассматривается следующее (Рис. 8.1):

Рис. 8.1. Двух-уровневая модель траектории системы

(1) сеть/граф верхнего $G = (H, V)$, где H - вершины (узлы) (временные/логические точки или состояния), V - множество дуг:

(2) морфологическая сруктура для каждой вершины $\mu \in H$: Λ^μ.

Таблица 8.1 содержит краткое описание рассматриваемых схем и задач.

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. chapter 3.
(ii) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapter 4.
(iii) M.Sh. Levin, Towards multistage design of modular systems. Electr. prepr., 13 p., June 19, 2013. Http://arxiv.org/abs/1306.4635 [cs.AI].

Таблица 8.1. Основные типы задач построения траекторий систем

Тип сети верхнего уровня	Тип схемы/задачи проектирования	Прикладной пример
1.Цепочка	(1.1) Последовательная траектория(и) (1.2) Последовательно-параллельная траектория(и)	1.Последовательные и последовательно-параллельные стратегии ранжирования [28, 336, 364] 2.Последовательные стратегии для Web-систем (провайдер) [354] 3.Проектирование start-up бригады (здесь)
2.Древовидная структура	(2.1) Древовидная траектория (2.2) Расширение сети верхнего уровня точками 'анализа', проектирование древовидных траекторий	Планирование лечения (здесь)
3.Ациклический граф	(3.1) Построение маршрута (цепочка, дерево, последовательно-параллельный граф, построение соответствующей траектории(ий) (3.2) Построение покрывающего дерева; решение задачи 2.2	Пример (здесь)
4.Ориентированный граф	(4.1) Построение маршрута (цепочка, дерево, последовательно-параллельный граф); design of corresponding system траектории(ий) (4.2) Построение покрывающего дерева; решение задачи 2.2 (4.3) Построение более простой (по структуре) покрывающей сети; решение задач 4.1 или 4.2	Упрощенная иллюстрация (здесь)

Рассматриваемая проектная комбинаторная схема включает следующее (i) определение множества временных/логических точек (состояний системы); (ii) определение сетевой структуры над указанным множество логических точек (например, цепочка, дерево, ориентированный граф); (iii) задание иерархической модели (морфологической структуры) системы для каждой логической точки; (iv) проектирование модульной системы для каждой логической точки (комбинаторный синтез на основе ИММП или задачи блочного рюкзака) для получения нескольких решений; (v) выбор системного решения для каждой логической точки с учетом качества решений и качества совместимости между соседними решениями (в рамках структуры над логическими точками), здесь также используется логический синтез.

В основном используется структура над логическим точками в виде цепочки (т.е., временной цепочки). В дополнение, могут рассматриваться более сложные случаи: (а) структура над логическими точками имеет вид дерева, (б) структура над логическими точками имеет вид ориентированного графа (орграф). В данной главе примеры иллюстрируют построение системных траекторий. Дополнительно, представлено краткое описание данного подхода для задач на основе нескольких областей.

8.2 Общие схемы

Предлагается следующая двух-уровневая комбинаторная схема для много-стадийного проектирования или построения траектории системы (Рис. 8.2) [336, 344, 374]:

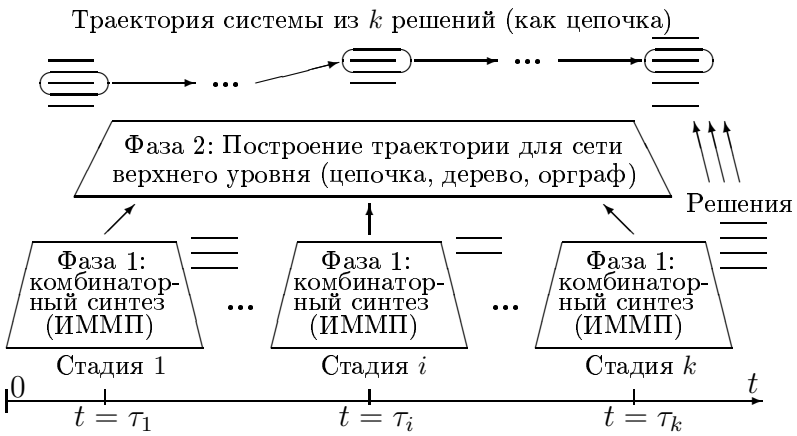


Рис. 8.2. Иллюстрация проектирования траектории

Фаза 0. Генерация общей структуры проектной задачи:

0.1. генерация временных/логических точек,

0.2. генерация сети верхнего уровня над логическими точками (цепочка, дерево, орграф),

0.3. формулирование подпроблем комбинаторного синтеза для каждой логической точки: морфологическая модель системы, проектные альтернативы (DAs), критерии для оценки DAs, оценки DAs и их совместимости).

Фаза 1. Иерархическое проектирование проектных решений для каждой логической точки (получение Парето-эффективных решений).

Фаза 2. Проектирование траектории системы как комбинации решений в логических точках. Результирующая системная траектория может иметь вид: (а) цепочка решений (последовательная или последовательно-параллельная траектория), (б) дерево решений, (в) орграф решений.

На Рис. 8.2 представлены следующие локальные решения (т.е., в логических точках): (а) стадия 1: S_1^1, S_2^1, S_3^1; (б) стадия i: S_1^2, S_2^2; (в) стадия k: $S_1^k, S_2^k, S_3^k, S_4^k$. Полученная системная траектория имеет вид: $\alpha =< S_2^1 \star ... \star S_1^i \star ... \star S_2^k >$. На Рис. 8.3 представлен иллюстративный пример для трех-стадийной траектории, пример траектории (цепочка) имеет вид: $\beta =< S_2^{\tau_1} \star S_3^{\tau_2} \star S_1^{\tau_3} >$.

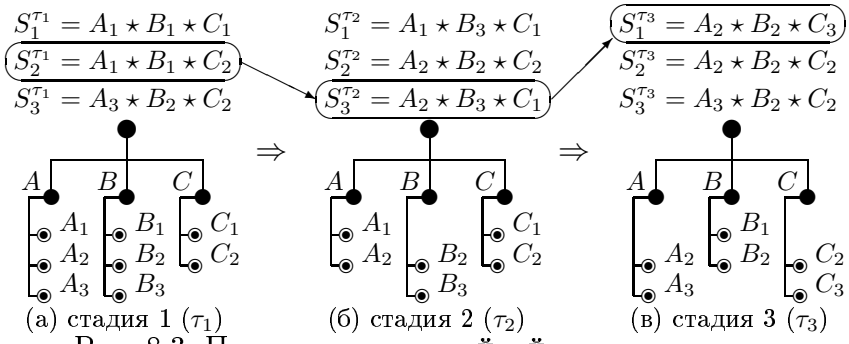
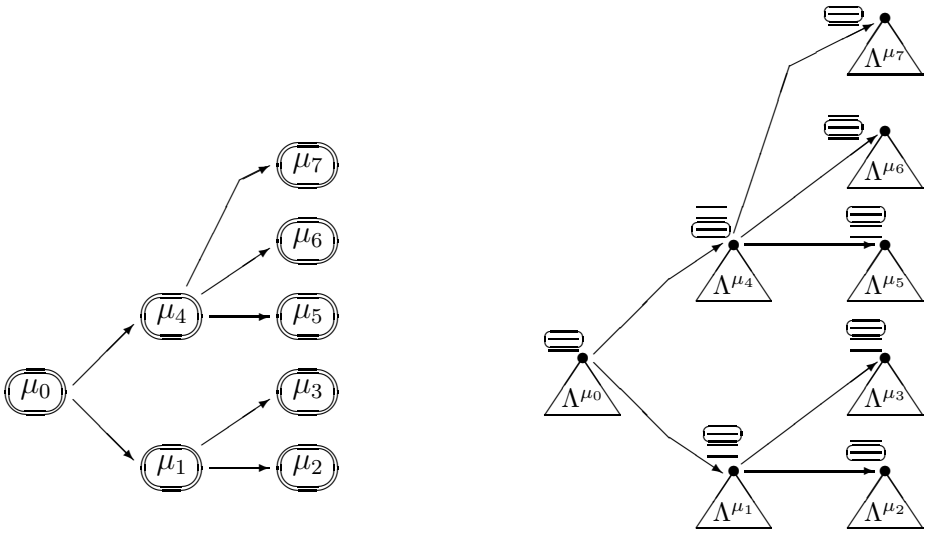


Рис. 8.3. Пример трех-стадийной траектории

Иллюстрация для древовидной системной траектории представлена на Рис. 8.4:



(а) сеть верхнего уровня (дерево) (б) системная траектория

Рис. 8.4. Иллюстрация для древовидной траектории

(i) восемь логических точек: $\{\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7\}$,

(ii) восемь соответствующих морфологических структур:

$\{\Lambda^{\mu_0}, \Lambda^{\mu_1}, \Lambda^{\mu_2}, \Lambda^{\mu_3}, \Lambda^{\mu_4}, \Lambda^{\mu_5}, \Lambda^{\mu_6}, \Lambda^{\mu_7}\}$.

При этом, представлены следующие системные решений: (a) точка μ_0: $S_1^{\mu_0}, S_2^{\mu_0}$; (b) точка μ_1: $S_1^{\mu_1}, S_2^{\mu_1}, S_3^{\mu_1}$; (c) точка μ_2: $S_1^{\mu_2}, S_2^{\mu_2}$; (d) точка μ_3: $S_1^{\mu_3}, S_2^{\mu_3}, S_3^{\mu_3}$; (e) точка μ_4: $S_1^{\mu_4}, S_2^{\mu_4}, S_3^{\mu_4}$; (f) точка μ_5: $S_1^{\mu_5}, S_2^{\mu_5}, S_3^{\mu_5}$; (h) точка μ_6: $S_1^{\mu_6}, S_2^{\mu_6}, S_3^{\mu_6}$; (i) точка μ_7: $S_1^{\mu_6}, S_2^{\mu_6}$. Таким образом, системная траектория (дерево) состоит из следующих локальных решений: $\gamma = \{S_1^{\mu_0}, S_1^{\mu_1}, S_2^{\mu_2}, S_1^{\mu_3}, S_3^{\mu_4}, S_1^{\mu_5}, S_2^{\mu_6}, S_1^{\mu_7}\}$. Очевидно, что структура системной траектории γ соответствует исходному дереву (Рис. 8.5).

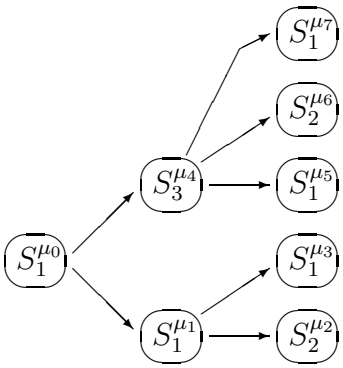


Рис. 8.5. Пример древовидной траектории

В случае древовидных траекторий представляется целесообразным использование дополнительных узлов как точек 'анализа/принятия решений' для проведения анализа реализации локального решения и выбора следующего шага (т.е., направление перехода). На Рис. 8.6 приведен пример такого расширенной древовидной сети с соответствующими точками 'анализа/принятия решений': a_0, a_1, a_4. Здесь результирующая траектория представляет собой цепочку, например (для Рис. 8.6): $\gamma' =< S^{\mu_0} \star S^{\mu_4} \star S^{\mu_5} >$.

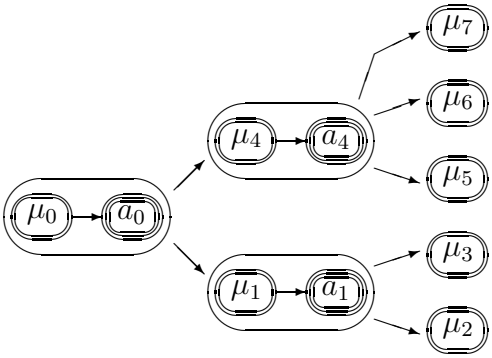


Рис. 8.6. Древовидная сеть с узлами 'анализа'

На Рис. 8.7 представлен пример орграфа с соответствующими системными траекториями. Здесь могут рассматриваться следующие ситуации (задачи):

(i) построение маршрута на основе исходного орграфа (например: последовательный маршрут в виде цепочки, последовательно-параллельный маршрут, древовидный маршрут);

(ii) построение покрывающего дерева для исходного графа и исследование предыдущей задачи (включая использование дополнительных узлов 'анализа').

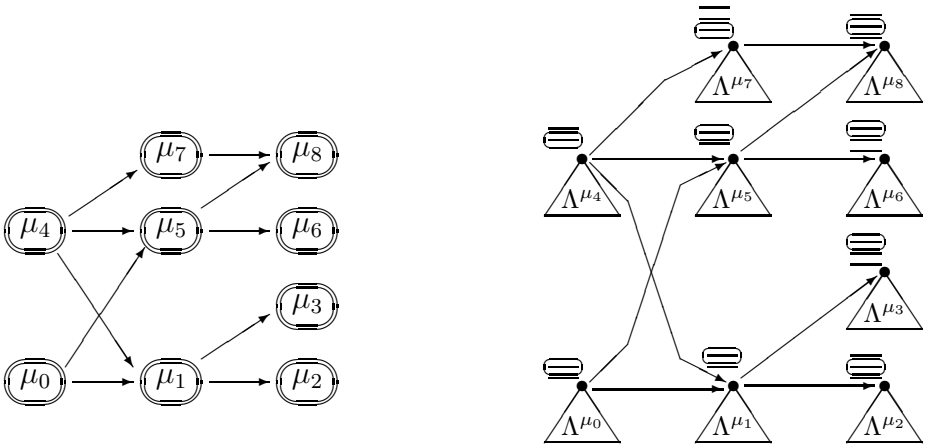
Можно привести следующие примеры маршрутов (Рис. 8.7):

(а) маршрут-цепочка (последовательный): $< S_2^{\mu_4} \star S_3^{\mu_7} \star S_2^{\mu_8} >$,

(б) древовидный маршрут основан на семи локальных решениях:

$\{S_2^{\mu_4}, S_1^{\mu_5}, S_1^{\mu_6}, S_2^{\mu_8}, S_1^{\mu_1}, S_2^{\mu_2}, S_1^{\mu_3}\}$,

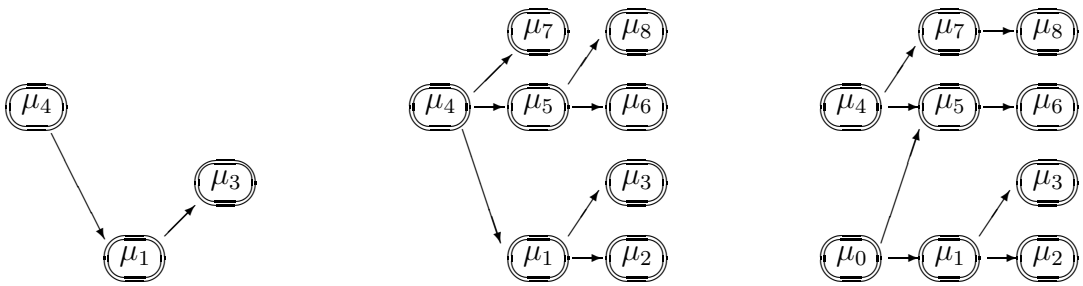
(в) последовательно-параллельный маршрут основан на четырех локальных решениях: $\{S_2^{\mu_4}, S_1^{\mu_5}, S_3^{\mu_7}, S_2^{\mu_8}\}$.



(а) сеть верхнего уровня (орграф) (б) системная траектория

Рис. 8.7. Иллюстрация системных траекторий на основе орграфа

Исходная сеть верхнего уровня (Рис. 8.7а) может аппроксимироваться более простыми покрывающими структурами: (а) цепочка (Рис. 8.8а), (б) аппроксимирующее покрывающее дерево (Рис. 8.8б), (в) покрывающая упрощенная сеть (Рис. 8.8в).



(а) цепочка $< \mu_4, \mu_1, \mu_3 >$ (б) покрывающее дерево (в) упрощеная сеть

Рис. 8.8. Аппроксимирующие структуры для сети верхнего уровня

8.3 Иллюстративные примеры

8.3.1 Четырех-стадийная траектория start-up группы

Рассматривается следующая общая древовидная структура для start-up группы:

0. Древовидная структура $S = L \star R \star E \star M$:

1. Лидер проекта L: нет L_0, частичное участие (неполный рабочий день), консультирование L_1, полной участие (полный рабочий день) L_2.

2. Исследователь R: нет R_0, частичное участие (неполный рабочий день), консультирование R_1, полной участие (полный рабочий день) R_2, два исследователя $R_3 = R_1 \& R_2$.

3. Инженер E: нет E_0, частичное участие (неполный рабочий день), консультирование E_1, полной участие (полный рабочий день) E_2, два инженера $E_3 = E_1 \& E_2$.

4. Менеджер M: нет M_0, частичное участие (неполный рабочий день), консультирование M_1, полной участие (полный рабочий день) M_2.

Эта структура исследуется для четырех последовательных временных точек:

Стадия 0 ($t = \tau_0$) (Рис. 8.9, приоритеты DAs показаны в круглых скобках; Таблица 8.2): генерация базовой идеи для нового родукта/системы, подготовка проекта предложения.

Стадия 1 ($t = \tau_1$) (Рис. 8.10, приоритеты DAs указаны в круглых скобках; Таблица 8.3): проектирование прототипа предлагемого продукта/системы, подготовка исследовательских материалов в виде научных статей, презентаций на конференциях, подготовка патента, поиск инвесторов.

Стадия 2 ($t = \tau_2$) (Рис. 8.11, приоритеты DAs указаны в круглых скобках; Таблица 8.4): проектирование предвартельной версии продукта/системы, анализ рынка (маркетинг), подготовка бизнес-плана, поиск инвесторов.

Стадия 3 ($t = \tau_3$) (Fig. 8.12, приоритеты DAs указаны в круглых скобках; Таблица 8.5): проектирование версии системы для реального приложения(ий), поиск реальных прикладных областей применения, маркетинг.

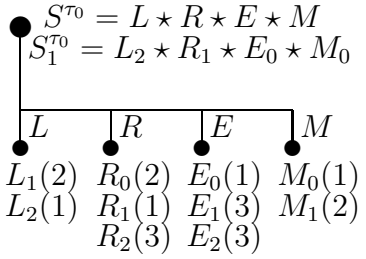


Рис. 8.9. Структура группы (τ_0)

Таблица 8.2. Совместимость (τ_0)

	R_0	R_1	R_2	E_0	E_1	E_2	M_0	M_1
L_1	1	2	1	3	3	2	3	2
L_2	1	3	2	3	3	2	3	2
R_0				3	3	2	3	1
R_1				3	3	1	3	3
R_2				3	3	1	3	2
E_0							3	3
E_1							3	3
E_2							3	2

Результирующее составное Парето-эффективное решение для τ_0 (Рис. 8.9) имеет вид (Рис. 8.13):

$S_1^{\tau_0} = L_2 \star R_1 \star E_0 \star M_0,\ N(S_1^{\tau_0}) = (3; 4, 0, 0)$.

Результирующие составные Парето-эффективные решения для τ_1 (Рис. 8.10) имеют вид (Рис. 8.13):

(а) $S_1^{\tau_1} = L_2 \star R_1 \star E_2 \star M_0,\ N(S_1^{\tau_1}) = (2; 4, 0, 0)$;

(б) $S_2^{\tau_1} = L_2 \star R_1 \star E_1 \star M_0,\ N(S_2^{\tau_1}) = (3; 3, 1, 0)$.

Результирующее составное Парето-эффективное решение для τ_2 (Рис. 8.11) имеет вид (Рис. 8.13):

$S_1^{\tau_2} = L_2 \star R_2 \star E_3 \star M_1,\ N(S_1^{\tau_2}) = (3; 4, 0, 0)$.

Результирующие составные Парето-эффективные решения для τ_3 (Рис. 8.12) имеют вид (Рис. 8.13):

(а) $S_1^{\tau_3} = L_1 \star R_2 \star E_3 \star M_2,\ N(S_1^{\tau_3}) = (2; 4, 0, 0)$;

(б) $S_2^{\tau_3} = L_2 \star R_2 \star E_3 \star M_2,\ N(S_2^{\tau_3}) = (3; 3, 1, 0)$.

Таблица 8.6 содержит оценки совместимости для полученных локальных решений. Предполагается, что локальные решения имеют одинаковые приоритеты (т.е., 1).

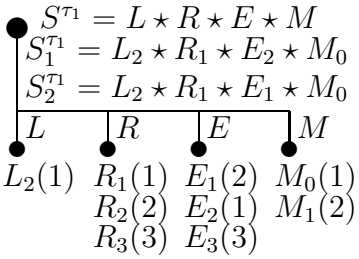


Рис. 8.10. Структура группы (τ_1)

Таблица 8.3. Совместимость (τ_1)

	R_1	R_2	R_3	E_1	E_2	E_3	M_0	M_1
L_2	3	2	2	3	3	1	3	1
R_1				3	2	2	3	1
R_2				2	2	1	3	3
R_3				1	2	3	2	2
E_1							3	3
E_2							3	2
E_3							3	2

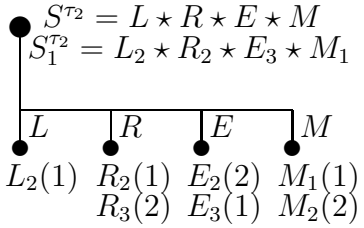


Рис. 8.11. Структура группы (τ_2)

Table 8.4. Совместимость (τ_2)

	R_2	R_3	E_2	E_3	M_1	M_2
L_2	3	2	2	3	3	2
R_2			2	3	3	2
R_3			2	3	3	2
E_2						2
E_3						2

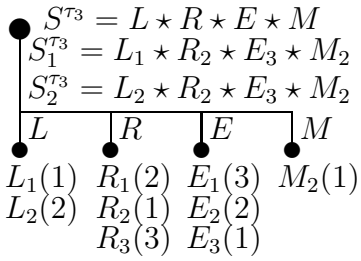


Рис. 8.12. Структура группы (τ_3)

Таблица 8.5. Совместимость (τ_3)

	R_1	R_2	R_3	E_1	E_2	E_3	M_2
L_1	2	2	2	2	3	3	3
L_2	2	3	2	2	3	3	3
R_1				3	2	3	2
R_2				3	3	3	3
R_3				3	3	2	3
E_1							2
E_2							3
E_3							3

Результирующая Парето-эффективная траектория имеет вид (Рис. 8.14):

$\alpha^{team} =< S_1^{\tau_0} \star S_2^{\tau_1} \star S_1^{\tau_2} \star S_1^{\tau_3} >,\ N(\alpha^{team}) = (3; 4, 0, 0)$.

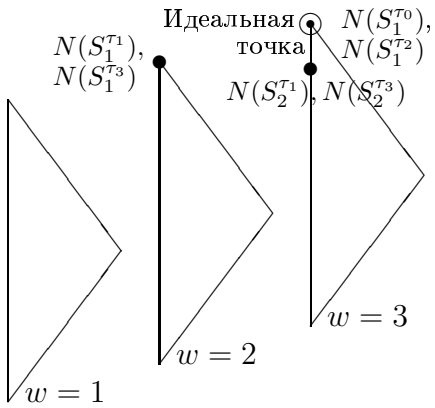


Рис. 8.13. Решетка качества

Таблица 8.6. Совместимость

	$S_1^{\tau_1}$	$S_2^{\tau_1}$	$S_1^{\tau_2}$	$S_1^{\tau_3}$	$S_2^{\tau_3}$
$S_1^{\tau_0}$	2	3	3	3	3
$S_1^{\tau_1}$			3	3	3
$S_2^{\tau_1}$			2	3	3
$S_1^{\tau_2}$				3	2

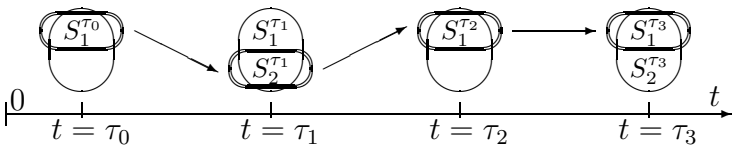


Рис. 8.14. Траектория группы

8.3.2 Древовидная траектория медицинского лечения

Рассматривается следующий пример плана медицинского лечения: (i) базовая древовидная структура плана лечения детской астмы из [383] (упрощенная версия); (ii) сеть верхнего уровня в виде дерева решений (Рис. 8.6); (iii) модифицированная древовидная структура плана лечения для каждого узла дерева решений. Рассматриваемая древовидная траектория плана лечения с узлами 'анализ/принтие решений' представлена на Рис. 8.15. Каждый узел этой траектории основан на упрощенной иерархической структуре базового плана лечения из [383].

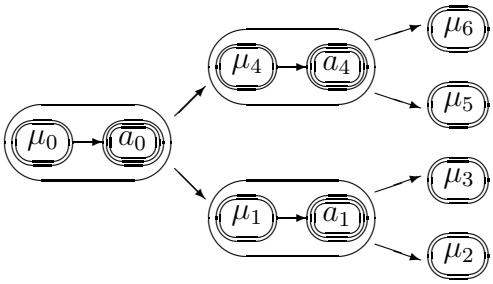


Рис. 8.15. Дерево решений плана лечения

Таким образом, рассматриваемая структура базового плана имеет следующий вид (приоритеты DAs указаны в круглых скобках) (Рис. 8.16):

0. План лечения $S = X \star Y \star Z$.

1. Основные подходы $X = J \star M$:

1.1. Физиотерапия J: нет $J_0(2)$, массаж $J_1(2)$, ингаляция $J_2(2)$, сауна $J_3(3)$, рефлексологическая терапия $J_4(3)$, лазерная терапия $J_5(3)$, массаж для специальных центров/точек $J_6(3)$, рефлексотерапия для специальных центров $J_7(4)$, гало-камеры и соляные шахты $J_8(2)$.

1.2. Лекарства M: нет $M_0(2)$, витамины $M_1(3)$, sodium chromoglycate (один месяц и два раза в год) $M_2(2)$, sodium chromoglycate (два месяца) $M_3(3)$, sodium chromoglycate (три месяца) $M_4(3)$.

2. Физиологическая и экологическая среда $Y = A \star B \star C$:

2.1. Физиологический климат A: нет $A_0(3)$, консультирование физиолога $A_1(2)$.

2.2. Домашняя экологическая среда B: нет $B_0(3)$, водная уборка $B_1(1)$, чистка книжной пыли $B_2(2)$, удаление хлопковых и шерстяных вещей $B_3(1)$, удаление ковров $B_4(1)$, исключение контактов с животными $B_5(2)$, уничтожение насекомых $B_6(2)$, удаление цветов $B_7(1)$, агрегированные альтернативы $B_8 = B_1 \& B_4 \& B_5 \& B_7(1)$.

2.2. Общая экологическая среда C: нет $C_0(4)$, улучшение среды обитания $C_1(1)$.

3. Режим, отдых $Z = O \star K$:

3.1. Режим O: нет $O_0(3)$, отдых в полдень $O_1(1)$, специальные физические действия (дренаж, отхаркивание мокроты) $O_2(1)$, спорт (бег, лыжи, плавание) $O_3(1)$, комфортная ванна $O_4(1)$, холодная ванна $O_5(2)$, исключение электронных игр $O_6(2)$, агрегированная альтернатива $O_7 = O_2 \& O_4(2)$, агрегированная альтернатива $O_8 = O_3 \& O_5(2)$.

3.2. Отдых K: нет $K_0(2)$, отдых в лесной среде $K_1(2)$, отдых около моря $K_2(3)$, отдых в горах $K_3(4)$, специальный лагерь $K_4(3)$, лечение в соляных шахтах $K_5(3)$.

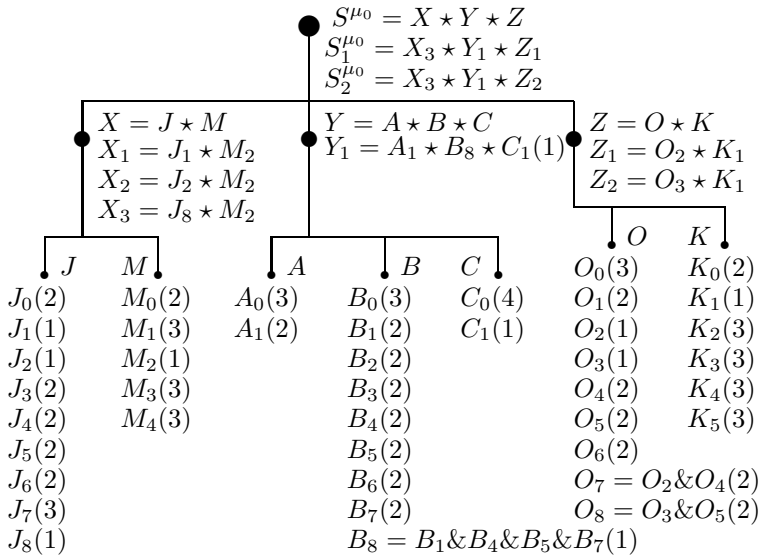


Рис. 8.16. Иерархическая модель плана лечения [344, 383]

Иерархическая модель (морфологическая структура) на Рис. 8.16 соответствует логической точке μ_0 (Рис. 8.15). Оценки совместимости для DAs представлены в Таблице 8.7, Таблице 8.8, Таблице 8.9 [344, 383] (упрощенные версии, для всех логических точек $\{\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6\}$). Оценки совместимости DAs на высшем иерархическом уровне представлены в Таблице 8.10 (μ_0). Таблица 8.11 содержит описание логических точек, включая ссылки на соответствующие морфологические структуры (Рис. 8.17, Рис. 8.18, Рис. 8.19, Рис. 8.20).

Для точки μ_0 (Рис. 8.16), Парето-эффективные DAs имеют вид:

(1) локальные Парето-эффективные решения для подсистемы X: $X_1 = J_1 \star M_2$, $N(X_1) = (3; 2, 0)$; $X_2 = J_2 \star M_2$, $N(X_2) = (3; 2, 0)$; $X_3 = J_8 \star M_2$, $N(X_3) = (3; 2, 0)$;

(2) локальные Парето-эффективные решения для подсистемы Y: $Y_1 = A_1 \star B_8 \star C_1$, $N(Y_1) = (3; 2, 1, 0)$;

(3) локальные Парето-эффективные решения для подсистемы Z: $Z_1 = O_1 \star K_2$, $N(Z_1) = (3; 2, 0, 0)$; $Z_2 = O_2 \star K_2$, $N(Z_2) = (3; 2, 0, 0)$.

(4) результирующие Парето-эффективные решения для системы S: (а) $S_1^{\mu_0} = X_3 \star Y_1 \star Z_1$, (б) $S_2^{\mu_0} = X_3 \star Y_1 \star Z_2$.

Для точки μ_1 (Рис. 8.17), результирующее Парето-эффективное решений имеет вид: $S_1^{\mu_1} = A_1 \star B_8 \star C_1$, $N(S_1^{\mu_1}) = (3; 3, 0, 0)$.

Для точки μ_2 (Рис. 8.18, и для точки μ_5), результирующие Парето-эффективные DAs имеют вид: (а) $S_1^{\mu_2} = O_7 \star K_1$, $N(S_1^{\mu_2}) = (3; 1, 1, 0)$; (б) $S_2^{\mu_2} = O_8 \star K_1$, $N(S_2^{\mu_2}) = (3; 1, 1, 0)$.

Для точки μ_3 (Рис. 8.19), результирующие Парето-эффективные DAs имеют вид:

(а) $S_1^{\mu_3} = J_1$, (б) $S_2^{\mu_3} = J_2$, (в) $S_3^{\mu_3} = J_8$.

Для точки μ_4 (Рис. 8.20, и для μ_6), результирующие Парето-эффективные DAs имеют вид: (а) $S_1^{\mu_4} = J_1 \star M_1$, $N(S_1^{\mu_2}) = (3; 2, 0)$; (б) $S_2^{\mu_4} = J_2 \star M_1$, $N(S_2^{\mu_2}) = (3; 2, 0)$; (в) $S_3^{\mu_4} = J_8 \star M_1$, $N(S_3^{\mu_2}) = (2; 2, 0)$.

Таблица 8.12 содержит описание точек 'анализ/принятие решений'. Пример финального древовидного решения представлен на Рис. 8.21.

Таблица 8.7. Совместимость

	M_0	M_1	M_2	M_3	M_4
J_0	0	3	3	3	3
J_1	3	3	3	3	3
J_2	3	3	3	3	3
J_3	3	3	3	2	2
J_4	3	3	3	2	2
J_5	3	3	3	2	2
J_6	3	3	3	3	3
J_7	3	3	3	3	3
J_8	3	2	2	2	2

Таблица 8.8. Совместимость

	K_0	K_1	K_2	K_3	K_4	K_5
O_0	0	3	3	3	3	3
O_1	3	3	3	3	3	3
O_2	3	3	3	3	2	3
O_3	3	3	3	3	3	3
O_4	3	2	2	2	2	2
O_5	3	3	3	3	3	3
O_6	3	3	3	3	3	3
O_7	3	3	3	3	3	3
O_8	3	3	3	3	3	3

Таблица 8.9. Совместимость

	C_0	C_1	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
A_0	0	3	0	3	3	2	3	2	3	3	3
A_1	3	3	2	3	3	2	3	2	3	3	3
C_0			0	3	3	2	3	2	3	3	3
C_1			2	3	3	2	3	2	3	3	3

Таблица 8.10. Совместимость

	Y_1	Z_1	Z_2
X_1	3	2	2
X_2	2	2	2
X_3	3	3	3
Y_1		3	3

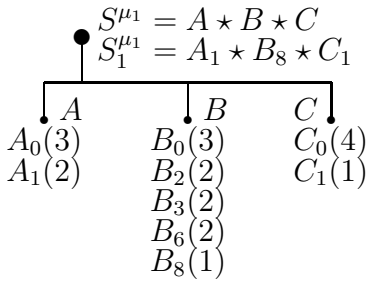


Рис. 8.17. План лечения для μ_1

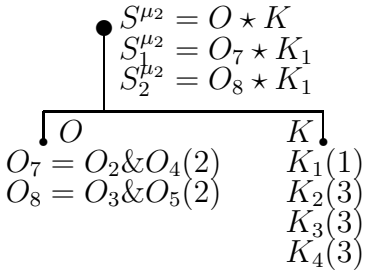


Рис. 8.18. План лечения для μ_2

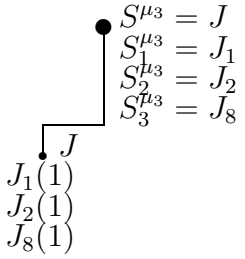


Рис. 8.19. План лечения для μ_3

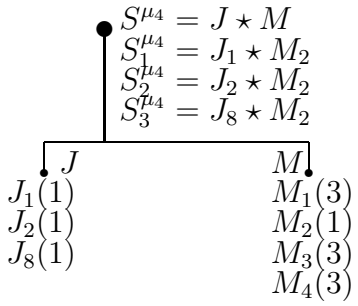


Рис. 8.20. План лечения для μ_4

Таблица 8.11. Логические точки

Логическая точка	Описание
μ_0	Базовое (полное) лечение (Рис. 8.16)
μ_1	Лечение средой (Рис. 8.17)
μ_2	Дополнительное лечение отдыхом (Рис. 8.18)
μ_3	Дополнительная физиотерапия (Рис. 8.19)
μ_4	Дополнительная физиотерапия и лечение лекарствами (Рис. 8.20)
μ_5	Дополнительное лечение средой (Рис. 8.18)
μ_6	Дополнительная физиотерапия и лечение лекарствами (Рис. 8.20)

Таблица 8.12. Точки 'анализ/решение'

Точка 'анализ/решение'	Описание
a_0	(i) хороший результат, переход к μ_1 (ii) результат недостаточен, переход к μ_4
a_1	(i) хороший результат, переход к μ_2 (ii) результат недостаточен, переход к μ_3
a_4	(i) хороший результат, переход к μ_5 (ii) результат недостаточен, переход к μ_6

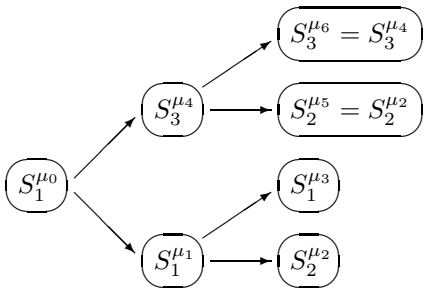


Рис. 8.21. Древовидная траектория

8.3.3 Упрощенный пример для орграфа

Рассмотрим модификацию предыдущего медицинского примера. Таблица 8.13 и Таблица 8.14 содержат описания логических точек и точек 'анализ/принятие решений'. Используются те же морфологические структуры. Дерево решения из Рис. 8.15 трансформируется в общий орграф (с обратной связью): Рис. 8.22. Пример решения в виде орграфа представлен на Рис. 8.23 (для конкретного пациента).

Таблица 8.13. Логические точки

Логическая точка	Описание
μ_0	Основное (полное) лечение (Рис. 8.16)
μ_1	Лечение средой (Рис. 8.17)
μ_2	Дополнительное лечение отдыхом (Рис. 8.18)
μ_3	Дополнительная физиотерапия (Рис. 8.19)
μ_4	Дополнительная физиотерапия и лечение лекарствами (Рис. 8.20)

Таблица 8.14. Точки 'анализ/принятие решений'

Точка 'анализ/ решение'	Описание
a_0	(i) хороший результат, переход к μ_1 (ii) результат недостаточен, переход к μ_4
a_1	(i) хороший результат, переход к μ_2 (ii) результат недостаточен, переход к μ_3
a_4	(i) хороший результат, переход к μ_2 (ii) результат на среднем уровне, переход к μ_1 (iii) результат недостаточен, переход к μ_0

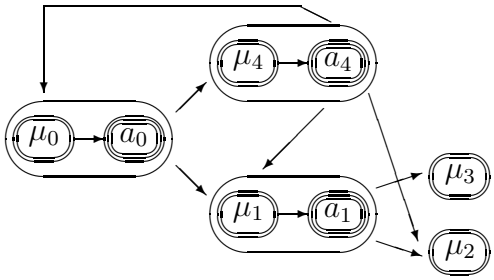


Рис. 8.22. Пример общего орграфа

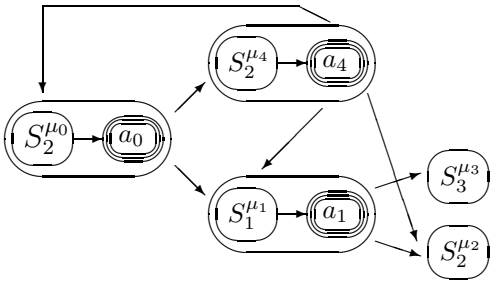


Рис. 8.23. Решение в виде орграфа

8.3.4 О задачах для нескольких областей

Много-стадийные стратегии могут рассматриваться и в случае комбинирования нескольких областей, например: (а) проектирование системы, тестирование системы; (б) медицинская диагностика, планирование лечения.

Сначала приведем предварительный пример. Рис. 8.24 иллюстрирует много-стадийную траекторию для много-стадийной диагностики или системного тестирования:

(i) точки $\{\theta_0, \theta_1, ..., \theta_k\}$ соответствуют диагостике/тестированию,

(ii) точки $\{a_0, a_1, ..., a_k\}$ соответствуют анализу/принятию решений,

(iii) морфологические структуры для тестовых точек: $\{\Lambda^{\theta_0}, \Lambda^{\theta_1}, ..., \Lambda^{\theta_k}\}$.

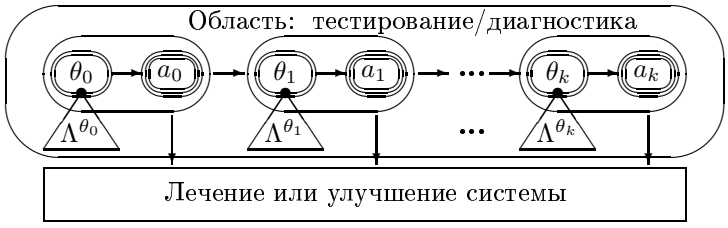


Рис. 8.24. Пример для диагностики - лечения

Теперь можно рассмотреть много-стадийные траектории для двух областей (Рис. 8.25): (а) тестирование системы, (б) проектирование системы. Следующие обозначения использованы:

(i) точки $\{\theta_1, \theta_2, ..., \theta_k\}$ соответствуют системному тестированию или диагностике,

(ii) точки $\{a_0, a_1, ..., a_k\}$ соответствуют процессу анализа/принятия решений,

(iii) точки $\{\xi_1, \xi_2, ..., \xi_q\}$ соответствуют системному проектированию или перепроектированию.

(iv) морфологические структуры для тестовых точек:

$\{\Lambda^{\theta_0}, \Lambda^{\theta_1}, ..., \Lambda^{\theta_k}\}$, and

(v) морфологические структуры для точек проектирования или перепроектирования:

$\{\Lambda^{\xi_0}, \Lambda^{\xi_1}, ..., \Lambda^{\xi_q}\}$.

Таким образом, комбинированная много-стадийная траектория включает две части: (а) много-стадийную траекторию для тестирования, (б) много-стадийную траекторию для улучшения/перепроектирования.

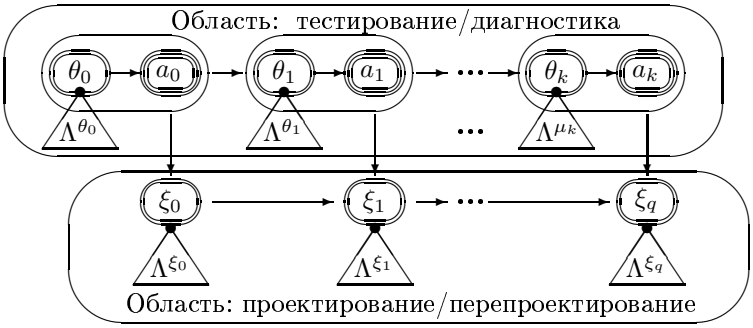


Рис. 8.25. Траектория: тестирование & проектирование

8.4 Выводы к главе

В данной главе описано много-стадийное проектирование модульных систем (проектирование траекторий систем). В будущем представляются важными следующие направления исследований:

1. исследование различных приложений (например, медицина, системы связи);

2. специальные исследования моделей с дополнительными точками 'анализа/принятия решений';

3. использование различных подходов к учету неопределенности;

4. исследование описанного подхода с учетом изменений внешних требований.

Глава 9

Схемы комбинаторной эволюции систем

В данной главе [1] приведено описание инженерных схем для комбинаторного моделирования эволюции модульных систем.

9.1 Предварительные сведения

В последние годы существенно возрастает важность задач моделирования эволюции и прогнозирования модульных систем. В Таблице 9.1 указаны различные направления исследований задач эволюции/развития систем. В данной главе описаны комбинаторные схемы для моделирования эволюции и прогнозирования модульных систем. На Рис. 9.1. представлен упрощенный иллюстративный пример двух-стадийной эволюции для составной системы $S = A \star B \star C$:

$$S_1 \Longrightarrow S_2 \Longrightarrow S_3.$$

При этом использованы следующие операции изменения:

1. Операции изменения для $S^1 \Longrightarrow S^2$: (a) изменение $B_1 \to B_3$, (b) изменение $C_1 \to C_2$;

2. Операции изменения для $S^2 \Longrightarrow S^3$: (a) изменение $A_1 \to A_2$, (b) изменение $B_3 \to B_2$, (c) изменение $C_2 \to C_3$; (d) расширение системы: добавление подсистемы D и элемента D_2.

Очевидно, что выделенные системные изменений могут быть использованы как основа для прогнозирования системы (как набор перспективных изменений системы). Такой набор перспективных изменений может расширятся и модифицироваться (например, на основе экспертных процедур).

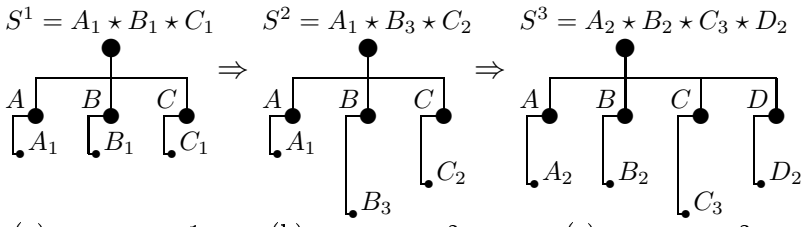
Предлагаемая комбинаторная схема для моделирования эволюции системы включает следующие стадии: (1) построение общей иерархическое модели исследуемой системы; (2) выделение изменений между поколениями системы (как некое множество операций изменения) и их оценивание (многокритериальный анализ включай использование интервальных оценок в виде мультимножеств, бинарных отношений над множеством операций изменения: совместимость, дополнительность, педшествование); (3) комбинаторный синтез (т.е., прогноз) системы как выбор перспективных операций изменения системы с учетом

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Composite Systems Decisions. Springer, 2006. chapter 5.
(ii) M.Sh. Levin, Towards combinatorial evolution of composite systems. *Expert Systems with Applications*, 40(4), 2013, pp. 1342-1351.
(iii) M.Sh. Levin, Towards Multistage Design of Modular Systems. Electr. prepr., 13 p., June 19, 2013. http://arxiv.org/abs/1306.4635 [cs.AI]
(iv) Levin M.Sh. Combinatorial evolution of composite systems, In: 16th Eur. Meeting on Cybern. and Syst. Res. Vol. 1, Vienna, 2002, pp. 275–280.

общих ограничений и совместимости между выбираемыми операциями (задача блочного рюкзака или морфологическое проектирование).

Таблица 9.1. Исследования эволюции/развития систем

Направление исследований	Источники
1.Эволюция систем с длинным жизненным циклом (программные пакеты)	[150]
2.Законы эволюции программ	[325]
3.Инженерный анализ изобретений и эволюции	[195]
4.Методы многокритериального принятия решений	[77, 111]
5.Системы анализа технологических изменений для эволюции мобильной связи (1G, 2G, 3G, 4G)	[539]
6.Патерны технологических систем	[498]
7. Анализ траекторий технологических улучшений	[156]
8.Эмерджентная эволюция	[445]
9.Эмерджентный синтез	[560]
10.Инновации в модульных системах	[320]
11.Моделирование версий в проектировании больших интегральных схем	[289]
12.Моделирование версий при управлении конфигурациями программ	[142]
13.Модель для эволюционного проектирования (ESD)	[512]
14.Методология перепроектирования	[464, 465, 587]
15.Архитектурные инновации как реконфигурация существующих технологий	[249]
16.Основы модульной эволюции	[136, 407]
17.Эволюция платформ систем связи на основе трансформации алгебраических сетей (AHL)	[203]
18.Системы планирования развития на основе оптимизации	[481]
19.Базы инженерных историй	[551]



(a) поколение 1 (b) поколение 2 (c) поколение 3

Рис. 9.1. Пример эволюции системы

Наш подход был использован для комбинаторного моделирования эволюции систем в нескольких приложениях (Таблица 9.2): (1) СППР КОМБИ (DSS COMBI) для многокритериального ранжирования [28, 329, 336, 364]; (2) электронные устройства для цифровой обработки сигналов [344]; (3) стандарты для передачи мультимедийной информации (JPEG, MPEG-1, MPEG-2, MPEG-4) [390]; (4) коммуникационный протокол ZigBee для беспроводных сенсорных сетей [393].

В данной главе предложенный комбинаторный подход иллюстрируется на примере трех учебных курсов автора (как поколений систем) в области многокритериального принятия решений и проектирования модульных систем [366]. При этом представлено следующее: (а) иерархические структуры рассматриваемых курсов, (б) изменения между курсами, (в) модульное (морфологическое) проектирование нового перспективного курса (прогноз), (г) агрегация полученных прогнозов в результирующее агрегированное решение.

Процесс проектирования решения основан на ИММП с интервальными оценками в виде мультимножеств. На стадии агрегации решений используется задача блочного рюкзака

(с интервальными оценками в виде мультимножеств). Получение указанных интервальных оценок в в виде мультимножеств основано на многокритериальном анализе и экспертных суждениях.

Таблица 9.2. Моделирование эволюции систем

Прикладная система	Исследуемая задача			Источник
	Эволюция	Прогнозирование	Агрегация	
1.СППР КОМБИ	Да	Нет	Нет	[28, 329, 336, 364]
2.Электронное устройство	Да	Да	Нет	[344]
3.MPEG-подобный стандарт для передачи мультимедийных данных	Да	Да	Нет	[390]
4.Протокол связи ZigBee	Да	Да	Да	[393]

9.2 Комбинаторная эволюция

Общая схема комбинаторной эволюции и прогнозирования для модульных систем представлена на Рис. 9.2 [344, 357, 366]:

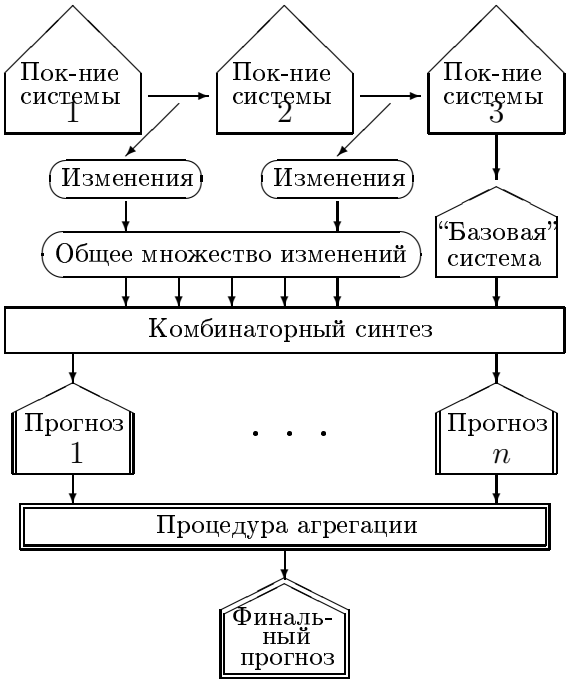


Рис. 9.2. Общая схема эволюции

Стадия 1. Генерация общей иерархической структуры для исследуемой системы (включая базовое множество альтернатив для компонентов системы - висячих вершин модели).

Стадия 2. Построение иерархической структуры для каждого поколения системы.

Стадия 3. Выделение системных изменений между соседними поколениями системы.

Стадия 4. Интеграция изменений системы для построения общего множество возможных изменений системы (включая использование экспертных процедур для добавление изменений системы).

Стадия 5. Анализ множества изменений системы: многокритериальное оценование, бинарные отношения над множеством (предшествование, дополнительность, совместимость).

Стадия 6. Комбинаторный синтез прогнозного варианта системы как выбор наиболее перспективных изменений системы с учетом следующего: (i) общее ресурсное ограничение(я), (ii) "полезность" для каждого изменение системы и соответствующая получаемая общая "полезность", (iii) совместимость между выбранными изменениями системы (в случае использования ИММП).

Стадия 7. Анализ полученного прогнозы(зов) и агрегация (в случае получения нескольких прогнозных вариантов) для формирования окончательного агрегированного решения.

9.3 Операции изменений

Приведем следующий набор типовых операций изменения для модульных систем [340, 344]:

I. Операции для DAs: (1.1.) изменение/улучшение DA O_1: $A_i \Rightarrow A'_i$, (1.2.) удаление DA O_2: A_i^-, (1.3.) добавление DA O_3: A_i^+, (1.4.) агрегация DAs O_4: $\{A_i\} \Rightarrow A^a = A_1 \& A_2 \& ....$

II. Операции для подсистем. (2.1.) изменение/улучшение части системы O_5, (2.2.) удаление части системы O_6, (2.3.) добавление части системы O_7, (2.4.) агрегация частей системы O_8.

III. Операции для структуры системы (O_9).

Таким образом, эволюция системы может быть рассмотрена как некое подмножество указанных операций. Для каждой операции можно исследовать набор характеристик (например, требуемые ресурсы, "полезность") или результирующая интервальная оценки в виде мультимножеств. Процесс трансформации системы может быть рассмотрен как оптимизацонная задача.

В качестве бинарных отношений над множеством операций изменений следующие три вида могут использоваться: (1) *эквивалентность* R^e; (2) *дополнительность* (*совместимость*) R^c; (3) *предшествование* R^p. Эти отношения могут использоваться как основа для структурных ограничений.

9.4 Пример для модульного курса

9.4.1 Иерархическая морфологическая модель

Общая иерархическая (древовидная) модель рассматриваемых курсов представлена на Рис. 9.3:

0. Общая схема курса $S = H \star B \star L$.

1. Методология $H = E \star G \star D \star I \star K$:

1.1. системный инжениринг (например, жизненный цикл продукции, проектирование на ранних стадиях, тестирование, техническое обслуживание, технология требований к системе) E,

1.2. схемы проектирования (например, схемы на основе принятия решений, иерархическое проектирование, морфологической анализ и его модификации, многодисциплинарная оптимизация, метод исследования пространства параметров, проектирования на основе моделей систем в виде грамматик, проектирование на основе аксиом, проектирование на основе имитационного моделирования и др.) G,

1.3. принятие решений (классификация задач принятия решений, общая схема, теория функций полезности, подход на основе правил Парето, методы порогов несравнимости, экспертные процедуры и др.) decision making D,

1.4. информационные системы (типы систем, распределенные системы, информационный поиск, индексация, близость информационных объектов и др.) I,

1.5. технология знаний (типы знаний, стадии обработки знаний, структуры знаний, операции над структурами знаний, экспертные суждения и др.) K.

2. Методы $B = F \star C \star Y \star A \star U$:

2.1. оптимизация (типы оптимизационных моделей, выпуклая оптимизация, нелинейная оптимизация, модели смешанного целочисленного оптимизации) F,

2.2. комбинаторная оптимизация (например, задача о рюкзаке, задача блочного рюкзака, задача о назначении/размещении, задача коммивояжера, задачи теории расписаний, раскраска графа, задачи о покрытии деревьями, задача о максимальной клике) C,

2.3. вопросы сложности (алгоритмическая сложность, примеры задач и их сложность) Y,

2.4. точные алгоритмы (например, метод ветвей и границ, алгоритмы на основе динамического программирования) A,

2.5. эвристики (например, жадные алгоритмы, локальная оптимизация, эволюционные вычисления, гибридные мета-эвристики) U.

3. Приложения $L = V \star Z \star R \star W \star J \star Q \star T$:

3.1. инвестиции (например, типы инвестиций, инвестиционный портфель) V,

3.2. задач управления персоналом (например, выбор персонала, назначение персонала на позиции) Z,

3.3. маркетинг (например, сценарии маркетинга, сегменты рынка, планирование, политический маркетинг) R,

3.4. программные комплексы (например, модульные пакеты программ, моделирование версий) W,

3.5. тестирование систем (например, непосредственное тестирование систем с известной структурой - glass-box testing, black-box testing, комбинаторное тестирование, планирование процессов тестирования) J,

3.4. сети (например, топология сетей, проектирование сетей, сетевые протоколы) Q,

3.5. телеметрия (например, сенсоры, телеметрические системы, сенсорные сети) T.

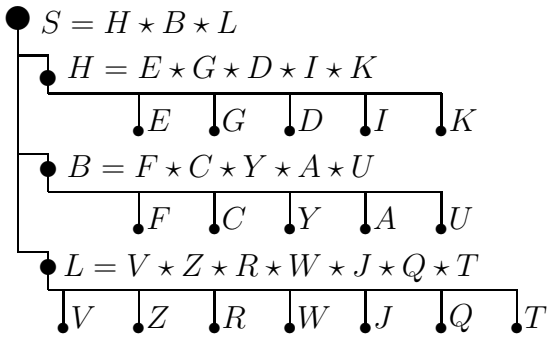


Рис. 9.3. Общая схема курсов

Для каждого компонента курса используются четыре типовых проектных альтернатив: отсутствие данного компонента X_0, краткое рассмотрение материала X_1, изучение на среднем уровне подробности X_2, подробное изучение материала X_3, подробное изучение материала, включая специальные индивидуальные работы/проекты (например, написание программ, разработка алгоритмов, исследование новых приложений, подготовка статей) X_4. Три упрощенных версии курсов рассматриваются:

(а) информационные технологии и принятие решений [20, 331, 332, 336]:

$S^1 = (E_0 \star G_1 \star D_3 \star I_2 \star K_1) \star (F_0 \star C_1 \star Y_0 \star A_0 \star U_1) \star (V_2 \star Z_2 \star R_1 \star W_1 \star J_0 \star Q_0 \star T_0)$;

(б) введение в системный инжениринг [336, 337, 344]:

$S^2 = (E_2 \star G_2 \star D_3 \star I_0 \star K_1) \star (F_1 \star C_1 \star Y_0 \star A_0 \star U_1) \star (V_0 \star Z_0 \star R_4 \star W_1 \star J_0 \star Q_0 \star T_1)$;

(в) проектирование систем [22, 336, 344, 345, 350, 353]:
$S^3 = (E_2 \star G_3 \star D_3 \star I_0 \star K_0) \star (F_2 \star C_3 \star Y_1 \star A_1 \star U_2) \star (V_0 \star Z_0 \star R_4 \star W_3 \star J_3 \star Q_4 \star T_4)$.

9.4.2 Постановка задачи, проектирование

Следует указать главные цели при построении нового курса:

(i) повышение уровня подготовки в области математических методов (часть системы B),

(ii) добавление реалистичных проектов студентов (использование альтернатив типа X_4),

(iii) приобретение некоторого первоначального опыта в организации start-up компаний (например, компоненты V, R).

Следующая схема решения используется:

1. Описание операций изменения между поколениями системы (т.е., курсами).

2. Построение системного "системного ядра" - "базовой" системы (здесь используется последнее поколение системы и экспертные суждения).

3. генерация интегрированного множества операций изменения.

4. Комбинаторный синтез для построения прогноза(ов) как расширение "базовой" системы.

5. Агрегация полученных решений (прогнозов) для получения агрегированного решения.

Рассматривается следующие изменения между поколениями системы:

(а) $S^1 \Rightarrow S^2$ (Таблица 9.3): $\overline{\Phi}^{1,2} = \{\Phi_1^{1,2}, ..., \Phi_7^{1,2}\}$;

(б) $S^2 \Rightarrow S^3$ (Таблица 9.4): $\overline{\Phi}^{2,3} = \{\Phi_1^{2,3}, ..., \Phi_{11}^{2,3}\}$.

Таблица 9.3. Изменения $S^1 \Rightarrow S^2$

	Изменение	Тип оператора
$\Phi_1^{1,2}$	$E_0 \to E_2$	добавление O_3
$\Phi_2^{1,2}$	$G_1 \to G_2$	улучшение O_1
$\Phi_3^{1,2}$	$I_2 \to I_0$	удаление O_2
$\Phi_4^{1,2}$	$F_0 \to F_1$	добавление O_3
$\Phi_5^{1,2}$	$V_2 \to V_0$	удаление O_2
$\Phi_6^{1,2}$	$Z_2 \to Z_0$	удаление O_2
$\Phi_7^{1,2}$	$R_1 \to R_4$	улучшение O_1

Таблица 9.4. Изменения $S^2 \Rightarrow S^3$

	Изменение	Тип оператора
$\Phi_1^{2,3}$	$G_2 \to G_3$	улучшение O_1
$\Phi_2^{2,3}$	$K_1 \to K_0$	удаление O_2
$\Phi_3^{2,3}$	$F_1 \to F_2$	улучшение O_1
$\Phi_4^{2,3}$	$C_1 \to C_3$	улучшение O_1
$\Phi_5^{2,3}$	$Y_0 \to Y_1$	добавление O_3
$\Phi_6^{2,3}$	$A_0 \to A_1$	добавление O_3
$\Phi_7^{2,3}$	$U_0 \to U_1$	улучшение O_1
$\Phi_8^{2,3}$	$W_1 \to W_3$	улучшение O_1
$\Phi_9^{2,3}$	$J_0 \to J_3$	добавление O_3
$\Phi_{10}^{2,3}$	$Q_0 \to Q_4$	добавление O_3
$\Phi_{11}^{2,3}$	$T_0 \to T_4$	добавление O_3

"Базовая" система может быть рассмотрена как модификация S^3 (экспертные суждения):

$S^k = (E_0 \star G_3 \star D_3 \star I_0 \star K_1) \star (F_2 \star C_3 \star Y_1 \star A_1 \star U_2) \star (V_1 \star Z_0 \star R_4 \star W_0 \star J_0 \star Q_3 \star T_0)$.

Для данной "базовой" системы можно рассмотреть дополнительное множество изменений (Таблица 9.5, экспертные суждения): $\overline{\Phi}^k = \{\Phi_1^k, ..., \Phi_9^k\}$. Очевидно, три пары операций являются альтернативными: (а) Φ_3^k и Φ_4^k, (б) Φ_5^k и Φ_6^k, (в) Φ_7^k и Φ_8^k.

Общее множество изменений основано на интеграции множеств $\overline{\Phi}^{1,2}$, $\overline{\Phi}^{2,3}$, $\overline{\Phi}^k$ как $\overline{\Phi}^g = \{\Phi_1^g, ..., \Phi_{12}^g\}$ (Таблица 9.6, экспертные суждения).

Таблица 9.5. Изменение “ядра” S^k

	Изменение	Тип оператора
Φ_1^k	$K_1 \to K_2$	улучшение O_1
Φ_2^k	$C_3 \to C_4$	улучшение O_1
Φ_3^k	$Y_1 \to Y_2$	улучшение O_1
Φ_4^k	$Y_1 \to Y_3$	улучшение O_1
Φ_5^k	$A_1 \to A_2$	улучшение O_1
Φ_6^k	$A_1 \to A_3$	улучшение O_1
Φ_7^k	$U_2 \to U_3$	улучшение O_1
Φ_8^k	$U_2 \to U_4$	улучшение O_1
Φ_9^k	$Q_3 \to Q_4$	улучшение O_1

Таблица 9.6. Общий список изменений

		Изменение	Тип операции
Φ_1^g	$\Phi_1^{1,2}$	$E_0 \to E_1$	добавление O_3
Φ_2^g	Φ_1^k	$K_1 \to K_2$	улучшение O_1
Φ_3^g	Φ_2^k	$C_3 \to C_4$	улучшение O_1
Φ_4^g	Φ_3^k	$Y_1 \to Y_2$	улучшение O_1
Φ_5^g	Φ_4^k	$Y_1 \to Y_3$	улучшение O_1
Φ_6^g	Φ_5^k	$A_1 \to Y_2$	улучшение O_1
Φ_7^g	Φ_6^k	$A_1 \to A_3$	улучшение O_3
Φ_8^g	Φ_7^k	$U_2 \to U_3$	улучшение O_1
Φ_9^g	Φ_8^k	$U_2 \to U_4$	улучшение O_1
Φ_{10}^g	$\Phi_9^{2,3}$	$J_0 \to J_3$	добавление O_3
Φ_{11}^g	Φ_7^k	$Q_3 \to Q_4$	улучшение O_3
Φ_{12}^g	$\Phi_4^{2,3}$	$T_0 \to T_4$	добавление O_3

Далее ИММП используется для построения перспективного курса (т.е., прогноз) $\widehat{S}^a = \widehat{H}^a \star \widehat{B}^a \star \widehat{L}^a$. Процесс проектирования реализуется как выбор и композиция операций изменения $\overline{\Phi}^g$. Проектируемый курс рассматривается как объединение двух частей: $\widehat{S}^a = \overline{S}^a \bigcup S^a$, где используются следующие части:

(i) фиксированная часть: $\overline{S}^a = \overline{H}^a \star \overline{B}^a \star \overline{L}^a$, где

$\overline{H}^a = G_3 \star D_3 \star I_0$, $\overline{B}^a = F_2$, $\overline{L}^a = V_1 \star Z_0 \star R_4 \star W_0$;

(ii) измененная часть: $S^a = H^a \star B^a \star L^a$, где базовая структура имеет вид: $H^a = E_0 \star K_1$, $B^a = C_3 \star Y_1 \star A_1 \star U_2$; $L^a = J_0 \star Q_3 \star T_0$.

Иерархическое проектирование направлено на изменение следующей части курса: $S^a = H^a \star B^a \star L^a$ (Рис. 9.4; интервальные оценки в виде мультимножеств для DAs указаны в круглых скобках, используются экспертные суждения).

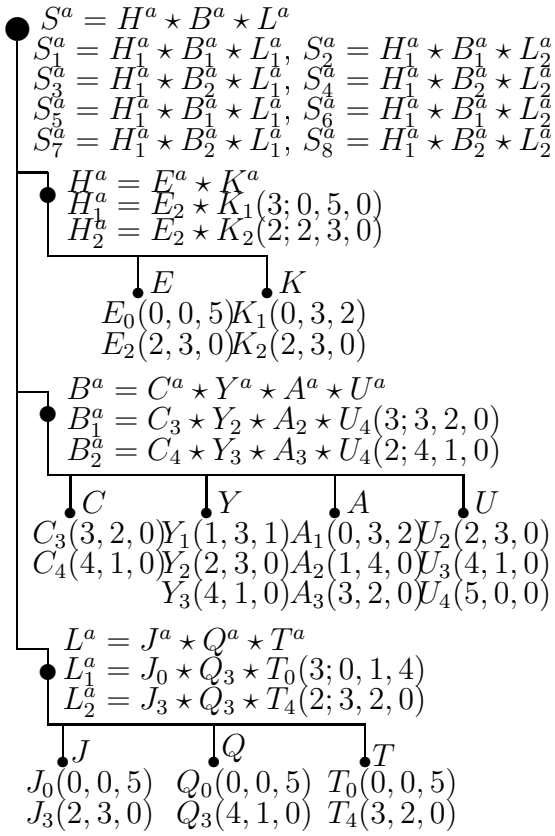


Рис. 9.4. Структура проектируемого курса

Оценки совместимости между DAs представлены в Таблице 9.7. (совместимость с элементами фиксированной части курса $\overline{S}^a$ не рассматривается для упрощения).

Полученные Парето-эффективные составные DAS для составных компонентов имеют вид:

(1) для H^a: $H_1^a = E_2 \star K_1$, $N(H_1^a) = (3; 0, 5, 0)$; $H_2^a = E_2 \star K_2$, $N(H_2^a) = (2; 2, 3, 0)$;

(2) для B^A: $B_1^a = C_3 \star Y_2 \star A_2 \star U_4$, $N(B_1^a) = (3; 3, 2, 0)$; $B_2^a = C_4 \star Y_3 \star A_3 \star U_4$, $N(B_2^a) = (2; 4, 1, 0)$;

(3) для L^a: $L_1^a = J_0 \star Q_3 \star T_0$, $N(L_1^a) = (3; 0, 1, 4)$; $L_2^a = J_3 \star Q_3 \star T_4$, $N(L_2^a) = (2; 3, 2, 0)$.

Таблица 9.7. Совместимость

	Y_1	Y_2	Y_3	A_1	A_2	A_3	U_2	U_3	U_4
C_3	3	3	3	3	3	3	2	3	3
C_4	2	3	3	3	2	2	2	3	3
Y_1				3	2	2	3	3	3
Y_2				2	3	2	3	3	3
Y_3				1	3	3	2	2	2
A_1							2	2	2
A_2							2	2	3
A_3							2	3	3

	Q_0	Q_3	T_0	T_4
J_0	3	3	3	1
J_3	1	2	1	3
Q_0			3	1
Q_3			3	2

	K_1	K_2
E_0	1	1
E_2	3	2

Рис. 9.5 иллюстрирует качество решений для L^a.

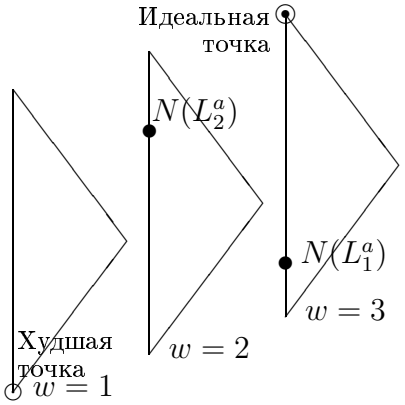


Рис. 9.5. Качество L^a

Для результирующей системы S^a получено восемь комбинаций ее частей (т.е., H^a, B^a, L^a):

$S_1^a = H_1^a \star B_1^a \star L_1^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_2^a = H_1^a \star B_1^a \star L_2^a = (E_2 \star K_1) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_3^a = H_1^a \star B_2^a \star L_1^a = (E_2 \star K_1) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_4^a = H_1^a \star B_2^a \star L_2^a = (E_2 \star K_1) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_5^a = H_2^a \star B_1^a \star L_1^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_6^a = H_2^a \star B_1^a \star L_2^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_7^a = H_2^a \star B_2^a \star L_1^a = (E_2 \star K_2) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_8^a = H_2^a \star B_2^a \star L_2^a = (E_2 \star K_2) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_3 \star Q_3 \star T_4)$.

9.4.3 Анализ и улучшение решений

Рассматривается подсистема $L^a = J^a \star Q^a \star T^a$ (Таблица 9.8). Предполагается следующий процесс улучшения для составного решения $L_2^a = J_3 \star Q_3 \star T_4$ (использованы бинарные переменные $\{y_{ij}\}$):

(1) две версии для элемента J_3: y_{11} (нет), y_{12} $((2,3,0) \Rightarrow (3,2,0))$, y_{13} $((2,3,0) \Rightarrow (4,1,0))$, y_{14} $((2,3,0) \Rightarrow (5,0,0))$;

(2) две версии для элемента Q_3: y_{21} (нет), y_{22} $((4,1,0) \Rightarrow (5,0,0))$;

(3) пять версия для элемента A_3: y_{31} (нет), y_{32} $((0,2,1,0) \Rightarrow (0,3,0,0))$, y_{33} $((0,2,1,0) \Rightarrow (1,2,0,0))$, y_{34} $((0,2,1,0) \Rightarrow (2,1,0,0))$, y_{35} $((0,2,1,0) \Rightarrow (3,0,0,0))$.

Таблица 9.9 содержит бинарные переменные (y_{ij}), операции улучшения (по компонентам) и их оценки (данные оценки имеют иллюстративный характер).

Таблица 9.8. Узкие места, улучшения

Составные DAs	Узкие места DA/IC	Операции улучшения w/e
L_2	J_3	$(2,3,0) \Rightarrow (3,2,0)$
L_2	J_3	$(2,3,0) \Rightarrow (4,1,0)$
L_2	J_3	$(2,3,0) \Rightarrow (5,0,0)$
L_2	Q_3	$(4,1,0) \Rightarrow (5,0,0)$
L_2	(J_3, Q_3)	$2 \Rightarrow 3$
L_2	(Q_3, T_4)	$2 \Rightarrow 3$
L_2	T_4	$(3,2,0) \Rightarrow (4,1,0)$
L_2	T_4	$(3,2,0) \Rightarrow (5,0,0)$

Таблица 9.9. Улучшения L_2

Операции улучшения	Интервальная оценка e_{ij}	Стоимость (a_{ij})
y_{11} (J_3, нет)	$(2,3,0)$	0
y_{12} (J_3, улучшение 1)	$(3,2,0)$	17
y_{13} (J_3, улучшение 2)	$(4,1,0)$	19
y_{14} (J_3, улучшение 3)	$(5,0,0)$	25
y_{21} (Q_3, нет)	$(4,1,0)$	0
y_{22} (Q_3, улучшение 1)	$(5,0,0)$	28
y_{31} (T_4, нет)	$(3,2,0)$	0
y_{32} (T_4, улучшение 1)	$(4,1,0)$	20
y_{33} (T_4, улучшение 2)	$(5,0,0)$	30

Оптимизационная задача улучшения имеет вид:

$$\arg\min_{M \in \overline{D}} \sum_{(i,j) \in S = \{(i,j) | y_{ij}=1\}} |\delta(M, e_{ij})| \quad s.t. \ \sum_{i=1}^{3} \sum_{j=1}^{q_i} a_{ij} y_{ij} \le b, \ \sum_{j=1}^{q_j} y_{ij} = 1, \ y_{ij} \in \{0,1\}.$$

Можно привести некоторые примеры решений:

(1) $b = 39$: $y_{13} = 1$ (J_3^2, улучшение 2), $y_{21} = 1$ (Q_3, нет), $y_{32} = 1$ (T_4^1, улучшение 1);

$L_2 \Rightarrow \widetilde{L_2^1} = J_3^2 \star Q_3 \star T_4^1$, $e(\widetilde{L_2^1}) = (4,1,0)$;

(2) $b = 73$: $y_{14} = 1$ (J_3^3, улучшение 3), $y_{22} = 1$ (Q_3^1, улучшение 1), $y_{33} = 1$ (T_4^2, улучшение 2);

$L_2 \Rightarrow \widetilde{L_2^2} = J_3^3 \star Q_3^1 \star T_4^2$, $e(\widetilde{L_2^2}) = (5,0,0)$.

9.4.4 Агрегация решений

Для полученных восьми решений используется упрощенный подход к агрегации на основе задачи блочного рюкзака (расширение “системного ядра”):

$S_1^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_2^a = (E_2 \star K_1) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_3^a = (E_2 \star K_1) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_4^a = (E_2 \star K_1) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_5^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_6^a = (E_2 \star K_2) \star (C_3 \star Y_2 \star A_2 \star U_4) \star (J_3 \star Q_3 \star T_4)$;
$S_7^a = (E_2 \star K_2) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_0 \star Q_3 \star T_0)$;
$S_8^a = (E_2 \star K_2) \star (C_4 \star Y_3 \star A_3 \star U_4) \star (J_3 \star Q_3 \star T_4)$.

Подрешение и надрешение представлены на Рис. 9.6 и Рис. 9.7.

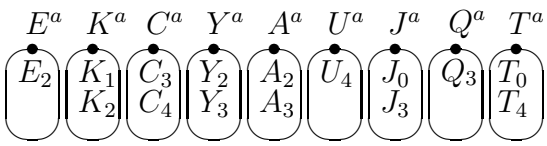


Рис. 9.6. Надрешение

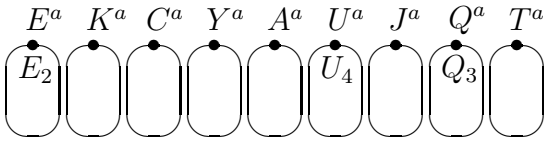


Рис. 9.7. Подрешение

Полученное подрешение содержит три элемента (эта комбинация будет использоваться как "системное ядро"). Процесс агрегации рассматривается на основе задачи блочного рюкзака как выбор DAs для подсистемы $\Theta = K \star C \star Y \star A \star J \star T$ (Рис. 9.8) (без учета совместимости). Используются следующие бинарные переменные: $\{x_{ij}\},\ i = \overline{1,6},\ j = \overline{1,2}$. Оптимизационная задача имеет вид:

$$\arg\min_{M \in \overline{D}} \sum_{(i,j) \in S = \{(i,j) | x_{ij} = 1\}} |\delta(M, e_{ij})| \quad s.t.\ \sum_{i=1}^{6} \sum_{j=1}^{2} a_{ij} x_{ij} \le b;\ \sum_{j=1}^{2} x_{ij} = 1;\ x_{ij} \in \{0,1\}.$$

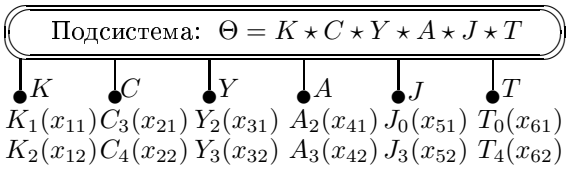


Рис. 9.8. Выбор DAs для подсистемы

Таблица 9.10 содержит иллюстративные оценки. Можно привести некоторые примеры результирующих решений:

(1) $b = 42$: $x_{11} = 1$ (K_1), $x_{21} = 1$ (C_3), $x_{31} = 1$ (Y_2), $x_{41} = 1$ (A_2), $x_{51} = 1$ (J_0), $x_{61} = 1$ (T_0), $\Theta_1 = K_1 \star C_3 \star Y_2 \star A_2 \star J_0 \star T_0$, $e(\Theta_1) = (1, 4, 0)$;

(2) $b = 101$: $x_{12} = 1$ (K_2), $x_{22} = 1$ (C_4), $x_{32} = 1$ (Y_3), $x_{42} = 1$ (A_3), $x_{52} = 1$ (J_3), $x_{62} = 1$ (T_4), $\Theta_2 = K_2 \star C_4 \star Y_3 \star A_3 \star J_3 \star T_4$, $e(\Theta_2) = (3, 2, 0)$.

Таблица 9.10. Оценки для агрегации

Выбор DA	Интервальная оценка e_{ij}	Стоимость (a_{ij})
$x_{11}\,(K_1)$	$(0, 3, 2)$	6
$x_{12}\,(K_2)$	$(2, 3, 0)$	12
$x_{21}\,(C_3)$	$(3, 2, 0)$	15
$x_{22}\,(C_4)$	$(4, 1, 0)$	20
$x_{31}\,(Y_2)$	$(2, 3, 0)$	11
$x_{32}\,(Y_3)$	$(4, 1, 0)$	22
$x_{41}\,(A_2)$	$(1, 4, 0)$	10
$x_{42}\,(A_3)$	$(3, 2, 0)$	16
$x_{51}\,(J_0)$	$(0, 0, 5)$	0
$x_{52}\,(J_3)$	$(2, 3, 0)$	13
$x_{61}\,(T_0)$	$(0, 0, 5)$	0
$x_{62}\,(T_4)$	$(3, 2, 0)$	18

9.5 Выводы к главе

В данной главе представлены общая схема для комбинаторного моделирования эволюции и прогнозирования модульной системы, включая иллюстративный пример для модульных учебных курсов. Представляется перспективным рассмотреть в будущем следующее:

1. исследование комбинаторной эволюции и прогнозирования различных прикладных систем;

2. использование различных подходов к учету неопределенности.

Глава 10

Приложения в программных системах

В данной главе [1] приведено описание четырех задач: (а) иерархическая композиция модульного пакета программ (иерархическая модель, проектирование, анализ результатов, улучшение), (б) иерархическое проектирование человеко-машинного интерфейса на примере интерфейса для системы СППР-КОМБИ (или DSS COMBI) [26, 27, 28, 29, 329, 330], (в) иллюстрация комбинаторной эволюции системы СППР-КОМБИ, (г) построение последовательно-параллельных стратегий для многокритериального ранжирования.

10.1 Модульный программный пакет

Рассматривается пример интеграции программного пакета из типовых модулей (Рис. 10.1):

0. Система $S = A \star B$.
1. Подсистема $A = M \star P$:
1.1. Система управления базой данных (Data Base Management System DBMS) (M).
1.2. Интегрированный пакет (Integrated Software IP) (P).
2. Подсистема $B = D \star H \star L$:
2.1. Система поддержки принятия решений СППР (Decision Support System DSS) (D)
2.2. Гипертекстовая система (Hypertext System HS) (H)
2.3. Интерфейсный пакет для системы искусственного интеллекта (Language for Artificial Intelligence LAI) (L).

Следует отметить, что последние годы подобные задачи становятся все более актуальными [461].

Исходные данные (характеристики/критерии, значения характеристик) для указанных программных пакетов были взяты из соответствующих профессиональных журналов: 'PC World', 'Microcomputers', 'InfoWorld', 'Expert Systems Users' и др. Рассматриваются следующие списки критериев:

1. DBMSs: (1) средства для развития (0 соответствует 'Нет', 1 соответствует 'Есть, Хорошо', 2 соответствует 'Превосходно'; далее используется аналогичная порядковая шкала);

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Modular system synthesis: Example for composite packaged software. IEEE Trans. on SMC, Part C, Vol. 35, No. 4, 2005, pp. 544-553.
(ii) Levin M.Sh. Hierarchical design of user interfaces. In: Human Computer Interaction. LNCS, Vol.876, Berlin: Springer, 1994, pp. 140-151.
(iii) Levin M.Sh., Combinatorial Engineering of Decomposable Systems. Springer, 1998. chapter 11.
(iv) Levin M.Sh., Composite Systems Decisions. Springer, 2006. chapter 7.
(v) M.Sh. Levin, Composite strategy for multicriteria ranking/sorting (methodological issues, examples). Electronic preprint. 24 pp., Nov. 9, 2012. http://arxiv.org/abs/1211.2245 [math.OC]

(2) скорость работы $(0, 1, 2)$; (3) подготовка отчетов $(0, 1, 2)$; (4) адаптивность чел.-маш. интерфейса $(0, 1)$; (5) стоимость (USD); (6) среда построения запросов $(0, 1, 2)$.

2. IPs: (1) графика $(0, 1, 2)$; (2) простота использования $(0, 1, 2)$; (3) база данных $(0, 1)$; (4) функция телекоммуникации $(0, 1, 2)$; (5) текстовый процессор $(0, 1, 2)$; (6) адаптивность чел.-маш. интерфейса $(0, 1)$; (7) стоимость (USD).

3. DSSs: (1) чел.-маш. интерфейс $(1, 2)$; (2) поддержка диалога $(0, 1, 2)$; (3) обучающая подсистема $(0, 1)$; (4) библиотека моделей $(0, 1, 2)$; (5) компоненты искусственного интеллекта $(0, 1)$; (6) наличие предыдущего опыта $(0, 1)$; (7) стоимость (USD).

4. HSs: (1) режим браузинга $(0, 1)$; (2) графика $(0, 1)$; (3) режим много-оконного интерфейса $(0, 1)$; (4) компоненты искусственного интеллекта $(0, 1)$; (5) стоимость (USD); (6) наличие предыдущего опыта $(0, 1)$; (7) уровень пользователя $(0, 1, 2)$.

5. LAIs: (1) стоимость (USD); (2) наличие предыдущего опыта $(0, 1, 2)$; (3) уровень пользователя $(0, 1)$; (4) качество сервиса $(0, 1, 2)$.

Таблица 10.1 содержит агрегативные критерии для подсистем (первый индекс соответствует компоненту, негативная монотонность указана в скобках).

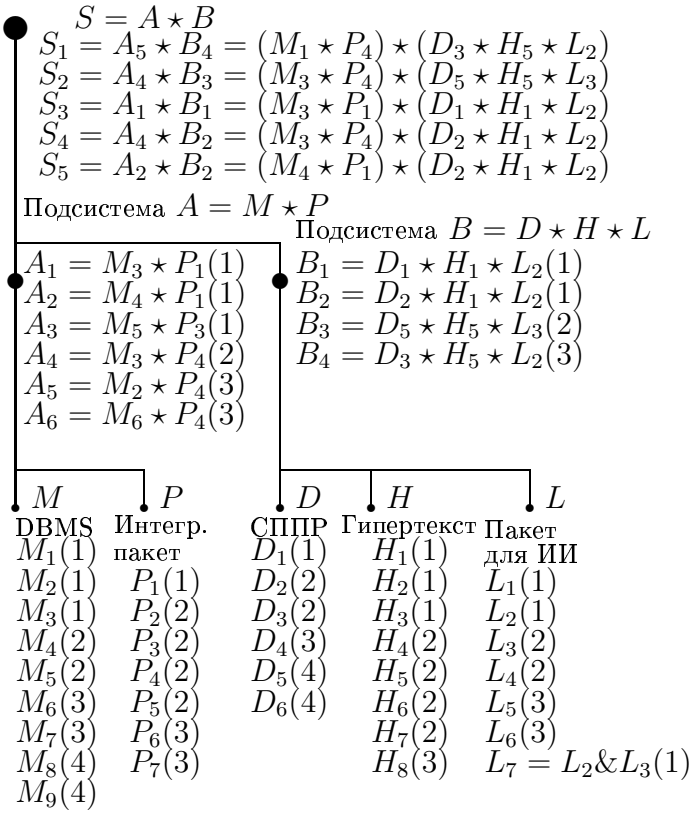


Рис. 10.1. Структура системы (приоритеты для DAs указаны в скобках)

Для висячих вершин модели системы выполняются следующие операции: (1) генерация DAs; (2) оценивание DAs по критериям (получение оценок); (3) ранжирование DAs на основе оценок. DAs и их оценки представлены в Таблицах 10.2, 10.3. Рис. 10.1 содержит вычисленные приоритеты для DAs.

Теперь можно рассмотреть следующий иерархический уровень модели системы: (1) задание совместимости между DAs; (2) композиция составных DAs; (3) оценивание DAs по критериям; (4) ранжирование DAs. Оценки совместимости DAs представлены в Таблице 10.4. DAs для подсистем A и B содержаться в Таблице 10.5. Рис. 10.2 иллюстрирует концентрическое представление DAs для B_1 и B_3.

Таблица 10.1. Агрегативные критерии

Критерии	Спецификации
F_{a1} Стоимость (-1)	$F_{m5} + F_{p7}$
F_{a2} Графика	$\min(F_{m3}, F_{p1})$
F_{a3} Простота использования	$\max(F_{m1}, F_{p2})$
F_{a4} Телекоммуникации	$\min(F_{m2}, F_{p4})$
F_{a5} Адаптивность чел.-маш. интерфейса	$\min(F_{m4}, F_{p6})$
F_{a6} Текстовый процессор	$\max(F_{m3}, F_{p5})$
F_{a7} Подготовка отчетов	$\max(F_{m3}, F_{p2})$
F_{a8} Скорость	$\max(F_{m2}, F_{p3})$
F_{a9} Среда построения запросов	F_{m6}
F_{b1} Стоимость (-1)	$F_{d7} + F_{h5} + F_{l1}$
F_{b2} Уровень пользователя	$\min(F_{h7}, F_{l3})$
F_{b3} Компоненты ИИ	$\max(F_{d5}, F_{h4})$
F_{b4} Опыт использования	$F_{d6} + F_{h6} + F_{l2}$
F_{b5} Поддержка диалога	F_{d2}

Таблица 10.2. Оценки для DAs

DAs	Критерии					
	1	2	3	4	5	6
M_1	2	2	2	2	482	3
M_2	2	2	2	2	490	3
M_3	1	1	1	1	486	2
M_4	1	1	1	1	237	2
M_5	1	1	1	1	502	3
M_6	1	1	1	1	451	0
M_7	1	1	1	1	333	0
M_8	0	0	0	0	240	0
M_9	0	0	0	0	180	0

DAs	Критерии						
	1	2	3	4	5	6	7
P_1	2	2	1	3	1	1	506
P_2	2	2	1	0	2	0	94
P_3	2	2	3	1	2	0	438
P_4	1	1	1	1	1	1	477
P_5	2	0	1	1	1	1	510
P_6	1	2	0	1	0	0	130
P_7	0	2	1	0	0	0	87

DAs	Критерии			
	1	2	3	4
L_1	1700	1	2	2
L_2	3000	1	2	2
L_3	2500	2	1	2
L_4	5000	0	1	2
L_5	2850	1	1	1
L_6	13570	0	1	2
$L_7 = L_2 \& L_3$	5500	2	2	2

Таблица 10.3. Оценки для DAs

DAs	Критерии						
	1	2	3	4	5	6	7
D_1	1	1	1	2	0	1	100
D_2	1	1	1	1	0	1	495
D_3	2	1	0	1	1	0	299
D_4	1	1	0	2	0	0	500
D_5	1	0	0	1	0	0	29
D_6	1	0	1	1	0	0	400

DAs	Критерии						
	1	2	3	4	5	6	7
H_1	1	1	1	1	350	0	2
H_2	1	1	1	1	600	1	2
H_3	1	1	1	1	300	1	1
H_4	1	1	1	0	500	1	0
H_5	1	0	1	1	700	0	1
H_6	0	0	1	0	100	0	1
H_7	0	0	0	0	100	1	1
H_8	1	1	1	1	470	1	1

Оценки совместимости между DAs для подсистем представлены в Таблице 10.6. Таблица 10.7 содержит составные DAs для результирующей системы. Это позволяет оценить составные DAs для результирующей системы по критериям и сравнить. Далее можно анализировать узкие места и акции улучшения (Таблица 10.8). В рассматриваемом примере все акции улучшения соответствуют улучшению Парето-эффективных решений. Улучшение совместимости (A_1, B_1) или (A_2, B_2) с 2 до 5 позволяет получить следующие два *идеальных* решения: $A_1 \star B_1$ и $A_2 \star B_2$. Рис. 10.3 иллюстрирует решетку качества системы и Парето-эффективные решения.

Таблица 10.4. Совместимость

	P_1	P_2	P_3	P_4	P_5	P_6	P_7
M_1	0	0	2	3	0	0	0
M_2	0	0	1	1	0	0	0
M_3	2	2	1	3	0	0	0
M_4	3	0	0	0	0	0	0
M_5	0	0	4	0	0	0	0
M_6	0	0	0	5	0	0	0
M_7	0	0	0	0	0	0	0
M_8	0	0	0	0	0	0	0
M_9	0	2	1	0	0	0	0

	D_1	D_2	D_3	D_4	D_5	D_6	L_1	L_2	L_3	L_4	L_5	L_6	L_7
H_1	1	3	0	0	0	0	0	3	0	0	0	0	0
H_2	0	0	0	0	0	0	0	0	0	0	0	0	0
H_3	0	0	0	0	0	0	1	1	1	2	2	2	1
H_4	0	0	0	0	0	0	0	0	0	1	1	1	0
H_5	1	0	4	0	5	0	0	4	5	1	1	1	4
H_6	0	0	0	0	0	0	0	0	0	1	1	1	0
H_7	0	0	0	0	0	0	0	0	0	1	1	1	0
H_8	0	0	0	0	0	1	0	0	0	0	0	0	0
D_1							1	1	1	2	2	2	1
D_2							0	3	0	0	0	0	0
D_3							0	4	0	0	0	0	0
D_4							0	0	0	0	0	0	0
D_5							0	0	5	0	0	0	0
D_6							0	0	0	1	1	1	0

Таблица 10.5. DAs для подсистем

DAs	Критерии									N
	1	2	3	4	5	6	7	8	9	
$A_1 = M_3 \star P_1$	992	2	2	3	1	1	4	3	2	$2;2,0,0,0$
$A_2 = M_4 \star P_1$	743	2	2	3	0	1	2	3	2	$3;1,1,0,0$
$A_3 = M_5 \star P_3$	940	2	2	1	0	2	3	3	3	$4;0,2,0,0$
$A_4 = M_3 \star P_4$	963	1	1	1	1	1	4	3	2	$3;1,1,0,0$
$A_5 = M_1 \star P_4$	959	1	1	1	1	1	4	2	3	$3;1,1,0,0$
$A_6 = M_6 \star P_4$	928	2	1	1	0	1	1	3	0	$5;0,1,1,0$
$B_1 = D_1 \star H_1 \star L_2$	3450	2	1	2	1					$1;3,0,0,0$
$B_2 = D_2 \star H_1 \star L_2$	3845	2	1	2	1					$3;2,1,0,0$
$B_3 = D_5 \star H_5 \star L_3$	3229	1	1	2	0					$5;0,2,0,1$
$B_4 = D_3 \star H_5 \star L_2$	3999	1	1	1	1					$4;1,2,0,0$

Обычно избыточность системных компонентов используется для повышения надежности системы. В случае программного обеспечения система с избыточностью содержит альтернативные модули. Рассмотрим пример для подсистемы B со следующей агрегативной альтернативой (избыточность на основе двух элементов) $L_7 = L_2 \& L_3$. Оценки для агрегативной альтернативы основываются на следующих 'агрегирующих' функциях на оценках элементов (Таблица 10.1): (1) сумма, (2) максимум, (3) минимум. Совместимость для агрегированной альтернативы L_7 будет равна минимуму соответствующих оценок включенных в нее (Таблица 10.4). Приоритет для L_7 будет равен 1. В результате получаем составное решений: $D_1 \star H_5 \star L_7$ with $N = (1;1,2,0,0)$. Это решение не является Парето-эффективным, но может рассматриваться как некая точка для дальнейшего улучшения (Таблица 10.8). Рис. 10.4 иллюстрирует составное решение с избыточностью.

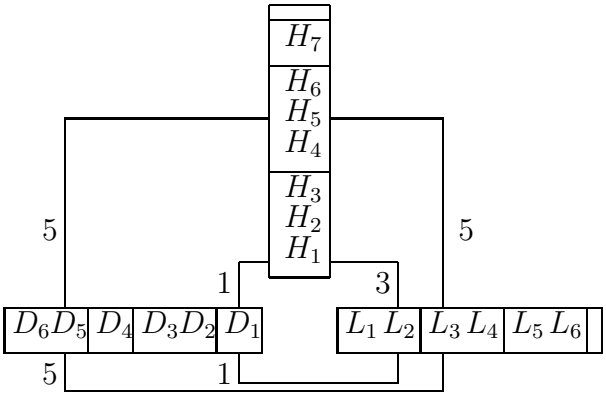


Рис. 10.2. Концентрическое представление DAs

Таблица 10.6. Совместимость

	A_1	A_2	A_3	A_4	A_5	A_6
B_1	2	1	0	0	1	0
B_2	1	2	0	3	0	0
B_3	0	0	0	4	0	0
B_4	0	0	0	0	5	0

Таблица 10.7. DAs для системы

Составные DAs	N
$S_1 = A_5 \star B_4 = (M_1 \star P_4) \star (D_3 \star H_5 \star L_2)$	$5; 0, 0, 2, 0$
$S_2 = A_4 \star B_3 = (M_3 \star P_4) \star (D_5 \star H_5 \star L_3)$	$4; 0, 2, 0, 0$
$S_3 = A_1 \star B_1 = (M_3 \star P_1) \star (D_1 \star H_1 \star L_2)$	$2; 2, 0, 0, 0$
$S_4 = A_4 \star B_2 = (M_3 \star P_4) \star (D_2 \star H_1 \star L_2)$	$3; 1, 1, 0, 0$
$S_5 = A_2 \star B_2 = (M_4 \star P_1) \star (D_2 \star H_1 \star L_2)$	$2; 2, 0, 0, 0$

Table 10.8. Узкие места и акции улучшения

Составные DAs	Узкие места		Акции w/r
	DAs	IC	
$A_1 = M_3 \star P_1$		(M_3, P_1)	$2 \Rightarrow 3$
$A_2 = M_4 \star P_1$	M_4		$2 \Rightarrow 1$
$A_2 = M_4 \star P_1$		(M_4, P_1)	$3 \Rightarrow 4$
$A_3 = M_5 \star P_3$		(M_3, P_4)	$2 \Rightarrow 3$
$A_3 = M_5 \star P_3$		M_5	$2 \Rightarrow 1$
$A_3 = M_5 \star P_3$	P_3		$2 \Rightarrow 1$
$A_4 = M_3 \star P_4$	P_4		$2 \Rightarrow 1$
$A_4 = M_3 \star P_4$		(M_3, P_4)	$3 \Rightarrow 4$
$A_5 = M_1 \star P_4$	P_4		$2 \Rightarrow 1$
$A_5 = M_1 \star P_4$		(M_1, P_4)	$3 \Rightarrow 4$
$A_6 = M_6 \star P_4$	P_4		$2 \Rightarrow 1$
$A_6 = M_6 \star P_4$	M_6		$3 \Rightarrow 2$
$B_2 = D_2 \star H_1 \star L_1$	D_2		$2 \Rightarrow 1$
$B_3 = D_5 \star H_5 \star L_3$	H_5		$2 \Rightarrow 1$
$B_3 = D_5 \star H_5 \star L_3$	L_3		$2 \Rightarrow 1$
$B_4 = D_3 \star H_5 \star L_2$	D_3		$2 \Rightarrow 1$
$B_4 = D_3 \star H_5 \star L_2$	H_5		$2 \Rightarrow 1$
$S_1 = A_5 \star B_4$	A_5		$3 \Rightarrow 2$
$S_1 = A_5 \star B_4$	B_4		$3 \Rightarrow 3$
$S_2 = A_4 \star B_3$	A_4		$2 \Rightarrow 1$
$S_2 = A_4 \star B_3$	B_3		$2 \Rightarrow 1$
$S_2 = A_4 \star B_3$		(A_4, B_1)	$4 \Rightarrow 5$
$S_3 = A_1 \star B_1$		(A_1, B_1)	$2 \Rightarrow 3$
$S_4 = A_4 \star B_2$	A_4		$2 \Rightarrow 1$
$S_4 = A_4 \star B_2$		(A_4, B_2)	$3 \Rightarrow 4$
$S_5 = A_2 \star B_2$		(A_4, B_2)	$2 \Rightarrow 3$
$D_1 \star H_5 \star L_7$	H_5		$2 \Rightarrow 1$

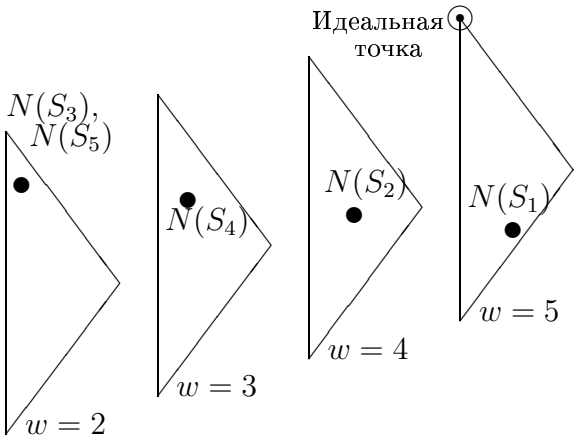


Рис. 10.3. Решетка качества и Парето-эффективные точки

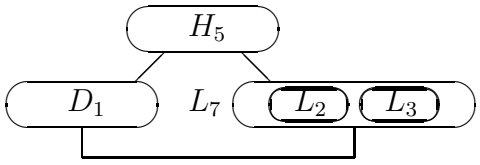


Рис. 10.4. Решение с избыточностью

Рассмотрим 3-стадийное модульное проектирование для подсистемы B. Предполагаем, что оценки совместимости между DAs улучшаются на каждой стадии. Могут использоваться два метода:

1. Агрегативный метод: Агрегация оценок совместимости и поиск составного решения на основе усреднения оценок.

2. Траекторный метод: Двух-уровневый процесс решения, состоящий из двух фаз:

2.1 Решение задачи проектирования (синтез составной подсистемы B) на каждой стадии.

2.2 Композиция составных решений, полученных на предыдущей фазе (для каждой стадии) в последовательность составных решений (траекторию).

Рассмотрим следующее улучшение совместимости:

(H_1, D_1): 2 (стадия 2), агрегированная оценка совместимости равна 2;
(H_1, D_2): 4 (стадия 2), 5 (стадия 3), агрегированная оценка совместимости равна 2;
(H_2, D_1): 3 (стадия 2), 5 (стадия 3), агрегированная оценка совместимости равна 3;
(H_5, D_3): 5 (стадия 2), агрегированная оценка совместимости равна 5;
(H_5, D_5): 5 (стадия 2), агрегированная оценка совместимости равна 5;
(H_1, L_2): 4 (стадия 2), 5 (стадия 3), агрегированная оценка совместимости равна 4;
(H_2, L_2): 2 (стадия 2), агрегированная оценка совместимости равна 2;
(H_5, L_2): 5 (стадия 2), агрегированная оценка совместимости равна 5;
(D_1, L_2): 2 (стадия 2), 3 (стадия 3), агрегированная оценка совместимости равна 2;
(D_2, L_2): 4 (стадия 2), 5 (стадия 3) агрегированная оценка совместимости равна 4;
(D_2, L_3): 3 (стадия 2), 4 (стадия 3), агрегированная оценка совместимости равна 3;
(D_3, L_2): 5 (стадия 2), агрегированная оценка совместимости равна 5;
(D_5, L_3): 5 (стадия 2), агрегированная оценка совместимости равна 5;

Рассмотрим агрегативный метод. Решаем задачу синтеза для случая агрегированных оценок совместимости. Получаем результирующие составные DAs:

(a) $B_1^a = D_3 \star H_5 \star L_2, \quad N(B_1^a) = (5; 1, 2, 0, 0)$;
(b) $B_2^a = D_1 \star H_1 \star L_2, \quad N(B_2^a) = (2; 3, 0, 0, 0)$;
(c) $B_3^a = D_1 \star H_2 \star L_2, \quad N(B_3^a) = (2; 3, 0, 0, 0)$.

Таким образом улучшение совместимости приводит к новому Парето-эффективному составному решению B_3^a.

Теперь рассмотрим траекторный метод. На первой стадии можно использовать Парето-эффективные составные DAs для подсистемы B из Таблицы 10.5:

$B_1^1 = D_1 \star H_1 \star L_2, \;\; N(B_1^1) = (1; 3, 0, 0, 0)$;
$B_2^1 = D_2 \star H_1 \star L_2, \;\; N(B_2^1) = (3; 2, 1, 0, 0)$;
$B_3^1 = D_5 \star H_5 \star L_3, \;\; N(B_3^1) = (5; 0, 2, 0, 1)$;
$B_4^1 = D_3 \star H_5 \star L_2, \;\; N(B_4^1) = (4; 1, 2, 0, 0)$.

На второй стадии составные DAs следующие:

(a) Парето-эффективные решения:
$B_1^2 = D_1 \star H_1 \star L_2, \;\; N(B_1^2) = (2; 3, 0, 0, 0)$;
$B_2^2 = D_2 \star H_1 \star L_2, \;\; N(B_2^2) = (4; 2, 1, 0, 0)$;
$B_3^2 = D_3 \star H_5 \star L_2, \;\; N(B_3^2) = (5; 1, 2, 0, 0)$;

(b) DAs, которые близки к Парето-эффективным решениям (эти решения доминируются указанными выше Парето-эффективными DAs):
$B_4^2 = D_5 \star H_5 \star L_3, \;\; N(B_2^2) = (5; 0, 2, 0, 1)$.

Составные DAs для стадии 3 следующие:

(a) Парето-эффективные решения:
$B_1^3 = D_1 \star H_1 \star L_2, \;\; N(B_1^3) = (2; 3, 0, 0, 0)$;
$B_2^3 = D_2 \star H_1 \star L_2, \;\; N(B_2^3) = (5; 2, 1, 0, 0)$;
$B_3^3 = D_1 \star H_2 \star L_2, \;\; N(B_3^3) = (2; 3, 0, 0, 0)$;

(b) DAs, которые близки к Парето-эффективным решениям (эти решения доминируются указанными выше Парето-эффективными DAs):
$B_4^3 = D_3 \star H_5 \star L_2, \;\; N(B_4^3) = (5; 1, 2, 0, 0)$;

$B_5^2 = D_5 \star H_5 \star L_3, \;\; N(B_2^2) = (5; 0, 2, 0, 1).$

Отметим, что DAs, близкие к Парето-эффективным DAs, могут быть полезны со следующей точки зрения. Изменение компонентов в траектории может потребовать некоторых усилий при этом необходимо решать дополнительную задачу верхнего уровня:

Скомбинировать траекторию с учетом качества составных DAs на каждой стадии и стоимости изменения компонентов (например, переобучение специалистов, совершенствование форматов файлов).

Получены следующие траектории (без изменений):

$\alpha_1 =< B_1^1, B_1^2, B_1^3 >, \alpha_2 =< B_2^1, B_2^2, B_2^3 >$.

Пример траектории с изменением компонентов следующий:

$\alpha' =< B_1^1, B_1^2, B_3^3 >$ с изменением $H_1 \to H_2$.

Другая стратегия может основываться на улучшении компонента. например, улучшение H_5 на стадии 3 приведет к траектории:

$\alpha'' =< B_3^1, B_3^2, B_4^3 >$.

Рис. 10.5 иллюстрирует метод траекторий.

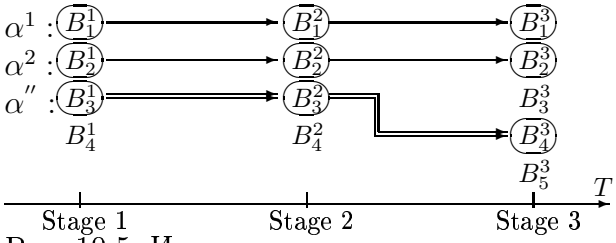


Рис. 10.5. Иллюстрация траекторного метода

10.2 Модульный пользовательский интерфейс

В данном разделе рассматривается иерархическое проектирование (ИММП) для пользовательского интерфейса [330, 336]. Пример базируется на разработанной автором с А.А. Михайловым (1984-1991 гг..) СППР КОМБИ (DSS COMBI) [28, 329, 330, 336, 364].

Рассматривается только базовый поток данных (обработки данных) в системе СППР КОМБИ, когда в качестве базового элемента обработки рассматривается отношение предпочтения и процесс решения моделируется как последовательная трансформация данных: (1) альтернативы, критерии, многокритериальные оценки альтернатив по критериям (R_0); (2) отношения предпочтения на альтернативах (R_1); (3) промежуточное линейное упорядочение альтернатив (R_2); (4) групповое ранжирование ($\widehat{R_2}$); (5) промежуточное групповое упорядочение альтернатив (R_3); (6) результирующие данные (ранжирование альтернатив (R_4).

Набор функциональных операций реализует указанную обработку:

$R_0 \Rightarrow R_1 \Rightarrow R_2 \Rightarrow \{R_3\} \Rightarrow R_4$.

На Рис. 10.6 представлена структура интерфейса СППР КОМБИ.

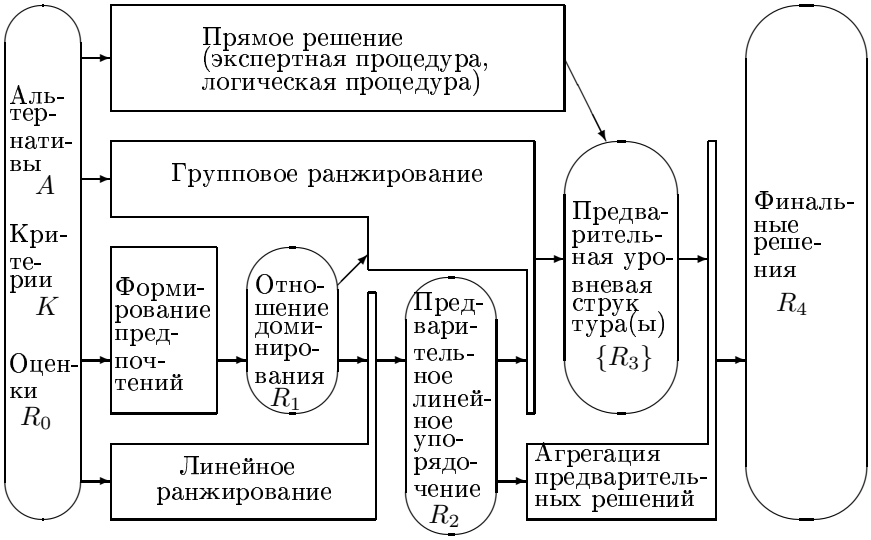


Рис. 10.6. Структура функционального графа-меню [28, 336]

Рассматриваются следующие операции редактирования данных: (a) ввод, (b) корректировка, (c) представление, (d) вывод, (e) импорт/экспорт.

В СППР КОМБИ реализованы следующие режимы: (a) выбор или создание новой прикладной задачи; (b) решение задачи (многокритериальное ранжирование) на основе различных методов; (c) анализ результатов; (d) помощь пользователю (обучение); (e) импорт/экспорт данных; (f) выход.

Таким образом, рассматриваются следующие компоненты пользовательского интерфейса: (1) управление режимами системы; (2) планирование/управление процессом решения; (3) операции обработки данных; (4) редактирование и представление данных.

Модель рассматриваемого фрагмента пользовательского интерфейса представлена на Рис. 10.7. Использованы следующие обозначения:

Y - представление главных режимов (решение, помощь, выход и др.);

Z - базовые композиции цветов для главных режимов (фон для экрана, фон для представляемых элементов, текст, схема);

J - представление функциональных схем для обработки данных;

L - представление функциональных операций для обработки данных ($R_v \to R_j$);

H - представление взаимосвязей между функциональными операциями;

K - специальные воздействия в части операций (например, звуковой сигнал);

X - цветовая композиция для представления части операций (фон, текст, схема);

E - представление схемы данных;

R - представление данных (число, таблица, блок-диаграмма);

Q - представление операций редактирования данных (ввод, коррекция и др.);

B - звук в части данных;

C - цветовая композиция для представления данных (фон, текст, схема).

Используемые критерии содержатся в Таблице 10.9. DAs (для висячих вершин модели) и их оценки по критериям представлены в Таблицах 10.10, 10.11, 10.12. Приоритеты для DAs указаны в скобках на Рис. 10.7.

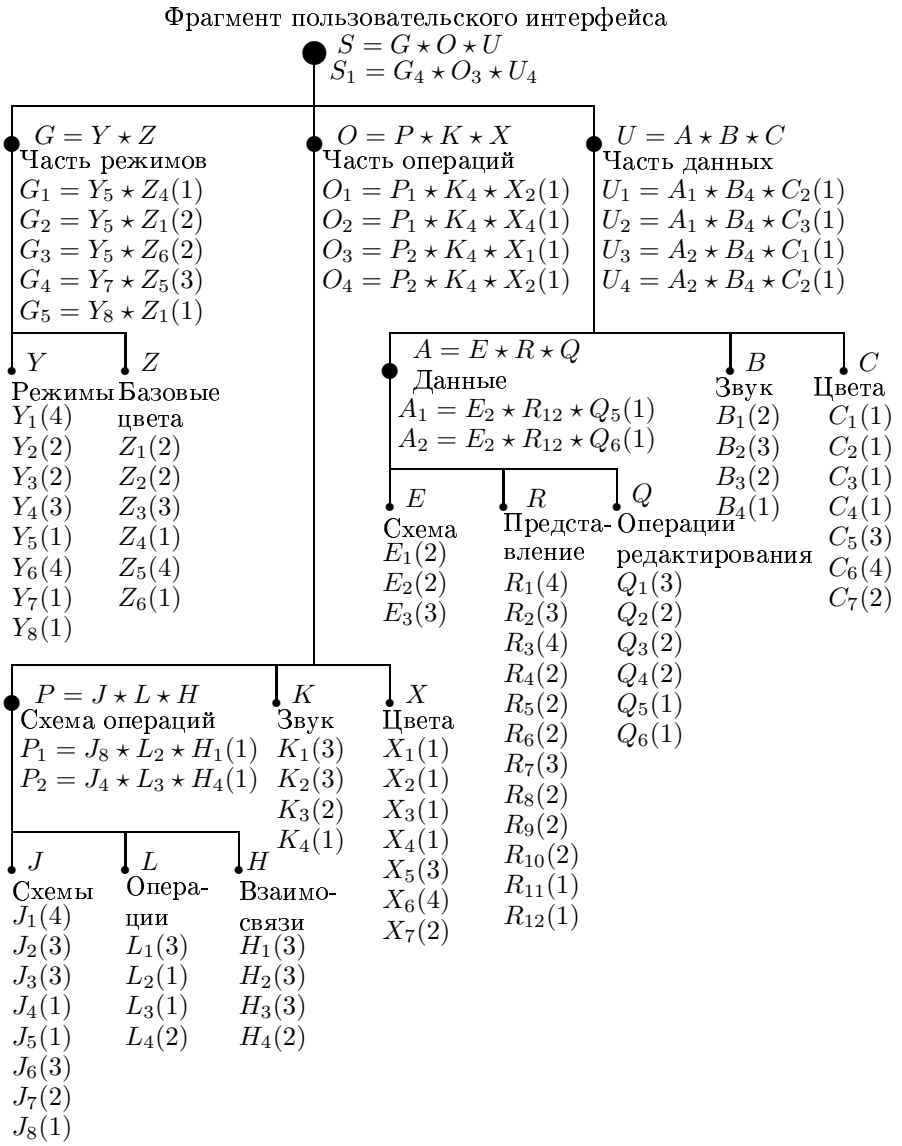


Рис. 10.7. Иерархическая схема фрагмента пользовательского интерфейса (приоритеты для DAs указаны в скобках)

Таблица 10.9. Критерии и их веса

Критерии	Веса											
	Y	J	L	H	E	R	Q	Z	X	C	K	B
1.Традиция (привычка)	5	3	5	5	5	3	2					
2.Объем информации	1	6	4	2	4		4					
3.Сложность разработки (отрицательно)	3	3	3	4	4	2	3				2	2
4.Удобство (Usability)	6	6	6	5	5	6	5	6	6	6		
5.Простота использования	6	6	6	5	5	6	5					
6.Возможность расширения	4	2	2	3	5	5	3					
7.Легкость обучения	2	6	6	4	6	6	4				3	3
8.Адаптивность								5	5	5	5	5
9.Эффективность											3	3

Таблица 10.10. DAs и их оценки (часть 1)

DAs	Критерии						
	1	2	3	4	5	6	7
Y_1 Командный язык	4	6	5	2	2	5	2
Y_2 Вертикальное меню	4	6	2	4	4	3	4
Y_3 Горизонтальное меню	4	6	2	4	4	3	4
Y_4 Блок-диаграмма	2	3	4	6	5	2	6
Y_5 Икона	3	4	3	5	6	3	6
Y_6 Меню-матрица	1	5	3	4	4	2	4
Y_7 Pop-Up меню	4	6	5	5	4	5	3
Y_8 Меню-каталог	3	6	4	5	4	5	4
J_1 Командный язык	4	6	5	2	2	5	2
J_2 Вертикальное меню	4	4	2	4	4	3	4
J_3 Горизонтальное меню	4	4	2	4	4	3	4
J_4 Блок-диаграмма	3	4	3	5	6	2	6
J_5 Икона	3	4	3	5	6	3	6
J_6 Меню-матрица	1	5	3	4	4	2	4
J_7 Pop-Up меню	3	6	5	5	4	5	3
J_8 Меню-каталог	3	6	4	5	4	5	4
L_1 Командный язык	4	6	5	2	2	5	2
L_2 Икона	3	3	3	5	6	3	6
L_3 Элементы блок-диаграммы	2	3	3	5	6	2	6
L_4 Элементы меню	3	6	5	5	4	3	3
H_1 Таблица (матрица)	4	5	4	3	2	4	3
H_2 Элемент (пара)	3	6	2	2	3	5	2
H_3 Список элементов	2	6	3	3	3	5	2
H_4 Граф	3	3	5	5	4	3	6

Таблица 10.11. DAs и их оценки (часть 2)

DAs	Критерии						
	1	2	3	4	5	6	7
E_1 Таблица связей	4	3	3	2	2	2	2
E_2 Блок-диаграмма	3	3	4	5	5	2	6
E_3 Каталог	3	6	5	5	4	5	3
R_1 Номера	3		1	1	3	1	4
R_2 Таблица (матрица)	3		2	3	3	2	4
R_3 Граф предпочтений	2		5	4	2	3	3
R_4 Vertical bar chart	3		3	3	4	3	4
R_5 Horizontal bar chart	3		3	3	4	3	4
R_6 Pie chart	3		3	3	4	3	4
R_7 Star chart	2		3	4	4	2	5
R_8 Графическое представление уровней (т.е., Парето-эффективных точек)	2		4	5	4	4	5
R_9 Анимация	2		6	5	3	3	6
R_{10} Table & bar chart	3		4	5	4	3	4
R_{11} Table & bar chart & pie chart	3		5	6	4	3	4
R_{12} Все версии	3		8	8	5	4	2
Q_1 Командный язык	4	6	5	2	2	5	2
Q_2 Вертикальное меню	4	5	2	4	4	3	4
Q_3 Горизонтальное меню	4	5	2	4	4	3	4
Q_4 Меню-матрица	2	6	3	4	4	3	5
Q_5 Список икон	3	4	3	5	6	3	6
Q_6 Pop-Up menu	3	6	4	5	4	5	3

Таблица 10.12. DAs их оценки (часть 3)

DAs	Критерии				
	3	4	7	8	9
Z_1 Black-blue-white-white		6		5	
Z_2 White-blue-yellow-black		5		2	
Z_3 Cyan-blue-white-black		3		6	
Z_4 Cyan-blue-cyan-black		4		4	
Z_5 Cyan-white-blue-black		5		5	
Z_6 Blue-white-black-black		6		4	
$X_1(C_1)$ Green-black-black		5		4	
$X_2(C_2)$ Blue-yellow-black		5		4	
$X_3(C_3)$ Blue-white-black		6		4	
$X_4(C_4)$ Blue-cyan-black		6		4	
$X_5(C_5)$ Blue-cyan-white		4		3	
$X_6(C_6)$ White-blue-black		3		2	
$X_7(C_7)$ Grey-black-black		4		4	
K_1 Нет	0		0	3	0
K_2 Старт	1		1	1	2
K_3 Конец	1		2	1	2
K_4 Процесс	3		2	2	1
B_1 Нет	0		0	3	0
B_2 Старт	1		1	1	1
B_3 Конец	1		2	1	1
B_4 Процесс	3		2	2	1

Таблицы 10.13, 10.14, 10.15 содержат оценки совместимости для DAs.

Составные DAs представлены в Таблице 10.16. Таблица 10.17 содержит узкие места и акции улучшения.

Таблица 10.13. Оценки совместимости

	L_1	L_2	L_3	L_4	H_1	H_2	H_3	H_4
J_1	5	0	0	0	1	2	2	1
J_2	0	0	0	5	1	2	2	1
J_3	0	0	0	5	1	2	2	1
J_4	1	1	5	1	3	2	2	4
J_5	0	5	0	0	3	2	2	3
J_6	1	2	0	4	4	2	2	2
J_7	1	2	0	5	0	1	1	3
J_8	2	5	0	4	5	2	2	2
L_1					4	3	3	2
L_2					5	3	3	4
L_3					3	2	2	5
L_4					4	3	3	3

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	E_1	E_2	E_3
R_1	3	3	3	4	4	4	0	0	2
R_2	3	3	3	5	4	4	3	5	1
R_3	3	3	3	4	4	4	2	5	2
R_4	3	4	2	4	3	2	4	4	4
R_5	3	2	4	4	3	2	4	4	4
R_6	3	3	4	4	4	3	4	4	4
R_7	3	3	4	4	4	3	4	4	4
R_8	3	3	4	4	4	3	4	4	4
R_9	3	3	4	3	4	4	4	4	4
R_{10}	3	4	4	5	4	4	4	5	4
R_{11}	3	4	4	5	4	4	4	5	4
R_{12}	5	5	5	5	5	5	5	5	5
Q_1							3	4	3
Q_2							4	3	5
Q_3							4	4	5
Q_4							3	5	3
Q_5							4	5	3
Q_6							4	5	3

Таблица 10.14. Оценки совместимости

	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
Z_1	3	1	1	3	4	3	3	4
Z_2	2	1	1	2	4	3	2	2
Z_3	2	4	4	4	4	4	4	3
Z_4	2	4	4	4	5	3	3	3
Z_5	2	4	4	4	3	4	3	3
Z_6	3	2	2	3	4	3	3	3

	K_1	K_2	K_3	K_4	P_1	P_2
X_1	5	5	5	5	4	5
X_2	5	5	5	5	5	5
X_3	5	5	5	5	4	4
X_4	5	5	5	5	5	4
X_5	5	5	5	5	4	4
X_6	5	5	5	5	4	4
X_7	5	5	5	5	4	4
K_1					5	5
...					...	...
K_4					5	5

Таблица 10.15. Оценки совместимости

	O_1	O_2	O_3	O_4	U_1	U_2	U_3	U_4
G_1	1	3	5	1	1	3	5	1
G_2	2	3	5	2	2	2	5	2
G_3	2	4	5	2	2	2	5	2
G_4	5	3	5	5	5	3	5	5
G_5	2	4	5	2	2	2	5	2
O_1					0	2	5	0
O_2					2	0	5	2
O_3					5	5	0	5
O_4					0	2	5	0

	B_1	B_2	B_3	B_4	A_1	A_2
C_1	5	5	5	5	4	5
C_2	5	5	5	5	5	5
C_3	5	5	5	5	5	4
C_4	5	5	5	5	4	4
...	...	...	...	...	...	...
C_7	5	5	5	5	4	4
B_1					5	5
...					...	...
B_4					5	5

Таблица 10.16. Составные DAs

DAs	N
$G_1 = Y_5 \star Z_4$	$5; 1, 1, 0, 0$
$G_2 = Y_5 \star Z_1$	$4; 2, 0, 0, 0$
$G_3 = Y_5 \star Z_6$	$4; 2, 0, 0, 0$
$G_4 = Y_7 \star Z_5$	$4; 2, 0, 0, 0$
$G_5 = Y_8 \star Z_1$	$4; 2, 0, 0, 0$
$P_1 = J_8 \star L_2 \star H_1$	$5; 2, 0, 1, 0$
$P_2 = J_4 \star L_3 \star H_4$	$4; 2, 1, 0, 0$
$O_1 = P_1 \star K_4 \star X_2$	$5; 3, 0, 0, 0$
$O_2 = P_1 \star K_4 \star X_4$	$5; 3, 0, 0, 0$
$O_3 = P_2 \star K_4 \star X_1$	$5; 3, 0, 0, 0$
$O_4 = P_2 \star K_4 \star X_2$	$5; 3, 0, 0, 0$

DAs	N
$A_1 = E_2 \star R_{12} \star Q_5$	$5; 2, 1, 0, 0$
$A_2 = E_2 \star R_{12} \star Q_6$	$5; 2, 1, 0, 0$
$U_1 = A_1 \star B_4 \star C_2$	$5; 3, 0, 0, 0$
$U_2 = A_1 \star B_4 \star C_3$	$5; 3, 0, 0, 0$
$U_3 = A_2 \star B_4 \star C_1$	$5; 3, 0, 0, 0$
$U_4 = A_2 \star B_4 \star C_2$	$5; 3, 0, 0, 0$
$S_1 = G_4 \star O_3 \star U_4$	$5; 3, 0, 0, 0$

Таблица 10.17. Узкие места и акции улучшения

Составные DAs	Узкие места		Акции	
	DAs	IC	w/r	Тип
$G_1 = Y_5 \star Z_4$	Z_4		$2 \Rightarrow 1$	1
$G_2 = Y_5 \star Z_1$		(Y_5, Z_1)	$4 \Rightarrow 5$	1
$G_3 = Y_5 \star Z_6$		(Y_5, Z_6)	$4 \Rightarrow 5$	1
$G_4 = Y_7 \star Z_5$		(Y_7, Z_5)	$4 \Rightarrow 5$	1
$G_5 = Y_8 \star Z_1$		(Y_8, Z_1)	$4 \Rightarrow 5$	1
$Y_2 \star Z_5$	Y_2		$2 \Rightarrow 1$	3
$Y_3 \star Z_5$	Y_3		$2 \Rightarrow 1$	3
$Y_5 \star Z_3$	Z_3		$2 \Rightarrow 1$	3
$Y_7 \star Z_3$	Z_3		$2 \Rightarrow 1$	3
$P_1 = J_8 \star L_2 \star H_1$	H_1		$3 \Rightarrow 1$	1
$P_2 = J_4 \star L_3 \star H_4$	H_4		$2 \Rightarrow 1$	2
$P_2 = J_4 \star L_3 \star H_4$		(J_4, H_4)	$4 \Rightarrow 5$	2
$A_1 = E_2 \star R_{12} \star Q_5$	E_2		$2 \Rightarrow 1$	1
$A_2 = E_2 \star R_{12} \star Q_6$	E_3		$2 \Rightarrow 1$	1

10.3 Эволюция системы СППР КОМБИ

В данном разделе представлен пример моделирования нескольких поколений (версий) СППР КОМБИ (DSS COMBI) [28, 329, 330, 336, 364]. Перечень поколения системы представлен в Таблице 10.18 (поколение 0 данной системы представляло собой программу на Фонтране на основе нескольких альтернативных алгоритмов).

Таблица 10.18. Поколения СППР КОМБИ

Поколение	Тип ЭВМ	Тип интерфейса	Область приложений	Обучение	Год	Ссылка	Презентация на конференции	Применение в образовании
1. КОМБИ (язык Pascal)	Мини-ЭВМ	На основе языка	Различные	Нет	1987	[26, 27] [29]	Нет	Нет
2. COMBI PC (язык Pascal)	PC	Граф-меню	Различные	Да	1988	[28]	SPUDM-89	Нет
3. COMBI PC (язык C)	PC	Граф-меню	Различные	Да	1989	[28, 329], [330, 336]	MCDM-90, EWHCI-93	[22, 331, 332], [345, 353]
4. COMBI PC (язык C)	PC	Граф-меню	Инвестиции	Да	1991	[329]	Нет	Нет

Модели системы представлены на Рис. 10.8, на Рис. 10.9, на Рис. 10.10, на Рис. 10.11 (материал опубликован в [336]). Рассмотрены четыре фазы развития системы.

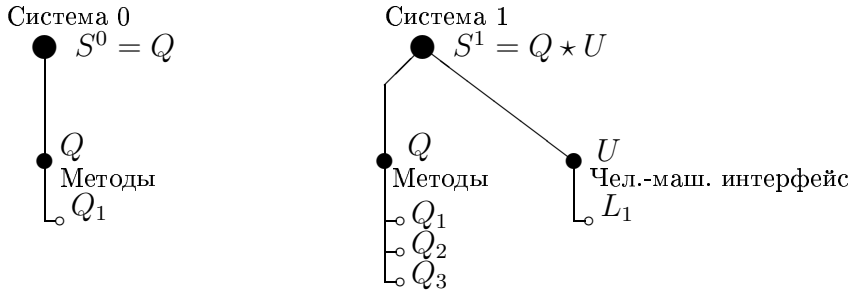


Рис. 10.8. Развитие СППР (1й шаг): фаза 1

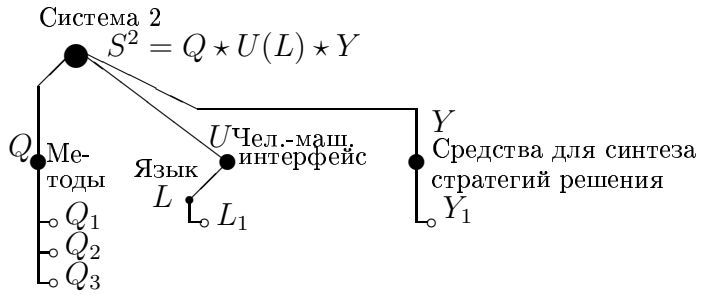


Рис. 10.9. Развитие СППР (2й шаг): фаза 2

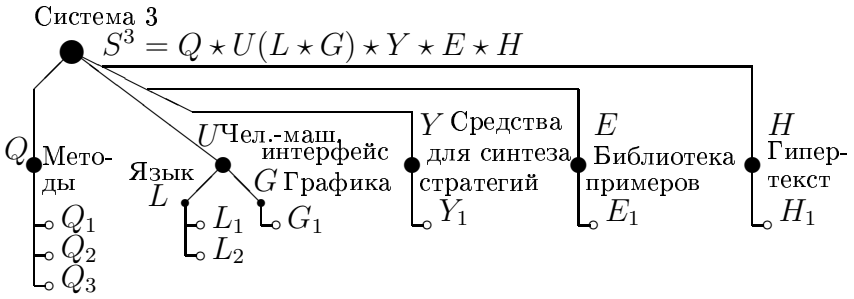


Рис. 10.10. Развитие СППР (3й шаг): фаза 3

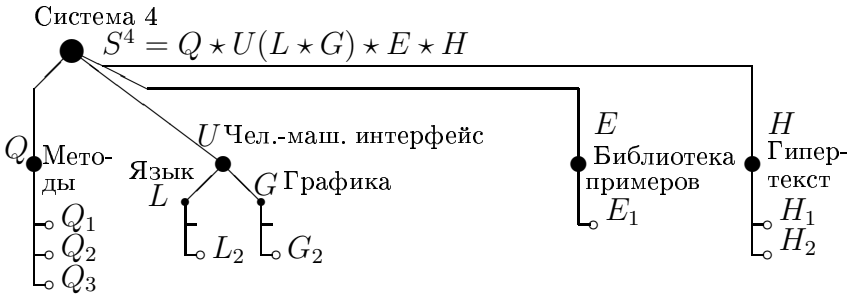


Рис. 10.11. Развитие СППР (4й шаг): фаза 4

Фазы 1, 2 и 3 основаны на расширении предыдущих версий системы (т.е., расширение структуры, добавление DAs). Фаза 4 заключается в расширении и изменении прежней версии, включая удаление DAs.

В указанных системных версиях следующие DAs были использованы: модификация метода ELECTRE (outranking technique) Q_1; метод функции полезности (utility function technique) Q_2; метод экспертной стратификации (т.е., эксперт задает уровень качества альтернативы) Q_3; командный язык L_1; меню методов L_2; граф-меню методов в виде морфологии G_1; графика для данных G_2; средства синтеза составных стратегий решения Y_1; база прикладных примеров E_1; гипертекстовая система(описание многокритериальных моделей/методов, примеры систем критериев, ссылки на источники и др.) H_1; гипертекстовая система по инвестициям H_2.

Фаза 4 основывалась на следующем:

(a) пользователи в основном не использовали режим синтеза стратегий решения (и соответствующие компонент системы был удален);

(b) потребовалось улучшить пользовательские свойства системы (usability) за счет дополнительных графических средств для представления информации.

10.4 Последовательно-параллельные стратегии

В данном разделе приведено описание композиции стратегий решения задачи многокритериального ранжирования альтернатив (sorting) на примере системы поддержки принятия решений СППР-КОМБИ.

Последовательно-параллельный процесс соответствует дерево-декомпозируемому графу (tree-decomposable graph), т.е. возможна декомпозиция исходного графа на базе дереве 'происхождения'. Другие дерево-декомпозируемые графы (например, деревья, частичные k-деревья) исследуются в ([100] и др.). Здесь используется структура типа *И-ИЛИ* дерева.

В данном разделе рассматривается синтез трех-этапной стратегии решения для многокритериального ранжирования. На Рис. 10.12 представлены два примера последовательно-параллельных стратегий для много-стадийной задачи.

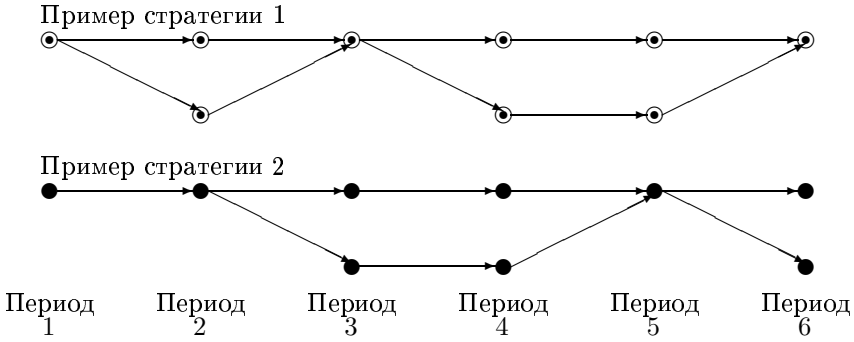


Рис. 10.12. Примеры много-стадийных последовательно-параллельных стратегий

Обычно планирование процесса принятия решений основывается на выборе и интеграции моделей из базы моделей. Такие процедуры часто требуют специальных знаний о моделях, включая базовые подмодели, их взаимосвязи и др. ([441] и др.). Здесь описана композиция схемы решения на основе ИММП, в том числе последовательно-параллельная композиция составных DAs для обработки данных с учетом IC между DAs. В этом приложении IC соответствует совместимости последовательных DAs и независимости параллельных DAs.

Обычно информационная часть систем поддержки принятия решений включает следующее:

(1) данные (альтернативы или базовые элементы, критерии, многокритериальные оценки альтернатив по критериям, отношения предпочтения);

(2) средства поддержки данных (система управления базой данных, интерфейс с коммерческими базами);

(3) вспомогательная информация для обучения и др.

Рассматриваются следующие типы данных (в основном, отношений предпочтения) $R_j (j = 0, ...3)$ (как в разделе 10.2). При этом предполагаются следующие базовые операции: (i) обработка данных (последовательная и/или параллельная); (ii) агрегация данных; (iii) параллелизация процесса решения на основе рассмотрения различных элементов (альтернативы, критерии, методы, эксперты).

Процесс решения можно представить как некую иерархию со следующими уровнями [28, 336]:

Уровень 1: алгоритмы и человеко-машинные процедуры для обработки данных.

Уровень 2: стратегии (*пошаговые* схемы обработки данных, в частности последовательно-параллельные).

Уровень 3: сценарии (комплексы стратегий, включая анализ результатов решения и обратную связь).

Далее приведены две базовые задачи проектирования стратегий [336]:

(а) композиция последовательной обработки данных (морфологическая клика);

(б) композиция параллельной обработки данных (максимальная клика с ограничением по общему весу).

Базовая схема (морфология) обработки данных (составная стратегия решения) состоит из трех стадий:

$$R_0 \Longrightarrow R_1 \Longrightarrow R_2 \Longrightarrow R_3$$

или $S = H \star T \star U$, где H соответствует формированию R_1 (некий алгоритм или процедура), T соответствует формированию R_2, U соответствует формированию R_3. Процесс проектирования стратегии решения для многокритериального ранжирования представлен на Рис. 10.13. Критерии для оценки процессов преобразования данных (локальных алгоритмов и процедур) представлены а Таблице 10.19 (включая веса важности критериев с привязкой к этапу преобразования данных). Для каждого критерия использована шкала [0...5]. Множество исследуемых в примере DAs (т.е., алгоритмов и процедур) и их характеристики представлены в Таблице 10.20.

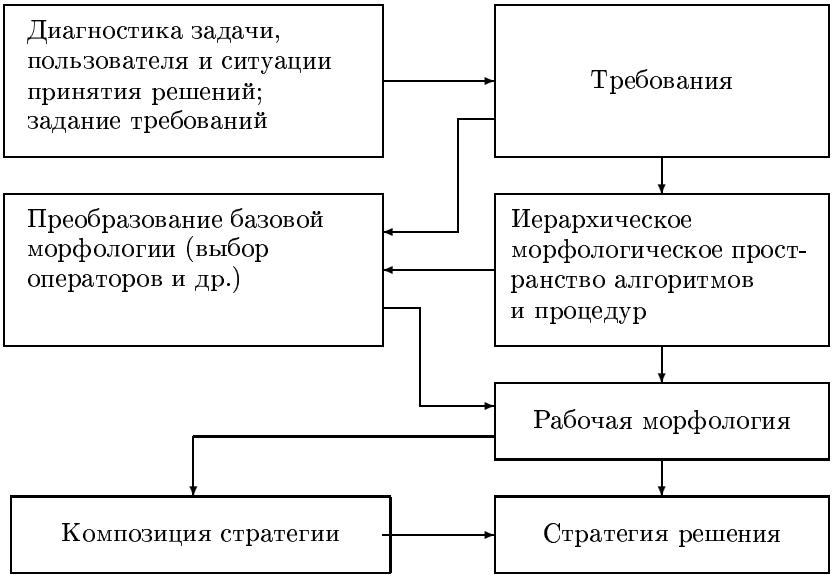


Рис. 10.13. Схема проектирования для стратегии решения

Таблица 10.19. Критерии

Критерии	Вес		
	H	T	U
1.Требуемые компьютерные ресурсы	-2	-1	-1
2.Требуемые человеческие ресурсы	-4	-3	-4
3.Качество ранжирования (надежность и др.)	5	5	5
4.Возможности для представления данных	4	4	4
5.Возможности анализа промежуточных данных	4	2	3
6.Полезность (простота обучения и др.)	5	5	5

Таблица 10.20. DAs и оценки

DAs	Описания	Критерии						Приоритет r
		1	2	3	4	5	6	
H_1	Парные сравнений	4	5	4	4	5	3	2
H_2	ELECTRE-подобный метод	3	3	4	4	5	3	1
H_3	Аддитивная функция полезности	2	3	2	3	1	4	3
H_4	Экспертная стратификация	5	5	4	4	4	3	3
T_1	Метод построчных сумм	2	0	2	2	2	4	3
T_2	Аддитивная функция полезности	2	3	2	3	1	4	3
T_3	Выделение максимальных элементов	3	0	4	4	4	3	1
T_4	Выделение Парето-эффект. элементов	3	0	4	4	4	3	1
T_5	Экспертная стратификация	5	5	4	4	4	3	2
U_1	Выделение максимальных элементов	3	0	4	4	4	3	1
U_2	Выделение Парето-эффект. элементов	3	0	4	4	4	3	1
U_3	Разбиение линейного ранжирования	4	3	3	2	3	3	3
U_4	Экспертная стратификация	5	5	4	4	4	3	2

Последовательная стратегия решения строится на основе морфологической клики (базовая версия ИММП). Для многокритериального ранжирования DAs используется СППР КОМБИ. Результирующие проритеты DAs представлены в Таблице 10.20. Таблица 10.21 содержит оценки совместимости DAs. В результате получаются следующие составные Парето-эффективные DAs: (а) $S_1 = H_4 \star T_5 \star U_4$, $N(S_1) = (5; 0, 2, 1)$; (б) $S_2 = H_2 \star T_1 \star U_3$, $N(S_2) = (5; 1, 0, 2)$; (в) $S_3 = H_2 \star T_3 \star U_1$, $N(S_3) = (4; 3, 0, 0)$.

Таблица 10.21. Совместимость DAs

	T_1	T_2	T_3	T_4	T_5	U_1	U_2	U_3	U_4
H_1	4	0	3	4	3	3	4	4	3
H_2	5	0	4	3	3	4	3	5	3
H_3	0	5	0	0	0	0	0	4	0
H_4	0	0	0	0	5	0	0	0	5
T_1						0	0	5	0
T_2						0	0	4	0
T_3						5	0	0	0
T_4						0	5	0	0
T_5						0	0	0	5

Для проектирования параллельных стратегий рассматриваются следующие DAs:

(1) H_1 (предполагается наличие 7 различных экспертов; 2-й индекс указывает параллельную версию; аналогичная ситуация рассматривается для H_4, T_5, U_4);

(2) H_2 (4 модификации методов, т.е. на основе использования различных порогов в методе порогов несравнимости - outranking technique);

(3) 3 последовательных стратегии S_1, S_2, S_3.

Таблица 10.22 содержит оценки независимости указанных DAs. Результирующие составные параллельные DAs, найденные на основе максимальной клики с порядковыми весами IC, представлены в Таблице 10.23.

Таблица 10.22. Независимость DAs

	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	H_{16}
H_{11}	.	0	4	1	4	1
H_{12}	0	.	3	4	2	3
H_{13}	4	3	.	3	4	3
H_{14}	1	4	3	.	2	4
H_{15}	4	2	4	2	.	2
H_{16}	1	3	3	4	2	.

	H_{21}	H_{22}	H_{23}	H_{24}	H_{25}
H_{21}	.	1	2	3	4
H_{22}	1	.	1	2	3
H_{23}	2	1	.	1	2
H_{24}	3	2	1	.	1
H_{25}	4	3	2	1	.

	S_1	S_2	S_3
S_1	.	3	3
S_2	3	.	2
S_3	3	3	.

Таблица 10.23. Составные параллельные стратегии

Базовые DAs	$(w; M)$	Параллельные DAs
1. H_1	$(4; 3)$	$H_{11}\&H_{13}\&H_{15}$
	$(3; 4)$	$H_{12}\&H_{13}\&H_{14}\&H_{16}$
	$(2; 5)$	$H_{12}\&H_{13}\&H_{14}\&H_{15}\&H_{16}$
2. $H_4(T_5, U_4)$	$(4; 3)$	$H_{41}\&H_{43}\&H_{45}$
	$(3; 4)$	$H_{42}\&H_{43}\&H_{44}\&H_{46}$
	$(2; 5)$	$H_{42}\&H_{43}\&H_{44}\&H_{45}\&H_{46}$
3. H_2	$(4; 2)$	$H_{21}\&H_{25}$
	$(2; 3)$	$H_{21}\&H_{23}\&H_{25}$
4. S	$(3; 2)$	$S_1\&S_2, S_1\&S_3$
	$(2; 3)$	$S_1\&S_2\&S_3$

Проектная морфология представлена на Рис. 10.14. Удаляем DAs, которые 'покрываюся' другими (например, $S_1\&S_3$ покрываются $S_1\&S_2\&S_3$). При этом предполагается, что

характеристики параллельных DAs (оценки, совместимость, требуемые ресурсы) соответствуют характеристикам базовых DAs.

Результирующие последовательно-параллельные стратегии представлены на Рис. 10.15. При этом параллельные стратегии включены в соответствующие последовательные стратегии.

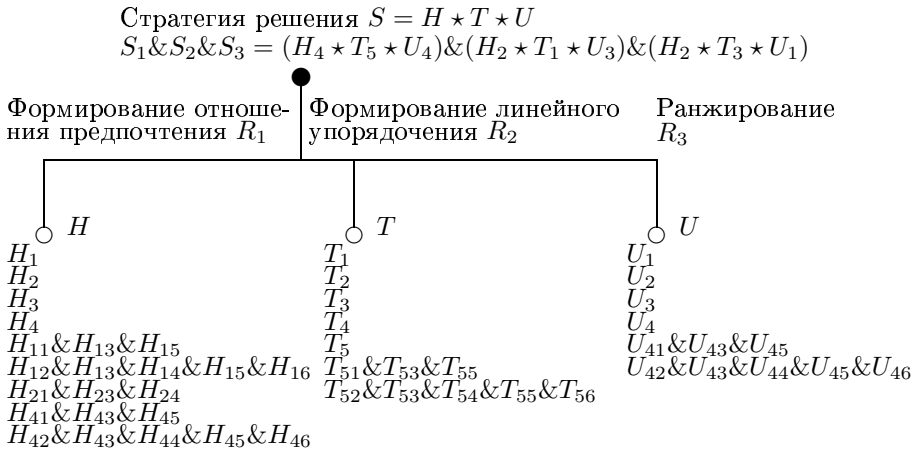


Рис. 10.14. Результирующая проектная морфология

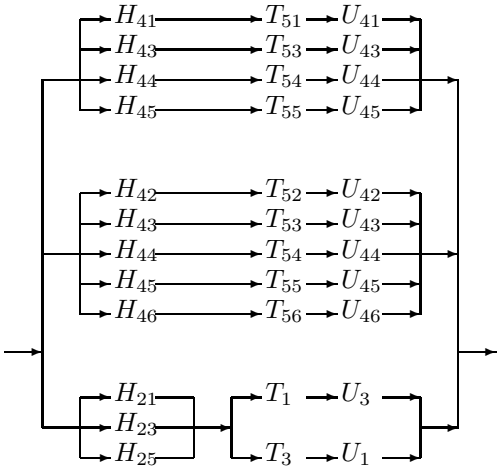


Рис. 10.15. Последовательно-параллельные стратегии

10.5 Выводы к главе

В данной главе представлены три задачи в области построения, модификации, анализа (эволюции) модульного программного обеспечения и задача построения последовательно-параллельных стратегий решения на примере многокритериального ранжирования. Представляется перспективным рассмотреть в будущем следующее:

1. исследование различных прикладных программных систем;

2. учет неопределенности параметров;

3. построение стратегий решения задач на основе графовых структур.

Глава 11

Морфологическая схема для задачи размещения

В данной главе [1] приведено описание решения задачи размещения элементов на плоскости с помощью морфологического подхода. В примере распределяются сотрудники двух проектов по комнатам.

11.1 Описание задачи

Задача размещения является ключевой для многих приложений, например [159, 427, 428, 448, 466]: (1) размещение оборудования в сетях; (2) размещение информации (информационных массивов и т.п.): (3) размещение датчиков (например, в крупных зданиях); (4) распределение частот по каналам связи; (5) размещение задач в вычислительных сетях (для выполнения); (6) проектирование электрических сетей; (7) проектирование стандартов; (8) проектирование строительных/архитектурных объектов и др. Более 50 базовых типов задач размещения (целевые функции, системные параметры, алгоритмы) приведены в [105].

Здесь представлена макроэвристика на основе ИММП для решения 'расширенной' задачи размещения [336]. В предлагаемой постановке, для каждого места (позиции) рассматривается множество локальных альтернативных решений по размещению объектов (т.е., DAs). В этом случае совместимость может основываться на ресурсных ограничениях и т.п. Пусть $H = \{1, ..., i, .., n\}$ – множество элементов для размещения (оборудование, персонал). Множество позиций обозначим $Q = \{1, ..., i, .., m\}$. Дополнительно используется следующая информация:

(1) взвешенные отношения на Q (близость);

(2) отношение на H;

(3) соответствие между элементами H и Q (веса, вектор оценок, отношение предпочтения, ограничения и др.);

(4) информация для некоторых локальных размещений двух типов:

(a) соответствие между составным элементом (как подмножество H, которое может быть использовано для размещения некоторого элемента из Q) и позициями Q; можно использовать следующие данные: веса, вектор оценок, отношение предпочтения ограничения, правила и др.;

[1] Материал главы основан на публикациях:
(i) Levin M.Sh., Combinatorial Engineering of Decomposable Systems. Springer, 1998. chapter 7.
(ii) Levin M.Sh., Combinatorial optimization in system configuration design. Autom. and Remote Control, vol. 70, no. 3, pp. 519-561, 2009.

(b) соответствие между локальными решением по размещению, например, как пара: $i_1 \in H$ назначено на $j_1 \in Q$ и $i_2 \in H$ назначено на $j_2 \in Q$; следующие данные могут быть использованы: веса, вектор оценок, отношение предпочтения ограничения, правила и др.;

В ситуации, когда несколько элементов H размещаются на одну и ту же позицию, можно построить новый агрегированный ('склеенный') элемент H с соответствующими агрегированными параметрами Предлагаемая задача имеет вид:

Разместить элементы множества H на позиции Q с учетом дополнительной информации (отношения, предпочтения, правила и др.).

Данная задача может быть сформулирована как задача выполнимости или как оптимизационная. Качество размещения элементов (решение) можно сформулировать как интегрированное 'качество' локальных решений (на основе оценок, предпочтений и т.п.) с учетом ограничений и правил. В комбинаторной оптимизации рассматриваются различные версии задачи размещений (например, задача о паросочетаниях, квадратичная задача о назначении, обобщенная задача о назначении и др.). В последние годы используются также подходы на основе методов искусственного интеллекта.

Рассматривается пример использования морфологического подхода (макроэвристики) на основе назначения персонала по комнатам: морфологический класс соответствует множеству сотрудников, которые могут быть размещены в конкретном служебном помещении (комнате). В примере используются следующие концепты: (1) исследовательские проекты; (2) персонал; (3) комнаты; (4) бинарные отношения на указанных выше множествах (концептах).

Пусть имеется 4 проекта: (а) крупный проект R_1 (три специалиста и секретарь); (б) проект R_2 (два специалиста); (в) проект R_3 (один специалист); (г) проект R_4 (один специалист). Один секретарь участвует в проектах R_1, R_2, R_4.

На Рис. 11.1, Рис. 11.2, Рис. 11.3 представлены структура проектов, план комнат (A, B, C, D, X, Y, Z) и их 'близости' (два уровня 'близости': пунктирная линия соответствует 'слабой близости') и описание персонала. Возможное размещение одного или двух сотрудников в комнатах указано в Таблице 11.1 в квадратных скобках.

Дополнительно рассматриваются правила:

Правило 11.1. Лидер проекта должен размещаться близко к остальным членам группы (исследователи, менеджер, секретарь).

Правило 11.2. Лидер проекта должен располагаться в большой комнате (один).

Правило 11.3. Менеджер проекта должен располагаться близко к лидеру и секретарю.

Правило 11.4. Менеджер проекта должен располагаться в небольшой комнате (один) или в большой комнате (вдвоем).

Правило 11.5. Исследователь, который реализует свой проект и руководит им как менеджер, должен располагаться в небольшой комнате (один) или в большой комнате (вдвоем).

Правило 11.6. Участники одного проекта должны располагаться в одной комнате или в близко расположенных комнатах.

Правило 11.7. Секретарь может располагаться в большой комнате (вдвоем) или в небольшой комнате (вдвоем).

Правило 11.8. курящие и некурящие сотрудники не могут располагаться в одной комнате.

Правило 11.9. Сотрудники, связанные отношением дружбы должны располагаться в одной комнате или близко расположенных комнатах.

Указанные правила учитываются при размещении сотрудников по комнатам (Таблица 11.1).

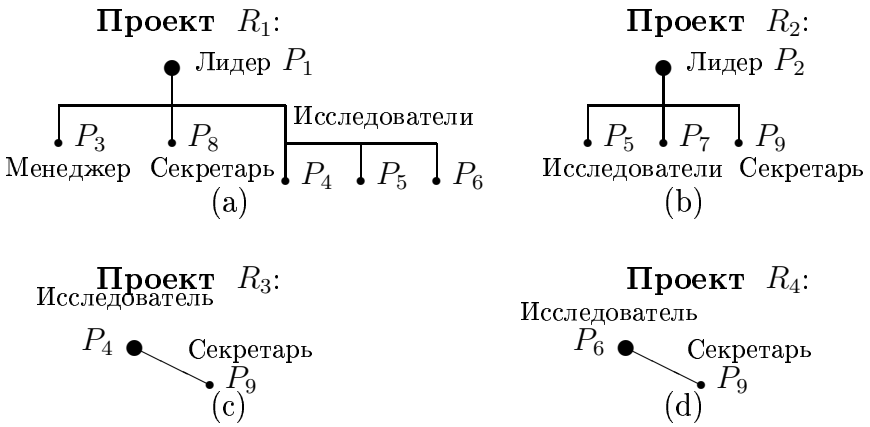


Рис. 11.1. Структура проектов

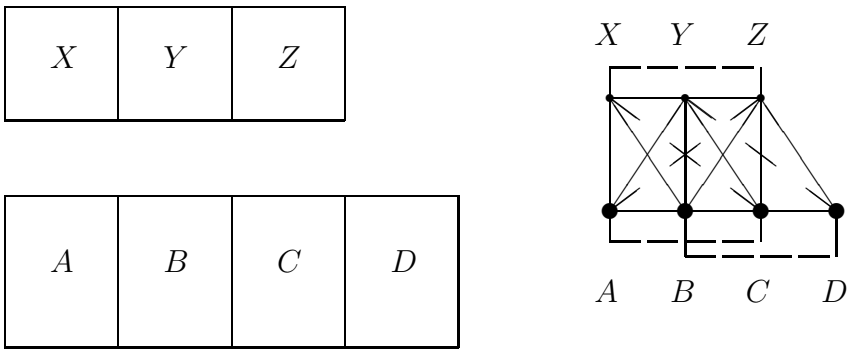


Рис. 11.2. План комнат и граф их близости

Таблица 11.1. Описание персонала

Пер-сонал	Роль	Проект	Куре-ние	Соответствующие комнаты	Друж-ба
P_1	Лидер кр. проекта	R_1	Да	$A[1]$, $B[1]$, $C[1]$, $D[1]$	P_2,P_3
P_2	Лидер проекта	R_2	Нет	$A[1]$, $B[1]$, $C[1]$, $D[1]$	P_1
P_3	Менеджер кр. проекта	R_1	Да	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_1,P_5 P_8
P_4	Исслед-ль	R_1,R_3	Нет	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_6
P_5	Исслед-ль	R_2	Да	$A[2]$, $B[2]$, $C[2]$, $D[2]$	P_3,P_8
P_6	Исслед-ль	R_1,R_4	Нет	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[1]$, $Y[1]$, $Z[1]$	P_4
P_7	Исслед-ль	R_1,R_2	Да	$A[2]$, $B[2]$, $C[2]$, $D[2]$	P_5,P_9
P_8	Секретарь	R_1	Да	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[2]$, $Y[2]$, $Z[2]$	P_3,P_5
P_9	Секретарь	Все	Да	$A[2]$, $B[2]$, $C[2]$, $D[2]$, $X[2]$, $Y[2]$, $Z[2]$	P_7

11.2 Составное решение

Во-первых, рассматривается 'дополнительность' сотрудников, которые могут располагаться вдвоем в одной комнате. Таблица 11.2 содержит вектор оценок и результирующую порядковую оценку:

1. Вектор оценок включает следующие бинарные элементы:

(1) курение (1 соответствует совместному курению или некурению, 0 - иначе);

(2) совместная работа (1 соответствует участия в одном проекте, 0 - иначе);

(3) отношение дружбы (1 соответствует дружбе, 0 - иначе).

2. Для результирующей оценки дополнительности используется шкала: $[0, 1, 2]$, где 0 - несовместимость, 1 - совместимость, 2 - хорошая совместимость.

Таблица 11.2. Дополнительность

	P_4	P_5	P_6	P_7	P_8	P_9
P_3	(010)/0	(101)/1	(010)/0	(110)/1	(111)/2	(100)/1
P_4		(000)/0	(111)/2	(010)/0	(010)/0	(010)/0
P_5			(000)/0	(111)/2	(100)/1	(110)/1
P_6				(010)/0	(010)/0	(010)/0
P_7					(110)/1	(111)/2
P_8						(110)/1

В результате получаются следующие хорошие агрегированные DAs:

$P_{38} = P_3 \& P_8$, $P_{46} = P_4 \& P_6$, $P_{79} = P_7 \& P_9$, $P_{57} = P_5 \& P_7$.

Итак, используются специальные оценки двух типов: (1) соответствие сотрудников комнатам, (2) совместимость локальных решений.

Локальное решение заключается в размещении сотрудника в комнате. При этом исследуется следующее: (а) словесное описание и правила, (б) многокритериальная оценка, (в) результирующая порядковая оценка. На Рис. 11.3 представлена результирующая структура составного решения. Предполагается, что порядковые оценки соответствия сотрудников комнатам получены на основе экспертных суждений (с учетом указанных выше правил и персональных предпочтений).

Таблица 11.3 содержит два отношения на множестве сотрудников, полученных на основе Рис. 11.1 и Таблицы 11.1: совместное участие проекте (p), и отношение дружбы (f).

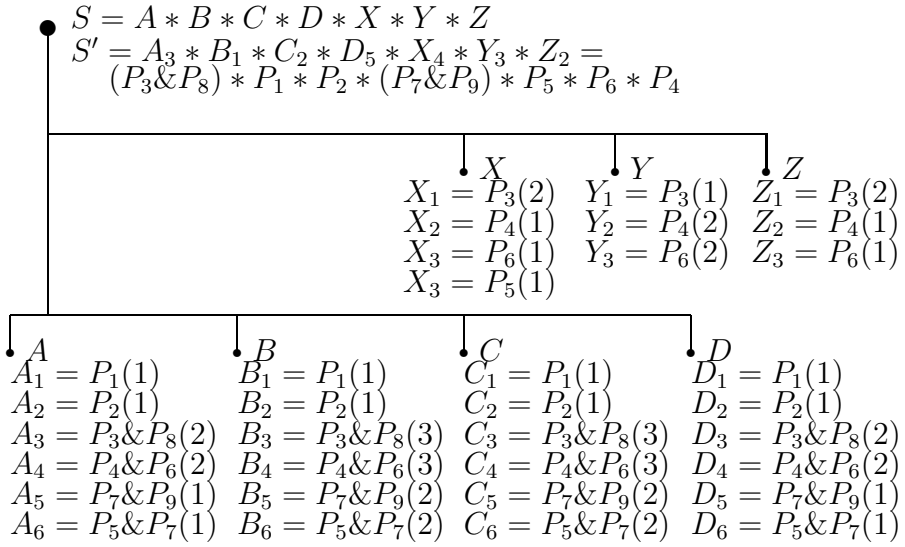


Рис. 11.3. Структура составного решения

Таблицы 11.4, 11.5, 11.6, 11.7, 11.8, 11.9 содержат оценки совместимости локальных решений. Используются следующие оценки:

1. Вектор, состоящий из двух компонентов:

(1.1) выполнение правил для совместных проектов:

(i) переменная равна 2, если выполняются правила 11.1, 11.3, 11.6;

(ii) переменная равна 1, если 'слабое' выполнение правил 11.1, 11.3, 11.6;

(iii) переменная обозначается $\triangle$ в противном случае (несовместимость и т.п.);

(1.2) выполнение правил для отношения дружбы:

(i) переменная равна 2, если выполняется правило 11.9;

(ii)переменная равна 1, если 'слабое' выполнение правила 11.9;

(iii) 0 - иначе.

2. Результирующая порядковая оценка:

(i) наилучшая совместимость:

(a) 5 (векторная оценка $(4,4)$ или $(8,0)$) или

(b) $\diamondsuit$ (независимость альтернатив DAs);

(ii) почти наилучшая совместимость 4 (векторная оценка равна $(4,2)$);

(iii) хорошая совместимость 3 (векторная оценка равна $(4,0)$ или $(2,2)$);

(iv) удовлетворительная совместимость 2 (векторная оценка равна $(1,1)$, $(2,0)$) $(0,2)$, $(2,1)$);

(v) плохая совместимость 1 (векторная оценка равна $(0,1)$, $(1,0)$);

(vi) несовместимость:

(a) 0 (векторная оценка равна $(0,0)$);

(b) $\triangle$ (если компонент векторной оценки обозначен $\triangle$).

В результате получается составное решение (Рис. 11.4):

$$S' = A_3 * B_1 * C_2 * D_5 * X_4 * Y_3 * Z_2 = (P_3 \& P_8) * P_1 * P_2 * (P_7 \& P_9) * P_5 * P_6 * P_4,$$

где $N(S') = (2; 5, 2)$.

Таблица 11.3. Два отношения

	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	f	p,f	p	p	p	-	p	-
P_2		-	-	p	-	p	p	-
P_3			p	p,f	p	-	p,f	-
P_4				p	p,f	-	p	p
P_5					p	f	p,f	-
P_6						-	p	p
P_7							-	p,f
P_8								-

Таблица 11.4. Совместимость

	B_1	B_2	B_3	B_4	B_5	B_6
A_1	$/\triangle$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	$/\diamondsuit$	$(2,0)/2$
A_2	$(0,2)/2$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(4,0)/3$	$(4,0)/3$
A_3	$(4,2)/4$	$/\diamondsuit$	$/\triangle$	$(8,0)/5$	$/\diamondsuit$	$(4,4)/5$
A_4	$(4,0)/3$	$/\diamondsuit$	$(8,0)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
A_5	$/\diamondsuit$	$(4,0)/3$	$/\diamondsuit$	$(4,0)/3$	$/\triangle$	$/\triangle$
A_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\triangle$	$/\triangle$

Таблица 11.5. Совместимость

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3
A_1	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
A_2	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$	$(2,0)/2$	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
A_3	$/\triangle$	$(4,0)/3$	$(4,0)/3$	$(4,4)/8$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
A_4	$(4,0)/3$	$/\triangle$	$/\triangle$	$(4,0)/3$	$(4,0)/3$	$/\triangle$	$/\triangle$
A_5	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$	$(0,2)/2$	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
A_6	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$	$/\triangle$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
B_1	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
B_2	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$	$(2,0)/2$	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
B_3	$/\triangle$	$(4,0)/3$	$(4,0)/3$	$(4,4)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
B_4	$(4,0)/3$	$/\triangle$	$/\triangle$	$(4,0)/3$	$(4,0)/3$	$/\triangle$	$/\triangle$
B_5	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$	$(0,2)/2$	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
B_6	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$	$/\triangle$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
C_1	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$	$(1,0)/1$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
C_2	$(0,1)/1$	$/\diamondsuit$	$/\diamondsuit$	$(1,0)/1$	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
C_3	$/\triangle$	$(2,0)/2$	$(2,0)/2$	$(2,2)/4$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
C_4	$(2,0)/2$	$/\triangle$	$/\triangle$	$(2,0)/2$	$(4,0)/3$	$/\triangle$	$/\triangle$
C_5	$/\diamondsuit$	$(1,0)/1$	$(1,0)/1$	$(0,2)/2$	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
C_6	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$	$/\triangle$	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
D_1	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$
D_2	$(0,0)/0$	$/\diamondsuit$	$/\diamondsuit$	$(0,0)/0$	$(0,1)/1$	$/\diamondsuit$	$/\diamondsuit$
D_3	$/\triangle$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$(2,0)/2$	$(2,0)/2$
D_4	$(0,0)/0$	$/\triangle$	$/\triangle$	$(0,0)/0$	$(2,0)/2$	$/\triangle$	$/\triangle$
D_5	$/\diamondsuit$	$(0,0)/0$	$(0,0)/0$	$/\diamondsuit$	$/\diamondsuit$	$(1,0)/1$	$(1,0)/1$
D_6	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$(1,1)/2$	$(1,0)/1$	$/\triangle$

Таблица 11.6. Совместимость

	C_1	C_2	C_3	C_4	C_5	C_6
A_1	$/\triangle$	$(0,1)/1$	$(2,1)/2$	$(2,0)/2$	$/\diamondsuit$	$(1,0)/1$
A_2	$(0,1)/1$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
A_3	$(2,1)/2$	$/\diamondsuit$	$/\triangle$	$(4,0)/3$	$/\diamondsuit$	$(2,2)/3$
A_4	$(2,0)/2$	$/\diamondsuit$	$(4,0)/3$	$/\triangle$	$(2,0)/2$	$(2,0)/2$
A_5	$/\diamondsuit$	$(2,0)/2$	$/\diamondsuit$	$(2,0)/2$	$/\triangle$	$/\triangle$
A_6	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$/\triangle$	$/\triangle$
B_1	$/\triangle$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	$/\diamondsuit$	$(2,0)/2$
B_2	$(0,2)/2$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(4,0)/3$	$(4,0)/3$
B_3	$(4,2)/4$	$/\diamondsuit$	$/\triangle$	$(8,0)/5$	$/\diamondsuit$	$(4,4)/5$
B_4	$(4,0)/3$	$/\diamondsuit$	$(8,0)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
B_5	$/\diamondsuit$	$(4,0)/3$	$/\diamondsuit$	$(4,0)/3$	$/\triangle$	$/\triangle$
B_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\triangle$	$/\triangle$

Таблица 11.7. Совместимость

	Y_1	Y_2	Y_3	Z_1	Z_2	Z_3
X_1	$/\triangle$	$(2,0)/2$	$(2,0)/2$	$/\triangle$	$(1,0)/1$	$(1,0)/1$
X_2	$(2,0)/2$	$/\triangle$	$(2,2)/3$	$(1,0)/1$	$/\triangle$	$(1,1)/2$
X_3	$(2,0)/2$	$(2,2)/3$	$/\triangle$	$(1,0)/1$	$(1,1)/2$	$/\triangle$
X_4	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$
Y_1				$/\triangle$	$(2,0)/2$	$(2,0)/2$
Y_2				$(2,0)/2$	$/\triangle$	$(2,2)/3$
Y_3				$(2,0)/2$	$(2,2)/3$	$/\triangle$

Таблица 11.8. Совместимость

	D_1	D_2	D_3	D_4	D_5	D_6
A_1	$/\triangle$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\diamondsuit$	$(0,0)/0$
A_2	$(0,0)/0$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(0,0)/0$	$(0,0)/0$
A_3	$(0,0)/0$	$/\diamondsuit$	$/\triangle$	$(0,0)/0$	$/\diamondsuit$	$(0,0)/0$
A_4	$(0,0)/0$	$/\diamondsuit$	$(0,0)/0$	$/\triangle$	$(0,0)/0$	$(0,0)/0$
A_5	$/\diamondsuit$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$/\triangle$
A_6	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$(0,0)/0$	$/\triangle$	$/\triangle$
B_1	$/\triangle$	$(0,1)/1$	$(2,1)/2$	$(2,0)/2$	$/\diamondsuit$	$(1,0)/1$
B_2	$(0,1)/1$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
B_3	$(2,1)/2$	$/\diamondsuit$	$/\triangle$	$(4,0)/3$	$/\diamondsuit$	$(2,2)/3$
B_4	$(2,0)/2$	$/\diamondsuit$	$(4,0)/3$	$/\triangle$	$(2,0)/2$	$(2,0)/2$
B_5	$/\diamondsuit$	$(2,0)/2$	$/\diamondsuit$	$(2,0)/2$	$/\triangle$	$/\triangle$
B_6	$(2,0)/2$	$(2,0)/2$	$(2,2)/3$	$(2,0)/2$	$/\triangle$	$/\triangle$
C_1	$/\triangle$	$(0,2)/2$	$(4,2)/4$	$(4,0)/3$	$/\diamondsuit$	$(2,0)/2$
C_2	$(0,2)/2$	$/\triangle$	$/\diamondsuit$	$/\diamondsuit$	$(4,0)/3$	$(4,0)/3$
C_3	$(4,2)/4$	$/\diamondsuit$	$/\triangle$	$(8,0)/5$	$/\diamondsuit$	$(4,4)/5$
C_4	$(4,0)/3$	$/\diamondsuit$	$(8,0)/5$	$/\triangle$	$(4,0)/3$	$(4,0)/3$
C_5	$/\diamondsuit$	$(4,0)/3$	$/\diamondsuit$	$(4,0)/3$	$/\triangle$	$/\triangle$
C_6	$(2,0)/2$	$(4,0)/3$	$(4,4)/5$	$(4,0)/3$	$/\triangle$	$/\triangle$

Таблица 11.9. Совместимость

	Z_1	Z_2	Z_3
A_1	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$
A_2	$(0,1)/1$	$/\diamondsuit$	$/\diamondsuit$
A_3	$/\triangle$	$(2,0)/2$	$(2,0)/2$
A_4	$(2,0)/2$	$/\triangle$	$/\triangle$
A_5	$/\diamondsuit$	$(1,0)/1$	$(1,0)/1$
A_6	$(1,1)/2$	$(1,0)/1$	$(1,0)/1$
B_1	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
B_2	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
B_3	$/\triangle$	$(4,0)/3$	$(4,0)/3$
B_4	$(4,0)/3$	$/\triangle$	$/\triangle$
B_5	$/\diamondsuit$	$(2,0)/1$	$(2,0)/1$
B_6	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
C_1	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
C_2	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
C_3	$/\triangle$	$(4,0)/3$	$(4,0)/3$
C_4	$(4,0)/3$	$/\triangle$	$/\triangle$
C_5	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
C_6	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
D_1	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$
D_2	$(0,2)/2$	$/\diamondsuit$	$/\diamondsuit$
D_3	$/\triangle$	$(4,0)/3$	$(4,0)/3$
D_4	$(4,0)/3$	$/\triangle$	$/\triangle$
D_5	$/\diamondsuit$	$(2,0)/2$	$(2,0)/2$
D_6	$(2,2)/3$	$(2,0)/2$	$(2,0)/2$

X_4 : P_5	Y_3 : P_6	Z_2 : P_4

A_3 : $P_3 \& P_8$	B_1 : P_1	C_2 : P_2	D_5 : $P_7 \& P_9$

Рис. 11.4. Составное решения

11.3 Выводы к главе

В главе описана задача размещения с учетом совместимости локальных назначения элементов на позиции. Использована новая схема решения в виде морфологического подхода. Представляется крайне перспективным применение предложенной постановки 'расширенной' задачи размещений и примененного подхода к решению в различных приложениях.

Глава 12

Приложения при управлении в умном доме

В данной главе [1] приведено описание двух примеров с использованием интервальных оценок в виде мультимножеств: (а) система управления умным домом (иерархическая модель, композиция, улучшение), (б) типовая система автоматизации здания (иерархическая модель, композиция, улучшение).

12.1 Предварительные сведения

Системы автоматизации зданий (building design automation systems BASs) становятся все более и более популярны [264, 288, 369, 398, 506]. BAS обеспечивают управление аппаратурой здания (включая функции мониторинга, выявления неисправностей, уменьшения потребления энергии и др.) с учетом надежности и уменьшения стоимости технического обслуживания. В последние два десятилетия, многие усилия направлены на исследования в области умных домов (smart homes): (1) архитектура [247, 264, 468, 566]; (2) проектирование [70, 264]; (3) управление [483, 578]; (4) динамика [473]; (5) радиосвязь [528]; (6) сенсоры [459]; (7) интеграция данных [257, 578]. В данной статье рассматривается модульное проектирование и улучшение системы управления умного дома и автоматизированной системы управления зданием. Рис. 12.1 иллюстрирует структуру нашего исследования: (1) модульное проектирование как композиция конфигурации системы; (2) модульное улучшение системы: (i) улучшение частей/компонентов системы, (ii) улучшение совместимости компонентов системы, (iii) изменение архитектуры системы. Таким образом, рассматриваются две задачи модульного проектирования: (а) иерархический комбинаторный синтез, (б) улучшение системы (в частности, upgrade).

[1] Материал главы основан на публикациях:
(i) Levin M.Sh., Modular design and improvement of the management system in the smart home with the use of interval multiset estimates. J. of Communications Technology and Electronics, vol. 58, no. 6, 584-593, 2013.
(ii) Левин М.Ш. Модульное проектирование и улучшение управления в умном доме с использованием интервальных оценок в виде мультимножеств. Электр. научн. ж. “Информационные процессы”, 12(2), 2012. С. 141-154.

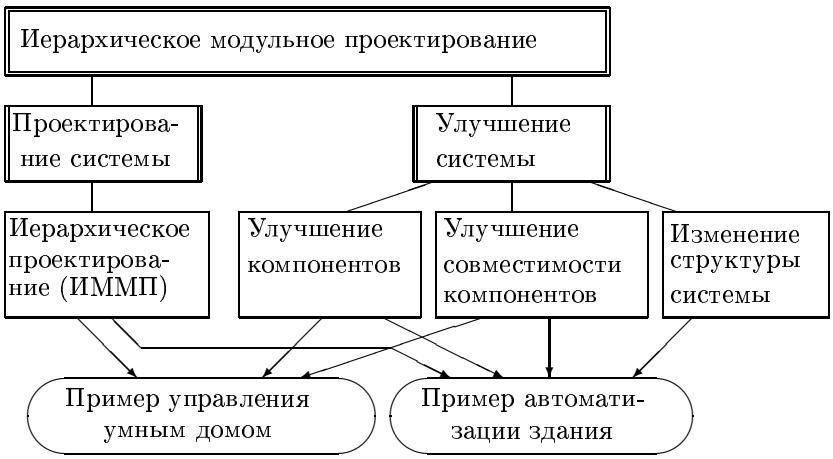


Рис. 12.1. Схема исследования

Следует отметить, что в последние годы существенно возросла важность вопросов перепроектирования систем (reengineering), т.е., процессы улучшения, трансформации, реконфигурации, расширения [70, 336, 344, 400, 491]. В данной главе рассматриваются две схемы улучшения [336, 344, 367, 400]: (1) стратегия улучшения узких мест как улучшение некоторого подмножества компонентов системы или совместимости между компонентами системы (стратегия 1); (2) расширение системы за счет добавления дополнительной подсистемы (стратегия 2).

В данной статье подробно исследованы два реальных прикладных примера: (а) модульная система управления для умного дома (Рис. 12.2, [393]), (б) модульная система автоматизации здания [398]. Структура системы автоматизации здания представлена на Рис. 12.3 ("SAC" соответствует "сенсору, актуатору, контроллеру" ("sensor, actuator, controller") [506]). Рассматриваемый пример системы автоматизации здания включает четыре части: (1) IP/KNX Gateway, (2) IP/WSN 6LoWPAN Gateway, (3) ZigBee Wireless Sensor Network, (4) KNX Field Bus Infrastructure.

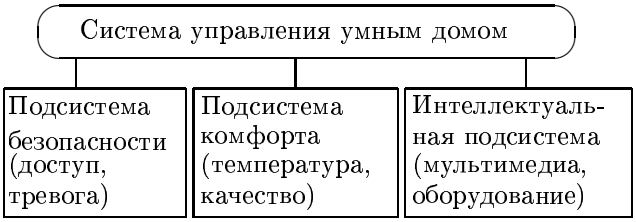


Рис. 12.2. Структура системы управления

Примеры включают следующее: иерархическая (древовидная) структура системы, проектные варианты (альтернативы) DA для частей/компонентов системы, процесс решения. Экспертное оценивание использовано для оценки DA (интервальные оценки на основе мультимножеств (interval multiset estimates [360]). Отметим, все оценки и примеры вычислений имеют иллюстративный характер, но могут быть успешно использованы как основа для реальных приложений.

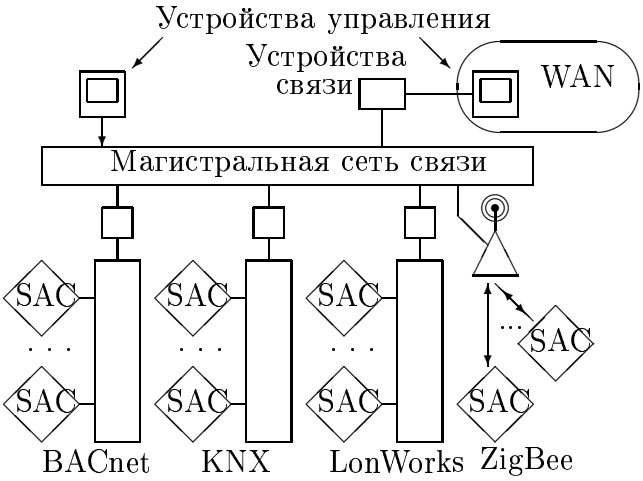


Рис. 12.3. Пример системы автоматизации здания

12.2 Примеры для системы управления умным домом

12.2.1 Иерархическая модель

Рассматривается следующая иерархическая структура системы управления умным домом (Рис. 12.4, интервальные оценки в виде мультимножеств представлены в круглых скобках):

0. Система управления $S = A \star B \star C$.
1. Подсистема безопасности $A = D \star E$.
1.1. Управление доступом $D = G \star H \star I$.
1.1.1. Оконные ставни G: ручной режим G_1, электрический режим G_2.
1.1.2. Дверной замок H: типовой H_1, электрический H_2.
1.1.3. Точка идентификации I: ключ I_1, PIN-код I_2, RFID I_3, биометрический способ I_4.
1.2. Управление тревогами $E = J \star K \star L$.
1.2.1. Сигнал тревоги J: сирена J_1, свет J_2.
1.2.2. Детектор присутствия K: инфракрасные лучи K_1, ультразвук K_2, движение K_3.
1.2.3. Способ сигнализации L: городской телефон L_1, радио L_2, Интернет L_3, GSM/SMS L_4.
2. Подсистема комфорта $B = M \star N$.
2.1. Температура $M = O \star P$.
2.1.1. Подогрев O: пол O_1, радиаторы O_2, потолок O_3, стена с подогревом O_4.
2.1.2. Кондиционер P: внешний P_1, внутренний P_2.
2.2. Качество среды $N = R \star F$.
2.2.1. Вентиляция R: по перекрытиям R_1, на рабочих местах R_2, центральная R_3.
2.2.2. Фильтрация воздуха F: локальная F_1, центральная F_2.
3. Интеллектуальная подсистема $C = Q \star T$.
3.1. Мультимедиа $Q = W \star V \star U$.
3.1.1. Видесистема W: монитор W_1, мультимедийный проектор W_2.
3.1.2. Аудио-система V: “2:1” V_1, “5:1” V_2, схема шумподавления Долби V_3.
3.1.3. Домашний сервер/ПК U: распределенный U_1, интегрированный U_2.
3.2. Домашнее оборудование $T = X \star Y \star Z$.
3.2.1. Духовой шкаф X: газовый X_1, электрический X_2.

3.2.2. Рефрижератор Y: с морозильником Y_1, с связью по Интернет Y_2.
3.2.3. Вакуумный очиститель Z: центральный Z_1, iLoc Z_2.

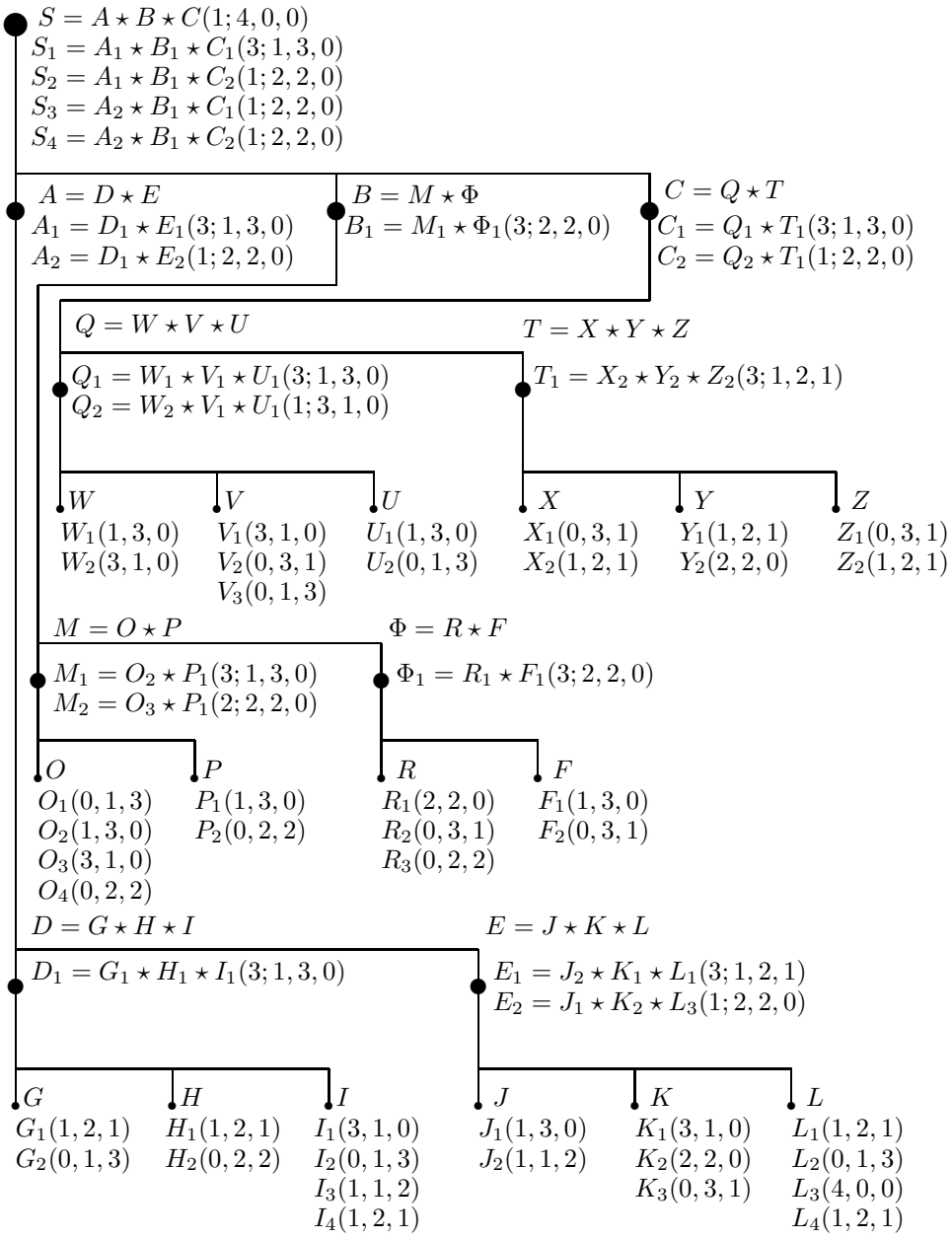


Рис. 12.4. Структура системы управления

Использованные оценки в виде мультимножеств основаны на нескольких критериях (стоимость, потребление энергии, надежность, длительность жизненного цикла [393]). Оценки совместимости между DAs представлены в Таблицах 12.1, 12.2 (порядковая шкала, экспертное оценивание).

Таблица 12.1. Совместимость

	H_1	H_2	I_1	I_2	I_3	I_4
G_1	3	3	3	2	1	1
G_2	3	3	3	3	3	3
H_1			3	1	1	1
H_2			1	3	3	3

	K_1	K_2	K_3	L_1	L_2	L_3	L_4
J_1	2	1	3	2	1	1	3
J_2	3	3	3	3	3	3	2
K_1				3	2	0	2
K_2				2	1	1	2
K_3				2	3	2	2

	O_1	O_2	O_3	O_4
P_1	3	3	2	1
P_2	2	3	1	2

Таблица 12.2. Совместимость

	R_1	R_2	R_3
F_1	3	3	2
F_2	2	2	3

	V_1	V_2	V_3	U_1	U_2
W_1	3	2	1	3	2
W_2	1	2	3	2	3
V_1				3	1
V_2				3	2
V_3				2	3

	Y_1	Y_2	Z_1	Z_2
X_1	2	2	3	2
X_2	3	3	2	3
Y_1			3	2
Y_2			3	3

12.2.2 Составные решения

Получены следующие Парето эффективные проектные альтернативы DAs:

(1) для части A (Рис. 12.5 иллюстрирует качество E):

$D_1 = G_1 \star H_1 \star I_1$, $N(D_1) = (3; 1, 3, 0)$;

$E_1 = J_2 \star K_1 \star L_1$, $N(E_1) = (3; 1, 2, 1)$; $E_2 = J_1 \star K_2 \star L_3$, $N(E_2) = (1; 2, 2, 0)$;

$A_1 = D_1 \star E_1(3; 1, 3, 0)$, $A_2 = D_1 \star E_2(1; 2, 2, 0)$.

(2) для части B:

$\Phi_1 = R_1 \star F_1$, $N(\Phi_1) = (3; 2, 2, 0)$;

$M_1 = O_2 \star P_1$, $N(M_1) = (3; 1, 3, 0)$; $M_2 = O_3 \star P_1$, $N(M_2) = (2; 2, 2, 0)$;

$B_1 = \Phi_1 \star M_1(3; 2, 2, 0)$.

(3) для части C:

$Q_1 = W_1 \star V_1 \star U_1$, $N(Q_1) = (3; 1, 3, 0)$; $Q_2 = W_2 \star V_1 \star U_1$, $N(Q_2) = (1; 3, 1, 0)$;

$T_1 = X_2 \star Y_2 \star Z_2$, $N(T_1) = (3; 1, 2, 1)$;

$C_1 = Q_1 \star T_1(3; 1, 3, 0)$, $C_2 = Q_2 \star T_1(1; 2, 2, 0)$.

Таким образом, получены следующие четыре составные решения (как комбинации):

$S_1 = A_1 \star B_1 \star C_1$, $S_2 = A_1 \star B_1 \star C_2$, $S_3 = A_2 \star B_1 \star C_1$, $S_4 = A_2 \star B_1 \star C_2$,

Следует заметить, что исходное множество комбинаций включает 1179648 возможных решений (т.е., $(2 \times 2 \times 4) \times (2 \times 3 \times 4) \times (4 \times 2) \times (3 \times 2) \times (2 \times 2 \times 2) \times (2 \times 2 \times 2)$).

12.2.3 Анализ и улучшение узких мест

Таблица 12.3 содержит примеры улучшения: узкие места (по элементам, по совместимости между элементами) и операции улучшения для составных решений (для компонента E).

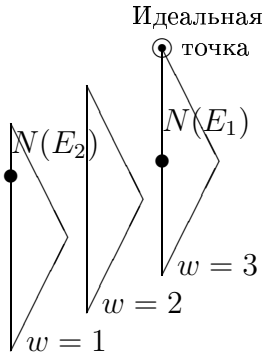


Рис. 12.5. Качество E

Таблица 12.3. Узкие места, операции улучшения

Составые DAs	Узкие места		Операции улучшения w/r
	DAs	IC	
$E_1 = J_2 \star K_1 \star L_1$	L_1		$(1, 2, 1) \Rightarrow (4, 0, 0)$
$E_1 = J_2 \star K_1 \star L_1$	J_2		$(1, 1, 2) \Rightarrow (4, 0, 0)$
$E_2 = J_1 \star K_2 \star L_3$	J_1		$(1, 3, 0) \Rightarrow (4, 0, 0)$
$E_2 = J_1 \star K_2 \star L_3$	K_1		$(3, 1, 0) \Rightarrow (4, 0, 0)$
$E_2 = J_1 \star K_2 \star L_3$		(J_1, L_3)	$1 \Rightarrow 3$
$E_2 = J_1 \star K_2 \star L_3$		(J_1, K_2)	$1 \Rightarrow 3$
$E_2 = J_1 \star K_2 \star L_3$		(K_2, L_3)	$1 \Rightarrow 3$

12.3 Пример для системы автоматизации здания

12.3.1 Структура системы и проектирование

В качестве примера рассматривается следующая иерархическая структура (Рис. 12.6; интервальные оценки в виде мультимножеств представлены в круглых скобках):

Система. Пример системы автоматизации здания $S = G \star Q \star T \star I$.
Подсистема 1. IP/KNX Gateway G: Gira HS3 G_1, EibPort G_2, EibPC G_3.
Подсистема 2. IP/WSN 6LoWPAN Gateway Q: Sensinode Nanorouter Q_1, Jennic Border-Router Q_2, Arch Rock PhyNet Router Q_3, Crossbow Ehernet Gateway (MIB600) Q_4.
Подсистема 3. ZigBee Wireless Sensor Network T: TI T_1, Ember T_2, Jennic T_3, Freescale T_4, Microchip T_5.
Подсистема 4. KNX Field Bus Infrastructure I: ABB I_1, Siemens I_2, SCHNEIDER Electric I_3, BOSCH I_4.

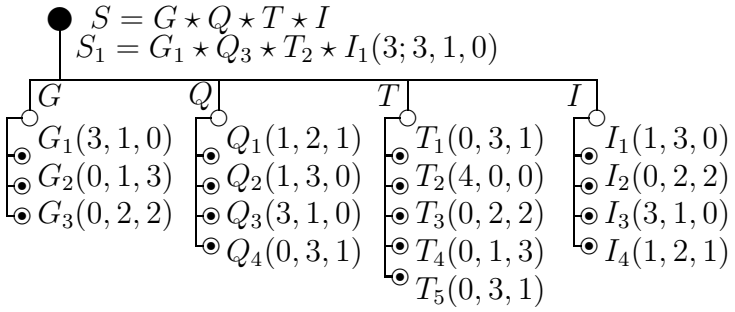


Рис. 12.6. Иерархическая структура системы

Использованные оценки в виде мультимножеств основаны на нескольких критериях (надежность, производительность, масштабируемость, техническое обслуживание, стоимость [398]). Оценки совместимости между DAs представлены в Таблице 12.4 (порядковая шкала, экспертное оценивание). Получено следующее Парето эффективное составное проектное решение: $S_1 = G_1 \star Q_3 \star T_2 \star I_1$, $N(S_1) = (3;3,1,0)$. Рис. 12.7 иллюстрирует качество указанного результирующего решения.

Таблица 12.4. Совместимость

	Q_1	Q_2	Q_3	Q_4	T_1	T_2	T_3	T_4	T_5	I_1	I_2	I_3	I_4
G_1	3	3	3	3	2	3	3	3	2	3	3	2	3
G_2	2	2	3	3	3	3	3	2	2	3	2	2	3
G_3	2	2	3	3	3	3	2	2	3	3	2	2	3
Q_1					2	3	3	2	2	3	2	2	3
Q_2					2	3	3	2	2	3	2	2	3
Q_3					2	3	3	2	2	3	3	2	3
Q_4					2	3	3	2	2	3	2	2	3
T_1										3	2	2	3
T_2										3	3	2	3
T_3										3	2	2	3
T_4										3	2	2	3
T_5										3	2	2	3

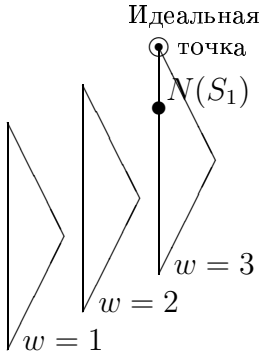


Рис. 12.7. Качество S

12.3.2 Расширение системы

Расширение системы реализуется как добавление пятой подсистемы HomeServer SW H: Domovea SW H_1, Eisbaer SW H_2, Misterhouse SW H_3, and Eibd-linKNX-KNXweb H_4. Расширенная структура системы представлена на Рис. 12.8. Таблица 12.5 содержит оценки совместимости (порядковая шкала, экспертное оценивание).

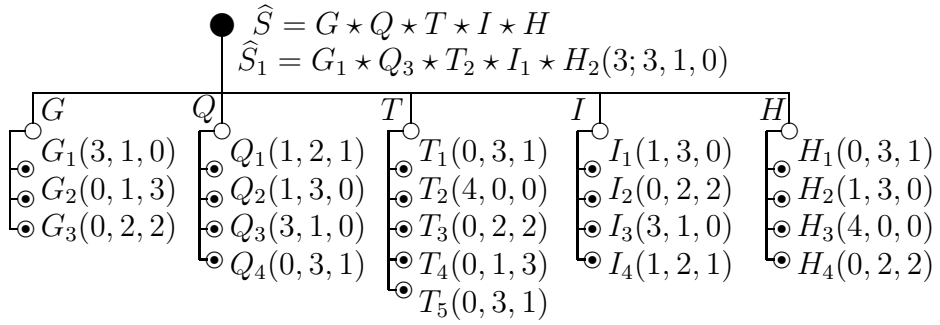


Рис. 12.8. Расширенная иерархическая структура системы

Таблица 12.5. Совместимость

	G_1	G_2	G_3	Q_1	Q_2	Q_3	Q_4	T_1	T_2	T_3	T_4	T_5	I_1	I_2	I_3	I_4
H_1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
H_2	3	1	1	1	1	3	1	1	3	1	1	1	3	3	1	2
H_3	3	1	1	1	1	3	1	1	3	1	1	1	2	1	1	1
H_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

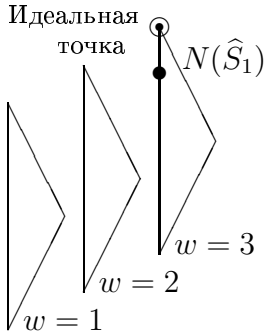


Рис. 12.9. Качество $\widehat{S}$

Получено следующее результирующее Парето эффективное составное решение: $\widehat{S}_1 = G_1 \star Q_3 \star T_2 \star I_2 \star H_2,\ N(\widehat{S}_1) = (3;3,1,0)$.

Рис. 12.9 иллюстрирует полученное составное проектное решение. Следует заметить, что решение $\widehat{S}_1$ является расширением составного проектного решения S_1 (стратегия 2).

12.3.3 Улучшение частей системы

Улучшение компонентов

Таблица 12.6 иллюстрирует процесс улучшения на основе компонентов решения (Рис. 12.6): S_1.

Таблица 12.6. Узкие места и операции улучшения

Составные DAs	Узкие места		Операции улучшения w/r
	DA	IC	
$S_1 = G_1 \star Q_3 \star T_2 \star I_1$	G_1		$(3,1,0) \Rightarrow (4,0,0)$
$S_1 = G_1 \star Q_3 \star T_2 \star I_1$	Q_3		$(3,1,0) \Rightarrow (4,0,0)$
$S_1 = G_1 \star Q_3 \star T_2 \star I_1$	I_1		$(1,3,0) \Rightarrow (4,0,0)$

Улучшение на основе задачи блочного рюкзака

Улучшение системы может быть реализовано как улучшение (upgrade) системных компонентов с учетом общего бюджетного ограничения. Такой подход основан на задаче блочного рюкзака. Здесь рассматривается исходное решения, включающее пять компонентов: $\overline{S} = G_2 \star Q_1 \star T_4 \star I_2 \star H_1$ (Рис. 12.10). Соответствующая задача блочного рюкзака имеет вид:

$$\max\ e(S) = \max\ M = \ \arg\min_{M \in D} \sum_{(i,j) \in S = \{(i,j) | x_{i,j} = 1\}} |\delta(M, e_{i,j})|,$$

$$s.t.\ \sum_{i=1}^{5} \sum_{j=1}^{q_i} a_{ij} x_{ij} \le b,\ \ \sum_{j=1}^{q_i} x_{ij} = 1,\ i = \overline{1,5};\quad x_{ij} \in \{0,1\}.$$

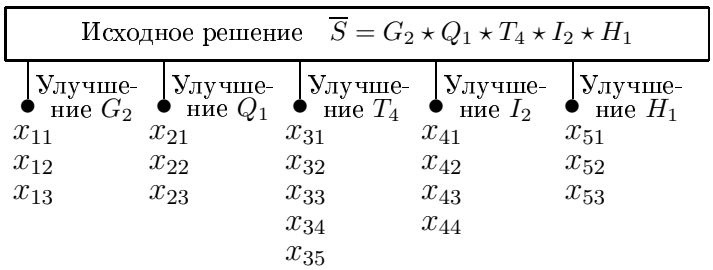


Рис. 12.10. Улучшение на основе задачи блочного рюкзака

Таблица 12.7 содержит бинарные переменные (x_{ij}), операции улучшения и их оценки (иллюстративные, экспертное оценивание). Приведем два примера решений:

(1) $b = 14$: $x_{32} = 1$ $(T_4 \to T_3)$, $\overline{S} \Rightarrow \widetilde{S}_1 = G_2 \star Q_1 \star T_3 \star I_2 \star H_1$, $e(\widetilde{S}_1) = (0, 2, 2)$;

(2) $b = 73$: $x_{13} = 1$ $(G_2 \to G_3)$, $x_{22} = 1$ $(Q_1 \to Q_2)$, $x_{33} = 1$ $(T_4 \to T_1)$, $x_{32} = 1$ $(I_2 \to I_4)$, $x_{52} = 1$ $(H_1 \to H_2)$, $\overline{S} \Rightarrow \widetilde{S}_2 = G_3 \star Q_2 \star T_1 \star I_4 \star H_2$, $e(\widetilde{S}_2) = (1, 2, 1)$.

Таблица 12.7. Оценки

Операции улучшения	Оценка e_{ij}	Цена (a_{ij})
x_{11} (нет)	$(0, 1, 3)$	0
x_{12} $(G_2 \to G_1)$	$(3, 1, 0)$	26
x_{13} $(G_2 \to G_3)$	$(0, 2, 2)$	16
x_{21} (нет)	$(1, 2, 1)$	0
x_{22} $(Q_1 \to Q_2)$	$(1, 3, 0)$	16
x_{23} $(Q_1 \to Q_3)$	$(3, 1, 0)$	20
x_{31} (нет)	$(0, 3, 1)$	0
x_{32} $(T_4 \to T_1)$	$(0, 3, 1)$	17
x_{33} $(T_4 \to T_2)$	$(4, 0, 0)$	28
x_{32} $(T_4 \to T_3)$	$(0, 2, 2)$	14
x_{33} $(T_4 \to T_5)$	$(0, 3, 1)$	17
x_{41} (нет)	$(0, 2, 2)$	0
x_{42} $(I_2 \to I_1)$	$(1, 3, 0)$	23
x_{43} $(I_2 \to I_3)$	$(3, 1, 0)$	27
x_{44} $(I_2 \to I_4)$	$(1, 2, 1)$	20
x_{51} (нет)	$(0, 3, 1)$	0
x_{52} $(H_1 \to H_2)$	$(1, 3, 0)$	21
x_{53} $(H_1 \to H_3)$	$(4, 0, 0)$	31

12.4 Выводы к главе

В данной главе представлены схемы модульного проектирования и улучшения/расширения для двух прикладных систем: управление умным домом, система автоматизации здания. Использованы две задачи комбинаторной оптимизации (морфологическая клика, задача блочного рюкзака). Данные оптимизационные модели базируются на интервальных оценках в виде мультимножеств (глава 4). Следует отметить, что представляется целесообразным применять и более сложные шкалы на основе задач оценивания, например: $P^{3,5}$, $P^{4,5}$. Можно указать следующие направления для исследований в будущем:

1. построение траекторий улучшения систем;

2. анализ рассмотренных задач проектирования и улучшения в режиме реального времени.

Глава 13

Приложения в системах связи

В данной главе [1] приведено описание семи примеров приложений в области коммуникационных систем: (а) задача последней мили (подключение пользователей к коммуникационным провайдерам), (б) модульное проектирование GSM-сети, (в) подключение пользователей к точкам доступа (на основе многокритериальной задачи о назначениях), (г) иерархическое моделирование и комбинаторная эволюция стандартов передачи мультимедийной информации (MPEG-1, MPEG-2, MPEG-4), (д) модульное проектирование протокола связи (ZigBee), (е) модульное проектирование стандартов передачи мультимедийной информации. (ж) комбинаторная эволюция и прогнозирование поколений беспроводных систем связи.

[1] Материал главы основан на публикациях:
(i) Levin M.Sh., Selection of user's connection in last mile problem. IEEE Trans. SMC - Part A, vol. 41, no. 2, pp. 370-374, 2011.
(ii) Levin M.Sh., Petukhov M.V., Connection of users with a telecommunications network: multicriteria assignment problem. J. of Commun. Technol. and Electr., vol. 55, no. 12, pp. 1532-1541, 2010.
(iii) M.Sh. Levin, Towards communication network development (structural system issues, combinatorial problems). *IEEE Region 8 Int. Conf. 'Sibircon 2010'*, vol. 1, pp. 204-208, 2010.
(iv) Levin M.Sh., Kruchkov O., Hadar O., Kaminsky E., Combinatorial systems evolution: Example of standard for multimedia information. Informatica (Lith.), vol. 20, no. 4, pp. 519-538, 2009.
(v) Levin M.Sh., Morphological methods for design of modular systems (a survey). Electr. prepr., 20 p., Jan. 9, 2012. http://arxiv.org/abs/1201.1712 [cs.SE]
(vi) Levin M.Sh., Improvement/extension of modular systems as combinatorial reengineering (survey). Electr. prepr., 24 p., Apr. 17, 2013. Http://arxiv.org/abs/1304.4965 [cs.AI]
(vii) Levin M.Sh., A modular approach to the communication protocol and standard for multimedia information: A review. J. of Communications Technology and Electronics, vol. 58, no. 6, 594-601, 2013.
(viii) Левин М.Ш. Модульный подход к коммуникационному протоколу и стандарту для мультимедийной информации. "Информационные процессы", 2012, вып. 12, ном. 4, С. 389-399.
(ix) Levin M.Sh. Note on evolution and forecasting of requirements: communication example. Electr. prepr., 8 p., May 22, 2016, http://arxiv.org/abs/1705.07558 [cs.NI]
(x) Levin M.Sh. Towards combinatorial modeling of wireless technology generations. Electr. prepr., 20 p., Aug. 29, 2017, http://arxiv.org/abs/1708.08996 [cs.NI]
(xi) Levin M.Sh. An example of combinatorial evolution and forecasting of requirements to communication systems. J. of Commun. Technol. Electr., vol. 62, no. 12, pp. 1499–1505, 2017.
(xii) Levin M.Sh. On combinatorial models of generations of wireless communication systems. J. of Commun. Technol. Electr., vol. 63, no. 6, pp. 667–679, 2018.
(xiii) Левин М.Ш. Пример комбинаторной эволюции и прогнозирования требования к коммуникационных системам. "Информационные процессы", 2017, вып. 17, ном. 2, С. 92-100.
(xiv) Левин М.Ш. О комбинаторных моделях поколений беспроводных систем связи. "Информационные процессы", 2017, вып. 17, ном. 3, С. 172-187.

13.1 Задача 'последняя миля'

Задача 'последняя миля' заключается в проектировании системы обслуживания (связи) домашних пользователей компьютеров провайдеров (фирмой - коммуникационным оператором) [160, 270, 344, 480]. При этом представляется целесообразным учитывать требования с двух сторон: (i) выгодность (полезность, прибыльность) для коммуникационного оператора и (ii) высокий уровень, качество и стоимость обслуживания для пользователей.

В данном разделе описывается подход к задаче 'последняя миля' на основе ИММП с учетом требований коммуникационного оператора [344, 356]. Рис. 13.1 иллюстрирует исследуемую задачу на примере: (a) коммуникационный оператор и (b) пользователи некоторого региона. В нашем случае рассматриваются следующие группы пользователей: (i) группа *компании* $A = E \star F$; (ii) группа *образовательные учреждения* $B = I \star J \star M$; (iii) группа *государственные организации* $C = O \star P \star R$; (iv) группа *производственные организации / заводы* $D = Y \star Z$. Таким образом, структура составного решения имеет вид (Рис. 13.2): $S = A \star B \star C \star D$.

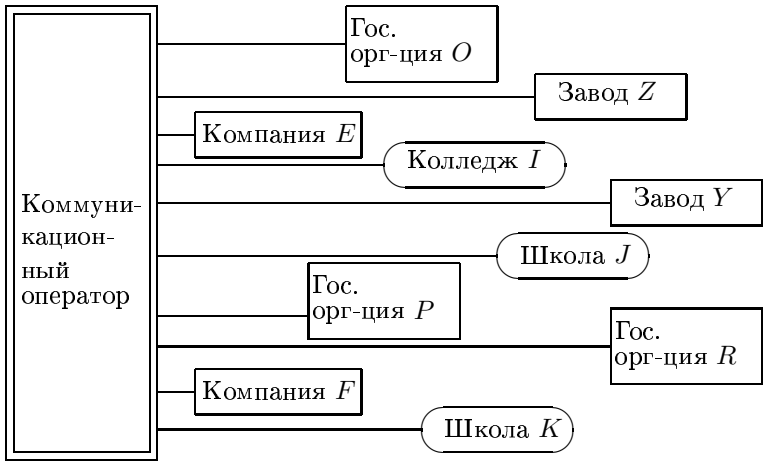


Рис 13.1. Иллюстративная схема

Далее, для каждого пользователя генерируем множество возможных альтернатив (проектных вариантов соединения, т.е., DAs) и вычисляем их приоритеты на основе критериев и оценок по критериям.

В нашем примере, рассматриваются следующее общее базовое множество DAs:

X_1 (ADSL): телефонная сеть,

X_2 (DACSIS): сеть кабельного телевидения,

X_3 Ethernet,

X_4 энергетическая сеть,

X_5 (FTTH): оптический кабель,

X_6 (LMDS): беспроводная связь (2.5–2.7 GHz),

X_7 (4G): беспроводная связь(800–2000 MHz), and

X_8 (WMAN, WiMAX): беспроводная связь (2.4, 3.5, 5.8 GHz).

Обычно рассматривается следующее множество основных критериев для многокритериального выбора вариантов связи: объем исходных инвестиций, стоимость оборудования, скорость передачи информации, время инсталяции системы связи (провайдеров), время инсталяции для пользователя, стоимость технического обслуживания, надежность, мобильность услуг, близость к существующей коммуникационной среде. В нашем исследовании для каждого пользователя рассматривается некоторое исходное множество вариантов (выбранных DAs) и их приоритеты (экспертные суждения) (Рис. 13.2). Таблица 13.1

и Таблица 13.2 содержат оценки совместимости для DAs (также на основе экспертных суждений).

В результате, получаем следующие результирующие составные решения:

1. $A_1 = E_7 \star F_7$, $N(A_1) = (3; 2, 0)$.

2. $B_1 = I_8 \star J_2 \star M_1$, $N(B_1) = (1; 3, 0, 0)$; $B_2 = I_8 \star J_1 \star M_1$, $N(B_2) = (1; 3, 0, 0)$; $B_3 = I_8 \star J_8 \star M_8$, $N(B_3) = (3; 1, 2, 0)$.

3. $C_1 = O_8 \star P_8 \star R_8$, $N(C_1) = (3; 2, 1, 0)$; $C_2 = O_7 \star P_7 \star R_7$, $N(C_2) = (3; 2, 1, 0)$; $C_3 = O_7 \star P_5 \star R_7$, $N(C_3) = (2; 3, 0, 0)$; $C_4 = O_8 \star P_5 \star R_7$, $N(C_4) = (2; 3, 0, 0)$; $C_5 = O_7 \star P_5 \star R_8$, $N(C_5) = (2; 3, 0, 0)$; $C_6 = O_8 \star P_5 \star R_8$, $N(C_6) = (2; 3, 0, 0)$.

4. $D_1 = Y_3 \star Z_5$, $N(D_1) = (2; 2, 0)$; $D_2 = Y_3 \star Z_8$, $N(D_2) = (2; 2, 0)$; $D_3 = Y_4 \star Z_4$, $N(D_3) = (3; 1, 1)$.

Далее имеется два пути:

1. Исследовать 54 возможных результирующих составных решения:

$S_1 = A_1 \star B_1 \star C_1 \star D_1$, $S_2 = A_1 \star B_1 \star C_2 \star D_1$, *и т.д.*

2. Сформулировать и решить дополнительную задачу морфологической клики на более высоком иерархическом уровне нашей модели для композиции компонентов A, B, C, and D с учетом соответствующих приоритетов DAs для A, B, C, and D и оценок их совместимости.

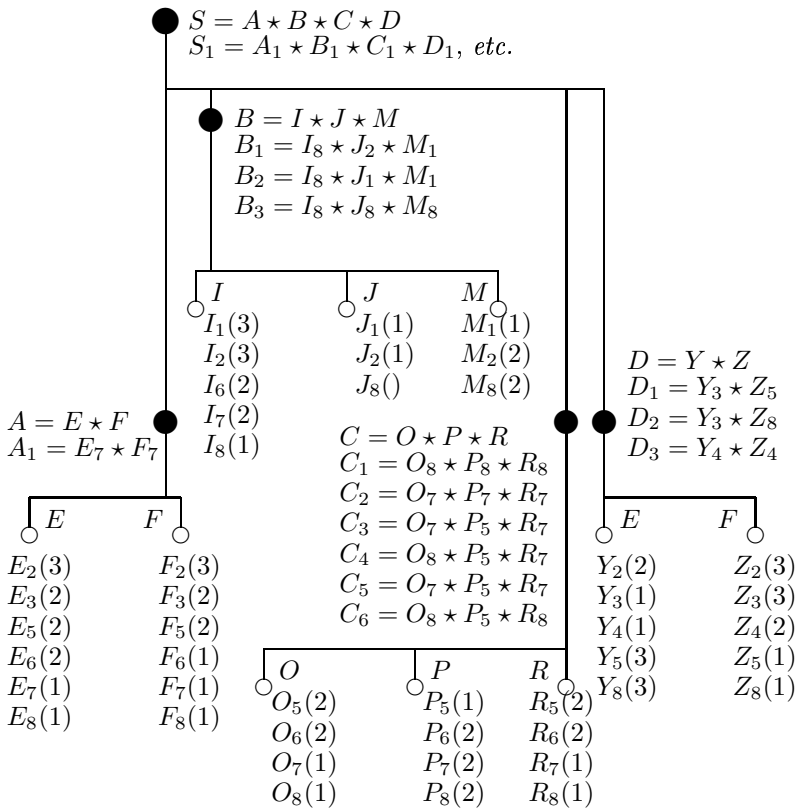


Рис 13.2. Структура составного решения

Таблица 13.1. Совместимость

	F_2	F_3	F_5	F_6	F_7	F_8
E_2	3	2	2	2	2	2
E_3	1	3	2	2	2	2
E_5	2	2	3	2	2	2
E_6	2	2	2	3	2	2
E_7	2	2	2	2	3	2
E_8	2	2	2	2	2	3

	J_1	J_2	J_8	M_1	M_2	M_8
I_1	3	2	1	3	2	1
I_2	2	3	2	2	3	2
I_6	1	2	2	1	2	2
I_7	1	2	2	3	2	2
I_8	1	2	3	1	2	3
J_1				3	2	1
J_2				2	3	2
J_8				1	2	3

	P_1	P_2	P_8	P_1	R_2	R_8	R_2	R_8
O_5	3	2	2	2	3	2	2	2
O_6	2	3	2	2	2	3	2	2
O_7	2	2	3	2	2	2	3	2
O_8	2	2	2	3	2	2	2	3
P_5					3	2	2	2
P_6					2	3	2	2
P_7					3	2	3	2
P_8					2	2	2	3

Таблица 13.2. Совместимость

	Z_2	Z_3	Z_4	Z_5	Z_8
Y_2	3	2	1	2	2
Y_3	2	3	1	2	2
Y_4	1	1	3	1	1
Y_5	2	2	1	3	3
Y_8	2	2	1	2	3

13.2 Модульное проектирование GSM сети

В настоящее время GSM сети являются наиболее широко используемыми [129, 215, 240]. Упрощенная древовидная модель GSM сети представлена на Рис. 13.3:

0. GSM сеть $S = A \star B$.
1. Переключающая подсистема (Switching SubSystem) SSS ($A = M \star L$).
1.1. Мобильный центр переключений/Регистр гостевых размещений (Mobile Switching Center/Visitors Location Register) MSC/VLR M : M_1 (Motorola), M_2 (Alcatel), M_3 (Huawei), M_4 (Siemens), M_5 (Ericsson).
1.2. Регистр домашнего размещения/Центр идентификации (Home Location Register/Authentification Center) HLR/AC L : L_1 (Motorola), L_2 (Ericsson), L_3 (Alcatel), L_4 (Huawei).
2. Базовая станция BSS ($B = V \star U \star T$).
2.1. Контроллер базовой станции (Base Station Controller) BSC V : V_1 (Motorola), V_2 (Ericsson), V_3 (Alcatel), V_4 (Huawei), V_5 (Nokia), V_6 (Siemens).
2.2. Приемо-передатчик базовой станции (Base Transceiver Station) BTS U : U_1 (Motorola), U_2 (Ericsson), U_3 (Alcatel), U_4 (Huawei), U_5 (Nokia).
2.3. Приемо-передатчики (Transceivers) TRx T : T_1 (Alcatel), T_2 (Ericsson), T_3 (Motorola), T_4 (Huawei), T_5 (Siemens).

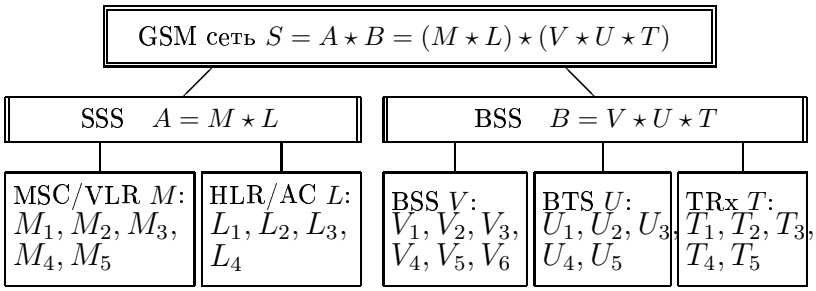


Рис. 13.3. Упрощенная структура GSM сети

Число исходных возможных системных комбинаций равно 3000 ($5 \times 4 \times 6 \times 5 \times 5$). Для оценки компонентов системы используются следующие критерии (веса важности критериев указаны в скобках):

1. M: максимальное число путей данных (1000 pathes) (C_{m1}, 0.2); максимальная нагрузка VLR (100000 подписчиков) (C_{m2}, 0.2); индекс цены (100000/цена (USD)) (C_{m3}, 0.2); энергетические потребности (1/(kWt)) (C_{m4}, 0.2); число коммуникаций и сигнализирующих интерфейсов (C_{m5}, 0.2).

2. L: максимальное число подписчиков (100000 подписчиков) (C_{l1}, 0.25); обеспечиваемый объем сервиса (C_{l2}, 0.25); надежность (шкала $[1, ..., 10]$) (C_{l3}, 0.25); интегрируемость с другими системами (шкала $[1, ..., 10]$) (C_{l4}, 0.25).

3. V: индекс цены (100000/цена (USD)) (C_{v1}, 0.25); максимальное число BTS (C_{v2}, 0.25); качество переключения каналов (C_{v3}, 0.25); пропускная способность (C_{v4}, 0.25).

4. U: максимальное число TRx (C_{u1}, 0.25); нагрузка (C_{u2}, 0.25); индекс цены (100000/цена (USD)) (C_{u3}, 0.25); надежность (шкала $[1, ..., 10]$) (C_{u4}, 0.25).

5. T: максимум энергетической текущей нагрузки (C_{t1}, 0.3); пропускная способность (C_{t2}, 0.2); индекс цены (100000/ цена (USD)) (C_{t3}, 0.25); надежность (шкала $[1, ..., 10]$) (C_{t4}, 0.25).

Оценки DAs приведены в Таблице 13.3, Таблице 13.4, Таблице 13.5 (использованы данные из каталогов и экспертное оценивание). В результате многокритериального ранжирования (модифицированная версия метода Electre). получаются приоритеты для DAs (указаны в таблицах). Экспертные оценки совместимости DAs представлены в Таблице 13.6.

Таблица 13.3. Оценки для M

DAs	Критерии					Приоритет r
	C_{m1}	C_{m2}	C_{m3}	C_{m4}	C_{m5}	
M_1	3.7	8.6	6	5.1	4	2
M_2	4.0	11	8	7	5	3
M_3	4.1	10	9	7	4	3
M_4	3.2	7	5	6	3	1
M_5	3.5	8.7	6.2	5	4	2

Таблица 13.4. Оценки для V, L

DAs	Критерии				Приоритет r
	C_{v1}	C_{v2}	C_{v3}	C_{v4}	
V_1	6	4	3	4	1
V_2	7	5	7	7	2
V_3	9	7	10	7	3
V_4	7	5	8	6	2
V_5	6	3	4	4	1
V_6	10	6	9	7	3

DAs	Критерии				Приоритет r
	C_{l1}	C_{l2}	C_{l3}	C_{l4}	
L_1	9	7	7	8	1
L_2	10	4	9	8	1
L_3	12	8	10	10	2
L_4	9	5	8	8	1

Таблица 13.5. Оценки для U, T

DAs	Критерии				Приоритет r
	C_{u1}	C_{u2}	C_{u3}	C_{u4}	
U_1	2	7	5	8	1
U_2	4	10	6	10	3
U_3	3	9	6	10	2
U_4	3	6	3	7	1
U_5	3	10	6	9	2

DAs	Критерии				Приоритет r
	C_{t1}	C_{t2}	C_{t3}	C_{t4}	
T_1	9	7	10	7	3
T_2	6	4	3	4	1
T_3	7	5	7	7	2
T_4	7	5	8	6	2
T_5	6	3	4	4	1

Таблица 13.6. Совместимость

	U_1	U_2	U_3	U_4	U_5	T_1	T_2	T_3	T_4	T_5
V_1	2	2	2	2	3	3	2	2	2	2
V_2	3	3	3	2	0	0	3	0	3	2
V_3	3	3	3	2	0	0	3	0	3	2
V_4	3	2	0	2	3	0	2	0	2	2
V_5	3	0	0	2	0	2	2	0	2	2
V_6	0	3	2	3	2	3	0	2	2	0
U_1						2	0	0	2	3
U_2						0	2	0	3	0
U_3						0	2	0	3	0
U_4						0	3	3	0	0
U_5						3	0	2	2	0

	L_1	L_2	L_3	L_4
M_1	3	2	0	3
M_2	2	3	2	0
M_3	0	2	3	2
M_4	2	3	3	3
M_5	3	3	0	3

Предварительная версия рассматриваемого примера была опубликована в [388].

Для подсистемы A, следующие Парето-эффективые составные DAs получены: (1) $A_1 = M_4 \star L_2$, $N(A_1) = (3; 2, 0, 0)$; (2) $A_2 = M_4 \star L_4$, $N(A_2) = (3; 2, 0, 0)$.

Для подсистемы B, следующие Парето-эффективые составные DAs получены: (1) $B_1 = V_5 \star U_1 \star T_5$, $N(B_1) = (2; 3, 0, 0)$; (2) $B_2 = V_5 \star U_4 \star T_2$, $N(B_2) = (2; 3, 0, 0)$; (3) $B_3 = V_1 \star U_5 \star T_1$, $N(B_3) = (3; 1, 1, 1)$, and (4) $B_4 = V_2 \star U_3 \star T_4$, $N(B_4) = (3; 0, 3, 0)$. Рис. 13.4 иллюстрирует качество решений для B.

Теперь получаются следующие результирующие решения для системы:

(1) $S_1^H = A_1 \star B_1 = (M_4 \star L_2) \star (V_5 \star U_1 \star T_5)$; (2) $S_2^H = A_1 \star B_2 = (M_4 \star L_2) \star (V_5 \star U_4 \star T_2)$;

(3) $S_3^H = A_1 \star B_3 = (M_4 \star L_2) \star (V_1 \star U_5 \star T_1)$; (4) $S_4^H = A_2 \star B_1 = (M_4 \star L_4) \star (V_5 \star U_1 \star T_5)$;

(5) $S_5^H = A_2 \star B_2 = (M_4 \star L_4) \star (V_5 \star U_4 \star T_2)$; (6) $S_6^H = A_2 \star B_3 = (M_4 \star L_4) \star (V_1 \star U_5 \star T_1)$;

(7) $S_7^H = A_1 \star B_3 = (M_4 \star L_2) \star (V_2 \star U_3 \star T_4)$; (8) $S_8^H = A_2 \star B_3 = (M_4 \star L_4) \star (V_2 \star U_3 \star T_4)$.

Чтобы получить интегрированную оценку качества системных решений, надо интегрировать оценки для частей A и B: $N(S_1^H) = (2; 5, 0, 0)$, $N(S_2^H) = (2; 5, 0, 0)$, $N(S_3^H) = (3; 3, 1, 1)$, $N(S_4^H) = (2; 5, 0, 0)$, $N(S_5^H) = (3; 3, 1, 1)$, $N(S_6^H) = (3; 3, 1, 1)$, $N(S_7^H) = (3; 2, 3, 0)$, $N(S_8^H) = (3; 2, 3, 0)$.

Далее, полученные системные решения могут анализироваться с целью выбора наилучшего решения (например, дополнительный многокритериальный анализ, экспертные процедуры).

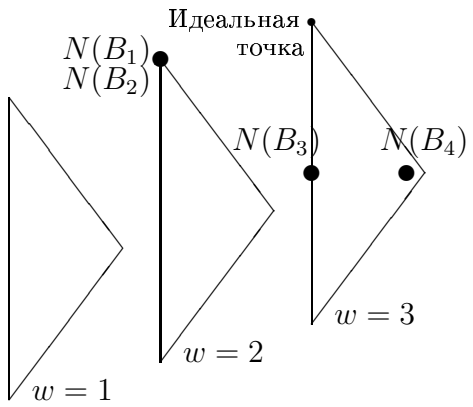


Рис. 13.4. Качество для B

13.3 Подключение пользователей к точкам доступа

13.3.1 Предварительные замечания

В последние годы возрастает важность задач подключения пользователей к беспроводным телекоммуникационным сетям [270, 344, 356]. В данной статье впервые рассматривается подключение пользователей к точкам доступа беспроводных телекоммуникационной сети на основе многокритериальной задачи о назначении. Рис. 13.5 иллюстрирует процесс подключения пользователей к точкам доступа сети.

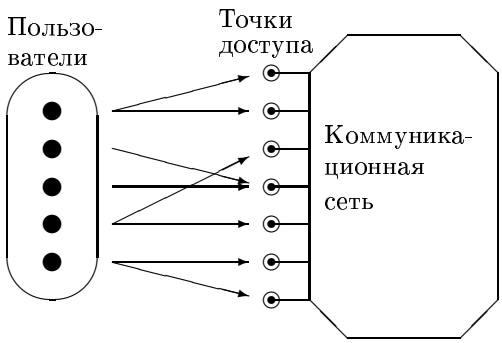


Рис. 13.5. Пользователи и точки доступа

В рамках предлагаемого многокритериального подхода используются следующие критерии: (а) максимум полосы пропускания, (б) число одновременно обслуживаемых пользователей, (в) требования надежности обслуживания и др. Многокритериальные задачи о назначении относятся к классу NP-трудных, поэтому предлагаются эвристические схемы решения. Вычисления проведены на основе программ, разработанных в среде MatLab (http://www.mathworks.com/).

13.3.2 Постановка задачи

В данной статье использованы следующие предположения:

1. Реальной территории (некой "области") соответствует карта местности, которая отражает рельеф.

2. Точки доступа располагаются на местности (т.е., на карте) и "покрывают" всю область.

3. Пользователи (клиенты) могут располагаться на территории произвольным (случайным) образом.

4. Для каждого клиента необходим доступ к сети, т.е., соответствующее подключение к одной (или хотя бы к одной в случае "расширенной" задачи подключения) точке доступа и каждое такое подключение описывается (оценивается) набором характеристик (критериев).

5. Точки доступа и клиенты имеют координаты на карте с учетом рельефа, т.е., (x, y, z).

Содержательная постановка задачи заключается в следующем:

Подключить максимальное число пользователей к точкам доступа телекоммуникационной сети так, чтобы "обобщенная" надежность соединений (т.е., передачи данных для всех пользователей) была максимальна и "обобщенная" длина связей была минимальна, и удовлетворялись все ограничения по требованиям пользователей (по качеству обслуживания): (а) по полосам пропускания, (б) по уровню защищенности данных.

Рассмотрим формальную постановку задачи. Обозначим множество пользователей $\Psi = \{1, ..., i, ..., n\}$ и множество точек доступа $\Theta = \{1, ..., j, ..., m\}$.

Каждый пользователь i описывается вектором: $(x_i, y_i, z_i, f_i, p_i, r_i, d_i)$, где (x_i, y_i, x_i) - координаты пользователя i, f_i - параметр, соответствующий требуемой полосе пропускания (от 1 Mbit/s до 10 Mbit/s), p_i - требуемый уровень CoS (класс, приоритет) обслуживания, r_i - требуемая для пользователя надежности передачи данных, d_i - уровень шифрования (т.е., уровень защиты информации).

Каждая точка доступа описывается так: $(x_j, y_j, z_j, f_j, n_j, r_j, d_j)$, где (x_j, y_j, x_j) - координаты точки доступа j, f_j - параметр, соответствующий максимально возможному трафику (т.е., максимум полосы пропускания) точки доступа j, n_j - максимальное число пользователей, которое может обслуживаться точкой доступа j, r_j - надежность канала передачи

данных, которая может обеспечиваться точкой доступа j, d_j - параметр, отвечающий за уровень защиты от взлома для данных, передаваемых через точку доступа j. Таблица 13.7 содержит описание шкал для используемых параметров.

Для каждой пары "пользователь-точка доступа" $(\forall(i,j), i \in \Psi, j \in \Theta)$ имеется набор характеристик: (1) уровень надежности r_{ij}, (2) характеристика расстояния и наличия препятствий (качество направления распространения сигнала) β_{ij}, (3) характеристика использования полосы пропускания f_{ij}, (4) класс обслуживания (приоритет) p_{ij}, (5) характеристика уровня защиты передачи данных d_{ij}. Таким образом получаем векторную характеристику $\forall(i,j)$, $i \in \Psi$, $j \in \Theta$: $\widehat{c_{ij}} = (r_{ij}, \beta_{ij}, f_{ij}, p_{ij}, d_{ij})$.

Подключение пользователя i к точке доступа j будет задаваться булевой переменной x_{ij} ($x_{ij} = 1$ при подключении i к j и $x_{ij} = 0$ в противном случае). Вариант подключения множества пользователей Ψ к множеству точек доступа Θ (т.е., решение задачи по подключению пользователей к точкам доступа) определяется набором значений булевых переменных (булевой матрицей) $X = ||x_{ij}||$, $i = \overline{1,n}$, $j = \overline{1,m}$.

Рассмотрим правила расчета характеристик для подключения пользователя i $(\forall i \in \Psi)$ доступа j $(\forall i \in \Theta)$:

(1) Надежность: $r_{ij} = \min\{r_i, r_j\}$.

(2) Расстояние: l_{ij}.

(3) Признак барьера: $e_{ij} = \begin{cases} 1, & \text{имеется препятствие/барьер,} \\ 0, & \text{препятствия нет.} \end{cases}$

(4) Интегрированный параметр, объединяющий барьер и расстояние ($L_{max} = \max_{\{(i,j)\}} l_{ij}$):

$$\beta_{ij} = \begin{cases} 0, & \text{если } ((l_{ij} > L_{max}/2)\&(e_{ij} = 1)) \text{ ,} \\ 5, & \text{если } ((l_{ij} < L_{max}/2)\&(e_{ij} = 1)) \text{ или } ((L_{max} \geq l_{ij} > L_{max}/2)\&(e_{ij} = 0)), \\ 10, & \text{если } ((l_{ij} < L_{max}/2)\&(e_{ij} = 0)). \end{cases}$$

Признак "связности" по β_{ij}: $\xi_{ij}^{\beta} = \begin{cases} 0, & \text{если } \beta_{ij} = 0, \\ 1, & \text{иначе.} \end{cases}$

(5) Уровень класса обслуживания (приоритет): $p_{ij} = p_i$.

(6) Требуемая/возможная полоса пропускания для соединения: f_{ij} (на начальной стадии $f_{ij} = f_i$). Рассматриваются три случая:

(а) при $p_{ij} = 1$: $f_{ij} = f_i$,

(б) при $p_{ij} = 2$: $f_{ij} = \begin{cases} f_i, & \text{если } (\max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j) \geq f_i, \\ \max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j, & \text{если } (\max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j) < f_i, \end{cases}$

(в) при $p_{ij} = 3$: $f_{ij} = \begin{cases} f_i, & \text{если } (\max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j) \geq f_i, \\ \max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j, & \text{если } (\max_j f_j - \frac{1}{m}\sum_{j=1}^{m} f_j) < f_i; \end{cases}$

Таблица 13.7. Параметры и описание шкал

Параметр	Шкала	Описание
p_i (пользователь)	$[1,3]$	при значении 1 все требования клиента должны удовлетворяться, соответствующие пользователи обслуживаются в первую очередь; при значении 2 возможно использование полосы пропускания в два раза уже, чем заказано пользователем, при большой загруженности сети требование пользователя рассматривается в вторую очередь, возможна пониженная надежность передачи данных; при значении 3 возможно подключение пользователя при любой доступной полосе пропускания.
r_i (пользователь)	$[1,10]$	при значении 1 возможна потеря информация пользователя (до 20 %, пример: просмотр видео-трансляций), при значении 10 потеря информация пользователя исключается (у пользователя максимальные требования по надежности)
r_j (точка доступа)	$[1,10]$	при значении 1 возможна потеря информация (до 20 %), при значении 10 потеря информация пользователя исключается.
d_i (пользователь)	$[1,10]$	при значении 1 информация пользователя является несекретной, значение 5 соответствует среднему уровню требования по защите при значении 10 информация является конфиденциальной,
d_j (точка доступа)	$[1,10]$	при значении 1 точка доступа имеет тривиальные меры защиты, при значении 10 точка доступа имеет самые строгие меры защиты данных от взлома.

Здесь учитывается то, что трафик может быть двух видов: (i) *elastic* (сервис пользователя использует только доступную полосу пропускания), (ii) *non-elastic* (пользователь имеет требования к определенной полосе пропускания для его сервисов).

(7) Показатель уровня защиты передачи данных : d_{ij}. Рассматриваются три случая:

(а) при $p_{ij}=1$: $d_{ij}=\begin{cases} d_j, & \text{если } d_j \geq d_i, \\ 0, & \text{если } d_j < d_i; \end{cases}$

(б) при $p_{ij}=2$: $d_{ij}=\begin{cases} d_j, & \text{если } d_j \geq d_i/2, \\ 0, & \text{если } d_j < d_i/2; \end{cases}$

(в) при $p_{ij}=3$: $d_{ij}=d_j$.

Признак “связности” по d_{ij}: $\xi^d_{ij}=\begin{cases} 0, & \text{если } d_{ij}=0, \\ 1, & \text{иначе.} \end{cases}$

Дополнительно для каждого пользователя определяется (на основе ξ^β_{ij}, ξ^d_{ij}) подмножество возможных для подключения точек доступа: $\forall i \in \Psi$ выделяется $\Theta_i \subseteq \Theta$. Очевидно, что при $|\Theta_i|=0$, пользователь i может быть исключен из дальнейшего рассмотрения. На Рис. 13.6 иллюстрирована схема обработки исходных данных.

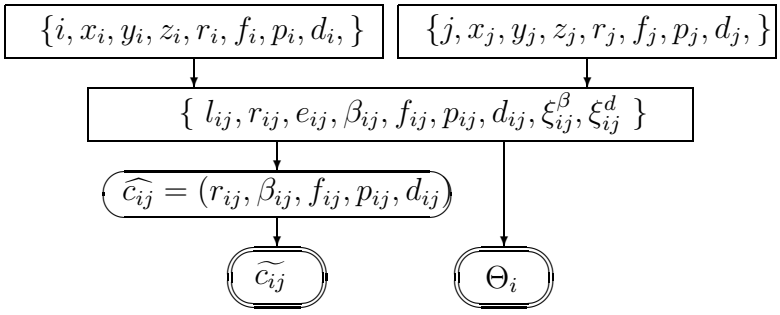


Fig. 13.6. Схема обработки данных

Будем использовать следующие набор обобщенных характеристик решения задачи (некий упрощенный аддитивный вариант для используемых критериев):

(а) обобщенная надежность $R(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij}\, x_{ij}$,

(б) обобщенный параметр качества распространения сигнала $B(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_{ij}\, x_{ij}$,

(в) обобщенное качество использования частотного спектра $F(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}\, x_{ij}$,

(г) обобщенная характеристика обслуживания $P(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}\, x_{ij}$,

(д) обобщенная характеристика уровня защиты передачи $D(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij}\, x_{ij}$.

В результате получаем векторную характеристику решения X:

$\overline{C(X)} = (R(X), B(X), F(X), P(X), D(X))$.

Далее вводятся ограничения:

(1) Ограничение на полосу пропускания точки доступа j: $\sum_{i=1}^{n} f_{ij}\, x_{ij} \le f_j \quad \forall j \in \Theta$, где f_j - максимум полосы пропускания точки доступа j.

(2) Ограничение на число пользователей, которые могут быть подключены к точке доступа j: $\sum_{i=1}^{n} x_{ij} \le n_j \quad \forall j \in \Theta$, где n_j - максимум числа пользователей для точки доступа j.

(3) Ограничение по подключению пользователя к точкам доступа (здесь предполагается, что каждый пользователь подключается только к одной точке доступа): $\sum_{j=1}^{m} x_{ij} \le 1 \quad \forall i \in \Psi$.

Таким образом, получается следующая многокритериальная задача:

$$\max\ R(X) = \sum_{i=1}^{n} \sum_{j \in \Theta_i} r_{ij}\, x_{ij}, \quad \max\ B(X) = \sum_{i=1}^{n} \sum_{j \in \Theta_i} \beta_{ij}\, x_{ij},$$

$$\max\ F(X) = \sum_{i=1}^{n} \sum_{j \in \Theta_i} f_{ij}\, x_{ij}, \quad \max\ P(X) = \sum_{i=1}^{n} \sum_{j \in \Theta_i} p_{ij}\, x_{ij},$$

$$\max\ D(X) = \sum_{i=1}^{n} \sum_{j \in \Theta_i} d_{ij}\, x_{ij}$$

$$s.t. \quad \sum_{i=1}^{n} f_{ij}\, x_{ij} \le f_j \ \ \forall j \in \Theta, \quad \sum_{i=1}^{n} x_{ij} \le n_j \ \ \forall j \in \Theta, \quad \sum_{j \in \Theta_i} x_{ij} \le 1 \ \ \forall i \in \Psi,$$

$$x_{ij} \in \{0, 1\}, \ \ \forall\, i = \overline{1, n}, \ \forall\, j = \overline{1, m}, \quad x_{ij} = 0, \ \ \forall\, i = \overline{1, n}, \ j \in \{\Theta \setminus \Theta_i\}.$$

13.3.3 Схема решения

Получившаяся многокритериальная обобщенная задача о назначении (размещении) относится к классу NP-трудных задач [207]. При поведении расчетов использовалась двухэтапная эвристика:

(i) упрощение задачи на основе отображения вектора характеристик соединения “пользователь - точка доступа” к порядковой оценке (на основе многокритериального ранжирования с использованием варианта метода порогов несравнимости ELECTRE [28, 364]),

(ii) решение получившейся однокритериальной задачи о назначении на основе жадного алгоритма;

Следует отметить, что исследуемая задача подключения пользователей имеет широкое применение для динамических режимов работы коммуникационных сетей (т.е., Ad Hoc сети, мобильные сети, mesh сети) и для решения рассматриваемой задачи имеется крайне ограниченный временной интервал. Такая ситуация приводит к необходимости использования очень быстрых эвристик (типа жадных алгоритмов).

13.3.4 Численный пример

Пример включает 30 пользователей и 6 точек доступа (Рис. 13.7). Исходные данные по точкам доступа и пользователям представлены в Таблице 13.8 и Таблице 13.9.

Таблица 13.8. Точки доступа

j	$x(j)$	$y(j)$	$z(j)$	$f(j)$	$n(j)$	$r(j)$	$d(j)$
1	50	157	10	30	4	10	10
2	150	165	10	30	5	15	8
3	72	102	10	21	6	10	9
4	140	112	10	21	5	8	8
5	45	52	10	35	10	10	10
6	147	47	10	30	5	15	7

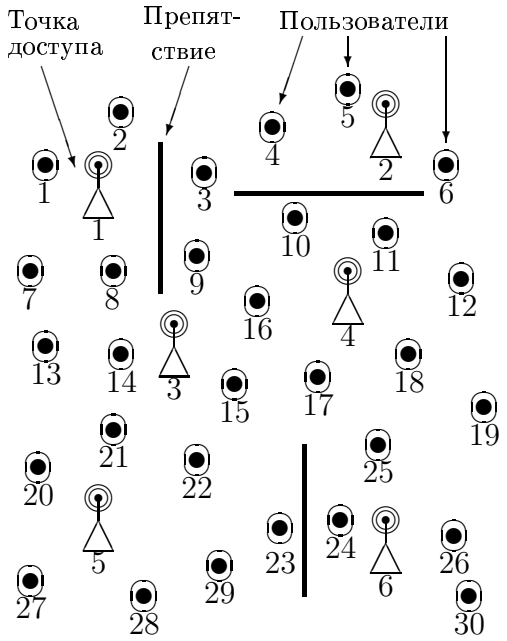


Рис. 13.7. Исходный пример

Таблица 13.9. Пользователи

i	$x(i)$	$y(i)$	$z(i)$	$f(i)$	$p(i)$	$r(i)$	$d(i)$
1	30	165	5	10	2	5	8
2	58	174	5	5	1	9	6
3	88	156	0	6	1	6	8
4	110	169	5	7	2	5	6
5	145	181	3	5	2	4	6
6	170	161	5	7	2	4	7
7	23	136	7	6	2	7	6
8	52	134	5	6	1	8	7
9	85	134	3	6	1	7	7
10	120	140	6	4	2	6	8
11	150	136	3	6	2	7	8
12	175	125	1	8	3	5	6
13	27	109	7	8	3	5	8
14	55	105	2	7	2	10	6
15	98	89	3	10	1	10	7
16	107	115	4	7	1	7	10
17	127	95	5	7	2	5	7
18	160	104	5	9	2	4	5
19	183	91	4	4	3	5	7
20	25	65	2	7	3	5	6
21	52	81	1	10	1	8	10
22	82	66	4	14	2	3	7
23	112	56	10	10	1	10	9
24	135	59	4	13	3	4	6
25	147	79	5	7	3	16	8
26	165	50	3	7	3	3	7
27	23	31	4	8	1	7	6
28	65	25	7	6	2	9	8
29	93	39	1	10	1	10	9
30	172	26	2	10	2	7	6

Рассчитанные параметры содержатся в Таблице 13.10, Таблице 13.11, Таблице 13.12, Таблице 13.13. Векторы $(\xi^{\beta}_{ij}, \xi^{d}_{ij})$ (таблица 13.12) определяют множества Θ_i, $(i = \overline{1,n})$: $\Theta_1 = \{1,3,5\}$, $\Theta_2 = \{1,2,3,5\}$, $\Theta_3 = \{1,2,3,4,5\}$, $\Theta_4 = \{1,2,3,4\}$, $\Theta_5 = \{2,4\}$, $\Theta_6 = \{2,4,6\}$, $\Theta_7 = \{1,3,5\}$, $\Theta_8 = \{1,3,4,5\}$, $\Theta_9 = \{1,2,3,4,5,6\}$, $\Theta_{10} = \{1,2,3,4,5,6\}$, $\Theta_{11} = \{2,3,4,5,6\}$, $\Theta_{12} = \{2,3,4,5,6\}$, $\Theta_{13} = \{1,3,4,5\}$, $\Theta_{14} = \{1,3,4,5\}$, $\Theta_{15} = \{1,3,4,5,6\}$, $\Theta_{16} = \{1,5\}$, $\Theta_{17} = \{2,3,4,5,6\}$, $\Theta_{18} = \{2,3,4,5,6\}$, $\Theta_{19} = \{2,3,4,6\}$, $\Theta_{20} = \{1,3,4,5\}$, $\Theta_{21} = \{1,5\}$, $\Theta_{22} = \{1,3,4,5,6\}$, $\Theta_{23} = \{1,3,5\}$, $\Theta_{24} = \{3,4,5,6\}$, $\Theta_{25} = \{2,3,4,6\}$, $\Theta_{26} = \{4,6\}$, $\Theta_{27} = \{1,3,4,5\}$, $\Theta_{28} = \{1,3,4,5,6\}$, $\Theta_{29} = \{1,3,5\}$, $\Theta_{30} = \{4,6\}$.

Расчет интегрированных показателей соответствия $\widetilde{c_{ij}}$ (таблица 13.14) осуществляется за счет отображения векторой оценки $\widehat{c_{ij}}$ (таблица 13.13) на порядковую шкалу [1,3] (5 соответствует наилучшему уровню соответствия, используется многокритериальное ранжирование на основе модификация метода ELECTRE [28, 364]). Дополнительно, в таблице 13.13 учтены множества $\{\Theta_i\}$ (т.е., $\widetilde{c_{ij}} = 0$ если $j \in \{\Theta \setminus \Theta_i\}$).

Таким образом получается и решается упрощенная (simplified) задача о назначении (с одним интегрированным критерием, каждый пользователь может подключаться только к одной точке доступа):

$$\max \sum_{i=1}^{n} \sum_{j \in \Theta_i} \widetilde{c_{ij}}\, x_{ij}$$

$$s.t. \quad \sum_{i=1}^{n} f_{ij}\, x_{ij} \le f_j \;\; \forall j \in \Theta, \quad \sum_{i=1}^{n} x_{ij} \le n_j \;\; \forall j \in \Theta, \quad \sum_{j \in \Theta_i} x_{ij} \le 1 \;\; \forall i \in \Psi,$$

$$x_{ij} \in \{0,1\}, \;\; \forall\, i = \overline{1,n}, \; \forall\, j = \overline{1,m}, \quad x_{ij} = 0, \;\; \forall\, i = \overline{1,n}, \; j \in \{\Theta \setminus \Theta_i\}.$$

Результаты решения задачи (т.е., вариант подключения пользователей к точкам доступа) представлены на Рис. 13.8 и в Таблице 13.15.

13.3.5 Обсуждение

Представленный в работе подход является первым шагом в использовании многокритериальных комбинаторных моделей при подключении пользователей к телекоммуникационной сети. Основное внимание было уделено новой постановке задачи и схеме решения. Очевидно, что следует провести дальнейшие исследования:

1. В области постановки задачи:

(1) Уточнение постановки: (а) состав и правила пересчета параметров пользователей и точек доступа, интегрированных показателей, правила, например, интегрированные показатели (барьер-расстояние β_{ij}), показатели качества распространения сигналов (f_{ij}); (б) множество критериев, (в) множество ограничений;

(2) возможность применения других режимов управления, например: многошаговые (multi-hop) схемы связи, режим P2P;

(3) учет мобильности точек доступа;

(4) учет режима расширения множества пользователей (т.е., подключение дополнительных пользователей в процессе работы системы);

(5) расширение использованной постановки задачи на основе возможности подключения каждого пользователя к нескольким точкам доступа.

2. В области моделей:

(1) использование более сложных оптимизационных моделей с учетом неопределенности, например, модели на основе размытых множеств, модели стохастической оптимизации;

(2) учет взаимосвязи "соседних" подключений (для учета возможных коллизий, взаимного влияния), например, на основе задачи квадратичного назначения или подхода иерархического морфологического проектирования (т.е., задачи морфологической клики).

3. В области методов решения: реализация серии схем решения (включая переборные методы, генетические алгоритмы, мета-эвристики) и проведение сравнительных вычислительных экспериментов на основе специальных тестовых числовых примеров.

Таблица 13.10. Матрица расстояний $||l_{ij}||$

i	Точки доступа j					
	1	2	3	4	5	6
1	22.11	120.10	75.88	122.20	114.10	166.25
2	19.44	92.57	73.52	102.92	122.79	155.16
3	39.31	63.44	57.20	68.85	112.98	124.35
4	61.39	40.51	77.19	64.61	133.94	127.59
5	98.23	18.17	107.79	69.53	163.37	134.20
6	120.17	21.00	114.50	57.67	165.2	116.40
7	34.34	130.30	59.72	119.47	86.89	152.66
8	23.62	102.91	38.07	90.85	82.45	128.91
9	42.46	72.35	35.24	59.65	91.50	107.06
10	72.15	39.26	61.35	34.64	115.69	96.92
11	102.42	29.83	85.38	26.93	134.65	89.33
12	129.34	48.02	105.92	38.41	149.37	83.36
13	53.31	135.18	45.64	113.08	59.85	135.10
14	52.85	112.65	19.03	85.66	54.53	109.05
15	83.53	92.35	29.90	48.39	65.02	64.92
16	71.06	66.22	37.82	33.67	88.59	79.12
17	98.98	73.85	55.67	21.98	92.73	52.24
18	122.20	62.02	88.16	22.11	126.31	58.68
19	148.60	81.25	111.71	48.23	143.53	57.17
20	95.67	160.28	60.35	124.49	25.16	123.58
21	76.56	129.39	30.36	93.73	31.16	101.30
22	96.65	120.25	37.84	74.27	40.01	67.99
23	118.51	115.43	60.96	62.61	67.12	36.14
24	129.87	107.22	76.51	53.57	90.47	18.00
25	124.57	86.20	78.61	34.10	105.63	32.39
26	157.24	116.19	106.78	67.22	120.22	19.54
27	129.00	184.72	86.48	142.43	31.00	125.17
28	132.88	163.81	77.38	114.90	33.73	84.95
29	125.91	138.59	67.01	87.29	50.54	55.33
30	179.19	140.96	125.86	92.11	129.88	33.62

Таблица 13.11. Матрица $||e_{ij}||$

i	Точки доступа j					
	1	2	3	4	5	6
1	0	1	0	1	0	1
2	0	0	1	1	0	1
3	1	0	0	0	0	0
4	1	0	1	1	1	1
5	1	0	1	1	1	1
6	1	0	1	0	1	0
7	0	1	0	1	0	1
8	0	1	0	1	0	1
9	1	1	0	0	0	1
10	1	1	0	0	0	0
11	1	1	0	0	0	0
12	1	1	0	0	0	0
13	0	1	0	0	0	1
14	0	1	0	0	0	1
15	1	1	0	0	0	1
16	1	1	0	0	0	0
17	1	1	0	0	0	0
18	1	1	0	0	0	0
19	1	1	0	0	1	0
20	0	1	0	0	0	1
21	0	1	0	0	0	1
22	0	1	0	0	0	1
23	0	1	0	1	0	1
24	1	1	1	0	1	0
25	1	1	0	0	1	0
26	1	1	1	0	1	0
27	0	1	0	0	0	1
28	0	1	0	0	0	1
29	0	1	0	0	0	1
30	1	1	1	0	1	0

Таблица 13.12. Матрица векторов $||(\xi_{ij}^{\beta}, \xi_{ij}^{d})||$

i	Точки доступа j					
	1	2	3	4	5	6
1	1,1	0,1	1,1	0,1	1,1	0,1
2	1,1	1,1	1,1	0,1	1,1	0,1
3	1,1	1,1	1,1	1,1	1,1	0,0
4	1,1	1,1	1,1	1,1	0,1	0,1
5	0,1	1,1	0,1	1,1	0,1	0,1
6	0,1	1,1	0,1	1,1	0,1	1,1
7	1,1	0,1	1,1	0,1	1,1	0,1
8	1,1	0,1	1,1	1,1	1,1	0,1
9	1,1	1,1	1,1	1,1	1,1	1,1
10	1,1	1,1	1,1	1,1	1,1	1,1
11	0,1	1,1	1,1	1,1	1,1	1,1
12	0,1	1,1	1,1	1,1	1,1	1,1
13	1,1	0,1	1,1	1,1	1,1	0,1
14	1,1	0,1	1,1	1,1	1,1	0,1
15	1,1	0,1	1,1	1,1	1,1	1,1
16	1,1	1,0	1,0	1,0	1,1	1,0
17	0,1	1,1	1,1	1,1	1,1	1,1
18	0,1	1,1	1,1	1,1	1,1	1,1
19	0,1	1,1	1,1	1,1	0,1	1,1
20	1,1	0,1	1,1	1,1	1,1	0,1
21	1,1	0,0	1,0	1,0	1,1	0,0
22	1,1	0,1	1,1	1,1	1,1	1,1
23	1,1	0,0	1,1	1,0	1,1	1,0
24	0,1	0,1	1,1	1,1	1,1	1,1
25	0,1	1,1	1,1	1,1	0,1	1,1
26	0,1	0,1	0,1	1,1	0,1	1,1
27	1,1	0,1	1,1	1,1	1,1	0,1
28	1,1	0,1	1,1	1,1	1,1	1,1
29	1,1	0,0	1,1	1,0	1,1	1,0
30	0,1	0,1	0,1	1,1	0,1	1,1

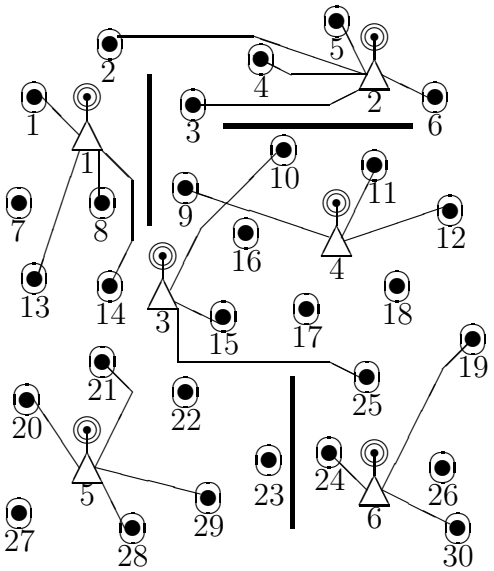


Рис. 13.8. Подключение пользователей

Таблица 13.13. Матрица $||\widehat{c_{ij}}|| = ||(r_{ij}, \beta_{ij}, f_{ij}, p_{ij}, d_{ij})||$

i	Точки доступа j					
	1	2	3	4	5	6
1	5, 10, 10, 2, 10	5, 0, 10, 2, 8	5, 10, 10, 2, 9	5, 0, 10, 2, 8	5, 5, 10, 2, 10	5, 0, 10, 2, 7
2	9, 10, 5, 1, 10	9, 5, 5, 1, 8	9, 5, 5, 1, 9	8, 0, 5, 1, 8	9, 5, 5, 1, 10	9, 0, 5, 1, 7
3	6, 5, 6, 1, 10	6, 10, 6, 1, 8	6, 10, 6, 1, 9	6, 10, 6, 1, 8	6, 5, 6, 1, 10	6, 0, 6, 1, 0
4	5, 5, 7, 2, 10	5, 10, 7, 2, 8	5, 5, 7, 2, 9	5, 5, 7, 2, 8	5, 0, 7, 2, 10	5, 0, 7, 2, 7
5	4, 0, 5, 2, 10	4, 10, 5, 2, 8	4, 0, 5, 2, 9	4, 5, 5, 2, 8	4, 0, 5, 2, 10	4, 0, 5, 2, 7
6	4, 0, 7, 2, 10	4, 10, 7, 2, 8	4, 0, 7, 2, 9	4, 10, 7, 2, 8	4, 0, 7, 2, 10	4, 5, 7, 2, 7
7	7, 10, 6, 2, 10	7, 0, 6, 2, 8	7, 10, 6, 2, 9	7, 0, 6, 2, 8	7, 10, 6, 2, 10	7, 0, 6, 2, 7
8	8, 10, 6, 1, 10	8, 0, 6, 1, 8	8, 10, 6, 1, 9	8, 5, 6, 1, 8	8, 10, 6, 1, 10	8, 0, 6, 1, 7
9	7, 5, 6, 1, 10	7, 5, 6, 1, 8	7, 10, 6, 1, 9	7, 10, 6, 1, 8	7, 10, 6, 1, 10	7, 0, 6, 1, 7
10	6, 5, 4, 2, 10	6, 5, 4, 2, 8	6, 10, 4, 2, 9	6, 10, 4, 2, 8	6, 5, 4, 2, 10	6, 5, 4, 2, 7
11	7, 0, 6, 2, 10	7, 5, 6, 2, 8	7, 5, 6, 2, 9	7, 10, 6, 2, 8	7, 5, 6, 2, 10	7, 10, 6, 2, 7
12	5, 0, 8, 3, 10	5, 5, 8, 3, 8	5, 5, 8, 3, 9	5, 10, 8, 3, 8	5, 5, 8, 3, 10	5, 10, 8, 3, 7
13	5, 10, 8, 3, 10	5, 0, 8, 3, 8	5, 10, 8, 3, 9	5, 5, 8, 3, 8	5, 10, 8, 3, 10	5, 0, 8, 3, 7
14	10, 10, 7, 2, 10	10, 0, 7, 2, 8	10, 10, 7, 2, 9	8, 10, 7, 2, 8	10, 10, 7, 2, 10	10, 0, 7, 2, 7
15	10, 5, 10, 1, 10	10, 0, 10, 1, 8	10, 10, 10, 1, 9	8, 10, 10, 1, 8	10, 10, 10, 1, 10	10, 5, 10, 1, 7
16	7, 5, 7, 1, 10	7, 5, 7, 1, 0	7, 10, 7, 1, 0	7, 10, 7, 1, 0	7, 10, 7, 1, 10	7, 10, 7, 1, 0
17	5, 0, 7, 2, 10	5, 5, 7, 2, 8	5, 10, 7, 2, 9	5, 10, 7, 2, 8	5, 5, 7, 2, 10	5, 5, 7, 2, 7
18	4, 0, 9, 2, 10	4, 5, 9, 2, 8	4, 10, 9, 2, 9	4, 10, 9, 2, 8	4, 5, 9, 2, 10	4, 10, 9, 2, 7
19	5, 0, 4, 3, 10	5, 5, 4, 3, 8	5, 5, 4, 3, 9	5, 10, 4, 3, 8	5, 0, 4, 3, 10	5, 10, 4, 3, 7
20	5, 5, 7, 3, 10	5, 0, 7, 3, 8	5, 10, 7, 3, 9	5, 5, 7, 3, 8	5, 10, 7, 3, 10	5, 0, 7, 3, 7
21	8, 10, 10, 1, 10	8, 0, 10, 1, 0	8, 10, 10, 1, 0	8, 5, 10, 1, 0	8, 10, 10, 1, 10	8, 0, 10, 1, 0
22	3, 5, 10, 2, 10	3, 0, 10, 2, 8	3, 10, 10, 2, 9	3, 10, 10, 2, 8	3, 10, 10, 2, 10	3, 5, 10, 2, 7
23	10, 5, 10, 1, 10	10, 0, 10, 1, 0	10, 10, 10, 1, 9	8, 5, 5, 1, 0	10, 10, 10, 1, 10	10, 5, 10, 1, 0
24	4, 0, 10, 3, 10	4, 0, 10, 3, 8	4, 5, 10, 3, 9	4, 10, 10, 3, 8	4, 5, 10, 3, 10	4, 10, 10, 3, 7
25	10, 0, 7, 3, 10	15, 5, 7, 3, 8	10, 10, 7, 3, 9	8, 10, 7, 3, 8	10, 0, 7, 3, 10	15, 10, 7, 3, 7
26	3, 0, 7, 3, 10	3, 0, 7, 3, 8	3, 0, 7, 3, 9	3, 10, 7, 3, 8	3, 0, 7, 3, 10	3, 10, 7, 3, 7
27	7, 5, 8, 1, 10	7, 0, 8, 1, 8	7, 10, 8, 1, 9	7, 5, 8, 1, 8	7, 10, 8, 1, 10	7, 0, 8, 1, 7
28	9, 5, 6, 2, 10	9, 0, 6, 2, 8	9, 10, 6, 2, 9	8, 5, 6, 2, 8	9, 10, 6, 2, 10	9, 5, 6, 2, 7
29	10, 5, 10, 1, 10	10, 0, 10, 1, 0	10, 10, 10, 1, 9	8, 10, 10, 1, 0	10, 10, 10, 1, 10	10, 5, 10, 1, 0
30	7, 0, 10, 2, 10	7, 0, 10, 2, 8	7, 0, 10, 2, 9	7, 10, 10, 2, 8	7, 0, 10, 2, 10	7, 10, 10, 2, 7

Таблица 13.14. Матрица $||\widetilde{c_{ij}}||$

i	Точки доступа j					
	1	2	3	4	5	6
1	3	0	3	0	3	0
2	2	1	1	0	2	0
3	1	1	1	1	1	0
4	1	1	2	1	0	0
5	0	1	0	1	0	0
6	0	1	0	1	0	1
7	1	0	2	0	1	0
8	2	0	2	1	2	0
9	1	1	1	1	1	1
10	2	1	1	1	1	1
11	0	1	2	1	3	1
12	0	3	3	3	3	2
13	3	0	3	3	3	0
14	3	0	3	3	3	0
15	3	0	3	3	3	2
16	2	0	0	0	3	0
17	0	1	3	1	2	1
18	0	1	2	1	3	1
19	0	1	2	1	0	1
20	3	0	3	2	1	0
21	3	0	0	0	3	0
22	2	0	1	1	2	1
23	3	0	3	0	3	0
24	0	0	3	3	3	2
25	0	3	3	3	0	3
26	0	0	0	1	0	1
27	3	0	2	1	3	0
28	3	0	3	3	3	2
29	3	0	3	0	3	0
30	0	0	0	3	0	3

Таблица 13.15. Вариант подключения

i	Точки доступа j					
	1	2	3	4	5	6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	1	0	0	0	0
4	0	1	0	0	0	0
5	0	1	0	0	0	0
6	0	1	0	0	0	0
7	0	0	0	0	0	0
8	1	0	0	0	0	0
9	0	0	0	1	0	0
10	0	0	1	0	0	0
11	0	0	0	1	0	0
12	0	0	0	1	0	0
13	1	0	0	0	0	0
14	1	0	0	0	0	0
15	0	0	1	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	1
20	0	0	0	0	1	0
21	0	0	0	0	1	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	1
25	0	0	1	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	1	0
29	0	0	0	0	1	0
30	0	0	0	0	0	1

13.4 Передача мультимедийной информации

13.4.1 Стандарты

На Рис. 13.9 представлена схема обработки информации: a - видео сигнал, b - звуковой (аудио) сигнал, c - сигнал синхронизации, d - тестовый сигнал. Эта схема используется как базовая для процесса стандартизации (приемная часть имеет аналогичную инверсную структуру для декодирования/декомпрессии).

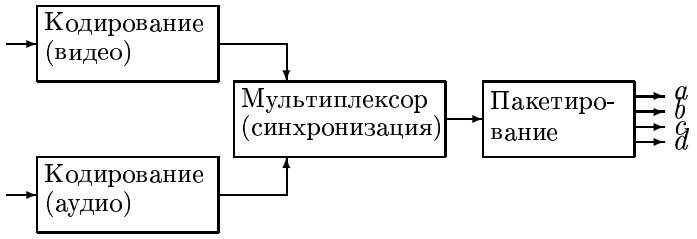


Рис. 13.9. Схема обработки информации

Анализ основных требований к процессу обработки информации основывается на следующем:

I. Исходные данные (аудио, видео, данные синхронизации, синтетическая информация).

II. Обработка (кодирование, передача по сети, декодирование).

III. Результирующие данные (для систем пользователя: человек, прикладная система).

Таким образом, требования можно рассматривать с учетом следующего:

1. Объект (т.е., данные определенного типа, процесс).

2. Базовая точка для анализа (т.е., стадия упомянутого цикла обработки информации: входная часть, выходная часть, промежуточная технологическая часть).

Дополнительно, целесообразно указать классификацию данных: (i) вид данных (аналоговые, цифровые), (ii) временной режим (реальное время, не-реальное время).

В общем случае, два типа требований исследуются:

1. Качество результатов (т.е., результирующая информация).

2. Затраты (стоимость, оценки времени и др.).

Качество результирующей картинки можно основывать на следующем: коэффициент PSNR, задержка, ошибки, латентность, потери (т.е., потери пакетов, потери кадров).

С точки зрения типовых изменений можно рассматривать операции изменения и требования к этим операциям:

I. Полезность операций:

1.1 Качество выхода.

1.2 Уровень требования к входу.

1.3 Уровень требования к технологических стадиям (т.е., кодирование, передача по сети, декодирование).

II. 'Стоимость' операций:

2.1 Требуемый дополнительный НИР.

2.2 Требуемая дополнительная разработка программного обеспечения.

2.3 Требуемая дополнительная разработка электроники.

2.4 Сложность дополнительного этапа производства.

2.5 Простота реализации в рамках существующих технологий.

В рамках данного исследования рассматривается линия стандартов MPEG [133, 194, 322, 579]. При этом основная цепочка эволюции следующая:

JPEG $\Rightarrow$ *MPEG-1* $\Rightarrow$ *MPEG-2* $\Rightarrow$ *MPEG-4*.

Структурные описания стандартов при обработке видео информации имеют следующие вид (Рис. 13.10):

I. Общая часть:

1.1. *Прикладной уровень:* (видео телефония, видео конференции, digital broadcast, кабельное и спутниковое TV, видео обслуживание по сети, DVD, DV camcorders, персональный стримминг и др.).

1.2. *Режим качества по времени и картинке*

1.2.1. Временной режим: задержка: 0.15 сек для “face-to-face”, 1.0 сек для “interactive”.

1.2.2. Качество картинки: хорошее качество, низкое качество.

1.3. *Формат*

1.3.1. Резолюция.

1.3.2. Цветовая декомпозиция: (luminance, chrominance).

1.4. *Базовые операции*

1.4.1. Трансформация.

1.4.2. Воспроизведение/свойства:

1.4.3. Стримминг видео, аудио, синхронизация, стримминг данных, тестирование, управление).

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1. Видео кодирование (методы трансформации в цифровой код).

2.1.2. Оценка движения (motion estimation) (вектор и т.п.).

2.1.3. Кодирующая структура (группа картинок (group of pictures GOP)).

2.2. *Принципы и структура*

2.2.1. Принципы:

2.2.1.1. Блок декомпозиции

2.2.1.2. Сканирование

2.2.2. Структура (базовая схема обработки, расширенная схема обработки, 'открытая структура', включая транскодирование).

2.3. *Алгоритмы.*

Алгоритмы в данном исследовании, в основном, не рассматриваются.

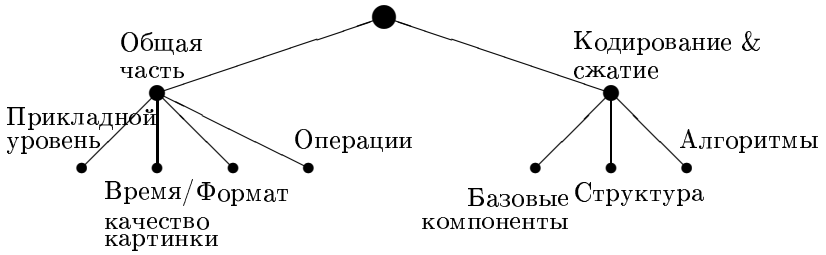


Рис. 13.10. Общая структура стандарта

Теперь можно рассмотреть стандарты MPEG на основе приведенных структур:

Описание MPEG-1

I. Общая часть:

1.1. *Прикладной уровень:* (a) битовая скорость передачи данных (bit rate): 64 Кбит / *сек* ... 2 Мбит / *сек*, (b) цифровая запоминающая среда (digital media storage), (c) компьютерные и коммуникационные сети.

1.2. *Режим качества по времени и картинке:* *1.2.1. Временной режим:* задержка. *1.2.2. Качество картинки:* без ошибки, низкий уровень качества.

1.3. *Формат:* *1.3.1. Резолюция:* низкая резолюция (CIF, QCIF). *1.3.2. Цветовая декомпозиция (по образцам):* 4:2:0

1.4. *Базовые операции*

1.4.1. Трансформация: (a) аудио информация, (b) информация о движущихся картинках (moving picture information), (c) комбинация аудио/видео информации (системный уровень).

1.4.2. Воспроизведение/свойства: (1) случайный доступ, (2) быстрый поиск в прямом /обратном направлении (fast forward/reverse search), (3) реверсивное воспроизведение (reverse playback), (4) аудио/видео синхронизация, (5) устойчивость к ошибкам (good error behavior), (6) задержка, (7) редактирование, (8) формат, (9) регулирование скорости (rate control).

1.4.3. Стримминг: аудио, видео, синхронизация, системный стрим, программный стрим.

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1. Видео кодирование: транскодирование: кодирование структуры коэффициентов словами переменной длины.

2.1.2. Оценки движения (motion estimation): от -1024 до 1023 полупикселей от -512 до 511.5 пикселей.

2.1.3. Кодирующая структура: (a) внутри кадра: I (закодировано внутри картинки - intrapicture coded), (b) между кадрами: P (закодировано с использованием прогноза компенсации движения, (c) B (закодировано на основе I и P), (d) D (в целях нахождения при быстром поиске). **2.2.** *Принципы и структура*

2.2.1. Принципы: (a) последовательность (заготовок инициализирует состояние декодера), (b) группа картинок (точка случайного доступа), (c) картинка (один кадр), (d) квант времени (ресинхронизация), (e) макроблок (16×16 картинка).

2.2.1.1. Блок декомпозиции: 16×16 (макро-блок), 8×8 (блок).

2.2.1.2. Сканирование: прогрессивное сканирование (zigzag).

2.2.2. Структура: предварительная обработка, кодирование, кодирование, последующая обработка (post-processing) открыты для улучшения (приоритет для передачи видео информации).

2.3. *Алгоритмы.*

Описание MPEG-2

I. Общая часть:

1.1. *Прикладной уровень:* (a) битовая скорость передачи данных; 4 Мбит / *сек* ... 80 Мбит / *сек*, (b) цифровое TV, высококачественная запоминающая среда, кабельное и спутниковое TV, видеообслуживание по сети.

1.2. *Режим качества по времени и картинке*

1.2.1. Временной режим: реальное время, небольшая задержка 500...700 *тсек*, задержка между аудио и видео сигналами ≈ 300 *тсек*.

1.2.2. Качество картинки: хорошее качество для базовых режимов: выявление ошибок, ресинхронизация, восстановление данных.

1.3. *Формат:* *1.3.1. Резолюция:* CIF, QCIF, полный D1, полу - D1, HDTV, высокая резолюция (NTSC, PAL). *1.3.2. Цветовая декомпозиция (по образцам):* 4:2:0, 4:2:2, 4:4:4 (для высокого профиля).

1.4. *Базовые операции*

1.4.1. Трансформация: *(i)* Кодирование: (a) аудио информация, (b) движущиеся картинки (moving picture information), (c) комбинация аудио/видео информации (системный уровень).

(ii) цифровой Долби (Dolby Digital), DVD 16×9 (TV) & масштабируемость: SNR (качество), пространственный, временной.

1.4.2. Воспроизведение/свойства: (1) случайный доступ, (2) бытсрый поиск в прямом/обратном направлении, (3) обратное воспроизведение, (4) аудио/видео синхронизация, (5) устойчивость к ошибкам, (6) задержка, (7) редактирование, (8) формат, (9) регулирование скорости, (10) масштабируемость.

Дополнения: совместимость, включая *вверх/вниз* (транскодирование разрешения), *вперед/назад.*

1.4.3. Стримминг: видео, аудио, системы, синхронизация, тестирование, управление цифровой запоминающей средой, интерфейс реального времени и транспортный поток.

Дополнительно: Маркер ресинхронизации (в начале каждого кадра).

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1. Видео кодирование: Новые VLC-таблицы для DCT (нелинейный). Компенсация движения кадра/области.

2.1.2. Оценки движения (motion estimation):

(a) от -2048 до 2047.5 полупикселей, от -1024 до 1023 пикселей.

(b) вектор движения для кадра/области.

2.1.3. Кодирующая структура: (a) внутри кадра: I (закодировано внутри картинки), (b) между кадрами: P (закодировано с использованием прогноза компенсации движения), (c) B (закодировано на основе I и P), (d) D (в целях нахождения при быстром поиске).

2.2. *Принципы и структура*

2.2.1. Принципы: 2.2.1.1. Декомпозиция блоков: 8×8. *2.2.1.2. Сканирование:* прогрессивное сканирование (zigzag) / чересстрочное сканирование.

2.2.2. Структура: предварительная обработка, кодирование, кодирование, последующая обработка (post-processing) открыты для улучшения (приоритет для передачи видео информации).

2.3. *Алгоритмы.*

Описание MPEG-4

I. Общая часть:

1.1. *Прикладной уровень:* (a) битовая скорость передачи данных: 24 *Кбит / сек* ... 1024 *Кбит / сек*, (b) DVD, цифровой кабель,записывающая видеокамера, WEB (персональный стримминг), защита приложений, интерактивный режим.

1.2. *Режим качества по времени и картинке: 1.2.1. Временной режим:* задержка 500...700 mсек. *1.2.2. Качество картинки:* переменное (зависит от использхования).

1.3. *Формат: 1.3.1. Резолюция:* от суб-QCIF к 'Studio'. *1.3.2. Цветовая декомпозиция (по образцам):* $4:2:0$, $4:2:2$, $4:4:4$.

1.4. *Базовые операции*

1.4.1. Трансформация

1.4.2. Воспроизведение/свойства: (1) случайные доступ, (2) быстрый поиск в прямом/обратном направлении, (3) реверсивное вопроизведение, (4) аудио/видео синхронизация, (5) устойчивость к ошибкам, (6) задержка, (7) редактирование, (8) формат, (9) регулирование скорости, (10) масштабируемость на базе объектов.

1.4.3. Стримминг: (a) данные стримминга, мультиплексировние и синхронизация данных, связанных с медийными объектами, (b) потоки реального времени, (c) позиция маркера ресинхронизхации - любая (задержка между аудио- и видеосигналом ≈ 300 mсек).

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1. Видеокодирование: (a) возможность использовать VLC-таблицы для нахождения ошибок, (b) возможность кодирования естественных и синтезированных данных, (c) высокая гибкость кодирования аудио- и видеообъектов.

2.1.2. Оценки движения (motion estimation):

2.1.3. Кодирующая структура:

2.2. *Принципы и структура*

2.2.1. Принципы: 2.2.1.1. Блочная декомпозиция: на основе объектов (VOB). *2.2.1.2. Сканирование:*

2.2.2. Структура: разделение данных движение и текста (приоритет обработки для каждого кадра).

2.3. *Алгоритмы:* алгоритмы могут включать специальные режимы для следующего: *1.* сцены, группы людей; *2.* графика, текст, рисунки.

13.4.2 Изменения стандартов

Теперь можно указать основные тенденции в развитии стандартов обработки мутльтимедийной информации:

1. Расширение приложений: (a) запоминающая среда, (b) вещание, broadcast, (c) WEB-приложения. В тоже время, : можно указать 2-ю линию развития по временным режимам: режим off line, режим on-line (реальное время), интерактивный режим.

2. Уменьшение задержки.

3. Улучшение качества:

3.1. Уменьшение ошибок.

3.2. Улучшение резолюций.

3.3. Улучшение качество синхронизации.

4. Обеспечение дополнительных приложений: *поиск*, *архивирование*, *индексация*, *мультимедийные базы данных.*

Теперь можно указать изменения стандартов:

Изменения: MPEG-1 $\Longrightarrow$ MPEG-2

I. Общая часть:

1.1. *Прикладной уровень:* цифровое TV, высококачественное хранение аудио- видеоинформации, кабельное/спутниковое TV, видео-сервис, высококачественная передача аудиоинформации, битовая скорость передачи данных (MP3, стерео и др.): 4 Мбит / сек ... 80 Мбит / сек.

1.2. *Режим качества по времени и картинке*

1.2.1. Временной режим:

(i) низкая задержка (от 0.15 до менее чем 0.15 для *F2F*),

(ii) задержка в реальном времени от 1 *сек* до менее чем 0.4 для интерактивного режима.

1.2.2. Качество картинки: (a) хорошее качество картинки, (b) специальные режимы способности системы противостоять ошибкам.

1.3. *Формат:*

1.3.1. Резолюция: D1, полу-D1, HDTV.

1.3.2. Цветовая декомпозиция (по образцам): 4 : 2 : 2 (для различных уровней и профилей качества).

1.4. *Базовые операции*

1.4.1. Трансформация: масштабируемость (2/3 уровня).

1.4.2. Воспроизведение/ свойства: вверх/вниз (транскодирование разрешения).

1.4.3. Стримминг: (1) тестирование, (2) управление цифровой запоминающей средой, (3) имитация программного обеспечения, (4) интерфейс реального времени, (5) транспортный поток, (6) 10 битовое видео (студийное качество).

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1 Видео кодирование: (a) таблица нелинейной квантификации с повышенной точностью для малых значений, (b) альтенативное сканирование для коэффициента DCT, (c) MV: $-2048...2047.5$ полупикселей, (d) новые VLC-таблицы для коэффициентов DCT, (e) сканирование кадра/области для DCT и MV, (f) не D-кадры.

2.1.2 Кодирующая структура: двойная компенсация основного движения (для P-кадров).

2.2. *Принципы и структура*

2.2.1 Принципы

2.2.1.1 Блочная декомпозиция: в масштабе: 1×1, 2×2, 4×4.

Изменения: MPEG-2 $\Longrightarrow$ MPEG-4

I. Общая часть:

1.1. *Прикладной уровень:* DVD, записывающая видеокамера, WEB (персональный стримминг), 24...1024 Кбит / сек (по сжатию: 1000 раз).

1.2. *Режим качества по времени и картинке*

1.2.1. Временной режим: (a) задержка 500...700 mсек, (b) задержка между аудио- и видеосигналами $\approx$ 300 mсек.

1.2.2. Качество картинки: использование VLC-таблиц для нахождения ошибок (возможно).

1.3. *Формат:*

1.3.1. Резолюция: формат файла от QCIF до 'Studio' (инструментарий для запоминания данных в файле).

1.3.2. Цветовая декомпозиция: $4:4:4$.

1.4. *Базовые операции*

1.4.1. Трансформация: масштабируемость на базе объектов.

1.4.2. Воспроизведение: пауза, медленно.

1.4.3. Стримминг: (a) стримминг данных, мультиплексирование и синхронизация данных, связанных с медийными объектами, передаваемыми с использованием протокола IP; (b) гибкость в установлении маркеров в любом месте кадра.

II. Кодирование/сжатие

2.1. *Базовые компоненты*

2.1.1 Видео кодирование:

2.1.2 Оценка движения (motion estimation): гибридное кодирование естественных изображений (т.е., на основе пикселей) и синтеза сцен (сгенерированных компьютером).

2.1.3 Кодирующая структура: высокая гибкость для представления объектов видео- и аудио данных.

2.2. *Принципы и структура:* разделение движения и текстурных (texture) данных (приоритет передачи каждого закодированного кадра).

2.3. *Алгоритмы:* сценический материал должен быть приемлемым для алгоритма, который может включать особые режимы для сцен переднего и заднего планов, групп людей и графики, текста и рисунков.

Теперь, можно построить модели структурных изменений (двух-стадийная эволюция) для некоторых частей стандартов MPEG: (a) *1.1* (Рис. 13.11a, Рис. 13.11b, Рис. 13.11c), (b) *1.3.1* (Рис. 13.12), (c) *1.4.3* (Рис. 13.13a, Рис. 13.13b), (d) *2.1.1* (Рис. 13.14), (e) *2.1.3* (Рис. 13.15). На Рис. 13.15, *D* соответствует быстрому поиску, *D'* соответствует возможному использованию модифицированного режима (*D*-кадр).

Улучшение коммуникационной технологии является существенной частью изменений в передаче мультимедийной информации. Укажем следующие основные проблемы в коммуникационных технологиях [133, 538, 579]:

1. Полная интеграция различных сетей (беспроводные, спутниковые, радиовещания, Интернет).

2. Качество сервиса (quality of service QoS) (задержка, дрожание задержки, потеря пакетов, интенсивность появления ошибочных битов, интенсивность появления группы ошибок, размер группы ошибок).

3. CBR или VBR.

4. Синхронизация видео, аудио и других данных.

5. Коммутация каналов или коммутация пакетов.

6. Мультиплексирование.

Таким образом, можно выделить два вида акций (действий, операций) изменения:

I. *Глобальные изменения:* новые сети, новое коммуникационное оборудование (новые генерации коммуникационных спутников,) коммуникации на основе лазеров и др.).

II. *Локальные изменения:* (a) использование использование VBR для CBR сетей, (b) эффективная буфферизация, (c) сглаживание (например, статистическое мультиплексирование).

Очевидно, что локальные улучшения могут значительно увеличить эффективность для существующих сетей (например, качество сервиса).

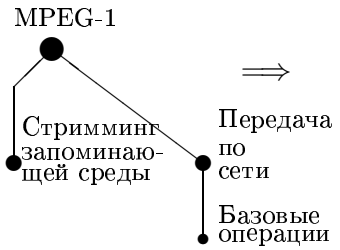


Рис. 13.11a. Прикладной уровень (*1.1*)

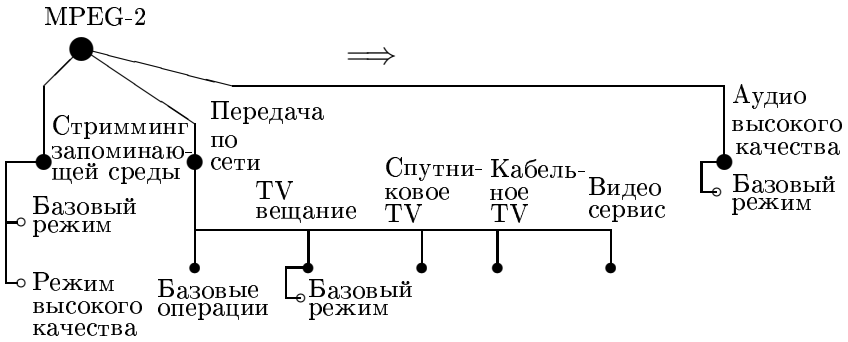


Рис. 13.11b. Прикладной уровень (*1.1*)

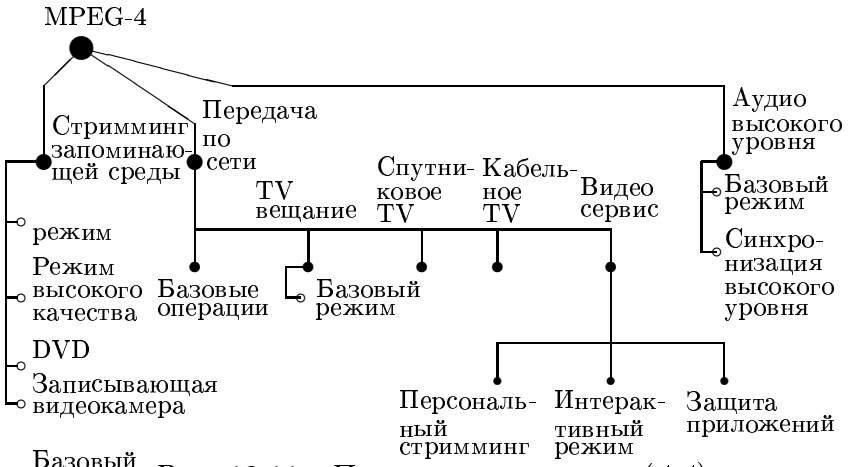


Рис. 13.11c. Прикладной уровень (*1.1*)

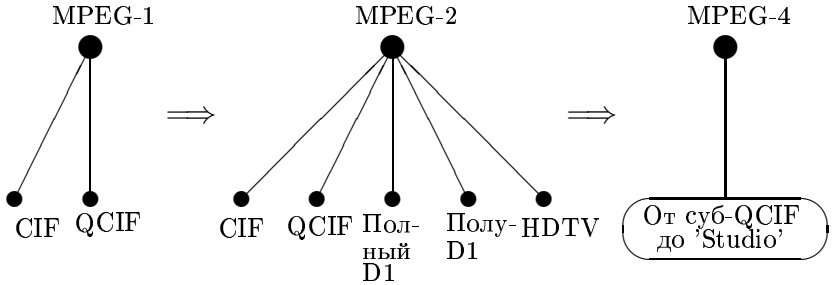


Рис. 13.12. Резолюция (*1.3.1*)

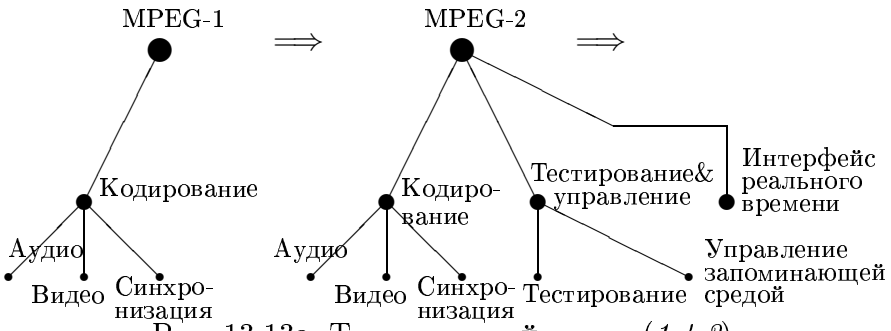


Рис. 13.13a. Транспортный поток (*1.4.3*)

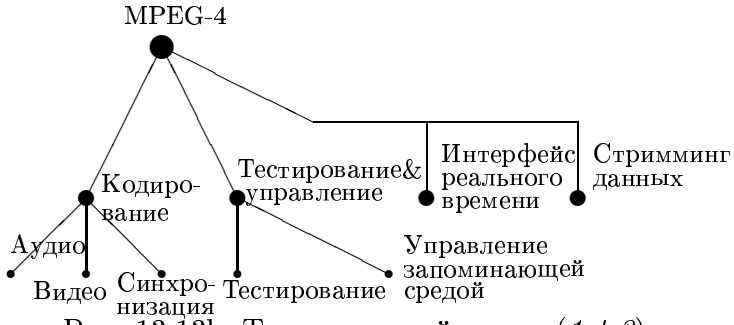


Рис. 13.13b. Транспортный поток (*1.4.3*)

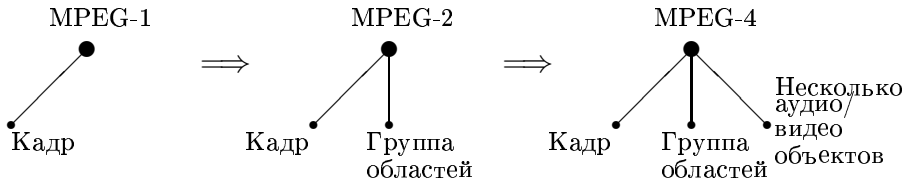


Рис. 13.14. Видео кодирование (*2.1.1*)

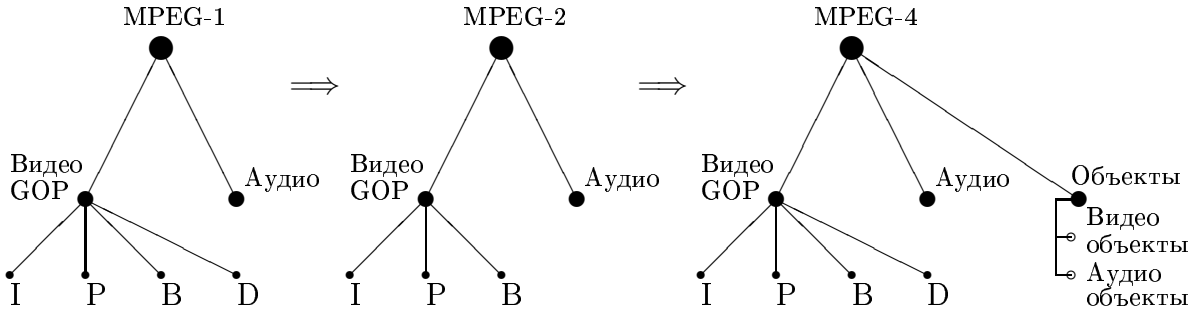


Рис. 13.15. Кодирующая структура/группы картинок GOP (*2.1.3*)

Теперь можно сформировать базовое множество операций изменения:

Группа 1. Расширение типов данных:

1.1. Хранение данных (интерфейс с прикладными системами): файлы, DVD, записывающая видеокамера (O_1).

1.2. Вещание (кодирование/декодирование): цифровое TV, кабельное TV, спутниковое TV, видео-конференции O_2.

1.3. Сеть: видео сервис по сети, персональный стримминг, защита приложений, интерактивный режим O_3.

Группа 2. Изменение резолюции:

2.1. Проектирование нового оборудование для улучшения качества картинок: O_4 для программного обеспечения, O_5 для электроники.

2.2. Изменение существующего оборудования для улучшения качества картинок: O_6 для программного обеспечения, O_7 для электроники.

2.3. Изменение существующей коммуникационной технологии для улучшения качества картинок: O_8 для программного обеспечения, O_9 для электроники.

Группа 3. Изменение временного режима:

3.1. Уменьшение задержки O_{10}.

3.2. Реализация режима реального времени (конференции, интерактивность) O_{11}.

Группа 4.

4.1. Уменьшение потерь O_{12}.

4.2. Уменьшение латентности O_{13}.

Группа 5. Передача:

5.1. Расширение сервисной информации (тестирование, управление и др.) O_{14}.

5.2. Дополнительные режимы (реальное время, стриминг данных) O_{15}.

5.3. Поддержка режимов поиска (например, при помощи D-кадров) O_{16}.

5.4. Поддержка режимов поиска с увеличением групп картинок (с 2 кадров до 7 кадров) O_{17}.

Группа 6. Объект & масштабируемость:

6.1. Реализация объектно-ориентированного подхода, т.е., семантический анализ передаваемой информации (от кадра к объекту/сцене, синхронизация маркеров для объектов, выявление глобального движения) O_{18}.

6.2. Реализация много-шкальности O_{19}.

6.3. Переменная блокировка O_{20}.

Рассмотрим три типа бинарных отношений на операциях изменения

(1) *эквивалентность* R^e;

(2) *дополнительность* (*совместимость*) R^c (Таблице 13.16);

(3) *условие следования для операций* $R^f : \alpha \to \beta$, например: $1 \to 6$, $1 \to 12$, $1 \to 14$ и др.

В Таблице 13.16 использованы следующие обозначения: $\star$ - отношение на базе объекта; $\diamondsuit$ - мягкое-жесткое (soft-hard) отношение; $\triangle$ - отношение по режиму обслуживания; $\spadesuit$ - отношение по временному режиму. Для *эквивалентно* R^e рассматриваются следующие пары операций: (O_4, O_5), (O_6, O_7), (O_8, O_9), (O_{18}, O_{20}). Некоторые структурные описания совместимости представлены на Рис. 13.16 и Рис. 13.17. Эти отношения могут быть использованы для структурных ограничений. Дополнительно, в Таблице 13.17 представлены оценки по времени для каждой операции изменения. При этом предполагаются три временные стадии:

(a) стадия 1 (прошлое) для интерполяционных задач;

(b) стадия 2 (ближайшее будущее) для краткосрочного прогноза (несколько лет: 2...4);

(c) стадия 3 (будущее) для долгосрочного прогноза (5...10 лет);

Очевидно, можно описать введенные операции изменения на основе двух типов свойств: *1.* эффективность (*полезность*), *2.* требуемые ресурсы. В результате, получаем многокритериальное описание и эти описания могут быть использованы в оптимизационных задачах (или задачах принятия решений) как целевые функции (критерии) или ограничения. Здесь рассмотрен список критериев:

Группа 1. Полезность: качество видео C_1; качество аудио C_2; качество синхронизации C_3; качество данных C_4; качество поиска C_5.

Группа 2. Технологическая среда: новые технологические возможности C_6.

Группа 3. 'Стоимость': дополнительный НИР C_7; требуемое новое программное обеспечение C_8; требуемое новое оборудование C_9; объем требуемого производства C_{10};

Веса важности критериев указаны в Таблице 13.18. Таблица 13.19 содержит оценки операций изменения по критериям и приоритеты (результат многокритериального ранжирования для трех временных стадий соответственно;

Таблица 13.16. Бинарное отношение *совместимость* R^c

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}
O_1																				
O_2																				
O_3										♠	♠			♠	♠	♠				
O_4					◊					⋆		⋆	⋆	⋆	⋆	⋆		⋆	⋆	⋆
O_5				◊						⋆		⋆	⋆	⋆	⋆	⋆			⋆	
O_6							◊													
O_7						◊														
O_8									◊											
O_9								◊												
O_{10}			♠	⋆	⋆						♠				♠					
O_{11}			♠							♠					♠					
O_{12}				⋆	⋆															
O_{13}				⋆	⋆															
O_{14}			♠	⋆	⋆											△				
O_{15}			♠	⋆	⋆					♠	♠									
O_{16}			♠	⋆	⋆									△						
O_{17}																				
O_{18}				⋆																⋆
O_{19}				⋆	⋆															
O_{20}				⋆														⋆		

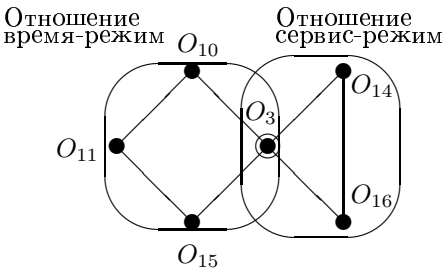


Рис. 13.16. *Совместимость* с базовым O_3

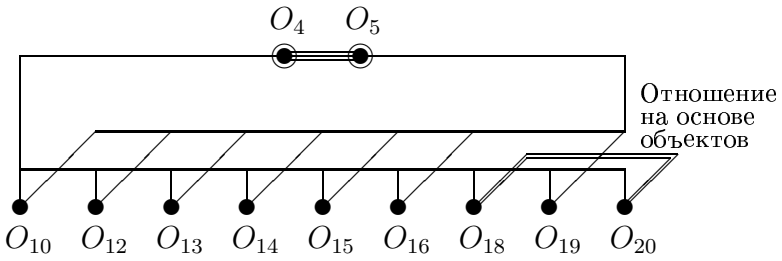


Рис. 13.17. *Совместимость* с базовыми узлами O_4 и O_5

Таблица 13.17. Соответствие временной стадии

	1: Прошлое	2: Близкое будущее	3: Будущее
O_1	$\star$	$\star$	$\star$
O_2	$\star$	$\star$	$\star$
O_3	$\star$	$\star$	$\star$
O_4	$\star$	$\star$	$\star$
O_5	$\star$	$\star$	$\star$
O_6	$\star$	$\star$	$\star$
O_7	$\star$	$\star$	$\star$
O_8	$\star$	$\star$	$\star$
O_9	$\star$	$-$	$\star$
O_{10}	$\star$	$\star$	$-$
O_{11}	$\star$	$\star$	$-$
O_{12}	$\star$	$\star$	$\star$
O_{13}	$\star$	$\star$	$\star$
O_{14}	$-$	$\star$	$\star$
O_{15}	$\star$	$\star$	$-$
O_{16}	$-$	$-$	$\star$
O_{17}	$-$	$\star$	$\star$
O_{18}	$-$	$\star$	$\star$
O_{19}	$-$	$\star$	$\star$
O_{20}	$-$	$\star$	$\star$

13.4.3 Численные примеры

Таблица 13.19 содержит результаты многокритериального ранжирования операция изменения для трех стадий в соответствии с Таблицей 13.18:

Стадия 1. Задача интерполяции. Здесь следующие операции являются лучшими:

1.1 O_6 (изменение существующих программ для улучшения качества картинок).

1.2 O_8 (изменение программ для существующей телекоммуникационной технологии).

1.3 O_{12} (уменьшение потерь).

Стадия 2. Планирование на несколько лет. Здесь следующие операции являются лучшими:

2.1 O_8.

2.2 O_9 (изменение оборудования для существующих телекоммуникационных технологий).

Стадия 3. Планирование на 5...10 лет. Здесь следующие операции являются лучшими:

3.1 O_1 (расширение интерфейса с прикладными системами (память)).

3.2 O_8.

3.3 O_9.

Таблица 13.18. Веса критериев

	Стадия 1	Стадия 2	Стадия 3
C_1	5	5	5
C_2	3	4	5
C_3	3	4	5
C_4	2	2	4
C_5	0	0	3
C_6	1	3	4
C_7	-1	-2	-4
C_8	-3	-4	-4
C_9	-4	-4	-4
C_{10}	-5	-5	-5

Таблица 13.19. Оценки по критериям и приоритеты

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	Приоритеты		p_i
O_1	5	5	5	5	5	5	5	3	4	5	2	2	1
O_2	5	5	5	5	–	4	5	4	4	5	2	2	2
O_3	–	–	–	5	5	5	5	5	1	2	2	2	2
O_4	4	4	4	4	5	5	5	5	2	3	2	2	2
O_5	4	4	4	4	2	4	5	4	5	4	2	2	2
O_6	3	3	3	2	0	2	3	4	0	1	1	2	2
O_7	3	3	3	0	0	2	3	2	4	3	2	2	2
O_8	5	5	5	5	5	5	5	5	0	4	1	1	1
O_9	5	5	5	5	5	5	5	2	5	5	2	1	1
O_{10}	3	3	5	3	1	4	5	4	3	2	2	2	2
O_{11}	–	–	–	–	3	5	4	4	5	5	3	3	3
O_{12}, O_{13}	5	5	5	4	1	2	5	5	1	2	1	2	2
O_{14}	3	3	5	5	5	5	5	5	3	3	2	2	2
O_{15}	–	–	–	5	5	5	5	5	1	3	3	3	2
O_{16}	–	–	–	–	5	5	5	4	–	–	3	3	3
O_{17}	5	–	–	–	3	4	3	5	3	4	2	2	2
O_{18}	5	2	4	4	5	5	5	5	2	3	2	2	2
O_{19}	3	3	3	4	–	5	4	4	4	4	2	2	2
O_{20}	5	–	–	4	3	3	4	5	2	4	2	2	2

В то же время, целесообразно исследовать следующие операции, которые являются близкими по важности к лучшим ("квази-лучшие"):

Стадия 1: O_1, O_4, O_9.

Стадия 2: O_1, O_6, O_{12}, O_{13}.

Стадия 3: O_4, O_{14}, O_{18}.

Рис. 13.18 содержит указанные операции изменения. Теперь можно сформулировать задачу:

Построить траектории из операций изменения с учетом их совместимости.

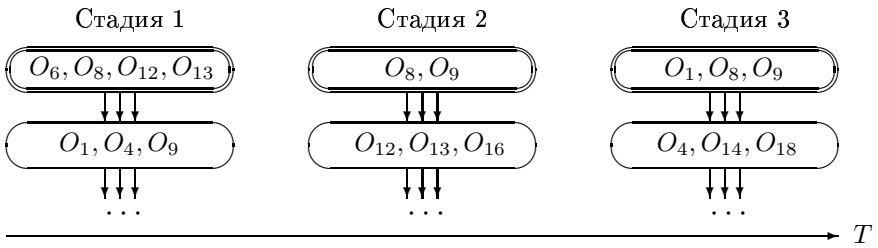


Рис. 13.18. Лучшие и 'квази-лучшие' операции изменения

Рассмотрим много-стадийную эвристику для выбора и координации операций изменения с учетом групп (кластеров) операций, которые получаются на основе введенных бинарных отношений (Рис. 13.16 и Рис. 13.17).

Фаза 1. Многокритериальное ранжирование операций изменения и выбор лучших и 'квази-лучших' операций для каждой временной стадии (Таблица 13.19, Рис. 13.18).

Фаза 2. Расширение множества операций на основе логических правил (т.е., бинарных отношений).

Фаза 3. Генерация альтернативных операций изменения для каждой временной стадии.

Фаза 4. Построение лучшей траектории с учетом взаимосвязи операций на различных временных стадиях.

Рассмотрим стадию 2. Здесь используются кластеры операций из Рис. 13.16 и Рис. 13.17. результирующее расширенное множество операций показано на Рис. 13.19.

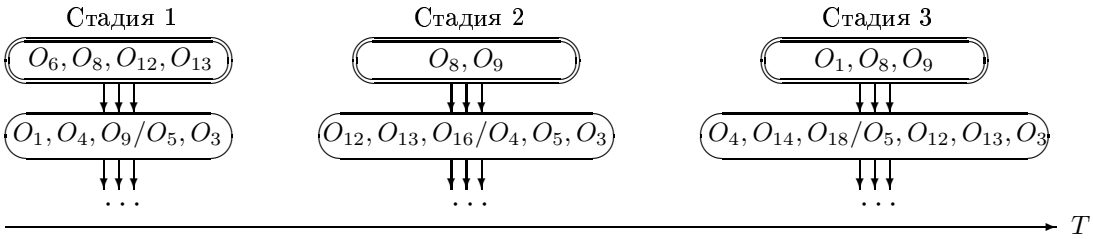


Рис. 13.19. Расширенные множества лучших и 'квази-лучших' операций

Теперь можно генерировать операции изменения (этот пример имеет иллюстративный характер, в реальной ситуации необходимо организовать специальную процедуру экспертного оценивания):

Стадия 1 (некоторые две исходные точки):
$A_1^1 = \{O_6, O_8, O_{12}, O_{13}, O_4, O_5, O_3\}$; $A_2^1 = \{O_6, O_8, O_1\}$.
Стадия 2:
$A_1^2 = \{O_8, O_9, O_{12}, O_{13}, O_4, O_5, O_{16}, O_3\}$; $A_2^2 = \{O_8, O_9\}$.
Стадия 3:
$A_1^3 = \{O_1, O_8, O_9, O_{12}, O_{13}, O_4, O_5, O_{14}, O_{18}, O_3\}$; $A_2^3 = \{O_1, O_8, O_9\}$; $A_3^3 = \{O_1\}$.

Рис. 13.20 иллюстрирует операции изменения и некоторые системные ('технологические') траектории. 'Технологические' траектории должны быть получены на основе лучших операций на каждой временной стадии и очень хорошей совместимости между операциями, выбранными для различных стадий. Эта задача соответствует ИММП. В примере рассматриваются порядковые оценки совместимости, полученные от экспертов (Таблица 13.20; 3 соответствует лучшей совместимости). Таким образом, получены траектории:

$$\alpha_1 =< A_1^1 \Rightarrow A_1^2 \Rightarrow A_1^3 >, \quad \alpha_2 =< A_2^1 \Rightarrow A_2^2 \Rightarrow A_3^3 > .$$

На финальной фазе можно проводить анализ полученных траекторий. Например, все траектории содержат на стадии 2 операции изменения, которые ориентированы на улучшение существующей коммуникационных сетей (O_8 и O_9).

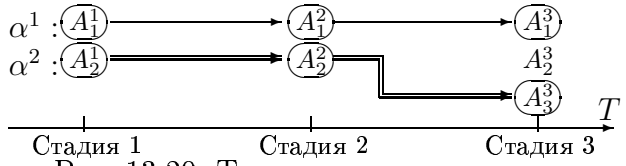


Рис. 13.20. Траектории изменения

Таблица 13.9. Совместимость

	A_1^2	A_2^2	A_1^3	A_2^3	A_3^3
A_1^1	3	1	3	2	0
A_2^1	2	3	2	2	1
A_1^2			3	2	0
A_2^2			3	3	3

13.4.4 Обсуждение

В данном разделе описан подход к комбинаторному моделирвоанию станартов передачи информации на примере стандартов типа MPEG. Такой подход представляется перспективным для применения к другим стандартам передачи информации.

13.5 Модульность в протоколах и стандартах

В течение последней декады, интенсивно исследуются различные вопросы протоколов для беспроводных сетей и стандарты обработки мультимедийной информации [79, 103, 121,

133, 194, 254, 260, 279, 285, 312, 474, 525, 579]. В последние несколько лет, появились исследования, направленные на многокритериальный анализ и выбор наилучшего протокола и стандарта [283, 302, 495]. Следует также отметить новый тренд в области коммуникационных протоколов и стандартов на основе их представления как комбинации элементов (например, правил, алгоритмов и др.) [211, 390, 393, 450]. Такой подход является основой для иерархической композиции (конфигурации) протоколов из элементов. Это направление близко к следующим современным подходам на основе композиции: адаптивные программы [424, 535], алгоритмы [260, 442], сервисы [299, 529].

В данном разделе описывается модульное проектирование (композиция) коммуникационного протокола и стандарта типа MPEG для обработки мультимедийной информации. [363, 390, 393]). Некоторые базовые подходы к использованию и исследованию коммуникационных протоколов и стандартов для обработки мультимедийной информации указаны в Таблице 13.21. Два представленных в данной статье два числовых примера являются модификациями примеров из статей [363, 390]. Эти примеры включают: иерархическую (древовидную) структуру системы, проектные альтернативы (DAs) для частей/компонентов системы, процесс решения. Экспертные суждения использованы для оценивания DAs. Оценки и процесс вычисления имеют иллюстративный характер, но могут быть использованы как основа для реальных приложений.

Таблица 13.21. Подходы к протоколам/стандартам

Основные задачи	Источник
1. Применение базового протокола для беспроводной связи	[79, 103, 279, 285],
2. Гибридный протокол для беспроводной связи	[121]
3. Динамический выбор протокола, обновление (upgrade)	[302, 493, 495, 515]
4. Реконфигурируемый стек протоколов	[130, 211, 593]
5. Само-организуемый стек протоколов	[56]
6. Динамическая архитектура протокола, общей схемы	[56, 450]
7. Комбинаторный синтез протокола ZigBee для беспроводных сенсорных сетей	[363]
8. Комбинаторная эволюций протокола ZigBee для беспроводных сенсорных сетей	[393]
9. Структурный подход к стандартам для мультимедийной информации	[132, 133, 194, 312], [322, 435, 579]
10. Многокритериальный выбор алгоритмов обработки для мультимедийной информации	[283]
11. Комбинаторная эволюция стандартов для мультимедийной информации	[390]

Традиционная схема обработки входной информации (например, сигнал, последовательность изображений, прикладная ситуация) имеет вид:

Входные данные $\Longrightarrow$ *Метод обработки* $\Longrightarrow$ *Результаты*

Очевидно, адаптация *метода обработки* (или алгоритма, набора правил обработки) может быть основана на специальном предварительном анализе *входа*, чтобы модифицировать *метод обработки* для соответствия имеющемуся *входу*. В работе [364], представлены следующие типовые случаи для модификации *метода обработки*: (1) использование базового (т.е., исходного) метода, (2) модификация имеющегося метод для получения метода, который соответствует имеющемуся *входу*, (3) выбор метода для применений из библиотеки методов с учетом типа имеющегося *входа*, (4) модульное проектирование метода на основе библиотеки компонентов методов с учетом типа имеющегося *входа*.

Рис. 13.21 and Рис. 13.22 иллюстрируют два указанных случая (3) и (4). Отметим, многокритериальный анализ алгоритмов для обработки последовательности изображений представлен в работе [283]: (а) определение набора типовых последовательностей изображений, (б) многокритериальный анализ алгоритмов для обработки последовательности изображений для выделения множества Парето-оптимальных алгоритмов для каждого типа последовательностей изображений. Такой анализ является базой для применения схемы, соответствующей Рис. 13.21, для обработки последовательностей изображений в реальном времени.

Аналогично, можно рассмотреть такой подход для коммуникационного протокола. Например, имеется стек (база) протоколов и в реальном времени осуществляется выбор наилучшего протокола для имеющейся конкретной входной ситуации [130, 211, 302, 493, 495, 593].

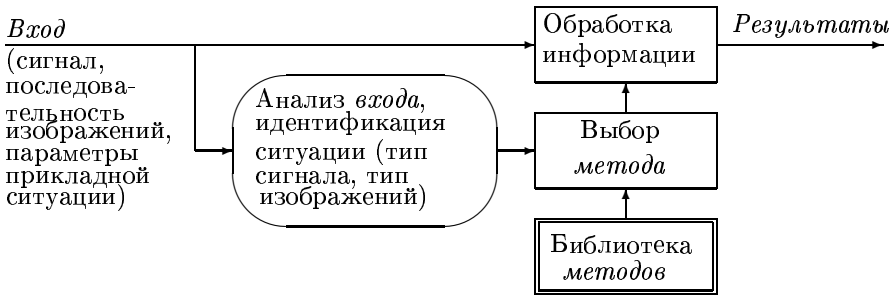


Рис. 13.21. Обработка на основе выбора метода

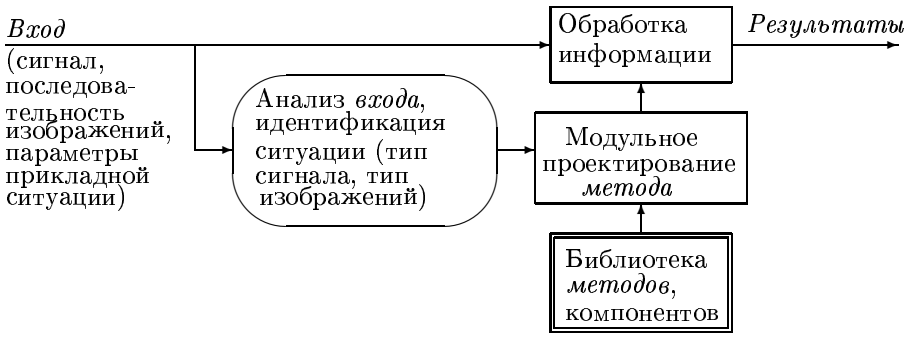


Рис. 13.22. Обработка на основе модульного проектирования метода

В работе [363], коммуникационный протокол ZigBee формируется на основе базовых компонентов (иерархическое модульное проектирование), т.е., реализуется схема, соответствующая Рис. 13.22.

13.5.1 Синтез коммуникационного протокола

Данный раздел содержит описание модульного проектирования коммуникационного протокола ZigBee. Используется упрощенная иерархическая модель протокола ZigBee (Рис. 13.23):

1. Разрешение интерференции $A = X \star J$:

1.1. Начальная процедура выявления канала X: X_0 (нет), X_1 (PAN координатор выбирает наилучший доступный RF канал / идентификация сети в начальный момент).

1.2. Желаемый канал J: J_0 (нет), J_1 (Выявление происходящей интерференции и использование нового операционного RF канала и/или идентификатора сети).

2. Автоматическое / распределенное адресное управление B: B_0 (нет), B_1 (Автоматическое назначение адреса устройства на основе использования иерархической, распределенной схемы), B_2 (Автоматическое назначение адреса устройства на основе использования стохастической схемы).

3. Групповая адресация I: I_0 (нет), I_1 (устройства могут назначаться для группы, целая группа может быть адресована на основе одной схемы).

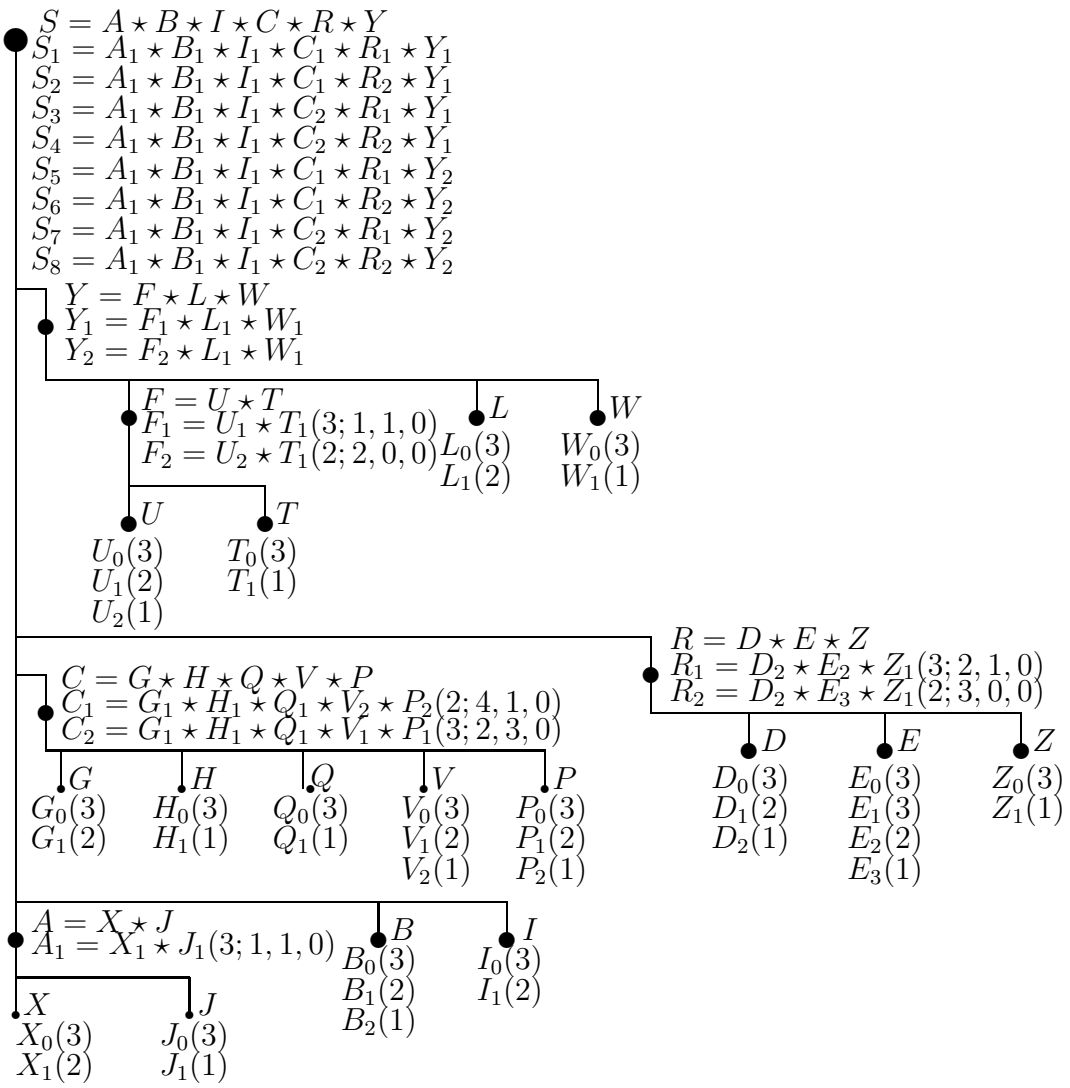


Рис. 13.23. Общая структура протокола ZigBee

4. Централизованный сбор данных $C = G \star H \star Q \star V \star P$:

4.1. Сбор данных на основе 6LoWPAN координатора G: G_0 (нет), G_1 (полная поддержка).

4.2 Сбор данных на основе других устройств H: H_0 (нет), H_1 (с помощью специального оборудования).

4.3. Маршрутизация “много-один” Q: Q_0 (нет), Q_1 (Сеть выявляет агрегатор за один шаг).

4.4. Поддержка режимов multicast/broadcast на основе 6LoWPAN V: V_0 (нет), V_1 (flooding), V_2 (путь к PAN координатору).

4.5. Маршрутизация источника P: P_0 (нет), P_1 (Агрегатор отвечает всем отправителям в экономном режиме).

5. Масштабируемость, размеры $R = D \star E \star Z$:

5.1. Масштабируемость сети D: D_0 (нет), D_1 (шкала сети ограничена адресующим алгоритмом; обычно, сети с десятками/сотнями устройств поддерживаются), D_2 (адресующий алгоритм уменьшает размер сети, поддерживаются сети с сотнями/тысячами устройств).

5.2. Размер сообщения E: E_0 (нет), E_1 (< 100 бит; точный размер зависит от используемого сервиса, который поддерживается), E_2 (большой размер сообщения, до возможного размеров буферов устройств получателя и отправителя, которые поддерживаются на основе фрагментации и ре-сборки), E_3 (большой размер сообщения, до возможного размеров буферов устройств получателя и отправителя, на основе использования 6LoWPAN фрагментации и ре-сборки).

5.3. Стандартизированная комиссия Z: Z_0 (нет), Z_1 поддержка на основе стандартной "старт-ап" процедуры и параметров для использования комиссионных средств в мультипользовательской среде.

6. Поддержка сети $Y = F \star L \star W$:

6.1. Робастная ячеистая сеть F: *6.1.1.* 6LoWPAN - подходы U: U_0 (нет), U_1 ("над-маршрут"), U_2 ("под-ячейка"). *6.1.2.* Таблицы соседства T: T_0 (нет), T_1 (на каждом устройстве).

6.2. поддержка библиотеки кластеров L: L_0 (нет), L_1 (типовое поведение между профилями).

6.3. Поддержка Веб сервисов W: W_0 (нет), W_1 (сжатые HTTP с XML данными).

Порядковые оценки для DAs (шкала $[1, 2, 3]$) представлены на Рис. 13.23 (в круглых скобках, экспертные суждения, иллюстративный характер). Оценки совместимости (порядковая шкала $[0, 1, 2, 3]$) представлены в Таблице 13.22 и Таблице 13.23 (экспертные суждения, иллюстративный характер).

Получаются следующие промежуточные составные DAs для подсистем: (1) $A_1 = X_1 \star J_1$, $N(A_1) = (3; 1, 1, 0)$; (2) $C_1 = G_1 \star H_1 \star Q_1 \star V_2 \star P_2$, $N(C_1) = (2; 4, 1, 0)$; (3) $C_2 = G_1 \star H_1 \star Q_1 \star V_1 \star P_1$, $N(C_2) = (3; 2, 3, 0)$; (4) $R_1 = D_2 \star E_3 \star Z_1$, $N(R_1) = (3; 2, 1, 0)$; (5) $R_2 = D_1 \star E_2 \star Z_1$, $N(R_1) = (2; 3, 0, 0)$; (6) $F_1 = U_2 \star T_1$, $N(F_1) = (3; 1, 1, 0)$; (7) $F_2 = U_2 \star T_1$, $N(F_1) = (2; 2, 0, 0)$; (8) $Y_1 = F_1 \star L_3 \star W_1$; (9) $Y_2 = F_2 \star L_3 \star W_1$.

Рис. 13.24 иллюстрирует "пространство" качества для подсистемы C.

Таблица 13.22. Оценки совместимости [363]

	H_0	H_1	Q_0	Q_1	V_0	V_1	V_2	P_0	P_1	P_2
G_0	3	2	3	2	3	1	3	0	1	1
G_1	3	3	3	3	3	3	2	3	3	2
H_0			2	1	1	2	1	1	1	1
H_1			0	3	1	3	3	1	3	2
Q_0					2	0	1	1	1	1
Q_1					0	3	3	1	3	2
V_0								2	1	1
V_1								1	3	2
V_2								1	2	3

	E_0	E_1	E_2	E_3	Z_0	Z_1
D_0	1	1	1	1	0	1
D_1	2	3	3	3	1	3
D_2	2	3	3	3	1	3
E_0					1	1
E_1					1	2
E_2					1	3
E_3					1	2

Таблица 13.23. Оценки совместимости [363]

	J_0	J_1
X_0	0	2
X_1	2	3

	T_0	T_1
U_0	0	1
U_1	1	3
U_2	2	2

	L_0	L_1	W_0	W_1
F_1	0	3	1	3
F_2	1	3	1	3
L_0			0	1
L_1			0	3

	B_0	B_1	B_2	I_0	I_1
A_1	0	3	3	1	3
B_0				1	1
B_1				1	3
B_2				1	3

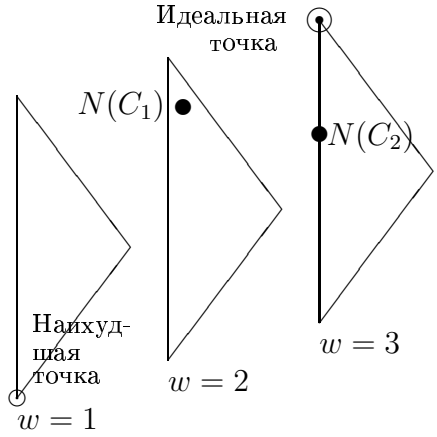


Рис. 13.24. Качество C

Для системы получаются восемь альтернативных решений: $S_1 = A_1 \star B_1 \star I_1 \star C_1 \star R_1 \star Y_1$, $S_2 = A_1 \star B_1 \star I_1 \star C_1 \star R_2 \star Y_1$, $S_3 = A_1 \star B_1 \star I_1 \star C_2 \star R_1 \star Y_1$, $S_4 = A_1 \star B_1 \star I_1 \star C_2 \star R_2 \star Y_1$, $S_5 = A_1 \star B_1 \star I_1 \star C_1 \star R_1 \star Y_2$, $S_6 = A_1 \star B_1 \star I_1 \star C_1 \star R_2 \star Y_2$, $S_7 = A_1 \star B_1 \star I_1 \star C_2 \star R_1 \star Y_2$, $S_8 = A_1 \star B_1 \star I_1 \star C_2 \star R_2 \star Y_2$.

Заметим, исходное множество возможных решений включает 1232272 комбинаций.

13.5.2 Композиция стандарта для мультимедийной информации

Данный раздел содержит описание модульного проектирования стандарта для обработки мультимедийной информации (MPEG 1, MPEG 2, MPEG 4). Рассматривается следующая технологическая схема: (1) *входные данные* (аудио, видео, данные синхронизации текста, синтетическая информация); (2) *обработка* (кодирование, передача по сети декодирование); (3) *выходные данные* (для человека, для прикладной системы). Очевидно, используются следующие требования (критерии): (i) качество результата (т.е., результирующей информации - после декодирования), (ii) затраты (стоимость, ошибки по времени и др.). Обычно, качество результирующих изображений оценивается на основе следующего: PSNR, задержки, ошибки, латентность, потери (например, потеря пакета, потеря кадра).

Обобщенная иерархическая структура стандарта типа MPEG имеет вид (Рис. 13.25; включая DAs для компонентов, оценки для DAs указаны в круглых скобках, порядковая шкала $[1, 2, 3]$) [390]):

0. Стандарт типа MPEG $S = A \star B$.

1. Общая часть $A = C \star B \star D \star E \star F$:

1.1. Прикладной уровень (видео-телефония, видео-конференция, цифровое вещание, цифровое хранение видео-информации, кабельное и спутниковое телевидение, видео-сервис по сети, DVD и др.) C: bit rate 64 kbit/s $C_1(2)$, bit rate $4 - 80$ Mbit/s $C_2(1)$.

1.2. Режим качества по времени и изображению $D = X \star Y$: *1.2.1.* режим по времени X: задержка $X_1(3)$, реальное время, малая задержка $X_2(1)$; *1.2.2.* качество изображения Y: низкое $Y_1(3)$, хорошее $Y_2(1)$.

1.3. Формат $E = U \star V$: *1.3.1.* Резолюция U: низкая $U_1(3)$, высокая U_1; *1.3.2.* декомпозиция цвета V: базовая $V_1(2)$, высокий профиль $V_2(1)$.

1.4. Базовые операции $F = T \star P \star M$: *1.4.1.* Трансформирование T: базовый режим $T_1(1)$, цифровой Dolby $T_2(2)$; *1.4.2.* Особенности P: базовый режим $P_1(1)$, с масштабируемостью $P_2(2)$; *1.4.3.* Передача потоков (видео, аудио, синхронизация, тестирование, управление) M: базовый режим $M_1(1)$, использование медиа объектов $M_2(2)$.

2. Кодирование/сжатие $B = G \star I \star J$.

2.1. Базовые компоненты $G = Q \star W \star Z$: *2.1.1.* видео кодирование (методы трансформации в цифровые коды) Q: кодирование словами переменной длины коэффициентов структуры $Q_1(1)$, VLC таблицы для DCT (нелинейные) $Q_2(2)$; *2.1.2.* оценка движений (вектор и др.) Z: от -1024 до 1023 пикселей (для половины) $Z_1(3)$, от -2048 до 2047 пикселей (для полной) $Z_2(1)$.

2.2. Принципы и структура $I = K \star L$:

2.2.1. Принципы $K = O \star H$: *2.2.1.1.* Блоковая декомпозиция O: 16 $times$16 (макроблок) и 8×8 (блок) $O_1(1)$, на основе объектов (VOB) $O_2(2)$; *2.2.1.2.* сканирование H: прогрессивное сканирование (zigzag) $H_1(1)$, альтернативное $H_2(2)$.

2.2.2. Структура (базовая схема обработки, расширенная схема обработки, 'открытая структура' включая транскодирование) L: базовый режим $L_1(1)$, разделение движения и данных текстуры $L_2(2)$.

2.3. Алгоритмы J: упрощенные $J_1(2)$ сложные $J_2(1)$.

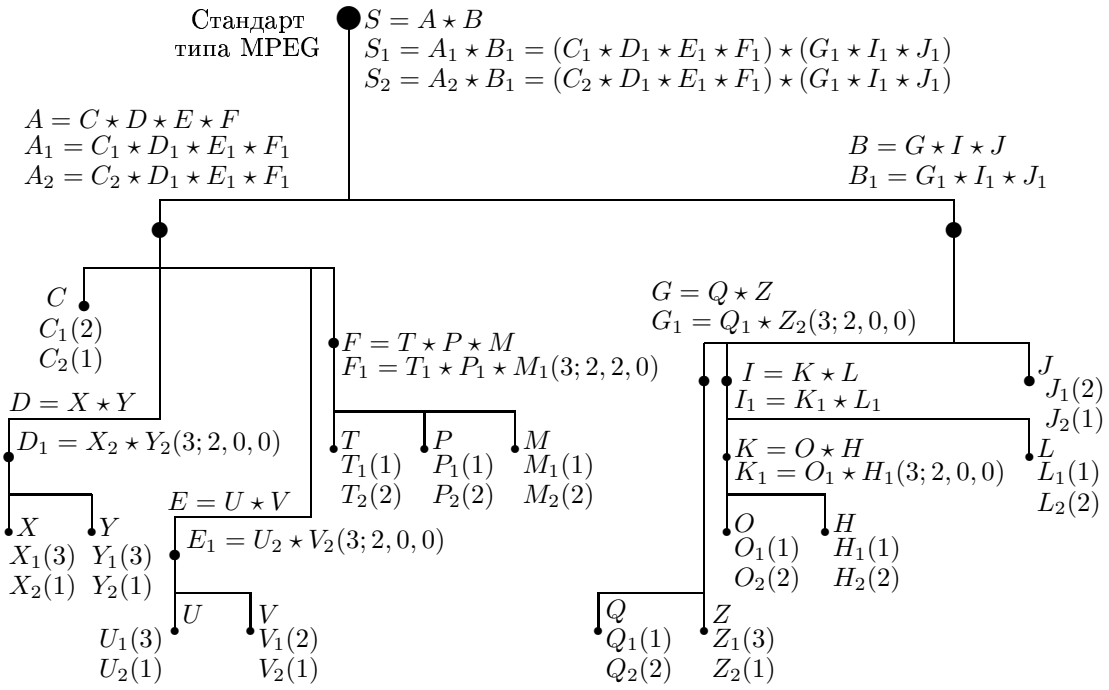


Рис. 13.25. Общая структура стандарта типа MPEG

Выше были использованы следующие обозначения: Dolbi Digital (формат Dolbi Digital), VLC (Variable-Length Coding), DCT (Discrete Cosine Transform).

Оценки совместимости приведены в Таблице 13.24 (экспертные суждения).

Получаются следуюшие промежуточные составные DAs для подсистем: (1) $D_1 = X_1 \star Y_1$, $N(D_1) = (3;2,0,0)$; (2) $E_1 = U_1 \star V_1$, $N(E_1) = (3;2,0,0)$; (3) $F_1 = T_1 \star P_1 \star M_1$, $N(F_1) = (3;3,0,0)$; (4) $G_1 = Q_1 \star Z_1$, $N(G_1) = (2;2,0,0)$; (5) $K_1 = O_1 \star H_1$, $N(K_1) = (2;3,0,0)$.

Таблица 13.24. Оценки совместимости

	Y_1	Y_2
X_1	2	3
X_2	3	3

	V_1	V_2
U_1	1	2
U_2	2	3

	P_1	P_2	M_1	M_2
T_1	3	2	3	1
T_2	1	3	1	3
P_1			3	2
P_2			1	3

	Z_1	Z_2
Q_1	3	3
Q_2	2	3

	H_1	H_2
O_1	3	2
O_2	2	3

Далее, комбинации для следующих более высоких иерархических уровней модели имеют вид: (а) $A_1 = C_1 \star D_1 \star E_1 \star F_1$, (б) $A_2 = C_2 \star D_1 \star E_1 \star F_1$, (в) $B_1 = G_1 \star I_1 \star J_1$.

Окончательно, получаются два альтернативных решения для системы:

$S_1 = A_1 \star B_1 = (C_1 \star D_1 \star E_1 \star F_1) \star (G_1 \star I_1 \star J_1)$;
$S_2 = A_2 \star B_1 = (C_2 \star D_1 \star E_1 \star F_1) \star (G_1 \star I_1 \star J_1)$.
Заметим, исходное множество возможных решений включает 16384 комбинации.

13.5.3 Обсуждение

В разделе представлен модульный подход к коммуникационному протоколу и стандарту для обработки мультимедийной информации. Представляется перспективным использование предложенного подхода к другим протоколам для беспроводной связи и применение модульного проектирования коммуникационных протоколов и стандартов обработки мультимедийной информации в в режиме реального времени.

13.6 Примеры улучшения/расширения сетей

13.6.1 Улучшение сети

Здесь представлен упрощенный пример улучшения телефонной сети московского региона [351]. Таблица 13.25 содержит перечень рассматриваемых регионов $\{A_1, ..., A_9\}$ и их оценки по параметрам (экспертное оценивание, порядковая шкала [1,10]): население P_1, уровень существующей коммуникационной инфраструктуры P_2, размер региона (площадь) P_3.

После кластеризации (использован агломеративный алгоритм для иерархическая кластеризация [347]), получаются следуюшие группы (кластеры) регионов: $G^1 = \{A_1\}$, $G^2 = \{A_2\}$, $G^3 = \{A_3, A_8\}$, $G^4 = \{A_4\}$, $G^5 = \{A_5, A_7, A_9\}$, and $G^6 = \{A_6\}$. Таблица 13.26 содержит пять типовых акций (варианта) развития/улучшения на уровне региона:

$\{D_1, D_2, D_3, D_4, D_5\}$ (D_1 - вариант соответствует отсутствию деятельности).

Указанные акции оцениваются по критериям (порядковая шкала [0,5], экспертное оценивание): общая полезность (C_1), сложность (C_2), перспективная полезность (C_3), требуемые затраты (C_4).

Таблица 13.25. Регионы, оценки

	Регион	Параметры		
		P_1	P_2	P_3
A_1	Центральный	10	10	9
A_2	Южный	8	9	10
A_3	Юго-западный	6	7	4
A_4	Юго-восточный	3	8	8
A_5	Северный	4	4	5
A_6	Северо-восточный	1	1	9
A_7	Северо-западный	4	5	4
A_8	Западный	5	6	4
A_9	Восточный	2	4	5

Таблица 13.26. Типовые акции улучшения

	Акция улучшения	Критерии			
		C_1	C_2	C_3	C_4
D_1	Нет	0	0	0	0
D_2	Новые линии	5	5	5	5
D_3	Ремонт линий	2	2	2	2
D_4	Расширение (новые линии и устройства)	3	3	3	3
D_5	Удаление старых линий	1	1	1	1

Двух этапная схема решения представлена на Рис. 13.26. Формирование составного плана улучшения (развития) основывается на следующей задаче блочного рюкзака (используется простейшая 'жадная' эвристика):

$$\max \sum_{i=1}^{6}\sum_{j=1}^{5} c_{ij}^1 x_{ij}, \quad \min \sum_{i=1}^{6}\sum_{j=1}^{5} c_{ij}^2 x_{ij}, \quad \max \sum_{i=1}^{6}\sum_{j=1}^{5} c_{ij}^3 x_{ij}, \quad \min \sum_{i=1}^{6}\sum_{j=1}^{5} c_{ij}^4 x_{ij}$$

$$s.t. \quad \sum_{i=1}^{6}\sum_{j=1}^{5} a_{ij} x_{ij} \le b, \quad \sum_{j=1}^{5} x_{ij} = 1 \ \forall i = \overline{1,6}, \quad x_{ij} \in \{0,1\}.$$

В результате получается следующее решение: $S_1 =< D_4^1 \star D_3^2 \star D_5^3 \star D_1^4 \star D_1^5 \star D_1^6 >$. В случае зависимости решений для соседних регионов, необходимо использовать комбинаторный синтез на основе ИММП.

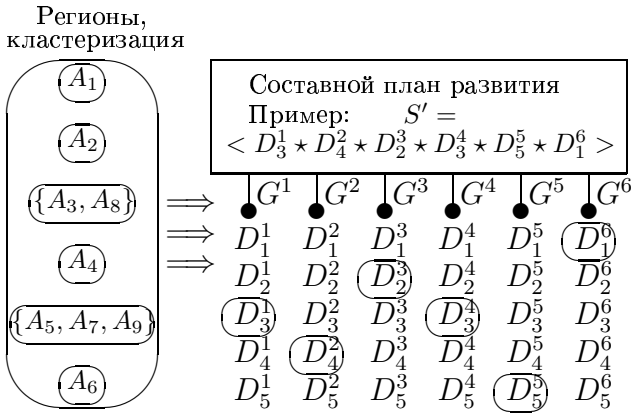


Рис. 13.26. Иллюстраций 2-х этапной схемы

13.6.2 Расширение сети

Рассматривается следующий пример для региональной коммуникационной сети [351]: (i) существует коммуникационная сеть для некоторого региона (множество конечных пользователей, множество точек доступа, линии подключения пользователей к точкам доступа), (ii) необходимо спроектировать дополнительную коммуникационную сеть для соседнего региона (как для дополнительного региона, т.е., расширение исходной системы). Могут быть применены две стратегии расширения:

Стратегия I. Проектирование дополнительной коммуникационной сети для нового региона.

Стратегия II. Проектирование новой "общей" коммуникационной сети для интегрированного региона, включающего существовавший регион и дополнительный регион.

Стратегия III. Проектирование коммуникационной сети для нового региона и модификация существующей коммуникационной сети для исходного региона.

Рассматриваются два региона (пользователи, точки доступа):

(i) исходный регион (17 пользователей, 3 точки доступа, Таблица 13.27, Таблица 13.28, Рис. 13.27);

(ii) дополнительный регион (11 пользователей, 3 точки доступа, Таблица 13.27, Таблица 13.28, Рис. 13.28).

Используются стратегии: *стратегия I* (независимое проектирование для регионов), *стратегия II* (интегрированное проектирование).

Для исходного региона используются следующие параметры: множество пользователей $\Psi = \{1, ..., i, ..., n\}$ $(n = 17)$, множество точек доступа $\Theta = \{1, ..., i, ..., m\}$ $(m = 3)$. Для каждого пользователя имеется вектор параметров $(x_i, y_i, z_i, f_i, p_i)$, где следующие компоненты (Таблица 13.27): координаты $(x_i, y_i, z_i$, требуемая ширина полосы частот f_i (шкала: 1 Mbit/s ... 10 Mbit/s), приоритет p_i (порядковая шкала [1,2,3], все требования пользователя удовлетворяются в случае $p_i = 1$), требуемая надежность r_i (порядковая шкала [1,10], 10 соответствует максимальной надежности. Параметры для точек доступа вводятся аналогично (с индексом j, Таблица 13.27), включая параметр n_j (максимальное число обслуживаемых пользователей).

Таким образом, каждая пара "пользователь - точка доступа" (т.е., $(i, j), i \in \Psi, j \in \Theta$) описывается так: (1) надежность $r_{ij} = min\{r_i, r_j\}$, (2) расстояние l_{ij}, (3) приоритет $p_{ij} =$

p_i, and (4) требуемая ширина полосы частот $f_{ij} = f_i$. Дополнительно, рассматривается параметр "связности": β_{ij} равен 1 если $l_{ij} \leq l$ и 0 в противном случае (L соответствует ограничению по расстоянию. Это параметр определяет для каждого пользователя $\forall i \in \Psi$ подмножество возможных точек доступа $\Theta_i \subseteq \Theta$. Подключение пользователя i к точке доступа j определяется Булевой переменной x_{ij} ($x_{ij} = 1$ в случае подключения/назначения i к j и $x_{ij} = 0$ в противном случае). Общее решение подключения (назначения) ($\Psi \Rightarrow \Theta$) определяется Булевой матрицей $X = ||x_{ij}||,\ i = \overline{1,n},\ j = \overline{1,m}$.

Таблица 13.27. Данные для исходного региона

(a) пользователи

i	x_i	y_i	z_i	f_i	p_i	r_i
1	30	165	5	10	2	5
2	58	174	5	5	1	9
3	95	156	0	6	1	6
4	52	134	5	6	1	8
5	85	134	3	6	1	7
6	27	109	7	8	3	5
7	55	105	2	7	2	10
8	98	89	3	10	1	10
9	25	65	2	7	3	5
10	52	81	1	10	1	8
11	65	25	7	6	2	9
12	93	39	1	10	1	10
13	172	26	2	10	2	7

(b) точки доступа

j	x_j	y_j	z_j	f_j	n_j	r_j
1	50	157	10	30	4	10
2	72	102	10	42	6	10
3	45	52	10	45	10	10

Таблица 13.28. Данные для дополнительного региона

(a) пользователи

i	x_i	y_i	z_i	f_i	p_i	r_i
14	110	169	5	7	2	5
15	145	181	3	5	2	4
16	170	161	5	7	2	4
17	120	140	6	4	2	6
18	150	136	3	6	2	7
19	175	125	1	8	3	5
20	183	91	4	4	3	5
21	135	59	4	13	3	4
22	147	79	5	7	3	16
23	172	26	2	10	2	7
24	165	50	3	7	3	3
25	127	95	5	7	2	5

(b) точки доступа

j	x_j	y_j	z_j	f_j	n_j	r_j
4	150	165	10	30	5	15
5	140	112	10	32	5	8
6	147	47	10	30	5	15

Формулировка задачи имеет вид:

$$\max \sum_{i=1}^{n} \sum_{j \in \Theta_i} r_{ij} x_{ij}, \quad \max \sum_{i=1}^{n} \sum_{j \in \Theta_i} f_{ij} x_{ij}, \quad \max \sum_{i=1}^{n} \sum_{j \in \Theta_i} p_{ij} x_{ij}$$

$$s.t. \quad \sum_{i=1}^{n} f_{ij} x_{ij} \leq f_j \ \forall j \in \Theta, \quad \sum_{i=1}^{n} x_{ij} \leq n_j \ \forall j \in \Theta, \quad \sum_{j \in \Theta_i} x_{ij} \leq 1 \ \forall i \in \Psi,$$

$$x_{ij} \in \{0,1\} \ \forall\ i = \overline{1,n},\ \forall\ j = \overline{1,m}; \quad x_{ij} = 0 \ \ \forall\ i = \overline{1,n},\ j \in \{\Theta \setminus \Theta_i\}.$$

Используется упрощенная двух-этапная эвристика: (i) для каждой пары (i,j) векторная оценка трансформируется (многокритериальное ранжирование, версия метода ELECTRE) в порядковую оценку (ii) решение полученной задачи о назначениях с одним критиерем (жадный алгоритм). Рис. 13.27 содержит полученное решений (подключение/назначение пользователей к точкам доступа).

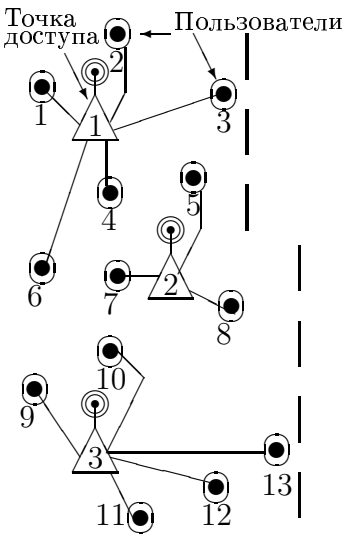


Рис. 13.27. Исходный регион

Рис. 13.28 включает два региона (исходный и дополнительный) и соответствуюшее подключение пользователей. Здесь решение получено отдельно для каждого региона (стратегия I): для исходного региона (подключение с Рис. 13.27) и для дополнительного региона.

Рис. 13.29 содержит результат интегрированного проектирования *стратегия II*). Следующие пользователи пере-подключены: 3, 13, 25.

Можно рассмотреть и другую общую эвристическую схему решения (для уменьшения размерности исходной задачи):

(а) подключение пользователей в исходном регионе,

(б) подключение пользователей в дополнительном регионе,

(в) выделение "пограничной" зоны и соответствующих пользователей (в исходном регионе и в новом регионе) и пере-решение задач для этого "подрегиона".

В примере (Рис. 13.28, Рис. 13.29) такое "пограничное" множество пользователей включает: $\{3, 5, 8, 13, 14, 17, 21, 25\}$.

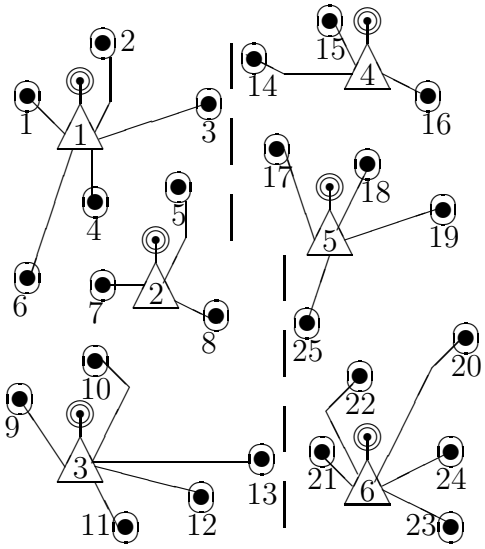


Fig. 13.28. Отдельное подключение

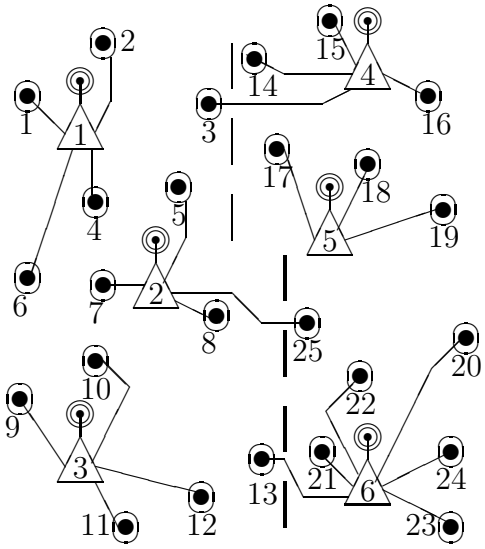


Fig. 13.29. Совместное подключение

13.6.3 Обсуждение

В данном разделе приведены две реалистичные задачи улучшения/расширения коммуникационных сетей. Следует отметить, указанные задачи могут рассматриваться как упро-

щенные. С общих позиций следует исследовать более сложные задачи модификации коммуникационных сетей и использовать известные модели (например, задачи модификации топологии сетей - augmentation problems), и новые подходы (например, модификация много-уровневых сетей, включая перераспределение узлов сети по уровням, модификацию топологии сети на каждом уровне, модификацию соединений узлов различных уровней).

13.7 Эволюция систем связи

В области систем связи комбинаторное моделирование эволюции было применено автором к двум задачам:

(1) комбинаторная эволюция и прогнозирование требования к коммуникационных системам [24, 376, 378];

(2) комбинаторное моделирование поколений беспроводных систем связи [25, 377, 380].

Далее в данном разделе [2] приведено краткое описание применения комбинаторного моделирования к поколениям беспроводных систем связи.

13.7.1 Поколения беспроводных систем связи

Базовая цепочка эволюции поколений беспроводных технологий представлена на Рис. 13.30. В таблице 13.29 приведены краткие описания поколений систем:

$1G \to 2G \to 3G \to 4G \to 5G \to 6G$.

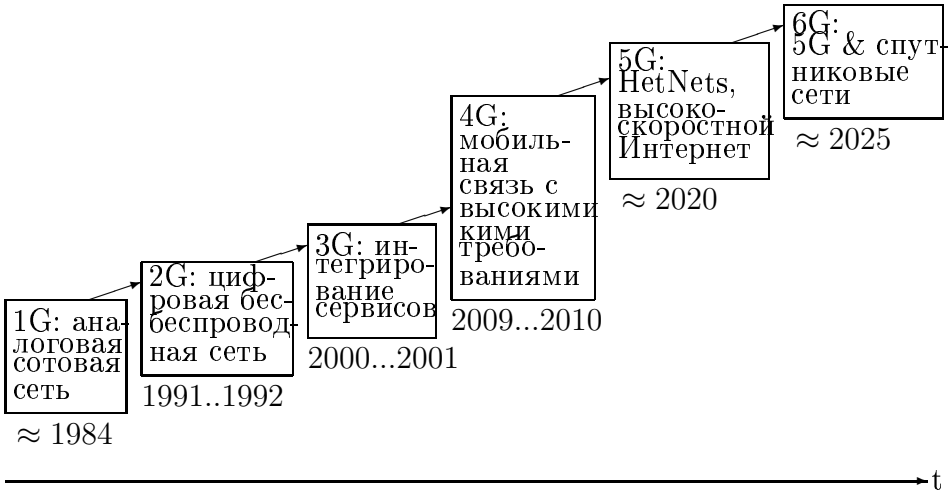


Рис. 13.30. Эволюция поколения беспроводных технологий связи

[2]Материал раздела непосредственно основан на публикациях:
(i) Levin M.Sh. On combinatorial models of generations of wireless communication systems. J. of Commun. Technol. Electr., vol. 63, no. 6, pp. 667–679, 2018.
(ii) Левин М.Ш. О комбинаторных моделях поколений беспроводных систем связи. “Информационные процессы”, 2017, вып. 17, ном. 3, С. 172-187.

Таблица 13.29. Поколения беспроводных мобильных систем связи

Ном.	Поколение	Описание
1.	1G технология (сотовые телефоны, аналоговая беспроводная связь)	AMPS (Analog Mobile Phone System) FDMA (Frequency Division Multiple Access)
2.	2G технология (цифровая беспроводная связь, GPRS, EDGE)	
2.1.	Базовая 2G технология	GSM стандарт: (a) TDMA (Time Division Multiple Access), (b) CDMA (Code Division Multiple Access)
2.2.	2.5G GPRS (General Packet Radio Service)	Область переключаемых пакетов: (a) WAP, (b) MMS, (c) SMS
2.3.	2.75G EDGE (Enhanced Data rates for GSM Evolution)	Расширенная версия GSM
3.	3G технология (Broad Band, IP Tech):	Улучшения (по сравнению 2.5G): (1) улучшение передачи аудио/видео данных, (2) увеличение в несколько раз скорости передачи данных (3) поддержка видео-конференций, (4) Web WAP браузинг с высокой скоростью, (5) IPTV (TV через Интернет)
3.1.	Базовая 3G технология	Версии: (1) W-CDMA, (2) GSM-EDGE, (3) UMTS, (4) DECT, (5) WiMAX, (6) CDMA 2000.
3.2.	3.5G-HSDPA (High-Speed Downlink Packet Access), W-CDMA	Цель: увеличение скорости передачи, реализация: AMC, MIMO, HARQ
3.3.	3.75G-HSUPA (High-Speed Uplink Packet Access), UMTS/ WCDMA uplink evolution	улучшение: P2P приложения
4.	4G технология:	
4.1.	Основная 4G (LTE, WiMAX, WiFi) (улучшенный WiMAX, 802.16, 3GPP, 3GPP2) (Интернет), сервис динамического доступа	Диапазон частот& сервисы; технологии: BDMA (Band Division Multiple Access), CDMA; сервисы: MMS, развлекательные программы, цифровое TV с высоким разрешением
4.2.	4.5G (LTE Advanced Pro) (4^+G)	Частотный спектр: 1800 MHz ... 2100 MHz, мобильный Интернет, скорость 100 Mbit/sec
5.	5G технологии::	
5.1.	Основная 5G (IPv4): WLAN, 802.16 (WMAN), WPAN (Интернет), LTE-E (Long-Term Evolution-Enhanced), гетерогенные сети (HetNet), асинхронный HetNet сценарий	Мобильная революция (скорость доступа), сотовые телефоны с очень широкой полосой, Services: OWA, OTP, Multimedia, развлекательные программы, радиовещание, цифровое TV и др.
5.2.	Мобильные облачные вычисления &5G	
5.3.	Mobile edge computing (MEC)&5G (периферийные вычисления)	3-уровневые гетерогенные MEC системы
6.	6G технология: 5G технология & спутниковые сети	Навигация, телекоммуникация, мониторинг земли и др.

13.7.2 Иерархическая структура системы связи

Автором рассматривается следующая иерархическая структура беспроводной мобильной системы связи (упрощенная версия) (Рис. 13.31):

0. Беспроводная мобильная система связи $S = B^1 \star B^2 \star B^3 \star B^4$:

1. Описание $B^1 = B^{11} \star B^{12}$:

1.1. технология B^{11}: аналоговая сотовая технология B_1^{11}, цифровая сотовая технология B_2^{11}, пакеты данных B_3^{11}, цифровая широкополосная передача пакетов данных B_4^{11}, высокоскоростная обработка на основе IP B_5^{11}, платформа на основе IP сети $B_6^{11} = B_4^{11} \& B_5^{11}$.

1.2. коммутация B^{12}: коммутация каналов B_1^{12}, коммутация пакетов B_2^{12}, коммутация каналов & коммутация пакетов $B_3^{12} = B_1^{12} \& B_2^{12}$.

2. Сервисы $B^2 = B^{21} \star B^{22}$:

2.1. сервис B^{21}: мобильный телефон (голос) B_1^{21}, оцифрованная речь B_2^{21}, SMS B_3^{21}, пакетированные данные B_4^{21}, оцифрованная речь & SMS & пакетированные данные $B_5^{21} = B_2^{21} \& B_3^{21} \& B_4^{21}$, интегрированные высоко-качественные аудио, видео данные B_6^{21}, динамический информационный доступ B_7^{21}, возможности ИИ B_8^{21}, динамический информационный доступ & возможности ИИ $B_9^{21} = B_7^{21} \& B_8^{21}$.

2.2. облачные вычисления B^{22}: нет B_1^{22}, облачные вычисления B_2^{22},

3. Передача данных & доступ $B^3 = B^{31} \star B^{32}$:

3.1. диапазон частот / скорость передачи / пропускная способность B^{31}: 2 kbps B_1^{31}, 64 kbps B_2^{31}, 400 kbps to 30 Mbps B_3^{31}, 3-5 Mbps, 100 Mbps (Wifi) B_4^{31}, 200 mbps to 1 Gbps B_5^{31}, approx 20 Gbps B_6^{31}.

3.2. многоканальная передача/технология доступа B^{32}: FDMA B_1^{32}, TDMA B_2^{32}, FDMA & TDMA $B_3^{32} = B_1^{32} \& B_2^{32}$, CDMA B_4^{32}, TDMA & CDMA $B_5^{32} = B_2^{32} \& B_4^{32}$, OFDMA B_6^{32}, LAS-CDMA B_7^{32}, OFDMA & LAS-CDMA $B_8^{32} = B_6^{32} \& B_7^{32}$.

4. Сети $B^4 = B^{41} \star B^{42} \star B^{43} \star B^{44}$:

4.1. опорная сеть B^{41}: PSTN B_1^{41}, GSM B_2^{41}, PSTN & GSM $B_3^{41} = B_1^{41} \& B_2^{41}$. пакет N/W B_4^{41}, Интернет B_5^{41}, пакет N/W & Интернет $B_6^{41} = B_4^{41} \& B_5^{41}$, спутниковая сеть B_7^{41},

пакет N/W & Интернет & спутниковая сеть $B_8^{41} = B_4^{41} \& B_5^{41} \& B_7^{41}$.

4.2. передача управления B^{42}: горизонтальная B_1^{42}, вертикальная B_2^{42}, горизонтальная & вертикальная $B_3^{42} = B_1^{42} \& B_2^{42}$.

4.3. гетерогенные (разнородные) сети (HetNets) B^{43}: нет B_1^{43}, агрегация различных сетей (HetNets) B_2^{43}.

4.4. космическая связь $B^{44} = B^{441} \& B^{442}$:

4.4.1. спутниковые сети B^{441}: нет B_1^{441}, телекоммуникационная сеть B_2^{441}, мониторинг земли B_3^{441}, навигация B_4^{441}, telecommunication & earth imaging & navigation телекоммуникационная сеть & мониторинг земли &навигация $B_5^{441} = B_2^{441} \& B_3^{441} \& B_4^{441}$.

4.4.2. спутниковые функции B^{442}: нет B_1^{442}, спутниковый роуминг B_2^{442}.

В результате, можно рассмотреть следующие модульные описания системы:

(1) $S^{1G} = (B_1^{11} \star B_1^{12}) \star (B_1^{21} \star B_1^{22}) \star (B_1^{31} \star B_3^{32}) \star (B_1^{41} \star B_1^{42} \star B_1^{43} \star (B_1^{441} \star B_1^{442}))$,

(2) $S^{2G} = (B_2^{11} \star B_3^{12}) \star (B_5^{21} \star B_1^{22}) \star (B_2^{31} \star B_4^{32}) \star (B_3^{41} \star B_1^{42} \star B_1^{43} \star (B_1^{441} \star B_1^{442}))$,

(3) $S^{3G} = (B_4^{11} \star B_2^{12}) \star (B_6^{21} \star B_1^{22}) \star (B_3^{31} \star B_5^{32}) \star (B_4^{41} \star B_1^{42} \star B_1^{43} \star (B_1^{441} \star B_1^{442}))$,

(4) $S^{4G} = (B_5^{11} \star B_3^{12}) \star (B_7^{21} \star B_1^{22}) \star (B_4^{31} \star B_5^{32}) \star (B_5^{41} \star B_3^{42} \star B_1^{43} \star (B_1^{441} \star B_1^{442}))$,

(5) $S^{5G} = (B_6^{11} \star B_3^{12}) \star (B_8^{21} \star B_2^{22}) \star (B_5^{31} \star B_5^{32}) \star (B_5^{41} \star B_3^{42} \star B_2^{43} \star (B_1^{441} \star B_1^{442}))$,

(6) $S^{6G} = (B_6^{11} \star B_3^{12}) \star (B_9^{21} \star B_2^{22}) \star (B_6^{31} \star B_8^{32}) \star (B_8^{41} \star B_3^{42} \star B_2^{43} \star (B_5^{441} \star B_1^{442}))$.

Рис. 13.31 Иерархическая модель беспроводной мобильной системы

13.7.3 Пример улучшения (прогнозирования)

Рассматривается иллюстративный пример двух-стадийного улучшения для варианта технологии 5G: $S^{5G} \Longrightarrow (S^{5G^{adv^1}} \rightarrow S^{5G^{adv^2}})$, где $S^{5G^{adv^1}}$ - система после стадии 1, $S^{5G^{adv^2}}$ - система после стадии 2. На Рис. 13.32 приведена схема двух-стадийного улучшения системы.

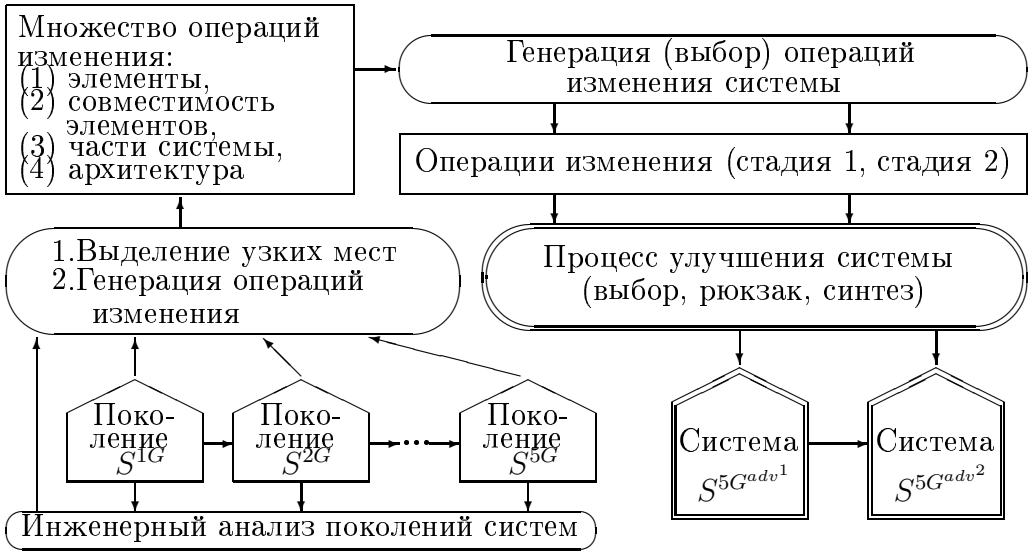


Рис. 13.32. Схема двух-стадийного улучшения системы

В качестве исходной версии системы используется следующая: $S^{5G} = (B_6^{11} \star B_3^{12}) \star (B_8^{21} \star B_2^{22}) \star (B_5^{31} \star B_5^{32}) \star (B_5^{41} \star B_3^{42} \star B_2^{43} \star (B_1^{441} \star B_1^{442}))$. При улучшении S^{5G} рассматриваются следующие компоненты (альтернативные варианты) системы (для изменения/улучшения): (1) сервис B_8^{21}; (2) диапазон частот / скорость передачи / пропускная способность B_5^{31}; (3) многоканальная передача/технология доступа B_5^{32}; (4) опорная сеть B_5^{41}; (5) спутниковая сеть B_1^{441}; (6) спутниковые функции B_1^{442}. Далее приведена две последовательные задачи улучшения (прогнозирования) (упрощенный случай, используется модель блочного рюкзака).

Задача 1. Структура составного улучшения $S^{5G} \Rightarrow S^{5G^{adv^1}}$ для стадии 1 приведена на Рис. 13.33. Таблица 13.30 содержатся описания соответствующих операций улучшения. Соответствующая задача блочного рюкзака для $S^{5G} \Rightarrow S^{5G^{adv^1}}$ имеет вид ($q_1 = 2$, $q_2 = 2$, $q_3 = 4$, $q_4 = 4$, $q_5 = 5$, $b_1^{constr} = 19.0$):

$$\max \sum_{\iota=1}^{6} \sum_{j=1}^{q_\iota} x_{\iota,j} c_{\iota,j} \quad s.t. \quad \sum_{\iota=1}^{6} \sum_{j=1}^{q_\iota} x_{\iota,j} b_{\iota,j} \leq b_1^{constr}, \ \sum_{j=1}^{q_\iota} x_{\iota,j} \leq 1 \ \forall \iota = \overline{1,6}, \ \forall j = \overline{1,q_\iota} \quad \forall x_{\iota,j} \in \{0,1\}.$$

Используется упрощенная эвристика (последовательная упаковка элементов по величине $\frac{c_{\iota,j}}{b_{\iota,j}}$.) Получается решение ($I_1^1 = U_2^1 \star U_2^2 \star U_3^3 \star U_3^4 \star U_2^6$):

$S^{5G^{adv^1}} = (B_6^{11} \star B_3^{12}) \star (B_9^{21} \star B_2^{22}) \star (B_6^{31} \star B_7^{32}) \star (B_7^{41} \star B_3^{42} \star B_2^{43} \star (B_2^{441} \star B_1^{442}))$.

Это решение используется как базовое для улучшения (прогнозирования) на стадии 2.

Задача 2. Структура составного улучшения $S^{5G^{adv^1}} \Rightarrow S^{5G^{adv^2}}$ для стадии 2 приведена на Рис. 13.34. Таблица 13.31 содержит описания соответствующих операций улучшения. Соответствующая задача блочного рюкзака для $S^{5G^{adv^1}} \Rightarrow S^{5G^{adv^2}}$ имеет вид ($q_1 = 2$, $q_2 = 2$, $q_3 = 4$, $q_4 = 2$, $b_2^{constr} = 17.5$):

$$\max \sum_{\iota=1}^{4} \sum_{j=1}^{q_\iota} y_{\iota,j} c_{\iota,j} \quad s.t. \quad \sum_{\iota=1}^{4} \sum_{j=1}^{q_\iota} y_{\iota,j} b_{\iota,j} < b_2^{constr}, \ \sum_{j=1}^{q_\iota} y_{\iota,j} \leq 1 \ \forall \iota = \overline{1,4}, \ \forall j = \overline{1,q_\iota} \quad \forall y_{\iota,j} \in \{0,1\}.$$

Используется упрощенная эвристика (последовательная упаковка элементов по величине $\frac{c_{\iota,j}}{b_{\iota,j}}$). Получается решение ($I_1^2 = V_2^1 \star V_2^2 \star V_2^3 \star V_1^4$):

$S^{5G^{adv^2}} = (B_6^{11} \star B_3^{12}) \star (B_9^{21} \star B_2^{22}) \star (B_6^{31} \star B_8^{32}) \star (B_8^{41} \star B_3^{42} \star B_2^{43} \star (B_3^{441} \star B_1^{442}))$.

Таким образом, получается следующая стратегия двух-стадийного улучшения (прогнозирования):

$\Upsilon : \ < S^{5G^{adv}} \Longrightarrow S^{5G^{adv^1}} \Rightarrow S^{5G^{adv^2}} >$.

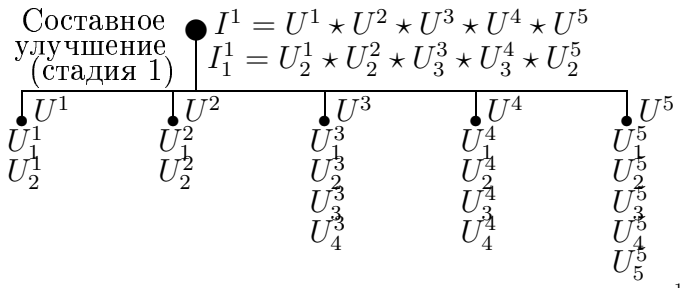


Рис. 13.33. Составное улучшение $S^{5G} \Rightarrow S^{5G^{adv^1}}$

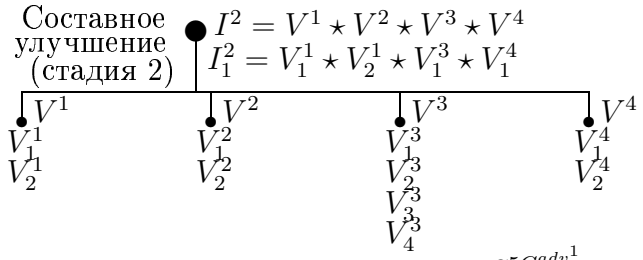


Рис. 13.34. Составное улучшение $S^{5G^{adv^1}} \Rightarrow S^{5G^{adv^2}}$

Таблица 13.30. Операции изменения для $S^{5G} \Rightarrow S^{5G^{adv^1}}$ (стадия 1)

Ном. ι	Компонент системы	Альтернатива	Операция изменения	Булева переменная	Полезность $c_{\iota,j}$	Стоимость $b_{\iota,j}$
1.	B^{21} (сервис)	B_8^{21}	U_1^1 : Нет	$x_{1,1}$	0.0	0.0
		B_8^{21}	$U_2^1 : B_8^{21} \rightarrow B_9^{21}$	$x_{1,2}$	2.0	3.0
2.	B^{31} (диапазон частот/ скорость передачи/ пропускная способность)	B_5^{31}	U_1^2 : Нет	$x_{2,1}$	0.0	0.0
		B_5^{31}	$U_2^2 : B_5^{31} \rightarrow B_6^{31}$	$x_{2,2}$	4.0	5.0
3.	B^{32} (многоканальная передача/технология доступа)	B_5^{32}	U_1^3 : Нет	$x_{3,1}$	0.0	0.0
		B_5^{32}	$U_2^3 : B_5^{32} \rightarrow B_6^{32}$	$x_{3,2}$	1.0	2.0
		B_5^{32}	$U_3^3 : B_5^{32} \rightarrow B_7^{32}$	$x_{3,3}$	3.6	4.0
		B_5^{32}	$U_4^3 : B_5^{32} \rightarrow B_8^{32}$	$x_{3,4}$	3.6	6.0
4.	B^{41} (опорная сеть)	B_5^{41}	U_1^4 : Нет	$x_{4,1}$	0.0	0.0
		B_5^{41}	$U_2^4 : B_5^{41} \rightarrow B_6^{41}$	$x_{4,2}$	3.6	6.0
		B_5^{41}	$U_3^4 : B_5^{41} \rightarrow B_7^{41}$	$x_{4,3}$	7.0	7.0
		B_5^{41}	$U_4^4 : B_5^{41} \rightarrow B_8^{41}$	$x_{4,4}$	9.0	12.0
5.	B^{441} (спутниковая сеть)	B_1^{441}	U_1^5 : Нет	$x_{5,1}$	0.0	0.0
		B_1^{441}	$U_2^5 : B_1^{441} \rightarrow B_2^{441}$	$x_{5,2}$	5.0	5.0
		B_1^{441}	$U_3^5 : B_1^{441} \rightarrow B_3^{441}$	$x_{5,3}$	5.6	7.0
		B_1^{441}	$U_4^5 : B_1^{441} \rightarrow B_4^{441}$	$x_{5,4}$	6.0	8.0
		B_1^{441}	$U_5^5 : B_1^{441} \rightarrow B_5^{441}$	$x_{5,5}$	14.0	20.0

Таблица 13.37. Операции изменения для $S^{5G^{adv^2}} \Rightarrow S^{5G^{adv^2}}$ (стадия 2)

Ном. ι	Компонент системы	Альтернатива	Операция изменения	Булева переменная	Полезность $c_{\iota,j}$	Стоимость $b_{\iota,j}$
1.	B^{32} (многоканальная передача/технологии доступа)	B_7^{32}	V_1^1 : Нет	$y_{1,1}$	0.0	0.0
		B_7^{32}	$V_2^1 : B_7^{32} \rightarrow B_8^{32}$	$y_{1,2}$	4.5	4.0
2.	B^{41} (опорная сеть)	B_7^{41}	V_1^2 : Нет	$y_{2,1}$	0.0	0.0
		B_7^{41}	$V_2^2 : B_7^{41} \rightarrow B_8^{41}$	$y_{2,2}$	6.5	7.0
3.	B^{441} (спутниковая сеть)	B_2^{441}	V_1^3 : Нет	$y_{3,1}$	0.0	0.0
		B_2^{441}	$V_2^3 : B_2^{441} \rightarrow B_3^{441}$	$y_{3,2}$	6.0	6.5
		B_2^{441}	$V_3^3 : B_2^{441} \rightarrow B_4^{441}$	$y_{3,3}$	6.5	7.5
		B_2^{441}	$V_4^3 : B_2^{441} \rightarrow B_5^{441}$	$y_{3,4}$	11.0	18.0
4.	B^{442} (спутниковые функции)	B_1^{442}	V_1^4 : Нет	$y_{4,1}$	0.0	0.0
		B_1^{442}	$V_2^4 : B_1^{442} \rightarrow B_2^{442}$	$y_{4,2}$	12.0	30.0

13.7.4 Обсуждение

В данном разделе приведено краткое описание комбинаторной эволюции и прогнозирования беспроводной системы связи. Подобные задачи в области современных поколений систем связи представляются крайне важными. Следует также отметить, что применение данного подхода к другим типам модульных систем является перспективным.

13.8 Выводы к главе

В данной главе представлены примеры применения комбинаторных инженерных схем для построения, модификации и комбинаторного моделирования эволюции различных частей коммуникационных систем (общая структура, протоколы, стандарты передачи информации, подключения пользователей к провайдерам и точкам доступа, поколения беспровод-

ных систем связи). В дальнейшем представлятся важным провести следующие исследования:

1. моделирование и построение структуры современных поколения систем связи, включая анализ эволюции систем;

2. применение модульного подхода к различным современным протоколам беспроводной связи;

3. исследование задач модульного проектирования коммуникационных протоколов и стандартов обработки мультимедийной информации в режиме реального времени;

4. моделирование и построение архитектуры современных сетей связи, включая вопросы модификации сетей;

5. исследование задач подключения пользователей к точкам доступа с учетом неопределенности (например, с использованием интервальных оценок в виде мультимножеств);

6. исследование задач проектирования и модификации много-уровневых коммуникационных систем.

Глава 14

Приложения в строительстве

В данной главе [1] приведено описание двух примеров композиции: (а) макротехнология бетона (композиция, производство, транспортирование), (б) иерархическая модель сейсмостойкого здания (типовой двухэтажный дом), оценивание дома и его укрепление.

14.1 Технология бетона

В данном разделе представлен пример иерархического построения технологии бетона [382, 344]. Иерархическая схема для технологии бетона включает:

(а) комбинаторное моделирование и анализ, включая две части: (i) бетон (альтернативные компоненты и их совместимость) и (ii) технология производства (альтернативы);

(б) адаптация технологии для определенной прикладной ситуации.

Наше рассмотрение близко к современным подходам в области интегрированного проектирования материалов и производственного процесса. Здесь рассматривается конкретная прикладная ситуация:

Найти 'наилучшую' комбинацию компонентов бетона и вариант производства с учетом целевого класса бетона и требований (ограничений).

Рассматривается следующая иерархическая структура (DAs указаны в скобках):

0. Технология бетона $S = D \times X \times U \times Z$
- **1.** Компоненты $D = A \times B$
  - *1.1* Крупные заполнители A $(A_0, A_1, A_2, A_3, A_4, A_5)$.
  - *1.2* Мелкие заполнители B $(B_0, B_1, B_2, B_3, B_4, B_5)$.
- **2.** Цемент X $(X_1, X_2, X_3, X_4, X_5)$.
- **3.** Производство U (U_1, U_2, U_3, U_4).
- **4.** Классы бетона (ветвь целей) Z $(Z_1, Z_2, Z_3, Z_4, Z_5)$.

Фактически, приведенная структура содержит стадию НИР (D, X и др.). В примере рассматриваются следующие DAs: A_0 (нет), A_1 (щебень), A_2 (гравий), A_3 (щебень из гравия), A_4 (доменный шлак), A_5 (топливная зола от тепловой электростанции) и т.д.; B_1

[1] Материал главы основан на публикациях:
(i) Levin M.Sh., Nisnevich M.L. Combinatorial scheme for management of life cycle: Example for concrete macrotechnology. J. of Intell. Manuf., 2001, Vol. 12, No. 4, pp. 393-401.
(ii) Levin M.Sh., Danieli M.A. Hierarchical decision making framework for evaluation and improvement of composite systems (example for building). Informatica (Lith.), 2005, Vol. 16, No. 2, pp. 213-240.
(iii) Levin M.Sh. Composite Systems Decisions, Springer, 2006, chapters 8, 9.

(песок), B_2 (дробленный песок), B_3 (просеянный щебень), B_4 (аналог A_4), B_5 (аналог A_5) и т.д.; X_1 (цемент '300'), X_2 (цемент '400'), X_3 (цемент '500'), X_4 (цемент '600'), X_5 (цемент '700'); U_1 (изготовление на заводе), U_2 (изготовление на строительной площадке, использование имеющейся смеси заполнителей), U_3 (изготовление на строительной площадке, смешивание заполнителей), U_4 (изготовление в пути на автомашине во время доставки); Z_1 (бетон '200' или B15), Z_2 (бетон '300' или B22.5), Z_3 (бетон '400' или B30), Z_4 (бетон '500' или B40), Z_5 (бетон '600' или B45).

Для U_1 производство бетона и железобетонных изделий (панелей, конструкций) осуществляется на заводе. Критерии указаны в Таблице 14.1 (веса критериев соответствуют целевому бетону Z_2).

Таблица 14.1. Критерии (веса для бетона '300')

Критерии		Компоненты			
		A	B	X	U
C_1	Стоимость	-1	-1	-2	-3
C_2	Позитивное влияние на экологию	2	2		
C_3	Транспортные затраты	-1	-1		
C_4	Качество	3	2	3	
C_5	Время изготовления бетона				2
C_6	Время изготовления изделий				2

Рассматривается композиция управляющей стратегии для получения бетона '300' (Z_2). Отметим, что композиция A_5 или B_5 ориентирована на получение 'легкого' бетона. Оценки DAs и результирующие приоритеты представлены в Таблице 14.2 и Таблице 14.3 (для компонентов A, B, X, U). Вычисление приоритетов может основываться на различных методах, в частности, была использована система СППР COMBI [28, 336, 364]. В тоже время, возможно использование простых методов, например, аддитивной свертки:

$$v(\alpha_i) = \sum_{j=1}^{n} \beta_j b_j(\alpha_i),$$

где i - номер альтернатива α, j - номер критерия, β_j - вес (важность) критерия j, $b_j(\alpha_i)$ - оценка альтернативы α_i по критерию j, $v(\alpha_i)$ - ранжирование альтернативы α_i.

Таким образом, вычисляются рейтинги (v) для DAs (для A):

$v(A_0) = (-1)0 + (2)0 + (-1)0 + (3)0 = 0,$
$v(A_1) = (-1)6 + (2)0 + (-1)1 + (3)5 = 8,$
$v(A_2) = (-1)4 + (2)1 + (-1)1 + (3)4 = 9,$
$v(A_3) = (-1)5 + (2)1 + (-1)1 + (3)5 = 11,$
$v(A_4) = (-1)3 + (2)3 + (-1)1 + (3)3 = 11,$
$v(A_5) = (-1)1 + (2)3 + (-1)2 + (3)4 = 15.$

Указанные рейтинги трансформируются в приоритеты. Оценки совместимости для компонентов D представлены в Таблице 14.4. Результирующие решения для $D = A \star B$ следующие:

$D_1 = A_5 \star B_5$, $D_2 = A_1 \star B_1$, $D_3 = A_1 \star B_2$, $D_4 = A_1 \star B_3$, $D_5 = A_2 \star B_1$, $D_6 = A_2 \star B_2$, $D_7 = A_2 \star B_3$, $D_8 = A_3 \star B_1$, $D_9 = A_3 \star B_2$, $D_{10} = A_3 \star B_3$, $D_{11} = A_4 \star B_4$.

При этом получаются следующие векторы качества:

$N(D_1) = (5; 1, 1, 0, 0)$, $N(D_2) = (5; 0, 1, 1, 0)$, $N(D_3) = (5; 0, 1, 0, 1)$,
$N(D_4) = (5; 0, 1, 0, 1)$, $N(D_5) = (5; 0, 1, 1, 0)$, $N(D_6) = (5; 0, 1, 0, 1)$,
$N(D_7) = (5; 0, 1, 0, 1)$, $N(D_8) = (5; 0, 1, 1, 0)$, $N(D_9) = (5; 0, 0, 1, 1)$,
$N(D_{10}) = (5; 0, 1, 0, 1)$, $N(D_{11}) = (5; 1, 1, 0, 0)$.

Таблица 14.2. Оценки компонентов

DAs	Критерии				
	C_1	C_2	C_3	C_4	При-теты
A_0	0	0	0	0	3
A_1	6	0	1	5	2
A_2	4	1	1	4	2
A_3	5	1	1	5	2
A_4	3	3	1	3	2
A_5	1	3	2	4	1

DAs	Критерии				
	C_1	C_2	C_3	C_4	При-теты
B_1	4	0	1	4	3
B_2	6	0	2	5	4
B_3	3	1	2	3	4
B_4	2	1	1	3	1
B_5	2	1	2	4	2

Таблица 14.3. Оценки компонентов

DAs	Критерии		
	C_1	C_2	При-теты
X_1	1	1	2
X_2	2	2	1
X_3	3	3	1
X_4	4	4	2
X_5	5	5	2

DAs	Критерии			
	C_1	C_2	C_3	При-теты
U_1	6	3	0	2
U_2	3	2	3	1
U_3	4	1	3	3
U_4	5	1	3	2

Очевидно, составные DAs D_1 и D_{11} являются Парето-эффективными решениями. Также будут рассматриваться некоторые другие составные DAs для следующей стадии (на основе экспертных суждений). Предполагается, что приоритеты полученных составных решений равны 1. Оценки совместимости DAs представлены в Таблице 14.4, в Таблице 14.5.

Таблица 14.4. Оценки совместимости

	B_1	B_2	B_3	B_4	B_5
A_0	0	0	0	0	0
A_1	5	5	5	3	0
A_2	5	5	5	3	0
A_3	5	5	5	3	0
A_4	3	3	3	5	0
A_5	0	0	0	0	5

	X_1	X_2	X_3	X_4	X_5
Z_1	5	4	3	2	1
Z_2	4	5	4	3	2
Z_3	2	4	5	4	3
Z_4	0	2	4	5	4
Z_5	0	0	2	4	5

	U_1	U_2	U_3	U_4
X_1	4	5	5	5
X_2	4	5	5	5
X_3	5	4	4	4
X_4	5	4	4	4
X_5	5	4	4	4

	U_1	U_2	U_3	U_4
Z_1	5	4	5	5
Z_2	5	4	5	5
Z_3	5	4	5	5
Z_4	4	4	5	5
Z_5	3	4	5	5

Таблица 14.5. Совместимость

	X_1	X_2	X_3	X_4	X_5
D_1	5	0	0	0	0
D_2	5	5	5	5	5
D_3	5	5	5	5	5
D_4	5	5	4	0	0
D_5	5	5	5	0	0
D_6	5	5	5	0	0
D_7	5	4	3	0	0
D_8	5	5	5	5	5
D_9	5	5	5	5	5
D_{10}	5	4	3	0	0
D_{11}	5	5	0	0	0

	Z_1	Z_2	Z_3	Z_4	Z_5
D_1	5	4	0	0	0
D_2	5	5	5	5	5
D_3	5	5	5	5	5
D_4	5	5	4	3	0
D_5	5	5	4	0	0
D_6	5	5	4	0	0
D_7	5	5	4	3	0
D_8	5	5	4	0	0
D_9	5	5	4	0	0
D_{10}	5	4	3	2	0
D_{11}	5	5	3	0	0

	U_1	U_2	U_3	U_4
D_1	5	5	5	3
D_2	5	5	5	5
D_3	5	5	5	5
D_4	4	5	3	2
D_5	5	5	5	5
D_6	5	5	5	5
D_7	4	3	3	2
D_8	4	5	5	5
D_9	5	5	5	5
D_{10}	4	3	3	2
D_{11}	3	3	3	3

Результирующие составные DAs для класса бетона Z_2 имеют вид (Парето-эффективные решения):

$S_1 = Z_2 \star X_2 \star U_2 \star D_2$; $S_2 = Z_2 \star X_2 \star U_2 \star D_3$; $S_3 = Z_2 \star X_2 \star U_2 \star D_4$;

$S_4 = Z_2 \star X_2 \star U_2 \star D_5$; $S_5 = Z_2 \star X_2 \star U_2 \star D_9$.
Вектор качества для указанных составных решений $N = (4; 3, 0, 0)$.
Можно рассмотреть акции улучшения для указанных решений $(S_1, ..., S_5)$:

1. Улучшение приоритетов для существующих DAs:
$A_1,\ 2 \Rightarrow 1$; $A_2,\ 2 \Rightarrow 1$; $A_3,\ 2 \Rightarrow 1$; $B_1,\ 3 \Rightarrow 1$; $B_2,\ 4 \Rightarrow 1$; $B_3,\ 3 \Rightarrow 1$.
2. Использование новых DAs, например, некоторые комбинации:
$A_6 = A_1 \& A_5$, $B_6 = B_1 \& B_3$, $U_5 = U_2 \& U_4$.
3. Улучшение совместимости для некоторых пар: $(U_2, Z_2),\ 4 \Rightarrow 5$.

Здесь к базовой проектной морфологии добавляются указанные комбинированные DAs с оценками (включая совместимости), которые равны лучшим оценкам составляющих их элементов: $A_6 = A_1 \& A_5$, приоритет равен 1; $B_6 = B_1 \& B_3$, приоритет равен 1; $U_5 = U_2 \& U_4$, приоритет равен 2. Теперь можно исследовать следующие комбинированные DAs для D: $D_{12} = A_6 \& B_6$; $D_{13} = A_6 \& B_3$; $D_{14} = A_1 \& B_6$. В результате можно получить следующие последовательно-параллельные DAs (для уровня Парето-эффективных решений) (Рис. 14.1): $S_6 = Z_2 \star X_2 \star U_5 \star D_{12}$; $S_7 = Z_2 \star X_2 \star U_5 \star D_{13}$; $S_8 = Z_2 \star X_2 \star U_5 \star D_{14}$.

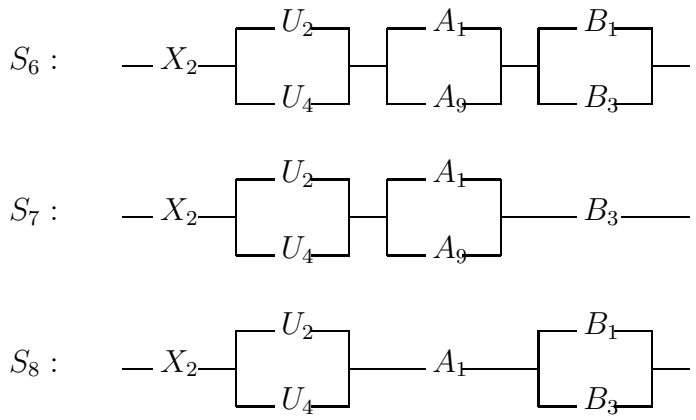


Рис. 14.1. Иллюстрация последовательно-параллельных DAs

14.2 Оценивание и улучшение здания

В данном разделе рассматривается пример оценивания и улучшения здания на основе ИММП [344, 385]. Общая схема включает следующие этапы:

I. Построение иерархической модели здания: *1.1.* Построение древовидное модели. *1.2.* Построение многокритериального (многофакторного) описания для каждого узла модели (т.е., части и компоненты здания), включая порядковые шкалы для каждого критерия.

II. Оценивание: *2.1.* Оценивание чатей здания по критериям. *2.2.* Пошаговой агрегирование информации для получения многокритериальных оценок для более высокого иерархического уровня модели (“снизу-вверх”).

III. Анализ здания и выявление узких мест: *3.1.* Анализ результирующей интегрированной оценки для системы, анализ оценок для частей здания и их взаимосвязей. *3.2.* Выявление узких мест как некоторых “слабых” частей зданий, его компонентов или их взаимосвязей.

IV. Проектирование акций улучшения для здания: *4.1.* Генерация / выбор множества возможных акций улучшения. *4.2.* Выбор / композиция наилучшего подмножества

акций улучшения с учетом проектных и технологических требований. *4.3.* Составление расписания реализации акций улучшения, которые были отобраны (сформированы) выше.

Здесь рассматривается конкретная ситуация для учета возможного землетрясения (шкала сейсмической интенсивности МСК-64). Рассматривается иерархическая модель следующего вида:

1. Здание S ([2, 3, 4, 5]).

1.1 Фундамент A ([3, 4, 5]).

1.2. Основная структура (каркас) B ([2, 3, 4, 5]).

1.2.1 Несущие конструкции D ([3, 4, 5]): *1.2.1.1.* Каркас E ([3, 4, 5]). *1.2.1.2.* Ядро жесткости G ([4, 5]). *1.2.1.3.* Лестничный пролет H ([3, 4, 5]).

1.2.2 Не несущие конструкции F ([2, 3, 4, 5]): *1.2.2.1.* Несущие стены I ([2, 3, 4, 5]). *1.1.2.2.* Разделительные стены J ([2, 3, 4, 5]).

1.3. Перекрытия C ([2, 3, 4, 5]).

Используется единая порядковая шкала для частей и компонентов здания: *1.* глобальные разрушения; *2.* локальные разрушения; *3.* трещины; *4.* небольшие трещины (типа 'волос'); *5.* отсутствие разрушений. Для каждой части здания применяется шкала, которая является частью указанной выше.

Множество акций улучшения (перепроектирования) включает следующее:

А. Внутренные акции.

1. Уменьшение веса: *1.1.* изоляционные материалы (например, теплоизоляция, звукоизоляция); *1.2.* несущие стены (каркас); *1.3.* не несущие стены; *1.4.* перекрытия.

2. Модификация статической схемы: *2.1.* проектирование ядра жесткости, *2.2.* увеличение статической неопределенности конструкции, (*2.2.1* проектирование шарнирных узлов, *2.2.2.* проектирование дополнительных опорных элементов, *2.2.3.* проектирование дополнительных узлов, *2.2.4.* проектирование дополнительных соединений.

3. Усиление структурных элементов и связей (проектирование дополнительных элементов): *3.1.* балок; *3.2.* колонн; *3.3.* стен; *3.4.* плит перекрытий; *3.5.* разделительных стен; *3.6.* соединений; *3.7.* перекрытий (купольных и др.); *3.8.* фундамента.

4. Дополнительные структурные системы и элементы: *4.1.* гибкая антисейсмическая обвязка; *4.2.* жесткая антисейсмическая обвязка (метал, бетон); *4.3.* металлическое ядро жесткости; *4.4.* бетонное ядро жесткости; *4.5.* стены жесткости.

В. Внешние решения.

Дополнительно, вводим три типа бинарных отношений на множестве акций улучшения: (1) эквивалентность акций R^e; (2) дополнительность R^c; (3) предшествование R^p.

Подход к оцениванию и улучшению здания иллюстрируется на примере двухэтажного здания (Рис. 14.2).

ИММП рассматривается для оценки здания (или проекта здания). Пусть для компонентов здания сгенерированы DAs (приоритеты с точки зрения сейсмостойкого строительства указаны в скобках):

Фундамент: A_1, ленточный фундамент (2),

A_2, фундаментная плита (1), A_3, фундамент, состоящий из изолированных частей (2).

Каркас: E_1, монолитный каркас (1), E_2, сборный каркас (2).

Ядро жесткости: G_1, монолитное ядро жесткости (1), G_2, сборное ядро жесткости (2).

Лестничный пролет: H_1, монолитный лестничный пролет (1), H_2, сборный лестничный пролет (2), H_3, составной лестничный пролет из монолитных и сборных элементов (3).

Стены с заполнителем: I_1, мелкие элементы (2), I_2, навесные стеновые панели (2), I_3, сборные панельные стены (1), I_4, каркасные перегородки (1).

Разделительные стены: J_1, сборные панельные стены (1), J_2, мелкие элементы (3), J_3, каркасные перегородки (2).

Перекрытия: C_1, монолитные плиты (1), C_2, составные плиты (3), C_3, сборные плиты (3).

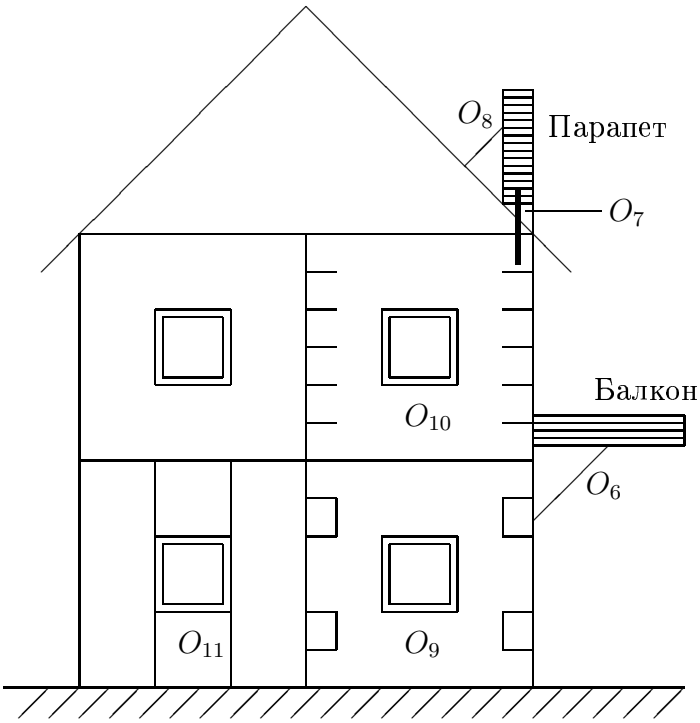


Рис. 14. 2. Схема здания и операции улучшения

Окончательно, получаются следующие составные DAs:

(a) $N = (3; 2, 1, 0)$: $S_1 = A_2 \times B_1 \times C_1$, $S_2 = A_2 \times B_3 \times C_1$, $S_3 = A_2 \times B_4 \times C_1$ (*результирующее качество* равно 2);

(b) $N = (2; 2, 1, 0)$: $S_4 = A_2 \times B_{13} \times C_1$ (*результирующее качество* равно 3).

Получается, что *результирующее качество* 1 является невозможным.

Далее исследуется следующий пример:

(a) $S^a = A_2 \times (E_1 \times G_1 \times H_1) \times (I_3 \times J_1) \times C_1$, *результирующее качество* равно 2.

(b) $S^b = A_1 \times (E_2 \times G_2 \times H_2) \times (I_3 \times J_1) \times C_3$, *результирующее качество* равно 3.

(c) $S^c = A_1 \times (E_2 \times G_1 \times H_1) \times (I_3 \times J_3) \times C_3$, *результирующее качество* равно 4.

Предполагается следующий список акций улучшения (операций):

Операционная группа I (каркас):

1. Увеличение габаритов и активное усиление O_1.
2. Увеличение активного усиления O_2.

Операционная группа II (узлы):

3. Увеличение уровня крепления в зоне узлов O_3.
4. Уменьшение шага для пересекающихся стержней в зоне узлов O_4.

Операционная группа III (консоль и консольный балкон):

5. Уменьшение проекции консоли O_5.
6. Дополнительная опора для консоли O_6.

Операционная группа IV (фронтон и парапетная стенка):

7. Фиксация нижней части O_7.
8. Проектирование трехмерной конструкции O_8.

Операционная группа V (связь между каркасом и стеновым заполнением)

9. Проектирование конструкций сдвига O_9.

10. Проектирование армированной решетки O_{10}.

11. Разделение стеновых заполнителей вспомогательным каркасом O_{11}.

Применение нескольких операций иллюстрируется на Рис. 14.2. Введенные выше бинарные отношения для указанных операций имеют вид:

(1) эквивалентность $R^e = \{(1,2),(1,3),(1,4),(2,3),(2,4),(5,6),(7,8),(9,10),$ $(9,11),(10,11)\}$, не-эквивалентность $\widetilde{R^e} = \{(3,4)\}$;

(2) дополнительность $R^c = \{(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),$ $(1,9),(1,10),(1,11),(2,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(1,10),$ $(1,11),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),(3,10),(3,11),(4,5),(4,6),$ $(4,7),(4,8),(4,9),(4,10),(4,11),(5,7),(5,8),(5,9),(5,10),(5,11),(6,7),$ $(6,8),(6,9),(6,10),(6,11),(7,9),(7,10),(7,11),(8,9),(8,10),(8,11)\}$, не-дополнительность $\widetilde{R^c} = \{(5,6),(7,8),(9,10),(9,11),(10,11)\}$;

(3) предшествование $R^p = \{(1,2)(1,3)(1,4)(1,5),(1,6),(1,7),(1,8),(1,9),$ $(1,10),(1,11),(2,3)(2,4)(2,5),(2,6),(2,7),(2,8),(2,9),(2,10),(2,11),$ $(3,5),(3,6),(3,7),(3,8),(3,9),(3,10),(3,11),(4,5),(4,6),(4,7),$ $(4,8),(4,9),(4,10),(4,11),(5,9),(5,10),(5,11),(6,9),(6,10),$ $(6,11),(7,9),(7,10),(7,11),(8,9),(8,10),(8,11)\}$.

Рассматриваются следующие критерии (соответствующие порядковые шкалы и веса критериев указаны в скобках):

Улучшение устойчивости к землетрясениям:

1. Уменьшение собственной массы $([0,...,4],3)$: K_1;

2. Увеличение допускаемой нагрузки $([1,...,5],5)$: K_2;

3. Повышение надежности $([1,...,5],5)$: K_3.

Качество архитектурных и плановых решений:

4. Фасад $([-1,0],3)$: K_4.

5. План $([-1,0],3)$: K_5.

6. Свободное пространство $([-1,0],3)$: K_6.

Эксплуатационные характеристики:

7. Теплотехника $([-1,0],1)$: K_7.

8. Акустика $([-1,0],1)$: K_8.

9. Риск пожара $([-1,0],4)$: K_9.

Затраты:

10. Материалы $([-1,0],3)$: K_{10}.

11. Стоимость $[1...4],4$: K_{11}.

12. Временные затраты $[0...3],3$: K_{12}.

Экспертные оценки по указанным критериям для рассматриваемых акций улучшения (операций) следующие:

O_1: -2, 5, 5, -1, -3, -2, 0, 0, 0, 5, 5, 5;

O_2: 0, 4, 4, 0, 0, 0, 0, 0, -1, 3, 3, 3;

O_3: 0, 2, 3, 0, 0, 0, 0, 0, 0, 1, 1, 1;

O_4: 0, 3, 4, 0, 0, 0, 0, 0, 0, 2, 3, 3;

O_5: 2, 2, 4, -3, -4, -1, -1, -1, 1, -2, -2, -3;

O_6: -1, 3, 5, -5, -1, -1, 0, 0, 2, 5, 4, 5;

O_7: 2, 3, 4, 0, 0, 0, 0, 0, 1, 1, 2, 2;

O_8: 1, 4, 5, -2, -1, -1, 0, 0, 1, 2, 3, 4;

O_9: 0, 2, 3, 0, 0, 0, 0, 0, 1, 1, 1, 1;

O_{10}: 0, 3, 4, 0, 0, 0, 0, 0, -1, 4, 3, 4;

O_{11}: 0, 2, 4, 0, 0, 0, -2, -1, 0, 5, 5, 5.

Очевидно, что можно провести анализ операций ([385], [344]): (i) кластеризация; (ii) отбор на основе задачи о рюкзаке; (iii) составление расписания выполнения операций и др.

Теперь можно рассмотреть процесс улучшения здания. Структура процесса (модель) основывается на бинарном отношении R^c: (a) операции для frame (например, O_1, O_2); (b) операции для узлов (например, O_3, O_4); (c) операции для парапета (например, O_5, O_6); (d) операции для балкона (например, O_7, O_8); (e) связь между каркасом и стеновым заполнением (e.g., O_9, O_{10}, O_{11}). При этом предшествование для указанных групп операция следующее: (a); (b); (c) и (d) параллельно; (e).

Бинарное отношение R^c является базовым, чтобы сгенерировать следующие агрегированные операции $O_1\&O_2$ и $O_3\&O_4$.

Бинарное отношение $\widetilde{R^c}$ представляет собой причину для исключения следующих агрегированных операций $O_5\&O_6$, $O_7\&O_8$, $O_9\&O_{10}$, $O_9\&O_{11}$, $O_{10}\&O_{11}$, $O_9\&O_{10}\&O_{11}$.

Структура процесса улучшения и сгенерированные операции представлены на Рис. 14.3.

Следует отметить, что все указанные операции улучшения являются совместимыми (по отношению R_c).

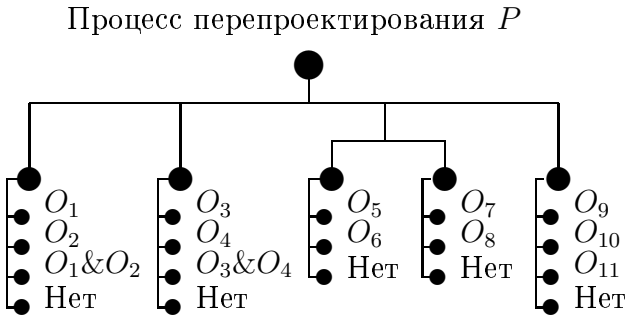


Рис. 14.3. Структура процесса улучшения

Таким образом, получается следующие иллюстративные примеры процесса улучшения (перепроектирования) P:

(а) $P_1 = (O_1\&O_2) \star (O_3\&O_4) \star (O_5\&O_7) \star O_{11}$;

(б) $P_2 = O_2 \star O_3 \star (O_6\&O_7) \star O_{10}$.

14.3 Выводы к главе

В данной главе представлены схемы модульного проектирования для двух задач в области строительных технологий: модульное проектирование макро-технологии бетона и иерархическое моделирование и улучшение (укрепление) типового двух-этажного дома (с точки зрения сейсмо-устойчивости). Приведенные задачи могут быть использованы как базовые аналоги для исследования близких задач и систем, например: модульное проектирование (композиция) новых материалов включая приложения в комбинаторной химии, исследование различных типов зданий. Можно указать следующие направления для исследований в будущем:

1. построение траекторий систем (т.е., много-стадийное проектирование);

2. анализ рассмотренных задач проектирования в режиме реального времени (в частности, для зданий);

3. учет неопределенности параметров.

Глава 15

Приложения в биомедицине

В данной главе [1] приведено описание двух примеров композиции: (а) плана лечения астмы у детей, (б) плана проведения иммунологического анализа.

15.1 План лечения

Рассматривается применение иерархического комбинаторного морфологического подхода к формированию плана лечения астмы у детей.

15.1.1 Структура плана, альтернативы, оценки

Используется следующая структура плана лечения:

0. План $S = X \star Y \star Z$.
  1. Базовое лечение $X = J \star M$.
    1.1. Физиотерапия J.
    1.2. Лекарственное лечение M.
  2. Психологическая и экологическая среда $Y = P \star H \star G$.
    2.1. Психологический климат climate P.
    2.2. Экологическая среда дома H.
    2.2. Общая экологическая среда G.
  3. Режим, отдых $Z = O \star K$.
    3.1. Режим O.
    3.2. Отдых K.

Исследуются следующие альтернативы:

J : нет J_o, массаж J_1, ингаляция J_2, сауна J_3, рефлексотерапия J_4, лазерная терапия J_5, массаж для специальных центров/точек J_6, рефлексотерапия для специальных центров J_7, галокамеры или "соляные шахты" J_8, $J_9 = J_1 \& J_2$, $J_{10} = J_4 \& J_5$.

M : нет M_o, витамины M_1, sodium chromoglycate (один месяц и два раза в год) M_2, sodium chromoglycate (два месяца) M_3, sodium chromoglycate (три месяца) M_4.

P : нет P_o, консультация у психолога P_1.

[1] Материал главы основан на публикациях:
(i) Levin M.Sh., Sokolova L.V. Hierarchical combinatorial planning of medical treatment. Comp. Meth. and Progr. in Biomedicine, 2004, Vol. 73, No. 1, pp. 3-11.
(ii) Levin M.Sh., Firer M. Hierarchical morphological design of immunoassay technology. Computers in Biology and Medicine, 2005, Vol. 35, No. 3, pp. 229-245.

H : нет H_o, влажная уборка H_1, очистка от книжной пыли H_2, отказ от ватных изделий (одеяла, подушки, матрасы) H_3, удаление ковров H_4, исключение контактов с домашними животными H_5, дезинфекция дома для устранения насекомых H_6, удаление цветов H_7.

G : нет G_o, улучшение места (расположения) жительства G_1.

O : нет O_o, дневной сон O_1, специальные медицинские процедуры (дренаж, откашливание) O_2, спорт (бег, лыжи, плавание) O_3, душ/ванна приятной температуры O_4, холодный душ/ванна O_5, запрет элестронных игр O_6, $O_7 = O_2 \& O_4$, $O_8 = O_3 \& O_5$.

K : нет K_o, отдых в лесной среде K_1, отдых около моря K_2, отдых в горах K_3, специальный лагерь K_4, лечение в соляных шахтах K_5.

Используются следующие критерии (веса критериев указаны в скобках):

Критерии для J: эффективность C_{j1} $(+5)$; стоимость C_{j2} (-4); временной период лечебного курса C_{j3} (-3); необходимость повторения лечебного курса C_{j4} (-3); доступность C_{j5} $(+5)$; побочные эффекты (осложнения) C_{j6} (-5).

Критерии для M: эффективность C_{m1} $(+5)$; стоимость C_{m2} (-4); временной период лечебного курса C_{m3} (-3); необходимость повторения лечебного курса C_{m4} (-3); доступность C_{m5} $(+5)$; побочные эффекты (осложнения) C_{m6} (-5).

Критерии для P: эффективность C_{p1} $(+5)$; стоимость C_{p2} (-6); доступность C_{p3} $(+5)$.

Критерии для H: эффективность C_{h1} $(+5)$; стоимость C_{h2} (-6); доступность C_{h3} $(+5)$.

Критерии для G: эффективность C_{g1} $(+5)$; стоимость C_{g2} (-6); доступность C_{g3} $(+5)$.

Критерии для O: эффективность C_{o1} $(+5)$; доступность C_{o2} $(+5)$; побочные эффекты (осложнения) C_{o3} (-3).

Критерии для K: эффективность C_{k1} $(+5)$; стоимость C_{k2} (-5); временной период лечебного курса C_{k3} (-5); необходимотсь повторения лечебного курса C_{k4} (-3); доступность C_{k5} $(+5)$; побочные эффекты (осложнения) C_{k6} (-3).

Таблицы 15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7 содержат оценки для DAs и результирующие приоритеты.

Таблица 15.1. DAs и оценки для J

DAs	C_1	C_2	C_3	C_4	C_5	C_6	При-теты
J_0	0	0	0	0	0	0	2
J_1	2	4	1	1	4	1	2
J_2	1	1	1	1	5	2	2
J_3	2	3	1	1	1	3	3
J_4	3	4	1	1	3	4	3
J_5	3	4	1	1	2	3	3
J_6	4	5	3	1	1	1	3
J_7	4	5	3	1	1	3	4
J_8	4	3	1	1	2	3	2
J_9	3	4	1	1	2	3	2
J_{10}	4	5	1	1	2	4	3

Таблица 15.2. DAs и оценки для M

DAs	C_1	C_2	C_3	C_4	C_5	C_6	При-теты
M_0	0	0	0	0	0	0	2
M_1	4	5	4	3	5	4	3
M_2	3	3	3	3	4	3	2
M_3	4	4	4	3	4	3	3
M_4	5	5	5	3	4	3	4

Таблица 15.3. DAs и оценки для P

DAs	C_1	C_2	C_3	При-теты
P_0	0	0	0	3
P_1	4	5	2	2

Таблица 15.4. DAs и оценки для G

DAs	C_1	C_2	C_3	При-теты
G_0	0	0	0	4
G_1	5	0	1	1

Таблица 15.5. DAs и оценки для H

DAs	C_1	C_2	C_3	При-теты
H_0	0	0	0	3
H_1	3	1	5	1
H_2	3	3	4	2
H_3	4	2	4	1
H_4	3	2	5	1
H_5	5	4	3	2
H_6	4	3	3	2
H_7	4	2	4	1

Таблица 15.6. DAs и оценки для O

DAs	C_1	C_2	C_3	При-теты
O_0	0	0	0	3
O_1	2	5	0	1
O_2	3	4	0	1
O_3	5	3	3	1
O_4	4	3	1	1
O_5	5	3	4	2
O_6	3	3	2	2
O_7	4	3	1	1
O_8	5	3	4	2

Таблица 15.7. DAs и оценки для K

DAs	C_1	C_2	C_3	C_4	C_5	C_6	При-теты
K_0	0	0	0	0	0	0	2
K_1	4	3	3	3	4	2	2
K_2	4	5	3	3	4	2	3
K_3	4	5	3	3	3	1	4
K_4	3	4	2	3	3	1	3
K_5	4	4	1	3	1	2	3

15.1.2 Совместимость и составные решения

Оценки совместимости для локальных DAs представлены в Таблице 15.8, в Таблице 15.9. Теперь можно вычислить наилучшие промежуточные составные решений:

1. Для X $(N(X) = (5; 0, 2, 0, 0)) : X_1 = J_o \star M_2,\ X_2 = J_1 \star M_o,\ X_3 = J_1 \star M_2,\ X_4 = J_2 \star M_o,$ $X_5 = J_2 \star M_2,\ X_6 = J_8 \star M_o,\ X_7 = J_9 \star M_o,\ X_8 = J_9 \star M_2.$

2. Для Y $(N(Y) = (5; 2, 1, 0, 0))$: $Y_1 = P_1 \star H_1 \star G_1,\ Y_2 = P_1 \star H_4 \star G_1,\ Y_3 = P_1 \star H_7 \star G_1.$

3. Для Z $(N(Z) = (5; 2, 0, 0, 0))$: $Z_1 = O_1 \star K_o,\ Z_2 = O_1 \star K_1,\ Z_3 = O_2 \star K_o,\ Z_4 = O_2 \star K_1,$ $Z_5 = O_3 \star K_o,\ Z_6 = O_3 \star K_1,\ Z_7 = O_4 \star K_o,\ Z_8 = O_4 \star K_1,\ Z_9 = O_7 \star K_o.$

На основе экспертных оценок выбираются следующие локальные составные DAs (приоритеты указаны в скобках): $X_6 = J_8 \star M_o(1),\ X_8 = J_9 \star M_2(2),\ Y_1 = P_1 \star H_1 \star G_1(1),\ Y_2 = P_1 \star H_4 \star G_1(1),\ Y_3 = P_1 \star H_7 \star G_1(1),\ Z_2 = O_1 \star K_1(2),\ Z_6 = O_3 \star K_1(1),\ Z_8 = O_4 \star K_1(1),\ Z_9 = O_7 \star K_o(1)$. Оценки совместимости для выбранных DAs представлены в Таблице 15.9. В результате, получаются две группы Парето-эффективных составных решений:

1. $N(S) = (5; 2, 1, 0, 0)$: $S_1 = X_8 \star Y_1 \star Z_6,\ S_2 = X_8 \star Y_1 \star Z_8,\ S_3 = X_8 \star Y_1 \star Z_9.$

2. $N(S) = (3; 3, 0, 0, 0)$: $S_4 = X_6 \star Y_1 \star Z_6,\ S_5 = X_6 \star Y_3 \star Z_8.$

Таблица 15.8. Оценки совместимости

	G_0	G_1	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
P_0	0	5	0	5	5	4	5	4	5	5
P_1	5	5	4	5	5	4	5	4	5	5
G_0			0	5	5	4	5	4	5	5
G_1			4	5	5	4	5	4	5	5

	Y_1	Y_2	Y_3	Z_2	Z_6	Z_8	Z_9
X_6	5	4	5	5	3	1	2
X_8	5	4	4	5	5	5	5
Y_1				5	5	5	5
Y_2				4	4	4	4
Y_3				5	5	5	5

Таблица 15.9. Оценки совместимости

	M_0	M_1	M_2	M_3	M_4
J_0	0	5	5	5	5
J_1	5	5	5	5	5
J_2	5	5	5	5	5
J_3	5	5	5	4	4
J_4	5	5	5	4	4
J_5	5	5	5	4	4
J_6	5	5	5	5	5
J_7	5	5	5	5	5
J_8	5	4	4	4	4
J_9	5	5	5	5	5
J_{10}	5	5	3	3	3

	K_0	K_1	K_2	K_3	K_4	K_5
O_0	0	5	5	5	5	5
O_1	5	5	5	5	5	5
O_2	5	5	5	5	4	5
O_3	5	5	5	5	5	5
O_4	5	4	4	4	4	4
O_5	5	5	5	5	5	5
O_6	5	5	5	5	5	5
O_7	5	4	4	4	4	4
O_8	5	3	5	3	5	4

Рис. 15.1 содержит иерархическую модель плана лечения, включая DAs, составные DAs и приоритеты (в скобках). Рис. 15.2 иллюстрирует дискретное пространство качества (решетку качества), идеальную точку и Парето-эффективные решения.

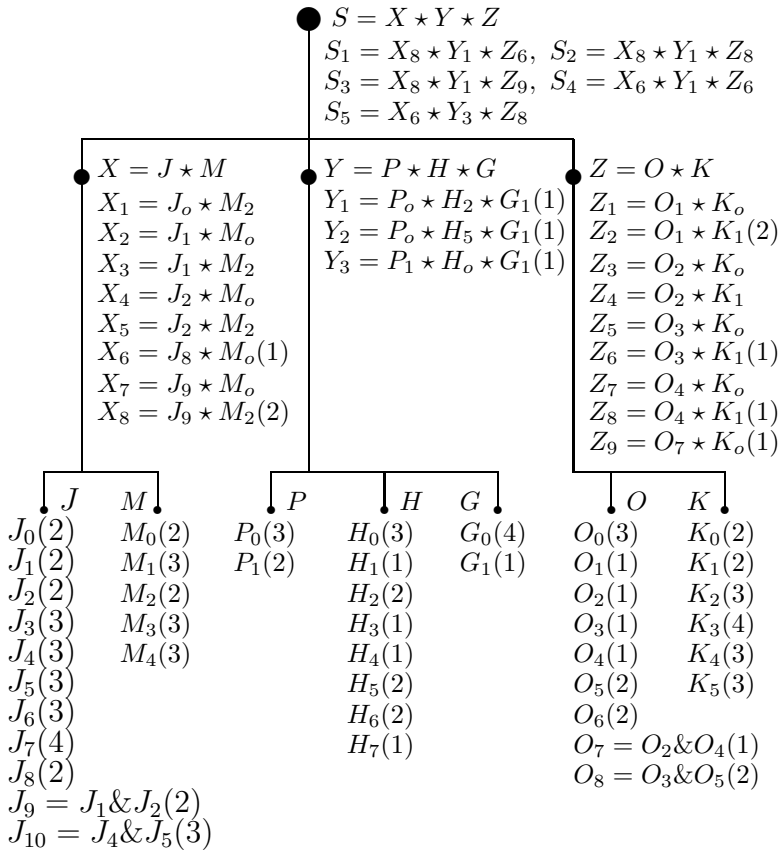


Рис. 15.1. Иерархическая модель плана лечения

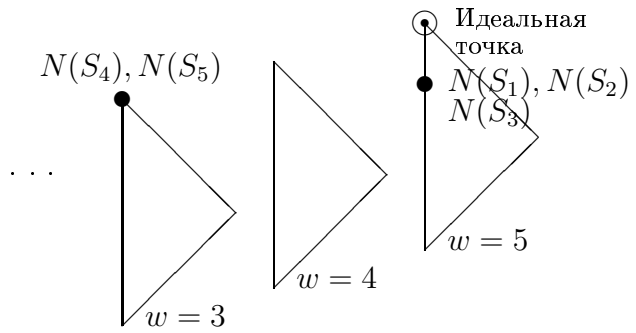


Рис. 15.2. Парето-эффективные точки

15.1.3 Улучшение, много-стадийный план

Улучшения основывается на следующем:

1. Выбор составных решений (некоторых планов лечения) для улучшения.

2. Улучшение локальных DAs.

3. Улучшение совместимости между локальными DAs.

Таблица 15.10 иллюстрирует процесс улучшения.

Таблица 15.10. Узкие места и улучшения

Составные DAs	Узкие места		Акции w/r
	DAs	IC	
$S_2 = X_8 \star Y_1 \star Z_6$	X_8		$2 \Rightarrow 1$
$S_2 = X_8 \star Y_1 \star Z_8$	X_8		$2 \Rightarrow 1$
$S_3 = X_8 \star Y_1 \star Z_9$	X_8		$2 \Rightarrow 1$
$S_4 = X_6 \star Y_1 \star Z_6$		(X_6, Z_6)	$3 \Rightarrow 5$
$S_5 = X_6 \star Y_3 \star Z_6$		$(X_6, Z_6$	$3 \Rightarrow 5$

Рассмотрим *траекторный метод* для формирования много-стадийного плана лечения (предварительная стадия, следующая стадия и т.д.). используются две фазы:

1. Формирование плана лечения для каждой.

2. Композиция траектории как последовательности решения для стадий.

Пример результирующей траектории имеет вид (Рис. 15.3):

$\beta_1 =< S_1^1, S_4^2, S_3^3 >,\ \beta_2 =< S_3^1, S_2^2, S_5^3 >.$

При этом использованы обозначения: S_j^i - составное решений, где i соответствует номеру стадии, j соответствует номеру решения на каждой стадии i.

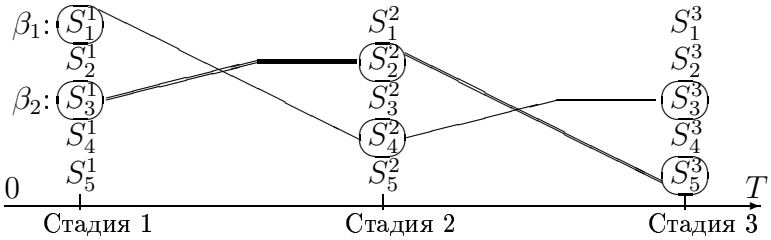


Рис. 15.3. Иллюстрация к моногостадийному плану

15.2 Технология иммунологического анализа

15.2.1 Структура и альтернативы

Структура проектируемой системы имеет вид:

0. Система $S = H \star B \star I \star D \star M$.
1. Стадия 1. $H = P \star Q$: *1.1. покрытие* P, *1.2. промывка* Q.
2. Стадия 2 (Формирование блоков) B.
3. Стадия 3 $I = O \star T$: *3.1. инкубация проб* O, *3.2. промывка* T.
4. Стадия 4. $D = J \star E$: *4.1. распознавание проб* J. *4.2. промывка* E.
5. Стадия 5 (Измерение) M.

Генерация альтернатив проводится на основе морфологических таблиц. Для компонента *coating* P данные содержатся в Таблице 15.11. В результате пять DAs выбираются: P_1, P_2, P_3, P_4, P_5. Приоритеты этих DAs представлены на Рис. 15.4 (экспертные суждения).

Таблица 15.12 содержит три DAs для Q: Q_1, Q_2, Q_3. Приоритеты этих DAs представлены на Рис. 15.4 (экспертные суждения).

Таблица 15.11. Морфологическая генерация DAs для P

Свойство	P_1	P_2	P_3	P_4	P_5
1. Биолог.:					
(a) Белок	*		*	*	
(b) Липид					*
(c) НК		*			
2. Биохим. (I):					
(a) Мало			*	*	*
(b) Много	*	*			
3. Биохим. (II):					
(a) Гидропонный					*
(b) Гидрофильный	*	*	*	*	
4. Концентрация $(\mu g/ml)$:					
(a) 0.5					
(b) 1		*			
(c) 2					*
(d) 4	*				
(e) 8			*	*	
(f) 16					
5. Три пластика:					
(a) PVP					
(b) PS	*	*	*	*	*
6. Предв. UV обраб. пластика:					
(a) Нет					
(b) Низк.		*			*
(c) Выс.	*		*	*	
7. Растворитель:					
(a) Solire					
(b) Вода					
(c) Фосфат	*		*		
(d) Карбонат				*	
(e) Метанол		*			*
8. Тип связи с твердой фазой:					
(a) Ковалентный			*	*	
(b) Не ковалентный	*	*			*
9. Время инкубации (h):					
(a) 0.5					
(b) 2					
(c) 8			*	*	
(d) 16	*	*			*
10. Температура инкубации (C^o):					
(a) 4	*		*	*	
(b) t окр. среды		*			*
(c) 37					
11. Специальный процесс смешения:					
(a) Нет	*	*	*	*	*
(b) Ротация					
(c) Встряхивание					

Таблица 15.13 содержит четыре DAs для B (*формирование блоков*): B_1, B_2, B_3, B_4. Использованы два критерия для оценивания DAs: (1) K_1 время (минимальное значение является лучшим; час), шкала: 0.5, 1.0, 2.0; (2) K_2 уровень фона или качество (минимальное значение является лучшим; %), шкала: 1%, 5%, 10%; Оценки и приоритеты (вычисления в виде многокритериального ранжирования) представлены в Таблице 15.16.

Таблица 15.12. Морфологическая генерация DAs для Q (аналогично для T/E)

Свойство	Q_1	Q_2	Q_3
1. Тип раствора:			
(a) Вода			*
(b) PBS/Детергент	*		
(c) PBS		*	
2. Число циклов:			
(a) 1			
(b) 2			*
(c) 3	*	*	
(d) 4			
3. Время каждого цикла (min):			
(a) 0.5			
(b) 1			
(c) 3	*	*	*
(d) 5			

Таблица 15.13. Морфологическая генерация DAs для B

Свойство	B_1	B_2	B_3	B_4
1. Раствор:				
(a) PBS/Глицин	*			
(b) PBS/Детергент			*	
(c) PBS-1% BSA		*		*
2. Время инкубации (час):				
(a) 0.5				
(b) 1	*		*	
(c) 2		*		*
3. Температура инкубации (C^o):				
(a) 4				*
(b) t окр. среды	*	*	*	
(c) 37				

Таблица 15.14. Оценки и приоритеты для B

DAs	K_1	K_2	При-теты
B_1	1	10	2
B_2	0.5	1	1
B_3	1	10	2
B_4	2	10	3

Таблица 15.15 содержит пять DAs для O (*инкубация пробы*): O_1, O_2, O_3, O_4, O_5. Приоритеты этих DAs (экспертные суждения) представлены на Рис. 15.4.

Таблица 15.15. Морфологическая генерация DAs для O

Свойства	O_1	O_2	O_3	O_4	O_5
1. Объем пробы:					
(a) 50	*				*
(b) 100		*		*	
(c) 200			*		
2. Время инкубации & температура:					
(a) 4				*	*
(b) t окр. среды	*	*	*		
(c) 37					
3. Нижний предел (концентрации $\mu g/ml$):					
(a) 0.01			*	*	*
(b) 1	*	*			
(c) 10					
4. Верх. предел (концентрации $\mu g/ml$):					
(a) 10		*			*
(b) 100	*		*	*	
(c) 1000					
5. Характеристики раствора (пропорция):					
(a) Чистый	*			*	
(b) Простая смесь		*			*
(c) Сложная смесь			*		
6. Тип пробы:					
(a) Сыворотка	*		*		
(b) Культ. жидкость		*			
(c) PBS или подобное				*	*
7. Характеристики пробы:					
(a) PBS					
(b) PBS/BSA/Tw	*		*	*	*
(c) PBS/Жив. сыворотка/Tw		*			

Таблица 15.16 содержит пять DAs для J (*инкубация пробы*): J_1, J_2, J_3, J_4. Приоритеты этих DAs (экспертные суждения) представлены на Рис. 15.4.

Таблица 15.16. Морфологическая генерация DAs для J

Свойство	J_1	J_2	J_3	J_4
1. Тип физического сигнала (использ. для анализа):				
(a) Радио	*			
(b) Флюресцент.		*		
(c) Биолог.			*	*
2. Время инкубации (час):				
(a) 0.5	*			
(b) 1		*		
(c) 2			*	
(d) 6				*
3. Температура инкубации (C^o):				
(a) 4		*		*
(b) t окр. среды	*		*	
(c) 37				
4. Характеристика раствора (пропорция):				
(a) PBS				
(b) PBS/BSA/Tw	*	*	*	*
(c) PBS/Жив. сыворотка/Tw				

Таблица 15.17 содержит пять DAs для M (*измерение*): M_1, M_2, M_3, M_4. Три критерия использованы для оценивания DAs:

(1) K_1 стоимость (порядковая шкала: высокая, средняя, низкая; вес критерия равен 3);

(2) K_2 время (часы, минимальное значение является лучшим; вес критерия равен 5);

(3) K_3 easy to perform (порядковая: сложно, средне, просто; вес критерия равен 5).

Оценки и результирующие приоритеты представлены в Таблице 15.18.

Таблица 15.17. Морфологическая генерация DAs для M

Свойство	M_1	M_2	M_3	M_4
1. Тип сгенерирован. сигнала:				
(a) Радиация	*			
(b) Флюресц.		*		
(c) Хим. иллюминисц.			*	
(d) Колометр.				*
2. Тип аналит. аппаратуры:				
(a) Счетчик-сцинтилятор	*			
(b) Флорометер		*		
(c) Светоиспускание			*	
(d) Спектрометер				*
3. Время измерения (час):				
(a) 0.1		*	*	
(b) 0.15				*
(c) 0.5				
(d) 1.0	*			

Таблица 15.18. Оценки и приоритеты для M

DAs	K_1	K_2	K_2	При-теты
M_1	Выс.	1.0	2	3
M_2	Сред.	0.1	3	1
M_3	Сред.	0.1	2	2
M_4	Низ.	0.15	3	1

Оценки совместимости для локальных DAs представлены в Таблице 15.19.

Таблица 15.19. Оценки совместимости

	P_1	P_2	P_3	P_4	P_5
Q_1	3	2	2	2	1
Q_2	3	1	2	1	3
Q_3	1	1	1	1	0

	O_1	O_2	O_3	O_4	O_5
T_1	2	2	2	2	2
T_2	2	2	1	2	1
T_3	2	2	1	2	1

	J_1	J_2	J_3	J_4
E_1	2	2	2	2
E_1	2	2	2	2
E_3	2	2	2	2

15.2.2 Составные решения для стадий

Теперь рассматриваются составные решения DAs для стадии 1 H, стадии 3 I, стадии 4 D. Для H получаются следующие DAs:

$H_1 = P_1 \star Q_1$, $N(H_1) = (3; 2, 0, 0)$, $H_2 = P_1 \star Q_2$, $N(H_2) = (3; 1, 1, 0)$,
$H_3 = P_1 \star Q_3$, $N(H_3) = (1; 1, 0, 1)$, $H_4 = P_2 \star Q_1$, $N(H_4) = (2; 1, 1, 0)$,
$H_5 = P_2 \star Q_2$, $N(H_5) = (1; 0, 2, 0)$, $H_6 = P_2 \star Q_3$, $N(H_6) = (1; 0, 1, 1)$,
$H_7 = P_3 \star Q_1$, $N(H_7) = (2; 1, 1, 0)$, $H_8 = P_3 \star Q_2$, $N(H_8) = (2; 0, 2, 0)$,
$H_9 = P_3 \star Q_3$, $N(H_9) = (1; 0, 2, 0)$, $H_{10} = P_4 \star Q_1$, $N(H_{10}) = (2; 1, 0, 1)$,
$H_{11} = P_4 \star Q_2$, $N(H_{11}) = (1; 0, 1, 1)$, $H_{12} = P_4 \star Q_3$, $N(H_{12}) = (1; 0, 0, 2)$,
$H_{13} = P_5 \star Q_1$, $N(H_{13}) = (1; 1, 0, 1)$, $H_{14} = P_5 \star Q_2$, $N(H_{14}) = (3; 0, 1, 1)$,
$H_{15} = P_5 \star Q_3$, $N(H_{15}) = (0; 0, 0, 2)$.

Во-первых, решение H_{15} исключается как недопустимое. Во-вторых, выбираются следующие наилучшие решения:

(а) идеальное решение H_1, приоритет равен 1;

(б) H_4 и H_7, приоритет равен 2;

(в) H_2, H_{10}, H_{11}, H_{13}, приоритет равен 3.

Для I получаются следующие DAs:

$I_1 = O_1 \star T_1$, $N(I_1) = (2; 2, 0, 0)$, $I_2 = O_1 \star T_2$, $N(I_2) = (2; 1, 1, 0)$,
$I_3 = O_1 \star T_3$, $N(I_3) = (2; 1, 0, 1)$, $I_4 = O_2 \star T_1$, $N(I_4) = (2; 2, 0, 0)$,
$I_5 = O_2 \star T_2$, $N(I_5) = (2; 1, 1, 0)$, $I_6 = O_2 \star T_3$, $N(I_6) = (2; 1, 1, 0)$,
$I_7 = O_3 \star T_1$, $N(I_7) = (2; 2, 0, 0)$, $I_8 = O_3 \star T_2$, $N(I_8) = (1; 1, 1, 0)$,
$I_9 = O_3 \star T_3$, $N(I_9) = (1; 1, 0, 1)$, $I_{10} = O_4 \star T_1$, $N(I_{10}) = (2; 2, 0, 0)$,
$I_{11} = O_4 \star T_2$, $N(I_{11}) = (2; 1, 1, 0)$, $I_{12} = O_4 \star T_3$, $N(I_{12}) = (2; 1, 0, 1)$,
$I_{13} = O_5 \star T_1$, $N(I_{13}) = (2; 2, 0, 0)$, $I_{14} = O_5 \star T_2$, $N(I_{14}) = (1; 1, 1, 0)$,
$I_{15} = O_5 \star T_3$, $N(I_{15}) = (1; 1, 0, 1)$.

Целесообразно выбрать следующие наилучшие решения (приоритет равен 1): I_1, I_4, I_7, I_{10}, I_{13}.

Для D получаются следующие решения:

$D_1 = J_1 \star E_1$, $N(D_1) = (2; 2, 0, 0)$, $D_2 = J_1 \star E_2$, $N(D_2) = (2; 1, 1, 0)$,
$D_3 = J_1 \star E_3$, $N(D_3) = (2; 1, 0, 1)$, $D_4 = J_2 \star E_1$, $N(D_4) = (2; 2, 0, 0)$,
$D_5 = J_2 \star E_2$, $N(D_5) = (2; 1, 1, 0)$, $D_6 = J_2 \star E_3$, $N(D_6) = (2; 1, 0, 1)$,
$D_7 = J_3 \star E_1$, $N(D_7) = (2; 2, 0, 0)$, $D_8 = J_3 \star E_2$, $N(D_8) = (2; 1, 1, 0)$,
$D_9 = J_3 \star E_3$, $N(D_9) = (2; 1, 0, 1)$, $D_{10} = J_4 \star E_1$, $N(D_{10}) = (2; 2, 0, 0)$,
$D_{11} = J_4 \star E_2$, $N(D_{11}) = (2; 1, 1, 0)$, $D_{12} = J_4 \star E_3$, $N(D_{12}) = (2; 1, 0, 1)$.

Далее выбираются следующие наилучшие решения (приоритет равен 1): D_1, D_4, D_7, D_{10}.

15.2.3 Структура системы и системные решения

На Рис. 15.4 представлена структура системы из пяти стадий: DAs и их приоритеты указаны в скобках. Компоненты Q, T, E являются эквивалентными.

0. Система: $S = H \star B \star I \star D \star M$.

1. Стадия 1: $H = P \star Q$. **Составные DAs:**

$H_1 = P_1 \star Q_1(1)$, $H_4 = P_2 \star Q_1(2)$,
$H_7 = P_3 \star Q_1(2)$, $H_2 = P_1 \star Q_2(3)$,
$H_{10} = P_4 \star Q_1(3)$, $H_{13} = P_5 \star Q_2(3)$.

1.1. Покрытие: P. DAs:
$P_1(1)$, $P_2(2)$, $P_3(2)$, $P_4(3)$, $P_5(3)$.

1.2. Промывка: Q. DAs:
$Q_1(1)$, $Q_2(2)$, $Q_3(3)$.

2. Стадия 2 (Формирование блоков): B. DAs:
$B_1(2)$, $B_2(1)$, $B_3(2)$, $B_4(3)$.

3. Стадия 3: $I = O \star T$. Составные DAs:
$I_1 = O_1 \star T_1(1)$, $I_4 = O_2 \star T_1(1)$,
$I_7 = O_3 \star T_1(1)$, $I_{10} = O_4 \star T_1(1)$, $I_{13} = O_5 \star T_1(1)$.

3.1. Инкубация проб: O. DAs:
$O_1(1)$, $O_2(1)$, $O_3(1)$, $O_4(1)$, $O_5(1)$.

3.2. Промывка: T. DAs:
$T_1(1)$, $T_2(2)$, $T_3(3)$.

4. Стадия 4: $D = J \star E$. Составные DAs:
$D_1 = J_1 \star E_1(1)$, $D_4 = J_2 \star E_1(1)$,
$D_7 = J_3 \star E_1(1)$, $D_{10} = J_4 \star E_1(1)$.

4.1. Анализ проб: J. DAs:
$J_1(1)$, $J_2(1)$, $J_3(1)$, $J_4(1)$.

4.2. Промывка: E. DAs:
$E_1(1)$, $E_2(2)$, $E_3(3)$.

5. Стадия 5 (Измерение): M. DAs:
$M_1(3)$, $M_2(1)$, $M_3(2)$, $M_4(1)$.

Рис. 15.4. Технология иммунологического анализа

Оценки совместимости для H, B, I, D, M представлены в Таблице 15.20. Оценки совместимости равны 3 для других пар: (1) B и I, D, M; (2) I и D, M; (3) D и M.

Таблица 15.20. Оценки совместимости

	B_1	B_2	B_3	B_4	I_1	I_4	I_7	I_{10}	I_{13}	D_1	D_4	D_7	D_{10}	M_1	M_2	M_3	M_4
H_1	1	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
H_4	1	0	0	0	2	2	3	3	3	2	2	3	3	2	2	3	3
H_7	1	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
H_2	1	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
H_{10}	1	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
H_{12}	0	0	0	0	2	2	3	3	3	2	2	3	3	2	2	3	3

В результате получаются 40 составных DAs для технологии иммунологического анализа ($N = (3; 5, 0, 0)$):

$S_1 = H_1 \star B_2 \star I_1 \star D_1 \star M_2$, $S_2 = H_1 \star B_2 \star I_1 \star D_4 \star M_2$,
$S_3 = H_1 \star B_2 \star I_1 \star D_7 \star M_2$, $S_4 = H_1 \star B_2 \star I_1 \star D_{10} \star M_2$,
$S_5 = H_1 \star B_2 \star I_4 \star D_1 \star M_2$, $S_6 = H_1 \star B_2 \star I_4 \star D_4 \star M_2$,
$S_7 = H_1 \star B_2 \star I_4 \star D_7 \star M_2$, $S_8 = H_1 \star B_2 \star I_4 \star D_{10} \star M_2$,
$S_9 = H_1 \star B_2 \star I_7 \star D_1 \star M_2$, $S_{10} = H_1 \star B_2 \star I_7 \star D_4 \star M_2$,
$S_{11} = H_1 \star B_2 \star I_7 \star D_7 \star M_2$, $S_{12} = H_1 \star B_2 \star I_7 \star D_{10} \star M_2$,
$S_{13} = H_1 \star B_2 \star I_{10} \star D_1 \star M_2$, $S_{14} = H_1 \star B_2 \star I_{10} \star D_4 \star M_2$,
$S_{15} = H_1 \star B_2 \star I_{10} \star D_7 \star M_2$, $S_{16} = H_1 \star B_2 \star I_{10} \star D_{10} \star M_2$,
$S_{17} = H_1 \star B_2 \star I_{13} \star D_1 \star M_2$, $S_{18} = H_1 \star B_2 \star I_{13} \star D_4 \star M_2$,
$S_{19} = H_1 \star B_2 \star I_{13} \star D_7 \star M_2$, $S_{20} = H_1 \star B_2 \star I_{13} \star D_{10} \star M_2$,
$S_{21} = H_1 \star B_2 \star I_1 \star D_1 \star M_4$, $S_{22} = H_1 \star B_2 \star I_1 \star D_4 \star M_4$,
$S_{23} = H_1 \star B_2 \star I_1 \star D_7 \star M_4$, $S_{24} = H_1 \star B_2 \star I_1 \star D_{10} \star M_4$,

$S_{25} = H_1 \star B_2 \star I_4 \star D_1 \star M_4$, $S_{26} = H_1 \star B_2 \star I_4 \star D_4 \star M_4$,
$S_{27} = H_1 \star B_2 \star I_4 \star D_7 \star M_4$, $S_{28} = H_1 \star B_2 \star I_4 \star D_{10} \star M_4$,
$S_{29} = H_1 \star B_2 \star I_7 \star D_1 \star M_4$, $S_{30} = H_1 \star B_2 \star I_7 \star D_4 \star M_4$,
$S_{31} = H_1 \star B_2 \star I_7 \star D_7 \star M_4$, $S_{32} = H_1 \star B_2 \star I_7 \star D_{10} \star M_4$,
$S_{33} = H_1 \star B_2 \star I_{10} \star D_1 \star M_4$, $S_{34} = H_1 \star B_2 \star I_{10} \star D_4 \star M_4$,
$S_{35} = H_1 \star B_2 \star I_{10} \star D_7 \star M_4$, $S_{36} = H_1 \star B_2 \star I_{10} \star D_{10} \star M_4$,
$S_{37} = H_1 \star B_2 \star I_{13} \star D_1 \star M_4$, $S_{38} = H_1 \star B_2 \star I_{13} \star D_4 \star M_4$,
$S_{39} = H_1 \star B_2 \star I_{13} \star D_7 \star M_4$, $S_{40} = H_1 \star B_2 \star I_{13} \star D_{10} \star M_4$.

15.3 Выводы к главе

В данной главе описаны две задачи комбинаторного планирования в биомедицине: планирование лечения детской астмы, планирование иммунологического анализа. Можно указать следующие направления для исследований в будущем:

1. исследование других типов болезней;

2. применение при планировании лечения двух-уровневого подхода, включащего уровень диагностики и уровень лечения (раздел 8.3.2);

3. анализ рассмотренных задач в режиме реального времени;

4. применение предложенного варианта комбинаторного планирования в различных биотехнологических процессах;

5. учет неопределенности параметров.

Глава 16

Приложения в телеметрии, сенсорных системах

В данной главе [1] представлены два прикладных примера композиции модульных систем: телеметрическая система и радио сенсор.

16.1 Телеметрическая система

Последние десятилетия важность различных телеметрических систем существенно возросла [78, 118]. Далее рассматривается комбинаторные инженерные схемы для телеметрической системы (Рис. 16.1) (используются интервальные оценки в виде мультимножеств): (а) модульного проектирования системы, (б) улучшение полученных решений (конфигураций системы), (в) агрегирование полученных решений (конфигураций). Базовый вариант анализа и синтеза телеметричекой системы (с порядковыми оценками альтерантив) представлен в [387]. Задача синтеза телеметрической системы с использованием интервальных оценок в виде мультимножеств описана в [361]. Оценки проектных альтернатив DAs и их совместимости основаны на экспертном оценивании.

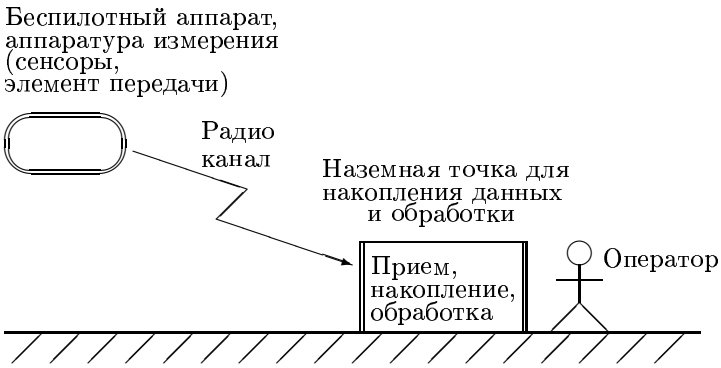


Рис. 16.1. Телеметрическая система

[1] Материал главы основан на публикациях:
(i) M.Sh. Levin, Composition of modular telemetry system with interval multiset estimates. Electronic preprint. 9 pp., July 25, 2012. http://arxiv.org/abs/1207.6051 [cs.SY]
(ii) Levin M.Sh., Khodakovskii I.A., Structural Composition of the Telemetry System. Autom. and Rem. Contr., 2007, vol. 68, no. 9, pp. 1654-1661.
(iii) Levin M.Sh., Fimin A.V., Configuration of alarm wireless sensor element. 2nd Int. Congress ICUMT-2010, Moscow, 924-928, 2010.
(iv) Levin M.Sh. Aggregation of composite solutions: strategies, models, examples. Electronic preprint, 72 pp., Nov. 29, 2011. http://arxiv.org/abs/1111.6983 [cs.SE]

Иерархическая структура телеметрической системы имеет вид (Рис. 16.2):

0. Телеметрическая система $S = A \star R \star L$:
1. Бортовое оборудование $A = D \star E \star F$:
1.1. энергетическая установка D,
1.2. сенсорные элементы E,
1.3. обработка данных F.
2. Радио канал R.
3. Наземный пункт $L = B \star O$:
3.1. наземное оборудование $B = U \star V$:
3.1.1. энергетическое оборудование U,
3.1.2. рабочее место (места) оператора(ов)) V,
3.2. оператор(ы) O.

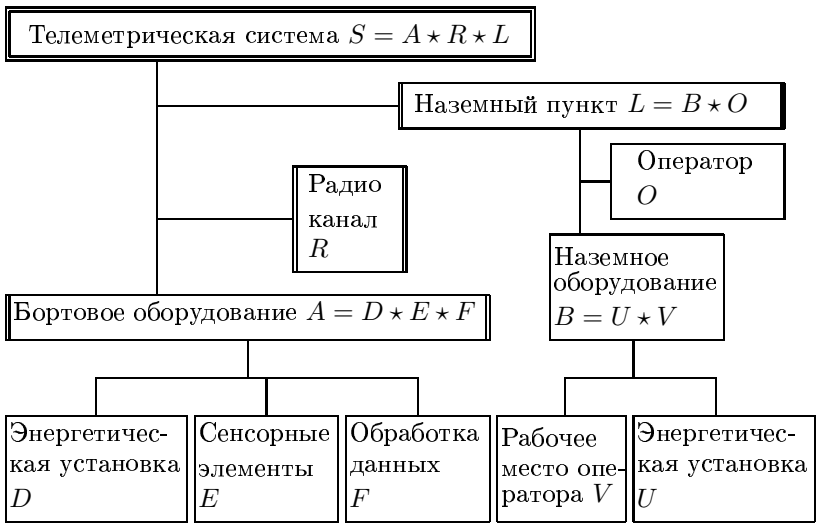


Рис. 16.2. Структура телеметрической системы

Рассматривается численный пример для бортовой телеметрической подсистемы из [387] (Рис. 16.3):

1. Бортовая подсистема $A = D \star E \star F$.
1.1. Энергетическая установка $D = X \star Y \star Z$:
1.1.1. стабилизатор X: X_1 (стандартный), X_2 (транзисторный), X_3 (высоко-стабильный),
1.1.2. главный источник Y: Y_1 (Li-ion), Y_2 (Cd-Mn), Y_3 (Li);
1.1.3. элемент emergency cell Z: Z_1 (Li-ion), Z_2 (Cd-Mn), Z_3 (Li).
1.2. Сенсорный элементы $E = I \star Q \star G$:
1.2.1. сенсоры ускорения I: I_1 (ADXL), I_2 (ADIS), I_3 (MMA);
1.2.2. сенсоры позиционирования Q: Q_1 (SS12), Q_2 (SS16), Q_3 (SS19), Q_4 (SS49), Q_5 (SS59), Q_6 (SS94);
1.2.3. глобальная система позиционирования (GPS) G: G_1 (EB), G_2 (GT), G_3 (LS), G_4 (ZX).
1.3. система обработки данных $F = H \star C \star W$:
1.3.1. устройство памяти H: H_1 (SRAM), H_2 (DRAM), H_3 (FRAM);
1.3.2. устройство обработки (CPU) C: C_1 (AVR), C_2 (ARM), C_3 (ADSP), C_4 (BM);
1.3.3. устройство записи данных W: W_1 (встроенное ADC), W_2 (внещнее ADC I2C), W_3 (внешнее ADC SPI), W_4 (внешнее ADC 2W), W_5 (внешнее ADC UART(1)).

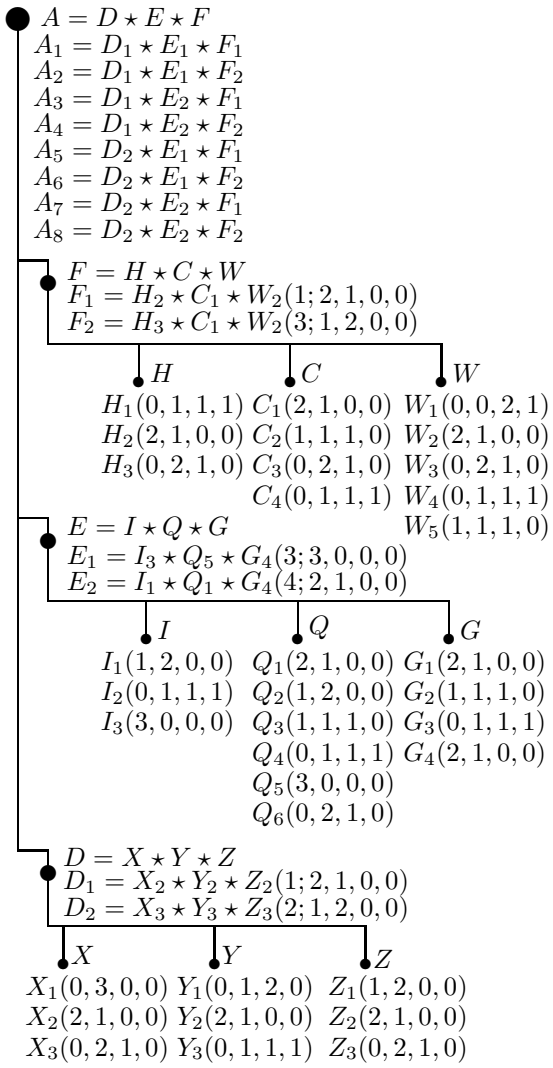


Рис. 16.3. Структура бортовой подсистемы

Интервальные оценки в виде мультимножеств для DAs показаны на Рис. 16.3 в круглых скобках (экспертное оценивание). Порядковые оценки совместимости проектных альтернатив представлены в Таблице 16.1 (экспертные оценки). Следует отметить, что исходное комбинаторное множество проектных решений включает 116640 возможных системных комбинаций (т.е., $(3 \times 3 \times 3) \times (3 \times 6 \times 4) \times (3 \times 4 \times 5)$).

Таблица 16.1. Оценки совместимости

	Y_1	Y_2	Y_3	Z_1	Z_2	Z_3
X_1	3	2	2	1	1	1
X_2	2	1	1	1	1	1
X_3	4	3	3	1	1	2
Y_1				2	1	1
Y_2				1	2	1
Y_3				1	1	2

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	G_1	G_2	G_3	G_4
I_1	4	4	4	4	4	4	3	4	4	4
I_2	3	3	3	3	3	3	2	3	3	1
I_3	3	3	3	3	2	2	1	1	1	3
Q_1							3	2	3	4
Q_2							1	1	3	1
Q_3							2	2	3	4
Q_4							2	2	2	4
Q_5							2	2	2	4
Q_6							2	2	2	4

	C_1	C_2	C_3	C_4	W_1	W_2	W_3	W_4	W_5
H_1	3	3	3	2	3	3	3	3	3
H_2	1	1	2	3	2	3	3	2	2
H_3	4	3	3	3	3	3	3	3	3
C_1					3	3	3	3	3
C_2					3	3	3	3	3
C_3					3	3	3	3	3
C_4					1	1	1	1	1

Получены следующие Парето-эффективные составные DAs:
(1) для D: $D_1 = X_2 \star Y_2 \star Z_2$, $N(D_1) = (1;2,1,0,0)$;

$D_2 = X_3 \star Y_3 \star Z_3$, $N(D_2) = (2; 1, 2, 0, 0)$;
(2) для E: $E_1 = I_3 \star Q_5 \star G_4$, $N(E_1) = (3; 3, 0, 0, 0)$;
$E_2 = I_1 \star Q_1 \star G_4$, $N(E_2) = (4; 2, 1, 0, 0)$;
(3) для F: $F_1 = H_2 \star C_1 \star W_2$, $N(F_1) = (1; 2, 1, 0, 0)$;
$F_2 = H_3 \star C_1 \star W_2$, $N(F_2) = (3; 1, 2, 0, 0)$.
Рис. 16.4 иллюстрирует качество решений для компонента F.

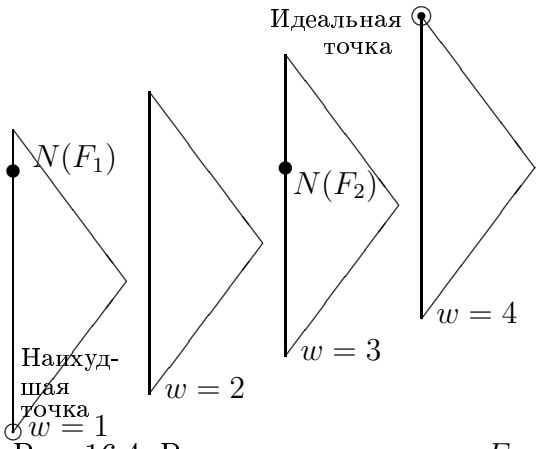


Рис. 16.4. Решетка качества для F

В результате, получаются (как комбинации) восемь решений (DAs) для исследуемой подсистемы:
$A_1 = D_1 \star E_1 \star F_1 = (X_2 \star Y_2 \star Z_2) \star (I_3 \star Q_5 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_2 = D_1 \star E_1 \star F_2 = (X_2 \star Y_2 \star Z_2) \star (I_3 \star Q_5 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_3 = D_1 \star E_2 \star F_1 = (X_2 \star Y_2 \star Z_2) \star (I_1 \star Q_1 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_4 = D_1 \star E_2 \star F_2 = (X_2 \star Y_2 \star Z_2) \star (I_1 \star Q_1 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_5 = D_2 \star E_1 \star F_1 = (X_3 \star Y_3 \star Z_3) \star (I_3 \star Q_5 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_6 = D_2 \star E_1 \star F_2 = (X_3 \star Y_3 \star Z_3) \star (I_3 \star Q_5 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_7 = D_2 \star E_2 \star F_1 = (X_3 \star Y_3 \star Z_3) \star (I_1 \star Q_1 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_8 = D_2 \star E_2 \star F_2 = (X_3 \star Y_3 \star Z_3) \star (I_1 \star Q_1 \star G_4) \star (H_3 \star C_1 \star W_2)$.

Теперь можно рассмотреть простейшие улучшения в виде реконфигураций полученных решений (т.е., по одному элементу или парной совместимости). Подсистема $F = H \star C \star W$ рассматривается как пример (Таблица 16.2).

Таблица 16.2. Узкие места, улучшения

Составные DAs	Узкие места DA/IC	Улучшение w/e
F_1	W_2	$(2,1,0,0) \Rightarrow (3,0,0,0)$
F_1	C_1	$(2,1,0,0) \Rightarrow (3,0,0,0)$
F_1	H_2	$(2,1,0,0) \Rightarrow (3,0,0,0)$
F_1	(H_2, W_2)	$1 \Rightarrow 3$
F_2	(H_3, W_2)	$3 \Rightarrow 4$
F_2	(C_1, W_2)	$3 \Rightarrow 4$
F_2	W_2	$(2,1,0,0) \Rightarrow (3,0,0,0)$
F_2	C_1	$(2,1,0,0) \Rightarrow (3,0,0,0)$
F_2	H_3	$(0,2,1,0) \Rightarrow (3,0,0,0)$

Теперь рассматривается гипотетический процесс улучшения (по элементам) для F_2 со бинарными переменными $\{y_{ij}\}$:
(1) две версии для элемента W_2: y_{11} (нет), y_{12} ($(2,1,0,0) \Rightarrow (3,0,0,0)$);
(2) две версии для элемента C_1: y_{21} (нет), y_{22} ($(2,1,0,0) \Rightarrow (3,0,0,0)$);

(3) пять версий для элемента H_3: y_{31} (нет), y_{32} $((0,2,1,0) \Rightarrow (0,3,0,0))$, y_{33} $((0,2,1,0) \Rightarrow (1,2,0,0))$, y_{34} $((0,2,1,0) \Rightarrow (2,1,0,0))$, y_{35} $((0,2,1,0) \Rightarrow (3,0,0,0))$.

В Таблице 16.3 представлены бинарные переменные (y_{ij}), улучшения и их оценки (оценки имеют иллюстративный характер). Задача улучшения (поиск решения в виде медианы для задачи типа блочного рюкзака) имеет вид:

$$\arg\min_{M \in \overline{D}} \sum_{(i,j) \in S = \{(i,j) | y_{ij}=1\}} |\delta(M, e_{ij})| \quad s.t. \ \sum_{i=1}^{3} \sum_{j=1}^{q_i} a_{ij} y_{ij} \le b, \ \sum_{j=1}^{q_i} y_{ij} = 1 \ \forall i = \overline{1,3}, \ y_{ij} \in \{0,1\}.$$

Таблица 16.3. Улучшения для F_2

Акция улучшения	Оценка e_{ij}	Стоимость (a_{ij})
y_{11} (W_2, нет)	$(2,1,0,0)$	0
y_{12} ($W_2 \Rightarrow W_2^1$, улучшение 1)	$(3,0,0,0)$	17
y_{21} (C_1, нет)	$(2,1,0,0)$	0
y_{22} ($C_1 \Rightarrow C_1^1$, улучшение 1)	$(3,0,0,0)$	15
y_{31} (H_3, нет)	$(0,2,1,0)$	0
y_{32} ($H_3 \Rightarrow H_3^1$, улучшение 1)	$(0,3,0,0)$	1
y_{33} ($H_3 \Rightarrow H_3^2$, улучшение 2)	$(1,2,0,0)$	7
y_{34} ($H_3 \Rightarrow H_3^3$, улучшение 3)	$(2,1,0,0)$	13
y_{35} ($H_3 \Rightarrow H_3^4$, улучшение 4)	$(3,0,0,0)$	22

Можно привести ряд примеров составных системных улучшений, являющихся решением рассмотренной оптимизационной модели:

(1) $b = 1$: $y_{11} = 1$ (W_2, нет), $y_{21} = 1$ (C_1, нет), $y_{32} = 1$ (H_3, улучшение 1);

$F_2 \Rightarrow \widetilde{F_2^1} = H_3^1 \star C_1 \star W_2$, $e(\widetilde{F_2^1}) = (2,1,0,0)$;

(2) $b = 45$: $y_{12} = 1$ (W_2, улучшение 1), $y_{22} = 1$ (C_1, улучшение 1), $y_{34} = 1$ (Z_1, улучшение 3),

$F_2 \Rightarrow \widetilde{F_2^2} = H_3^3 \star C_1^1 \star W_2^1$, $e(\widetilde{F_2^2}) = (3,0,0,0)$.

Далее рассматривается упрощенная стратегия агрегирования (расширение “ядра системы” на основе задачи блочного рюкзака) для восьми полученных выше решений:

$A_1 = (X_2 \star Y_2 \star Z_2) \star (I_3 \star Q_5 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_2 = (X_2 \star Y_2 \star Z_2) \star (I_3 \star Q_5 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_3 = (X_2 \star Y_2 \star Z_2) \star (I_1 \star Q_1 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_4 = (X_2 \star Y_2 \star Z_2) \star (I_1 \star Q_1 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_5 = (X_3 \star Y_3 \star Z_3) \star (I_3 \star Q_5 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_6 = (X_3 \star Y_3 \star Z_3) \star (I_3 \star Q_5 \star G_4) \star (H_3 \star C_1 \star W_2)$,
$A_7 = (X_3 \star Y_3 \star Z_3) \star (I_1 \star Q_1 \star G_4) \star (H_2 \star C_1 \star W_2)$,
$A_8 = (X_3 \star Y_3 \star Z_3) \star (I_1 \star Q_1 \star G_4) \star (H_3 \star C_1 \star W_2)$.

Подрешение и надрешение представлены на Рис. 16.5 и Рис. 16.6.

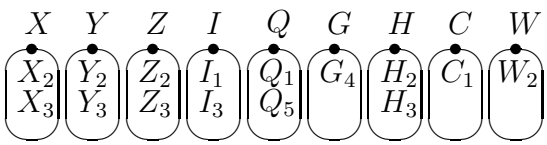


Рис. 16.5. Надрешение

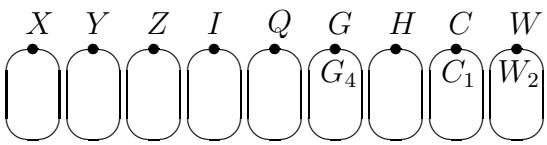


Рис. 16.6. Подрешение

Полученное подрешение содержит три элемента и эта комбинация будет рассматриваться как “ядро системы system”). Таким образом, процесс агрегирования представляет собой некую задачу блочного рюкзака для выбора (и добавления) DAs для подсистемы: $\Theta = X \star Y \star Z \star I \star Q \star H$ (Рис. 16.7) (без учета совместимости). Используются следующие бинарные переменные: $\{x_{ij}\},\ i = \overline{1,6},\ j = \overline{1,2}$.

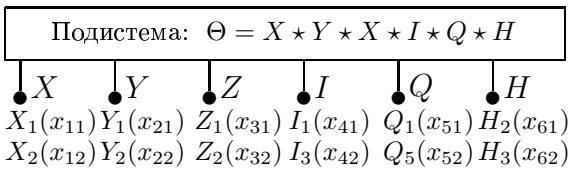


Рис. 16.7. Выбор DAs для подсистемы

Модель имеет вид:

$$\arg\min_{M \in \overline{D}} \sum_{(i,j) \in S = \{(i,j) | x_{ij} = 1\}} |\delta(M, e_{ij})|, \quad s.t. \sum_{i=1}^{6} \sum_{j=1}^{2} a_{ij} x_{ij} \le b;\ \sum_{j=1}^{2} x_{ij} = 1\ \forall i = \overline{1,6},\ x_{ij} \in \{0,1\}.$$

Оценки представлены в Таблице 16.4 (иллюстративный характер).

Таблица 16.4. Оценки для агрегации

Выбор DA	Оценки e_{ij}	Стоимость (a_{ij})
$x_{11}\,(X_2)$	$(2,1,0,0)$	11
$x_{12}\,(X_3)$	$(0,2,1,0)$	4
$x_{21}\,(Y_2)$	$(2,1,0,0)$	10
$x_{22}\,(Y_3)$	$(0,1,1,1)$	2
$x_{31}\,(Z_2)$	$(2,1,0,0)$	12
$x_{32}\,(Z_3)$	$(0,2,1,0)$	6
$x_{41}\,(I_1)$	$(1,2,0,0)$	7
$x_{42}\,(I_3)$	$(3,0,0,0)$	20
$x_{51}\,(Q_1)$	$(2,1,0,0)$	14
$x_{52}\,(Q_5)$	$(3,0,0,0)$	21
$x_{61}\,(H_2)$	$(2,1,0,0)$	13
$x_{62}\,(H_3)$	$(0,2,1,0)$	5

Некоторые примеры решений, соответствующие указанной модели, имеют вид:

(1) $b = 42$: $x_{12} = 1$ (X_3), $x_{22} = 1$ (Y_3), $x_{32} = 1$ (Z_3), $x_{41} = 1$ (I_1), $x_{51} = 1$ (Q_1), $x_{62} = 1$ (H_3), $\Theta_1 = X_3 \star Y_3 \star Z_3 \star I_1 \star Q_1 \star H_3$, $e(\Theta_1) = (0,2,1,0)$;

(2) $b = 53$: $x_{11} = 1$ (X_2), $x_{21} = 1$ (Y_2), $x_{32} = 1$ (Z_3), $x_{41} = 1$ (I_1), $x_{51} = 1$ (Q_1), $x_{62} = 1$ (H_3), $\Theta_2 = X_2 \star Y_2 \star Z_3 \star I_1 \star Q_1 \star H_3$, $e(\Theta_2) = (1,2,0,0)$;

(3) $b = 87$: $x_{11} = 1$ (X_2), $x_{21} = 1$ (Y_2), $x_{31} = 1$ (Z_2), $x_{42} = 1$ (I_3), $x_{52} = 1$ (Q_5), $x_{61} = 1$ (H_2), $\Theta_3 = X_2 \star Y_2 \star Z_2 \star I_3 \star Q_5 \star H_2$, $e(\Theta_3) = (2,1,0,0)$.

16.2 Беспроводной сенсор

В последние годы очень повысилась важность сенсорных сетей и систем [47, 152, 494, 525, 588]. Можно рассматривать упрощенную 3-уровневую архитектуру сенсорных систем a sensor system (Рис. 16.8): (i) сенсоры и локальные сенсорные сети (уровень сенсорных подсистем), (ii) коммуникационная сеть (транспортный уровень), (iii) подсистема управления (уровень управления: анализ информации, интеграция информации, принятие решений и управление) [494, 588]. Здесь рассматривается иерархическое модульное проектирование

радио сенсора на уровне 1 указанной 3-уровневой архитектуры. В качестве прикладного примера рассмотрен противопожарный сенсор.

В данном разделе описана составная комбинаторная инженерная схема модульного проектирования (Рис. 16.9): выбор проектных альтернатив (DAs) для частей системы (многокритериальное ранжирование), комбинаторый синтез (композиция) конфигураций системы (ИММП), агрегация полученных конфигураций с целью формирования результирующего решения как агрегированной конфигурации системы (построение "ядра системы" и его расширеное на основе рюкзачной модели).

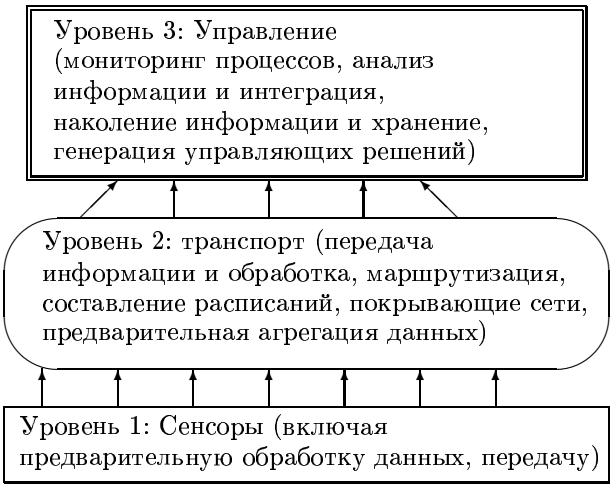


Рис. 16.8. Сенсорная система (три уровня)

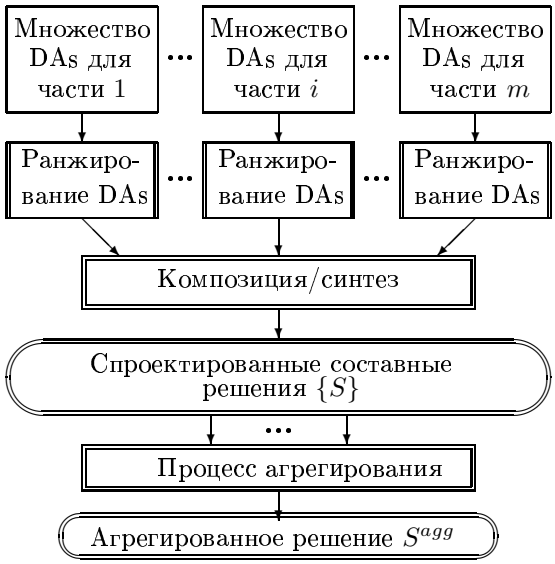


Рис. 16.9. Выбор, композиция, агрегация

Рассматривается следующая иерархическая структура радио сенсора (Рис. 16.10):

0. Радио сенсор $S = H \star W$.

1. Электроника $H = M \star U \star Z$.

1.1. Микроэлектронные компоненты $M = R \star P \star D \star Q$.

1.1.1. Радио R:

Chipcon CC2420 Radio $R_1(3)$, Chipcon CC1000 Radio $R_2(4)$, Semtech XE1205 Radio $R_3(2)$, Infineon TDA5250 Radio $R_4(1)$.

1.1.2. Микропроцессор P:

Atmel ATmega128 with 10-bit ADC $P_1(3)$, Atmel AVR AT90S2313 $P_2(1)$, Texas Instruments MSP430F16 with 12-bit ADC/DAC $P_3(2)$.

1.1.3. DAC/ADC D:

Atmel ATmega128L embedded 10-bit ADC $D_1(2)$, Texas Instruments MSP430F16 embedded 12-bit ADC/DAC $D_2(1)$, Analog Devices 14-bit AD679 $D_3(3)$.

1.1.4. Память Q:

No external memory $Q_1(4)$, 4 Kb EEPROM $Q_2(3)$, 128 Kb Flash $Q_3(2)$, 1 Mb Flash $Q_4(1)$.

1.2. Энергетическое устройство U:

2800 mAh NiMh Battery $U_1(1)$, 1500 mAh Li-Ion Battery $U_2(2)$.

1.3. Сенсор Z:

Edwards 284b-pl Heat Detector $Z_1(1)$, 123 Security Systems Photoelectric 2-Wire Smoke $Z_2(2)$, Multisensing Fire Detector $Z_3(3)$.

2. Программы $W = Y \star O$.

2.1. Сенсорные программы Y:

Zigbee/802.15.4 & Application $Y_1(3)$, TinyOS BMAC & Application $Y_2(1)$, Ad-Hoc software & Application $Y_3(2)$.

2.2. Операционная система O:

без OS, простая среда запуска $O_1(1)$, TinyOS $O_2(2)$.

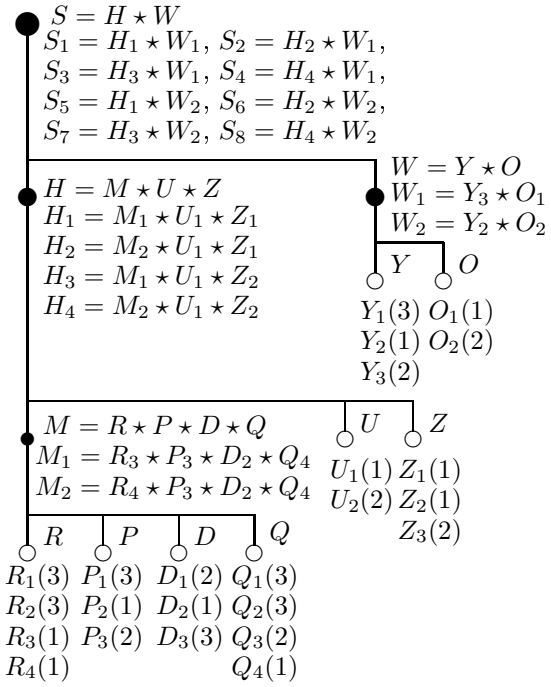


Рис. 16.10. Структура радио сенсора

При оценивании DAs используются следующие критерии (веса важности критериев указаны в скобках, '−' соответствуют случаю, когда минимальное значение является наилучшим): стоимость C_1 (-100), радиус C_2 (1), энергетические требования C_3 (-80), скорость/частота C_4 (1), достоверность C_5 (10), объем памяти C_6 (0.5), длительность функционирования C_7 (1000). Оценки DAs по критериям представлены в Таблице 16.5 (экспертное оценивание).

Таблица 16.5. Оценки DAs по критериям

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	Приоритеты
R_1	13	80	25	250				3
R_2	11	160	29	76				3
R_3	6	600	25	76				1
R_4	8	200	17	64				1
P_1	8		8	16				3
P_2	2.5		5	10				1
P_3	11		2	12				2
D_1	0		2	150	10			2
D_2	0		1	200	12			1
D_3	4		4	250	14			3
Q_1	0		0	0		0		3
Q_2	1		2	3		1024		3
Q_3	3		3	2		131072		2
Q_4	3		3	2		1048576		1
U_1	3					2800		1
U_2	10					1500		2
Z_1	10				2			1
Z_2	25				5			1
Z_3	50				16			3
Y_1	100					15000	5	3
Y_2	50					6000	6	1
Y_3	100					4000	11	2
O_1	0					2000	4	1
O_2	0					4500	0	2

Результирующие приоритеты DAs представлены на Рис. 16.10 (в круглых скобках) и в Таблице 16.5. Данные приоритеты получены в результате применения многокритериального ранжирования (модификация метода порогов несравнимости ELECTRE).

Таблица 16.6 содержит оценки совместимости между DAs. Используемые оценки, в основном, имеют иллюстративный характер. Для компонентов M, U и S рассматриваются одинаковые оценки совместимости между локальными DAs.

Таблица 16.6. Оценки совместимости

	P_1	P_2	P_3	D_1	D_2	D_3	Q_1	Q_2	Q_3	Q_4
R_1	3	3	3	3	3	3	3	3	3	3
R_2	3	3	3	3	3	3	3	3	3	3
R_3	3	3	3	3	3	3	3	3	3	3
R_4	3	3	3	3	3	3	3	3	3	3
P_1				3	0	1	3	3	3	3
P_2				0	0	1	3	3	3	3
P_3				0	3	1	3	3	3	3
D_1							3	3	3	3
D_2							3	3	3	3
D_3							3	3	3	3

	O_1	O_2
Y_1	1	2
Y_2	0	3
Y_3	3	2

Комбинаторный синтез для подсистем сенсора приводит к следующим результатам. Получаются Парето-эффективные составные DAs для подсистем:

(а) $W_1 = Y_3 \star O_1, \;\; N(W_1) = (3; 1, 1, 0)$;

(б) $W_2 = Y_2 \star O_2, \;\; N(W_2) = (3; 1, 1, 0)$;

(в) $M_1 = R_3 \star P_3 \star D_2 \star Q_4, \;\; N(M_1) = (3; 3, 1, 0)$.

(г) $M_2 = R_4 \star P_3 \star D_2 \star Q_4, \;\; N(M_1) = (3; 3, 1, 0)$.

Решения M_1 и M_2 иллюстрируются на Рис. 16.11 и на Рис. 16.12.

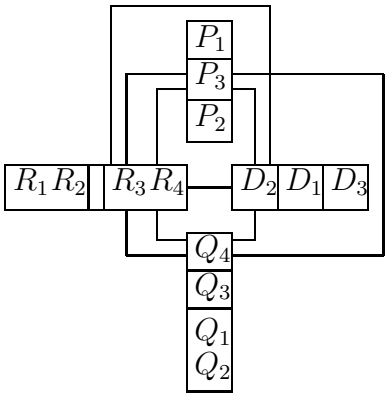


Рис. 16.11. Концентрическое представление

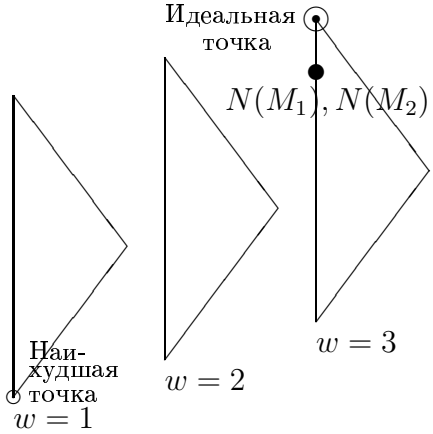


Рис. 16.12. Решетка качества

Для подсистемы H решения имеют вид:

$H_1 = M_1 \star U_1 \star Z_1 = (R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)$;

$H_2 = M_2 \star U_1 \star Z_2 = (R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_2)$;

$H_3 = M_1 \star U_1 \star Z_1 = (R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)$;

$H_4 = M_1 \star U_1 \star Z_2 = (R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_2)$.

Окончательно, получаются восемь системных решений:

$S_1 = H_1 \star W_1 = (M_1 \star U_1 \star Z_1) \star (Y_3 \star O_1) = ((R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_3 \star O_1)$;

$S_2 = H_2 \star W_1 = (M_2 \star U_1 \star Z_1) \star (Y_3 \star O_1) = ((R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_3 \star O_1)$;

$S_3 = H_3 \star W_1 = (M_1 \star U_1 \star Z_2) \star (Y_3 \star O_1) = ((R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_3 \star O_1)$;

$S_4 = H_4 \star W_1 = (M_2 \star U_1 \star Z_2) \star (Y_3 \star O_1) = ((R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_3 \star O_1)$;

$S_5 = H_1 \star W_2 = (M_1 \star U_1 \star Z_1) \star (Y_2 \star O_2) = ((R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_2 \star O_2)$;

$S_6 = H_2 \star W_2 = (M_2 \star U_1 \star Z_1) \star (Y_2 \star O_2) = ((R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_1)) \star (Y_2 \star O_2)$;

$S_7 = H_3 \star W_2 = (M_1 \star U_1 \star Z_2) \star (Y_2 \star O_2) = ((R_3 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_2)) \star (Y_2 \star O_2)$;

$S_8 = H_4 \star W_2 = (M_2 \star U_1 \star Z_2) \star (Y_2 \star O_2) = ((R_4 \star P_3 \star D_2 \star Q_4) \star (U_1 \star Z_2)) \star (Y_2 \star O_2)$.

Следует отметить, что исходное комбинаторное множество решений включает 5184 $(4 \times 3 \times 3 \times 4 \times 2 \times 3 \times 3 \times 2)$ возможных составных решений.

В рассматриваемом примере, имеется восемь решений: S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8. Подструктура этих решений представлена на Рис. 16.13. Эта подструктура используется как “ядро” для последующего расширения. Надструктура представлена на Рис. 16.14.

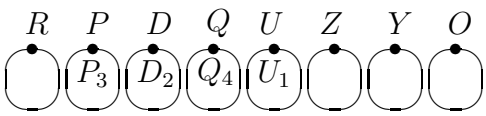


Рис. 16.13. Подструктура (“ядро”)

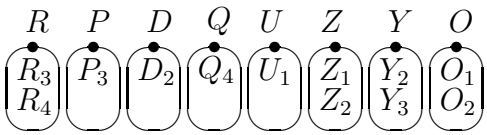


Рис. 16.14. Надструктура решений

Процедура агрегации реализована в виде стратегии расширения на основе задачи блочного рюкзака. Таблица 4 содержит альтернативы (для стратегии расширения) (DAs) и их оценки (порядковая шкала, экспертное оценивание). Рассмотренные DAs соответствуют надрешению (Рис. 16.14).

Таблица 16.7. Проектные альтернативы

κ	DAs	Бинарная переменная	Стоимость a_{ij}	Польза c_{ij}
1	R_3	x_{11}	2	3
2	R_4	x_{12}	3	4
3	Z_1	x_{21}	4	3
4	Z_2	x_{22}	6	3
5	Y_2	x_{31}	7	3
6	Y_3	x_{32}	8	2
7	O_1	x_{41}	1	3
8	O_2	x_{42}	1	2

Предполагается, что альтернативы для расширения являются совместимыми. Получающаяся задача блочного рюкзака имеет вид:

$$\max \sum_{i=1}^{4} \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \ \sum_{i=1}^{4} \sum_{j=1}^{q_i} a_{ij} x_{ij} \leq b, \ \ \sum_{j=1}^{q_i} x_{ij} = 1 \ \ \forall i = \overline{1,4}, \ \ x_{ij} \in \{0,1\}.$$

Очевидно, $q_1 = 2$, $q_2 = 2$, $q_3 = 2$, $q_4 = 2$. Для решения используется простейший жадный алгоритм на основе упорядочения альтернатив по c_i/a_i). Можно привести следующие примеры получаемых решений:

(1) $b^1 = 14$: $(x_{11} = 1, x_{21} = 1, x_{31} = 1, x_{41} = 1)$, $S^{agg}_{b^1} = R_3 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$;

(2) $b^2 = 15$: $(x_{12} = 1, x_{21} = 1, x_{31} = 1, x_{41} = 1)$, $S^{agg}_{b^2} = R_4 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$.

16.3 Выводы к главе

В данной главе были представлены два примера комбинаторных схем в области телеметрии и сенсорных систем: проектирование и улучшение телеметрической системы и радио сенсора. Следует отметить следующие возможные перспективные направления исследований:

1. исследование задач динамического проектирования и улучшения систем в телеметрии,

2. исследование комбинаторной эволюции телеметрических систем и их компонентов,

3. анализ и проектирование телеметрической системы, которая включает несколько бортовых аппаратов и/или несколько наземных пунктов,

4. исследование и решение задач проектирования на уровне транспортирования сенсорных систем (например, формирование расписаний),

5. использование схем выявления узких мест в модульных системах на уровне сбора и обработки информации сенсорных систем.

Глава 17

Приложения в менеджменте

В данной главе [1] приведены описания нескольких комбинаторных схем в области организационного управления, маркетинга, планирования исследований, экономики, электронной торговли.

17.1 Планирование исследования в геологии

В данном разделе описывается комбинаторное планирование процесса геологического изучения и промышленного освоения месторождений нефти, газа и конденсата [30, 379] на примере нескольких месторождений полуострова Ямал [34] При оценивании параметров используются порядковые шкалы, а также интервальные оценки в виде мультимножеств.

17.1.1 Исследование месторождений

Общий план исследований включает пять частей (Рис. 17.1):

$$S = A^1 \star A^2 \star A^3 \star A^4 \star A^5,$$

где A^1 соответствует плану для месторождения Харосовей, A^2 - для месторождения Арктическое, A^3 - для месторождения Нейтинское, A^4 - для месторождения Крузенштернское, A^5 - для месторождения Бованенковское.

Схема решения включает два этапа:

1. Иерархическое комбинаторное формирование плана исследований для месторождений (с целью сокращения объема здесь приведены данные по двум месторождениям: Арктическое и Крузенштернское).

2. Композиция общего плана исследования для региона.

[1] Материал главы основан на публикациях:
(i) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Springer, 1998, chapter 11.
(ii) Левин М.Ш., Комбинированная схема формирования стратегии маркетинга. Бизнес информатика, 2009, вып. 2, С. 42-51.
(iii) Levin M.Sh., Towards electronic shopping of composite product. Electronic preprint. 10 pp., March 3, 2012. http://arxiv.org/abs/1203.0648 [cs.SE]
(iv) Levin M.Sh., Morphological approach to electronic shopping. IEEE Region 8 Int. Conf. "Sibircon-2008", Novosibirsk, pp. 280-285, 2008.
(v) Левин М.Ш., Фимин А.В. Комбинаторная схема для анализа политических кандидатов и их стратегий. Электронный научн. ж. "Информационные процессы", 2009, вып. 9, ном. 2, С. 83-92.
(vi) Левин М.Ш., Пороскун В.И. Комбинаторное планирование исследований нефти и газа. В: "Геологические исследования и применения", 1997, вып. 4, М.: Институт минеральных ресурсов РАН, С. 49-55.
(vii) Levin M.Sh. Combinatorial framework for planning of geological exploration. Electr. prepr., 14 pp., Jan. 22, 2018, http://arxiv.org/abs/1801.07229 [cs.AI]

Рассматривается планирование действий (мероприятий, акций, операций) по геологическому изучению и промышленному освоению месторождений нефти и газа. Предполагается 4-уровневая иерархия геологических объектов: (a) продуктивный пласт, залежь (нижний уровень); (b) группа продуктивных пластов, залежей; (c) месторождение нефти/газа; (d) нефтегазоносный район (группа месторождений).

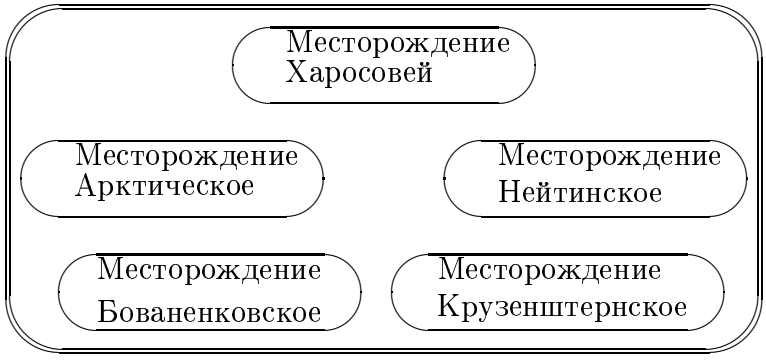


Рис. 17.1. Месторождения на полуострове Ямале (регион)

Характеристики для оценивания указанного геологического объекта следующие:

1. признак наличия залежи ('3' соответствует наличию залежи, '2' соответствует перспективному горизонту в пределах месторождения, '1' соответствует перспективному горизонту в пределах ловушки);

2. глубина залегания, м;

3. тип флюида (классификационный фактор: (i) газ, (ii) газ и конденсат, (iii) нефть);

4. величина запасов (газ - млн. куб. м, нефть - тыс. т.);

5. средний дебит работающих скважин (куб. м в сутки);

6. сложность горно-геологических условий (1 - простые, 2 - сложные, 3 - очень сложные);

7. надежность (риск) получения ожидаемого результата ([0...100]);

8. достоверность оценки запасов (ресурсов) (C_1 - 20%, C_2 - 50%, C_3 - 80%, и т.д.);

9. удаленность от технологической базы (газо-нефтепровода) (км).

Для каждого геологического объекта рассматривается восемь альтернативных исследовательских (проектных) действий (вариантов - DAs), представленных в Таблице 17.1.

Таблица 17.1. Типы проектных вариантов геологического исследования

Обозначение	Содержание геологического исследования
X_1	Консервация работ
X_2	Доразведка залежи (дополнительное исследование)
X_3	Независимая эксплуатация геологического объекта (газ)
X_4	Независимая эксплуатация геологического объекта (нефть)
X_5	Независимая эксплуатация геологического объекта (нефть, газ)
X_6	Совместная эксплуатация геологического объекта (газ)
X_7	Совместная эксплуатация геологического объекта (нефть)
X_8	Совместная эксплуатация геологического объекта (нефть, газ)

Далее для каждого геологического объекта выбирается некоторое подмножество альтернативных действий из указанного списка (с соответствующим индексом) на основе экспертного оценивания. Это предварительный выбор альтернатив исследования на нижнем уровне. Отобранные локальные DAs являются базой для комбинирования составных акций исследования (составных DAs) для групп геологических объектов и, далее, для месторождений.

Каждый компонент стратегии (геологической объект, группа объектов, стратегия) обозначается буквой (уровень эффективности или приоритет ι указан для каждого компонента в скобках). Процесс решения для каждой задачи композиции основывается на особенностях ситуации принятия решений, включая следующее: (а) эксперт(ы), знания и опыт эксперта(ов), наличие оборудования; (б) наличие и объем исходной информации об объекте; (в) ресурсы для решения задач (например, время, компьютеры, коммуникации). Предполагается, что эксперт готов к экспертным суждениям следующих типов: (1) выбор DA для каждого геологического объекта, (2) ранжирование DAs для каждого геологического объекта, (3) оценивание совместимости между DAs (по некой порядковой шкале).

Исследовательская стратегия для месторождения нефти и газа Арктическое представлена на Рис. 17.2. Таблица 17.2 содержит факторы для задания совместимости между элементами стратегии (здесь и далее C5+в - содержание в газе углеводородов C5 и выше). Оценки совместимости между DAs (экспертное оценивание) представлены в Таблице 17.3 и Таблице 17.4. Составные DAs для групп объектов ТП24-НП3 (D), ТП14-ТП13 (W) приведены в Таблице 17.5.

Полученные 6 вариантов исследовательской стратегии для месторождения Арктическое имеют вид:

$$A_1^2 = W_1 \star D_1 \star B_3(1),\ A_2^2 = W_2 \star D_1 \star B_3(1),\ A_3^2 = W_1 \star D_2 \star B_3(1),$$

$$A_4^2 = W_2 \star D_2 \star B_3(1),\ A_5^2 = W_3 \star D_1 \star B_3(1),\ A_6^2 = W_3 \star D_2 \star B_3(1).$$

Таблица 17.6 содержит примеры узких мест и возможные акции улучшения. Рис. 17.3 иллюстрирует качество составных решения для компонента W.

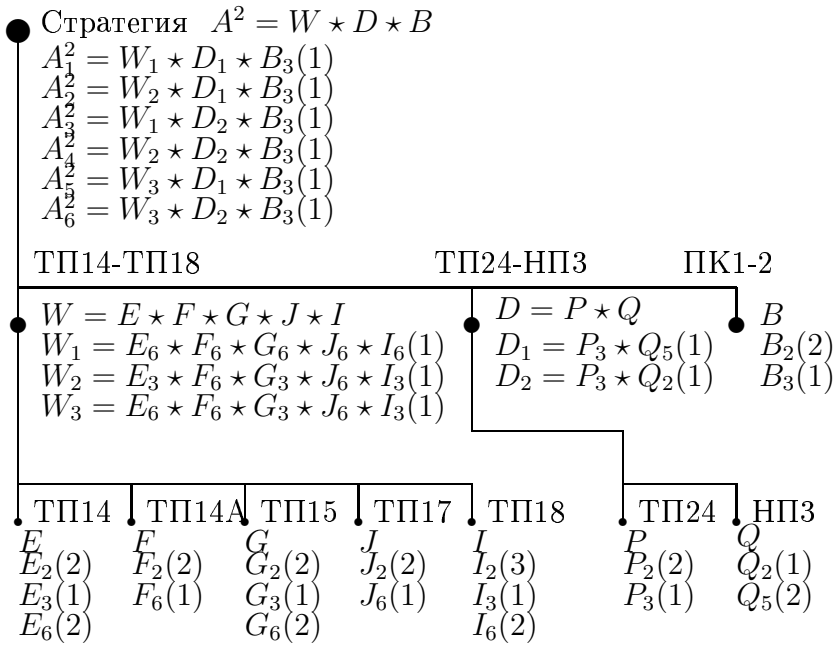


Рис. 17.2. Стратегия для месторождения Арктическое

Таблица 17.2. Факторы совместимости

DAs	Факторы
1.ТП14 E & ТП14A F	Запасы, близость, C5+в
2.ТП14 E & ТП15 G	Запасы, близость, C5+в
3.ТП14 E & ТП17 J	Запасы, близость
4.ТП14 E & ТП18 I	Запасы, близость
5.ТП14A F & ТП15 G	Запасы, близость
6.ТП14A F & ТП17 J	Запасы, близость
7.ТП14A F & ТП18 I	Близость
8.ТП15 G & ТП17 J	Запасы, близость, C5+в
9.ТП15 G & ТП18 I	Запасы, близость, C5+в
10.ТП17 J & ТП18 I	Запасы, близость, C5+в
11.ТП24 P & НП3 Q	Близость, C5+в

Таблица 17.3. Совместимость между DAs для групп ТП14-ТП18

	F_2	F_6	G_2	G_3	G_6	J_2	J_6	I_2	I_3	I_6
E_2	2	3	2	3	4	1	0	2	3	3
E_3	3	4	2	4	4	1	2	3	2	2
E_6	3	4	3	4	4	1	4	3	3	4
F_2			2	3	4	2	3	2	3	2
F_6			1	3	4	3	3	2	3	1
G_2						2	3	2	1	3
G_3						2	4	2	3	1
G_6						2	4	2	3	1
J_2								2	1	1
J_6								1	3	4

Таблица 17.4. Совместимость между DAs для групп ТП24-НП3

	Q_2	Q_5
P_2	2	3
P_3	3	4

Таблица 17.5. Составные DAs

Составные DAs	N
$D_1 = P_3 \star Q_5$	$4; 1, 1, 0$
$D_2 = P_3 \star Q_2$	$3; 2, 0, 0$
$W_1 = E_6 \star F_6 \star G_6 \star J_6 \star I_6$	$4; 2, 3, 0$
$W_2 = E_3 \star F_6 \star G_3 \star J_6 \star I_3$	$2; 5, 0, 0$
$W_3 = E_6 \star F_6 \star G_3 \star J_6 \star I_3$	$3; 4, 1, 0$

Таблица 17.6. Узкие места и акции улучшения

Составные DAs	Узкие места		Акция
	DAs	IC	w/r
$D_1 = P_3 \star Q_5$	Q_5	(P_3, Q_2)	$2 \Rightarrow 1$
$D_2 = P_3 \star Q_2$			$3 \Rightarrow 4$
$W_1 = E_6 \star F_6 \star G_6 \star J_6 \star I_6$	E_6		$2 \Rightarrow 1$
$W_1 = E_6 \star F_6 \star G_6 \star J_6 \star I_6$	G_6		$2 \Rightarrow 1$
$W_1 = E_6 \star F_6 \star G_6 \star J_6 \star I_6$	I_6		$2 \Rightarrow 1$
$W_3 = E_6 \star F_6 \star G_3 \star J_6 \star I_3$	E_6		$2 \Rightarrow 1$

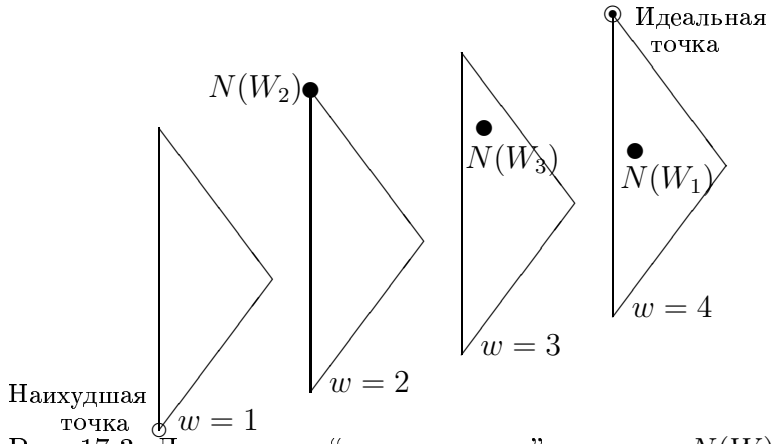


Рис. 17.3. Дискретное “пространство” качества $N(W)$

Исследовательская стратегия для месторождения Крузенштернское представлена на Рис. 17.4. Таблица 17.7 содержит факторы для задания совместимости между элементами

стратегии. Оценки совместимости между DAs (экспертное оценивание) представлены в Таблицах 17.8, 17.9. Составные DAs для B и H представлены в Таблице 17.10. Полученные 2 варианта исследовательской стратегии для месторождения Крузенштернское имеют вид: $A_1^4 = B_1 \star H_1$, $A_2^4 = B_2 \star H_2$. Рис. 17.5 иллюстрирует качество составных решения для компонента H.

Таким образом, получены следующие стратегии исследования для месторождений (Рис. 17.6):

0. Общая составная стратегия: $S = A^1 \star A^2 \star A^3 \star A^4 \star A^5$

1. Стратегия для месторождения Харосовей: A_1^1.

2. Стратегии для месторождения Арктическое: A_1^2, A_2^2, A_3^2, A_4^2, A_5^2, A_6^2.

3. Стратегия для месторождения Нейтинское: A_1^3.

4. Стратегии для месторождения Крузенштернское: A_1^4, A_2^4.

5. Стратегии для месторождения Бованенковское: A_1^5, A_2^5.

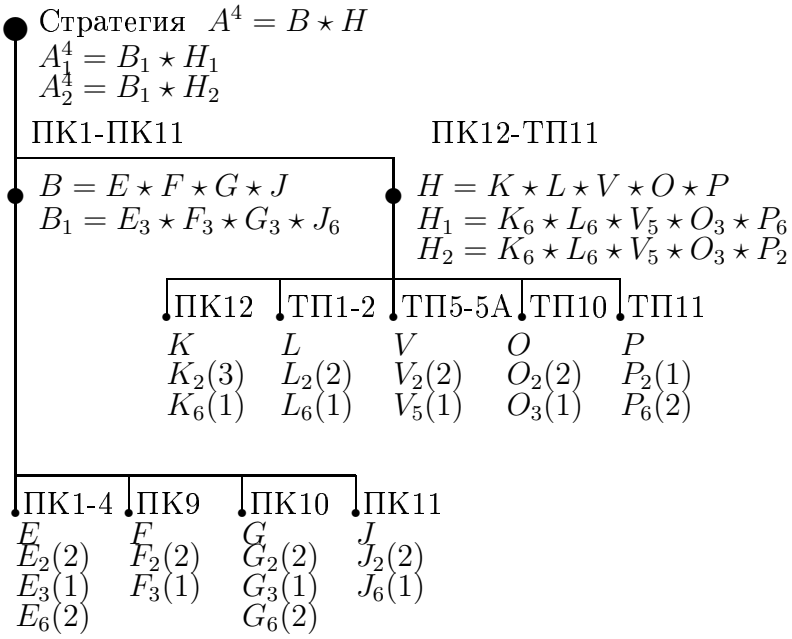


Рис. 17.4. Стратегия для месторождения Крузенштернское

Таблица 17.7. Факторы совместимости

DAs	Факторы
1.ПК1-4 E & ПК9 F	Запасы, близость
2.ПК1-4 E & ПК10 G	Запасы, близость
3.ПК1-4 E& ПК11 J	Запасы, близость
4.ПК9 F & ПК10 G	Запасы, близость
5.ПК9 F & ПК11 J	Запасы, близость
6.ПК10 G & ПК11 J	Запасы, близость
7.ПК12 K & ТП1-2 L	Запасы, близость
8.ПК12 K & ТП5-5А V	Запасы, близость
9.ПК12 K & ТП10 O	Запасы, близость
10.ПК12 K & ТП11 P	Близость, C5+в
11.ТП1-2 L & ТП5-5А V	Запасы, близость, C5+в
12.ТП1-2 L & ТП10 O	Запасы, близость, C5+в
13.ТП1-2 L & ТП11 P	Запасы, близость, C5+в
14.ТП5-5А V & ТП10 O	Запасы, близость, C5+в
15.ТП5-5А V & ТП11 P	Запасы, близость, C5+в
16.ТП10 O & ТП11 P	Запасы, близость, C5+в

Таблица 17.8. Совместимость между DAs для групп ПК1-ПК11

	F_2	F_3	G_2	G_3	G_6	J_2	J_3	J_6
E_2	2	1	2	1	2	2	1	2
E_3	4	3	4	3	1	4	3	1
E_6	1	4	1	4	2	1	3	4
F_2			3	4	2	3	4	2
F_3			3	4	4	3	4	4
G_2						3	4	4
G_3						4	4	4
G_6						3	4	4

Таблица 17.9. Совместимость между DAs для групп ПК12-ТП11

	L_2	L_6	V_2	V_5	O_2	O_3	P_2	P_6
K_2	4	3	2	4	3	1	4	3
K_6	1	4	3	4	3	4	3	4
L_2			2	3	4	2	3	4
L_6			2	4	4	4	3	4
V_2					4	4	2	3
V_5					3	4	3	4
O_2							4	4
O_3							3	4

Таблица 17.10. Составные DAs

Составные DAs	N
$B_1 = E_3 \star F_3 \star G_3 \star J_3$	$3; 4, 0, 0$
$H_1 = K_6 \star L_6 \star V_5 \star O_3 \star P_6$	$4; 4, 1, 0$
$H_2 = K_6 \star L_6 \star V_5 \star O_3 \star P_2$	$3; 5, 0, 0$

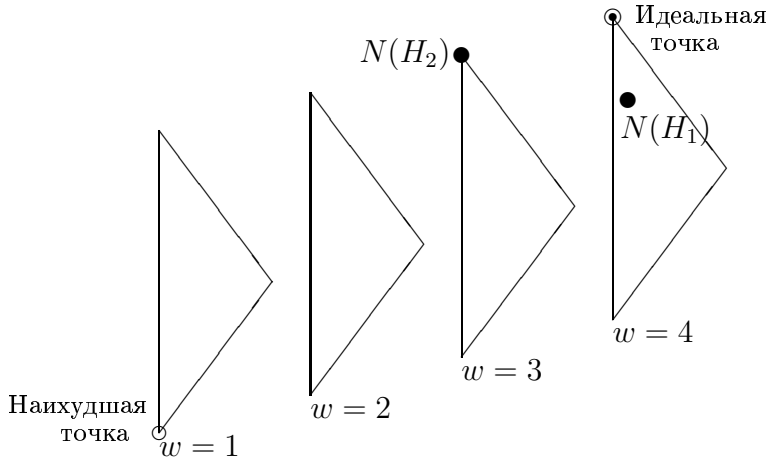


Рис. 17.5. Дискретное “пространство” качества $N(H)$

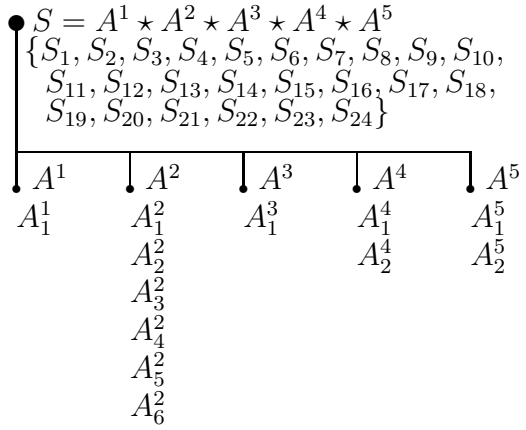


Рис. 17.6. Составная стратегия для региона

В результате, получается 24 составных стратегий исследования для рассматриваемого региона (без анализа совместимости):

$S_1 = A_1^1 \star A_1^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_2 = A_1^1 \star A_1^2 \star A_1^3 \star A_2^4 \star A_1^5$,
$S_3 = A_1^1 \star A_1^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_4 = A_1^1 \star A_1^2 \star A_2^3 \star A_2^4 \star A_1^5$,
$S_5 = A_1^1 \star A_2^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_6 = A_1^1 \star A_2^2 \star A_1^3 \star A_2^4 \star A_1^5$,

$S_7 = A_1^1 \star A_2^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_8 = A_1^1 \star A_2^2 \star A_2^3 \star A_2^4 \star A_1^5$,
$S_9 = A_1^1 \star A_3^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_{10} = A_1^1 \star A_3^2 \star A_1^3 \star A_2^4 \star A_1^5$,
$S_{11} = A_1^1 \star A_3^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_{12} = A_1^1 \star A_3^2 \star A_2^3 \star A_2^4 \star A_1^5$,
$S_{13} = A_1^1 \star A_4^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_{14} = A_1^1 \star A_4^2 \star A_1^3 \star A_2^4 \star A_1^5$,
$S_{15} = A_1^1 \star A_4^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_{16} = A_1^1 \star A_4^2 \star A_2^3 \star A_2^4 \star A_1^5$,
$S_{17} = A_1^1 \star A_5^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_{18} = A_1^1 \star A_5^2 \star A_1^3 \star A_2^4 \star A_1^5$,
$S_{19} = A_1^1 \star A_5^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_{20} = A_1^1 \star A_5^2 \star A_2^3 \star A_2^4 \star A_1^5$,
$S_{21} = A_1^1 \star A_6^2 \star A_1^3 \star A_1^4 \star A_1^5$, $S_{22} = A_1^1 \star A_6^2 \star A_1^3 \star A_2^4 \star A_1^5$,
$S_{23} = A_1^1 \star A_6^2 \star A_2^3 \star A_1^4 \star A_1^5$, $S_{24} = A_1^1 \star A_6^2 \star A_2^3 \star A_2^4 \star A_1^5$.

Теперь можно провести дополнительный анализ полученных стратегий для окончательного выбора наилучшей стратегии (например, многокритериальное ранжирование, экспертное оценивание). С другой стороны, возможно агрегирование полученных решений для формирования результирующей агрегированнной региональной стратегии.

17.1.2 Агрегирование решений

В рассматриваемом примере имеется 24 решения из предыдущего раздела: $S_1, \ldots, S_{24}$. Подструктура этих решений представлена на Рис 17.7. Эта структура далее используется как “ядро” для дальнейшего расширения. На Рис. 17.8 представлена надструктура.

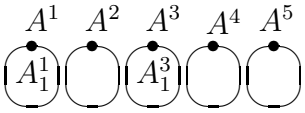


Рис. 17.7. Подструктура (“ядро”)

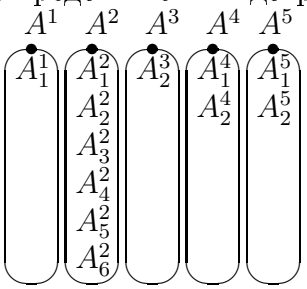


Рис. 17.8. Надструктура

Таблица 17.11 содержит проектные варианты для расширения “ядра”, включая их оценки (использованы порядковые шкалы, экспертные оценки).

Таблица 17.11. Варианты расширения

κ	Варианты	Двоичная переменная	Цена a_{ij}	Польза c_{ij}
1	A_1^2	x_{11}	4	4
2	A_2^2	x_{12}	6	6
3	A_3^2	x_{13}	3	2
4	A_4^2	x_{14}	3	3
5	A_5^2	x_{15}	4	3
6	A_6^2	x_{16}	5	3
7	A_1^4	x_{21}	3	4
8	A_2^4	x_{22}	3	3
9	A_1^5	x_{31}	3	3
10	A_2^5	x_{32}	4	4

Предполагается, что проектные варианты являются совместимыми. Задача агрегирования (стратегия расширения) основана на модели блочного рюкзака:

$$\max \sum_{i=1}^{3} \sum_{j=1}^{q_i} c_{ij} x_{ij} \quad s.t. \ \sum_{i=1}^{3} \sum_{j=1}^{q_i} a_{ij} x_{ij} \le b, \ \sum_{j=1}^{q_i} x_{ij} = 1 \ \forall i = \overline{1,3}, \ x_{ij} \in \{0,1\}.$$

В модели $q_1 = 6$, $q_2 = 2$, $q_3 = 2$. При использовании простейшего жадного алгоритма (на основе упорядочения элементов по c_i/a_i) получаются следующие решения (рассмотрены четыре варианта ограничения):

(1) $b^1 = 9$: $(x_{14} = 10,\ x_{21} = 1,\ x_{31} = 1)$,

$S^{agg}_{b^1} = A^1_1 \star A^2_4 \star A^3_1 \star A^4_1 \star A^5_1 = R_3 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$;

(2.1.) $b^2 = 10$: $(x_{14} = 1,\ x_{21} = 1,\ x_{32} = 1)$,

$S^{agg1}_{b^2} = A^1_1 \star A^2_4 \star A^3_1 \star A^4_1 \star A^5_2 = R_3 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$;

(2.2.) $b^2 = 10$: $(x_{11} = 1,\ x_{21} = 1,\ x_{31} = 1)$,

$S^{agg2}_{b^2} = A^1_1 \star A^2_1 \star A^3_1 \star A^4_1 \star A^5_1 = R_3 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$;

(3) $b^3 = 11$: $(x_{11} = 1,\ x_{21} = 1,\ x_{32} = 1)$,

$S^{agg}_{b^3} = A^1_1 \star A^2_1 \star A^3_1 \star A^4_1 \star A^5_2 = R_4 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$;

(4) $b^4 = 11$: $(x_{12} = 1,\ x_{22} = 1,\ x_{31} = 1)$,

$S^{agg}_{b^4} = A^1_1 \star A^2_2 \star A^3_1 \star A^4_2 \star A^5_1 = R_4 \star P_3 \star D_2 \star Q_4 \star U_1 \star Z_1 \star Y_2 \star O_1$.

17.1.3 Использование интервальных оценок

Шкала интервальных оценок (в виде частично упорядоченного множества) для используемой задачи оценивания $P^{3,4}$ представлена на Рис. 17.9. Часть месторождения Арктическое (Рис. 17.10) используется для примера использования интервальных оценок.

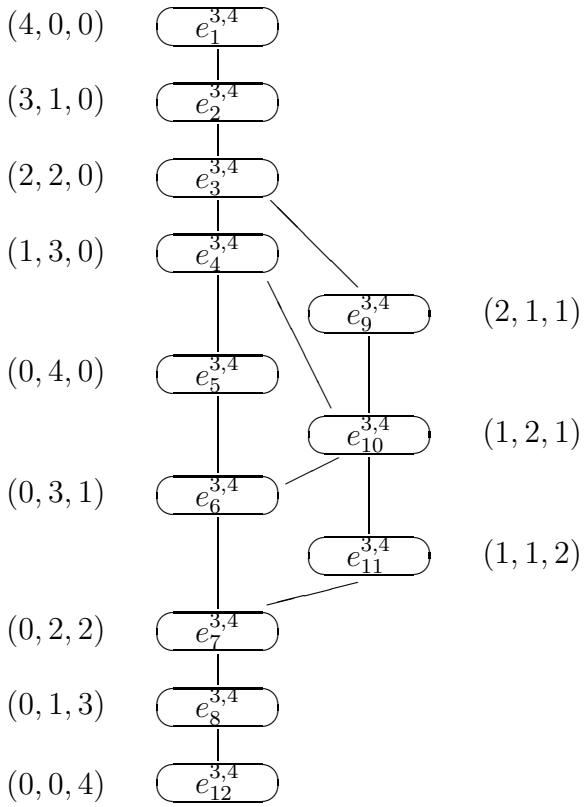


Рис. 17.9. Оценки для задачи оценивания $P^{3,4}$

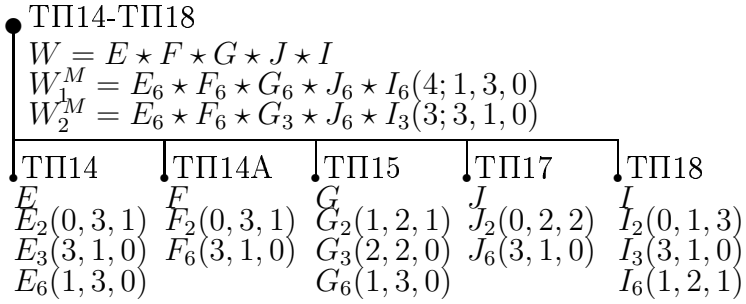


Рис. 17.10. Месторождение Арктическое (интервальные оценки)

Интервальные оценки для локальных DAs приведены на Рис. 17.10 (в скобках). Используются оценки совместимости DAs из Таблицы 17.3. Пусть имеются два решения:

$W_1^M = E_6 \star F_6 \star G_6 \star J_6 \star I_6, \quad N(W_1^M) = (w(W_1^M); e(W_1^M)) = (4;1,3,0);$

$W_2^M = E_6 \star F_6 \star G_3 \star J_6 \star I_3, \quad N(W_2^M) = (w(W_2^M); e(W_2^M)) = (3;3,1,0).$

Оценки $e(W_1^M) = (1,3,0)$, $e(W_2^M) = (3,1,0)$ являются медианами по оценкам соответствующих компонентов.

17.2 Стратегии политического маркетинга

17.2.1 Предварительные сведения

Методы маркетинга стали применять в области политики давно [306, 422, 513]. В последние примерно 15 лет сформировалось направление: маркетинг в политике (political marketing) [69, 250, 324, 449, 463, 503, 504], где широко используются методы маркетинга для анализа политических "объектов" и ситуаций и построения стратегий проведения политических акций и кампаний. Очевидно, что при этом применяются традиционные подходы из области маркетинга, например: задачи сегментации рынков, построение стратегий. Здесь предлагается составная много-стадийная комбинаторная схема многокритериального анализа политических кандидатов и построения стратегий проведения их политических акций. Четыре комбинаторные задачи используются как вспомогательные (базовые "модули"): (а) многокритериальное ранжирование, (б) кластеризация (группировка), (в) задача о назначении, (г) блочная задача о рюкзаке. Схема включает следующие стадии: (1) кластеризацию (т.е. построение типовых групп объектов, например, сегментов электората, групп типовых кандидатов); (2) определение соответствия кандидатов группам избирателей (на основе задачи о назначении), в результате для каждого кандидата формируется набор соответствующих ему групп избирателей, кроме того на этой стадии может быть выявлено отсутствие соответствующих групп избирателей для некоторых кандидатов; (3) решение задачи блочного рюкзака для формирования комплексной стратегии для каждого кандидата с учетом его ресурсных ограничений; здесь предполагается исходный набор из четырех "элементарных" действий политического кандидата (организация встреч с избирателями, подготовка рекламных роликов для телевидения, организация общественных диспутов, корректировка политической программы) и для каждого сегмента электората выбирается подмножество исходного набора действий с учетом общего ресурсного ограничения. Задача многокритериального ранжирования носит вспомогательный характер и может быть использована в рамках применения многокритериальной версии задачи о назначении, многокритериальной версии задачи блочного рюкзака, а также для определения предпочтений групп избирателей (при использовании ранжировок и/или частичных порядков). Данная составная схема является типовой в рамках 4-х уровневой архитектуры для задач комбинаторной оптимизации, предложенной в [355]. Численный

пример реализует один из возможных вариантов расчета и иллюстрирует предложенный подход. Расчеты проводились на основе программ в среде MATLAB (MatWorks, Inc.. http://www.matworks.com).

17.2.2 Составная схема решения

В данной работе рассматриваются два базовых исходных множества: (1) *элементы* (т.е. политические кандидаты) и (2) *позиции* (т.е. группы типовых избирателей). Используется комбинаторная схема, включающая четыре стадии (Рис. 17.11):

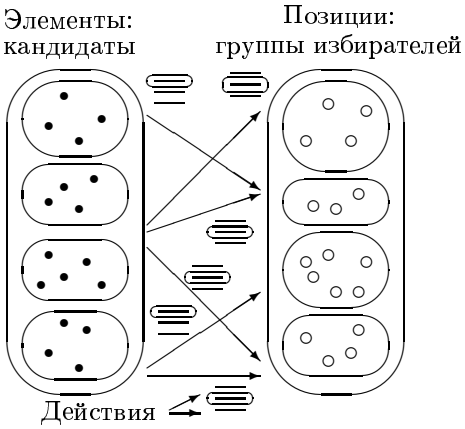


Рис. 17.11. Комбинаторная схема

1. кластеризация множества *элементов* (для уменьшения размерности задачи);

2. кластеризация множества *позиций* (для уменьшения размерности задачи);

3. назначение кластерам *элементов* кластеров *позиций*, т.е., получение пар *кластер элементов - кластер позиций*;

4. выбор для каждой полученной пары подмножества операций из некоторого заданного множества операций (действий) (*задача блочного рюкзака* или *многокритериальная задача блочного рюкзака*).

17.2.3 Пример

В рамках численного примера предлагается иллюстративный расчет на основе условных политических кандидатов, включая стадии (Рис. 17.12): (1) группировка политических кандидатов для получения типовых политических кандидатов (задача кластеризации); (2) определение соответствия типовых кандидатов группам избирателей (задача о назначении); (3) формирование комплексной стратегии для типового политического кандидата, комплексная стратегия включает набор действий данного кандидата для каждого соответствующего данному кандидату сегмента электората с учетом общего ограничению по ресурсу (для данного кандидата) (задача о блочном рюкзаке). Данные являются иллюстративными.

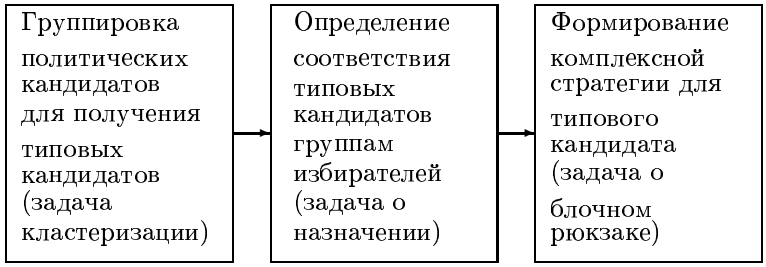


Рис. 17.12. Блок-схема расчета

Рассматриваемый пример основан на данных президентских выборов в России в 1996 году, т.е., рассматриваются 10 кандидатов: В.А. Брынцалов, Ю.П. Власов, М.С. Горбачев, Б.Н. Ельцин В.В. Жириновский, Г.А. Зюганов, А.И. Лебедь, С.Н. Федоров, М.Л. Шаккум, Г.А. Явлинский. В расчетах указанные выше политические кандидаты рассматриваются в закодированном виде и используются следующие обозначения (порядок изменен): $A = \{A_1, ..., A_i, ..., A_{10}\}$. В качестве характеристик кандидатов рассматриваются только личностные характеристики: (1) харизма (C_1), (2) политический опыт (C_2), (3) радикальность (C_3), (4) образование (C_4) и (5) репутация (C_5). Таблица 17.12 содержит оценки исследуемых альтернатив (т.е. примеров политических кандидатов) по личностным характеристикам (на основе экспертных оценок в порядковой шкале [1,2,3,4,5]). Таким образом, для каждого кандидата A_i имеется вектор оценок по критериям $\{C_1, C_2, C_3, C_4, C_5\}$: $\overline{y_i} = (y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5})$.

Таблица 17.12. Оценки альтернатив

	C_1	C_2	C_3	C_4	C_5
A_1	4	4	3	3	4
A_2	3	5	4	5	2
A_3	5	2	5	2	4
A_4	3	4	3	4	2
A_5	5	4	5	4	3
A_6	3	2	2	5	5
A_7	3	5	3	3	1
A_8	2	3	2	3	1
A_9	3	1	1	3	1
A_{10}	4	2	2	4	1

Таблица 17.13. Оценки групп политиков

	C_1	C_2	C_3	C_4	C_5
G_1	2	3	2	3	1
G_2	4.5	2	3.5	3	2.5
G_3	3.125	4.5	3.25	3.75	2
G_4	3	1.5	1.5	4	3
G_5	3.875	4.125	3.625	3.75	3

Далее проводится расчет на основе задачи кластеризации с возможным пересечением получаемых кластеров. В результате получаются 5 групп политиков (типовых политиков) ($p = \overline{1,5}$):

Группа (кластер) 1: $G_1 = \{A_8\}$ (слабый политик).

Группа (кластер) 2: $G_2 = \{A_3, A_{10}\}$ (харизматичные неопытные политики).

Группа (кластер) 3: $G_3 = \{A_1, A_2, A_4, A_7\}$ (опытные образованные политики).

Группа (кластер) 4: $G_4 = \{A_6, A_9\}$ (неопытные кандидаты).

Группа (кластер) 5: $G_5 = \{A_1, A_2, A_4, A_5\}$ (харизматичные опытные политики).

Заметим, каждая группа соответствует "типовому" политику. В Таблице 17.13 приведены средние оценки по характеристикам для каждой группы политиков, т.е. имеются векторы оценок следующего вида: $\widetilde{y_p} = (\widetilde{y_{p1}}, \widetilde{y_{p2}}, \widetilde{y_{p3}}, \widetilde{y_{p4}}, \widetilde{y_{p5}})$.

Следует заметить, что данный пример кластеризации носит иллюстративный характер. В реальных прикладных ситуациях следуют проводить кластеризацию групп избирателей для получению укрупненных сегментов электората. В примере предполагаются заданными следующие группы избирателей (сегменты электората, аналог сегментов рынка) ($q = \overline{1,5}$): (1) центристский E_1, (2) коммунистически-аграрный E_2, (3) национал-патриотический E_3, (4) политически апатичный электорат (молодежь и др.) E_4 и (5) буржуазный E_5. В таблице 17.14 представлены предпочтения указанных групп избирателей (экспертные оценки по тому же набору характеристик), т.е., для каждого кандидата E_q имеется вектор оценок по критериям $\{C_1, C_2, C_3, C_4, C_5\}$: $\overline{z_q} = (z_{q1}, z_{q2}, z_{q3}, z_{q4}, z_{q5})$.

Таблица 17.14. Предпочтения избирателей

	C_1	C_2	C_3	C_4	C_5
E_1	4	5	1	4	4
E_2	3	2	4	3	5
E_3	5	4	5	3	5
E_4	5	1	2	4	5
E_5	3	5	1	5	5

Таблица 17.15. Соответствие $G_p - E_q$

	E_1	E_2	E_3	E_4	E_5
G_1	41	34	46	34	43
G_2	53.5	53	69.5	56	54
G_3	61.25	52.625	71.125	51.625	63.875
G_4	49	45	55.5	50.5	53
G_5	66.75	60.625	80.25	60.75	69.625

Теперь можно построить матрицу соответствия, т.е. численное выражение соответствия для каждой пары "группа кандидатов - группы избирателей" $G_p - E_q$ $(p = \overline{1,5},\ q = \overline{1,5})$: $c_{p,q}$. Значение указанного соответствия для каждой пары вычисляется как сумма произведений значения характеристики кандидата (среднее значение для группы кандидатов из Таблицы 17.13) на значение предпочтения группы избирателей (из Таблицы 17.14) (т.е. величина типа скалярного произведения векторов): $c_{p,q} = \sum_{\xi=1}^{5} \widetilde{y_{p\xi}}\ z_{q\xi}$. В результате получается Таблица 17.15. Заметим, что расчет этих комплексных характеристик может проводиться и другими способами (например, на основе ранжировок, бинарных отношений предпочтения, на основе экспертных оценок). На основе решения задачи о назначении получаются базовые соответствия между "типовыми" политиками (G_p) и типовыми группами избирателей (E_q). При этом в модели предполагаются следующие ограничения:

(1) каждому типовому кандидату соответствуют не более 3-х групп избирателей,

(2) каждой группе избирателей соответствуют не более 2-х типовых кандидатов.

Таким образом, используемая версия задачи о назначении имеет вид:

$$\max \sum_{p=1}^{5}\sum_{q=1}^{5} c_{p,q}\ x_{p,q} \quad s.t. \quad \sum_{p=1}^{5} x_{p,q} \le 2\ \forall q = \overline{1,5}, \quad \sum_{q=1}^{5} x_{p,q} \le 3\ \forall p = \overline{1,5}, \quad x_{p,q} \in \{0,1\}\ \forall p,\ \forall q.$$

Результат решения задачи о назначении представлен в Таблице 17.16: (1) для кандидата G_1 множество возможных сегментов электората пусто $|\Psi(G_1)| = 0$, (2) для кандидата G_2 множество возможных сегментов электората $\Psi(G_2) = \{E_2, E_4\}$, (3) для кандидата G_3 множество возможных сегментов электората $\Psi(G_3) = \{E_1, E_3, E_5\}$, (4) для кандидата G_4 множество возможных сегментов электората $\Psi(G_4) = \{E_2, E_4\}$, (5) для кандидата G_5 множество возможных сегментов электората $\Psi(G_5) = \{E_1, E_3, E_5\}$.

Таблица 17.16. Результирущее соответствие $G_p - E_q$

	E_1	E_2	E_3	E_4	E_5
G_1	0	0	0	0	0
G_2	0	1	0	1	0
G_3	1	0	1	0	1
G_4	0	1	0	1	0
G_5	1	0	1	0	1

Теперь можно рассмотреть задачу формирования комплексной стратегии для каждого типа кандидатов. Рассмотрим типового кандидата "опытный образованный политик" G_3, для которого имеется следующий набор групп избирателей (т.е. сегментов электората): $\{E_1, E_3, E_5\}$. В качестве элементарных стратегий (действий) используется следующее множество (общее для всех типов кандидатов): (1) организация встреч с избирателями O_1, (2) подготовка рекламных роликов и показ их по телевидению O_2, (3) организация общественных диспутов O_3, (4) корректировка политической программы под конкретную группу избирателей O_4. Рис. 17.13 иллюстрирует рассмотренную ситуацию (верхний индекс в обозначении действия соответствует сегменту электората). В качестве характеристик указанных действий $(\{O_v, v = \overline{1,4}\})$ используются следующие: (i) требуемый ресурс $r_{v,3}$, (ii) полезность данного действия v кандидата p по отношению к группе избирателей q: $e_{v,p,q}$, т.е., эта характеристика индивидуальна для каждой пары (G_p, E_q).

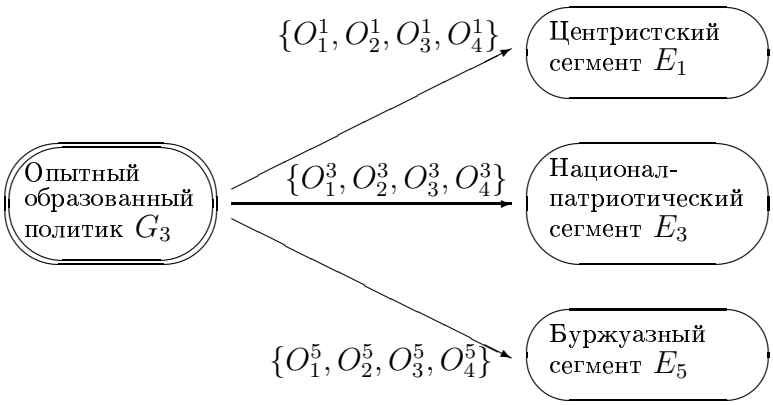


Рис. 17.13. Соответствие: кандидат - сегменты

В Таблице 17.17 приведены значения характеристик действий для типового кандидата G_3 (на основе экспертного оценивания с использованием порядковых шкал: шкала $[1, 10]$ для ресурса и шкала $[1, 5]$ для полезности; для каждого сегмента электората оценки являются индивидуальными). Таким образом, можно сформулировать задачу формирования комплексной стратегии для типового кандидата G_3 на основе модели блочного рюкзака (случай одной целевой функции):

Таблица 17.17. Действия и характеристики для G_3

	$r_{v,3}$	$e_{v,3,1}$	$e_{v,3,3}$	$e_{v,3,5}$
O_1	5	3	5	1
O_2	10	4	5	3
O_3	2	3	2	5
O_4	3	4	3	5

$$\max \sum_{v=1}^{4} \sum_{\forall q \in \{1,3,5\}} e_{v,3,q}\ x_{v,q}$$

$$s.t.\ \sum_{v=1}^{4} \sum_{\forall q \in \{1,3,5\}} r_{v,3}\ x_{v,q} \le b_3,\ 1 \le \sum_{v=1}^{4} x_{v,q} \le 4,\ \forall q \in \{1,3,5\},\ x_{v,q} \in \{0,1\}.$$

В данной модели предполагается, что для каждой группы избирателей выбирается хотя бы одно действие, значение b_3 соответствует общему ограничению по ресурсу для рассматриваемого кандидата G_3. Примеры решений имеют вид:

(i) при ограничении по ресурсу $b_3 \le 18$: $\{O_3^1, O_4^1, O_1^2, O_3^3, O_4^3\}$,

(ii) при ограничении по ресурсу $b_3 \le 28$: $\{O_1^1, O_3^1, O_4^1, O_1^2, O_3^2, O_4^2, O_3^3, O_4^3\}$.

Очевидно, приведенную задачу можно решать для каждого кандидата.

Таблица 17.18 содержит ряд перспективных задач/моделей для совершенствования предложенного похода.

Таблица 17.18. Перспективы использования задач/моделей

Стадия	Использованная задача/модель	Перспективные задачи/модели
1.Кластеризация	Иерархическая кластеризация (агломератвный алгоритм)	1.Другие методы кластеризации [268, 430] 2.Модификация алгоритма иерархической кластеризации [347]
2.Определение соответствия кандидатов группам избирателей	Задача о назначении	1.Обобщенная задача о назначении [291] 2.Многокритериальная задача о назначении [505] 3.Квадратичная задача о назначении [120] 4.Расширенная задача размещения (с набором правил и др.) [336, 349] 5.Модели теории игр и многоагентных систем [460, 517] 6.Динамические модели (задачи реконфигурации и др.) [344, 349]
3.Формирование комплексной стратегии кандидата	Задача блочного рюкзака	1.Многокритериальная задача блочного рюкзака [31] 2.Задачи комбинаторного синтеза на основе морфологической клики [344, 349] 3.Динамические модели (задачи реконфигурации и др.) [344, 349]

17.3 Проектирование бригады

Рассмотрим пример проектирования бригады исполнителей (управления персоналом) [336]. Структура бригады и DAs представлены на Рис. 17.14 (приоритеты DAs указаны в скобках). Критерии и их веса представлены в Таблице 17.19 (используется порядковая шкала [1,10]). Таблица 17.20 содержит оценки DAs по критериям и Таблица 17.21 содержит оценки совместимости между DAs. В результате получаются два Парето-эффективных решения:

$$S_1 = A_2 \star B_1 \star C_2,\ N(S_1) = (3; 2, 1, 0); \quad S_2 = A_3 \star B_1 \star C_3,\ N(S_2) = (3; 2, 1, 0).$$

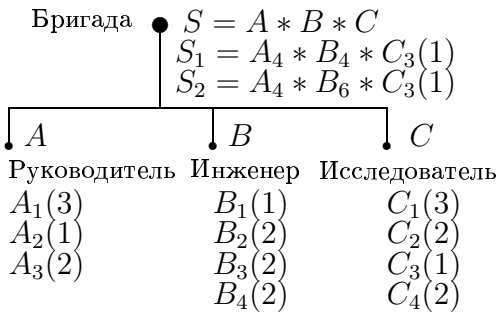


Рис. 17.14. Структура бригады

Таблица 17.19. Критерии

Критерии	Веса		
	A	B	C
1.Профессиональный опыт	3	3	3
2.Опыт работы на Западе		2	
3.Требуемая зарплата	-1	-3	-3
4.Творческие способности			4
5.Знание рынка	4		

Таблица 17.20. DAs и оценки

DAs	Критерии				
	1	2	3	4	5
A_1	5		10		3
A_2	15		20		5
A_3	10		15		4
B_1	15	5	7		
B_2	6	6	3		
B_3	10	0	9		
B_4	3	0	3		
C_1	15		10	3	
C_2	5		7	5	
C_3	4		4	7	
C_4	6		6	4	

Таблица 17.21. Совместимость

	B_1	B_2	B_3	B_4	C_1	C_2	C_3	C_4
A_1	2	3	3	1	2	0	0	3
A_2	3	0	0	1	0	3	0	0
A_3	3	0	1	1	1	0	3	0
B_1					0	3	3	0
B_2					1	1	0	3
B_3					0	0	2	0
B_4					0	0	0	3

17.4 Композиция продукта при электронной торговле

В данном разделе описан пример проектирования пользователем покупаемого продукта. Многие товары имеют сложную конфигурацию и покупатель может получить электронную услугу по самостоятельной генерации товара на основе его/ее потребностей и предпочтений. В качестве примеров составных товаров можно привести компьютер, автомобиль, частный дом. На Рис. 17.15 представлена упрощенная структура автомобиля:

1. корпус A: седан A_1, универсал A_2, джип A_3, пикап A_4, корпус спортивного автомобиля A_5;

2. двигатель B: дизель B_1, бензиновый B_2, электрический B_3, водородный B_4;

3. оборудование C: базовая версия C_1, компьютерный контроль C_2, компьютерный контроль & GPS-связь C_3.

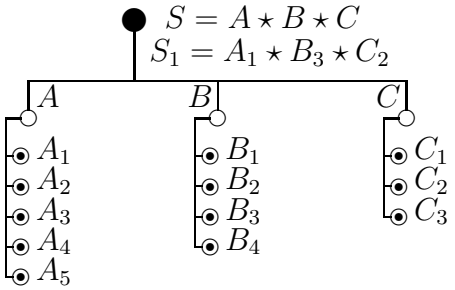


Рис. 17.15. Структура автомобиля

Теперь рассмотрим составную акцию в рамках электронной торговли. Упрощенная структура (*покупка автомобиля*) представлена на Рис. 17.16:

1. Место производства A (отечественный A_1, иностранный A_2).

2. Конфигурация автомобиля B (минимальная B_1, средняя B_2, максимальная B_3).

3. Способ оплаты C (кредит C_1, наличные C_2, безналичные C_3, в рассрочку C_4).

4. Место покупки D (автомагазин D_1, дилер D_2, прямо от производителя D_3).

5. Степень амортизации E (новый E_1, со вторых рук E_2).

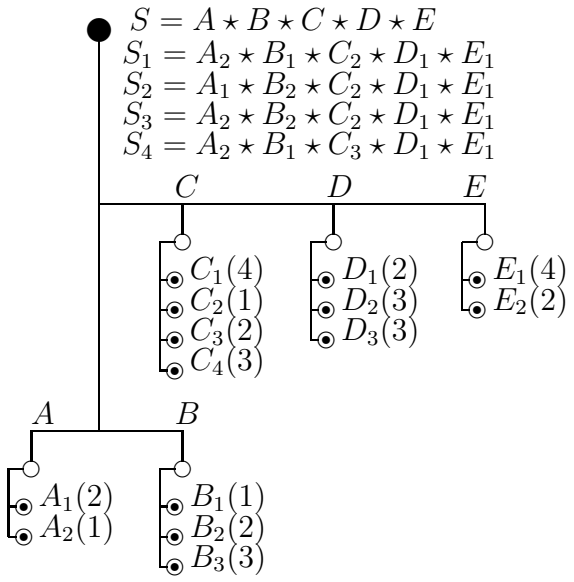


Рис. 17.16. Структура составной акции

Используются следующие критерии ('+' соответствует положительной ориентации порядковой шкалы $[1,5]$, '-' соответствует отрицательной ориентации шкалы):

(a) цена $K_{a1}, -$, престиж брэнда $K_{a2}, +$, useful life $K_{a3}, +$, уровень потребности в техобслуживании $K_{a4}, -$, надежность $K_{a5}, +$;

(b) стоимость $K_{b1}, -$, престиж бренда $K_{b2}, +$, удобство обновления (upgrade-ability) $K_{b3}, +$;

(c) риск кредита $K_{c1}, -$, стоимость эксплуатации $K_{c2}, +$, доступность $K_{c3}, +$;

(d) надежность $K_{d1}, -$, стоимость $K_{d2}, +$, качество сервисного обслуживания $K_{d3}, +$, гарантия $K_{d4}, -$;

(e) стоимость $K_{e1}, -$, уровень потребности в техобслуживании $K_{e2}, +$, гарантия $K_{e3}, +$.

Таблица 17.22 содержит порядковые для DAs по указанным критериям (экспертные оценивание). Результирующие приоритеты для DAs получены на основе многокритериального ранжирования для каждой части системы (шкала $[1,4]$). Приоритеты указаны в скобках на Рис. 16.4. Таблица 17.23 содержит оценки совместимости между DAs (шкала $[0,5]$). В результате получены составные DAs :

1. Идеальное решения: $S_1 = A_2 \star B_1 \star C_2 \star D_1 \star E_1$, $N(S_1) = (5;5,0,0,0)$. Это решение является одновременно Парето-эффективным.

2. Следующий уровень качества решений (*почти-оптимальные* включает следующие решения: $S_2 = A_1 \star B_2 \star C_2 \star D_1 \star E_1$, $N(S_2) = (5;4,1,0,0)$ $S_3 = A_2 \star B_2 \star C_2 \star D_1 \star E_1$, $N(S_3) = (5;4,1,0,0)$.

3. Дополнительно, целесообразно рассмотреть решение: $S_4 = A_2 \star B_1 \star C_3 \star D_1 \star E_1$, $N(S_4) = (4;4,1,0,0)$.

Очевидно, что $N(S_1) \succ N(S_2)(N(S_3)) \succ N(S_4)$. Рис. 17.17 представляет решетку качества для результирующих решений.

Таблицы 17.22. Оценки альтерантив

DAs	Критерии				
	1	2	3	4	5
A_1	2	3	3	3	2
A_2	4	5	5	5	4

DAs	Кр-рии		
	1	2	3
B_1	2	3	5
B_2	3	4	3
B_3	4	5	2

DAs	Кр-рии		
	1	2	3
C_1	5	5	4
C_2	1	4	3
C_3	1	4	2
C_4	5	3	4

DAs	Кр-рии			
	1	2	3	4
D_1	4	4	4	5
D_2	2	3	2	2
D_3	3	3	1	2

DAs	Кр-рии		
	1	2	3
E_2	4	2	5
E_1	2	4	1

Таблица 17.23. Совместимость

	B_1	B_2	B_3	C_1	C_2	C_3	C_4	D_1	D_2	D_3	E_1	E_1
A_1	5	5	5	2	5	4	3	5	4	0	5	4
A_2	5	5	5	5	5	5	5	5	4	3	5	4
B_1				5	5	5	5	5	4	3	5	4
B_2				5	5	5	5	5	3	1	5	3
B_3								5	2	1	5	2
C_1								4	1	0	4	1
C_2								5	5	2	5	5
C_3								4	1	4	5	2
C_4								3	0	0	4	0
D_1											5	1
D_2											3	4
D_3											1	4

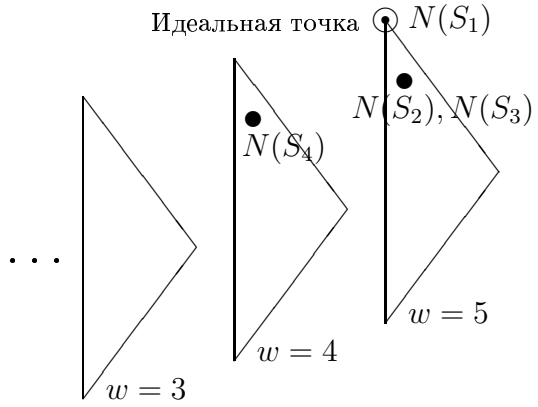


Рис. 17.17. Решетка качества

17.5 Комбинаторное формирование инвестиций

Для формирования портфеля инвестиций часто используются модели рюкзачного типа, многокритериальное принятие решений и вероятностный анализ (включая оценивание рисков). Здесь приводится упрощенный иллюстративный пример из [336] по комбинаторному построению портфеля инвестиций на основе ИММП. Рис. 17.18 содержит структуру и альтернативы составного портфеля инвестиций (приоритеты DAs указаны в скобках). Таблица 17.24 содержит критерии. DAs и их оценки представлены в Таблице 16.2.

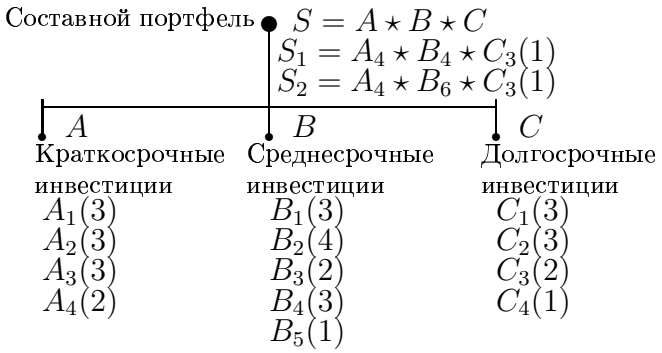


Рис. 17.18. Структура составного портфеля инвестиций

В Таблице 17.26 представлены оценки совместимости. Таблица 17.27 содержит Парето-эффективные составные DAs. Таблица 17.28 содержит узкие места и акции улучшения. На Рис. 17.19 проиллюстрированы точки, соответствующие результирующим составным DAs.

Таблица 17.24. Критерии

Критерии	Веса		
	A	B	C
1.Прибыль (0,...,100)	5	7	6
2.Риск (-; 0,...,100)	4	5	4
3.Престиж (0,1,2)	4	3	3
4.Возм-ть продолжения (0,...,5)	3		
5.Возм-ть основать нов. компанию (0,1,2)		4	4
6.Получение нового опыта (0,...,7)		2	2
7.Орг-ция нового рынка (0,...,7)		4	4
8.Возм-ть получить *ИМЯ* (0,...,7)	2	2	2
9.Связь с пред. деятельностью (0,...,7)	3	2	3

Таблица 17.25. DAs и оценки

DAs	Критерии								
	1	2	3	4	5	6	7	8	9
A_1 Гос. облигации	10	0	2	2				0	1
A_2 Банк. депозит	12	0	2	2				2	0
A_3 Спекуляция на бирже	30	40	3	1				2	0
A_4 Нефт. акции	20	25	3	1				2	0
B_1 Гос. облигации	15	0	2		0	0	0	0	1
B_2 Банк. депозит	18	0	2		0	0	0	0	1
B_3 Недвижимость	20	10	4		1	2	1	6	3
B_4 Драгоценности	30	30	4		0	1	1	2	2
B_5 Акции биотех. комп.	28	20	4		2	6	5	6	7
C_1 Гос. облигации	20	0	2		0	0	0	0	1
C_2 Банк. депозит	24	0	2		0	0	0	0	1
C_3 Антиквариат	30	20	4		1	1	1	5	2
C_4 Акции аэро-косм. комп.	26	12	4		1	3	2	4	4

Таблица 17.26. Совместимость

	B_1	B_2	B_3	B_4	B_5	C_1	C_2	C_3	C_4
A_1	4	3	3	3	5	4	3	5	5
A_2	5	4	4	3	5	5	4	5	4
A_3	5	5	4	3	2	5	5	4	3
A_4	5	5	5	4	2	5	5	4	3
B_1						4	3	5	5
B_2						4	4	5	5
B_3						5	5	4	4
B_4						5	5	3	3
B_5						5	5	4	3

Таблица 17.27. Составные DAs

DAs	N
$S_1 = A_4 * B_3 * C_1$	$5; 0, 2, 1, 0$
$S_2 = A_2 * B_5 * C_1$	$5; 1, 0, 2, 0$
$S_3 = A_2 * B_3 * C_4$	$4; 1, 1, 1, 0$
$S_4 = A_2 * B_5 * C_4$	$3; 2, 1, 0, 0$

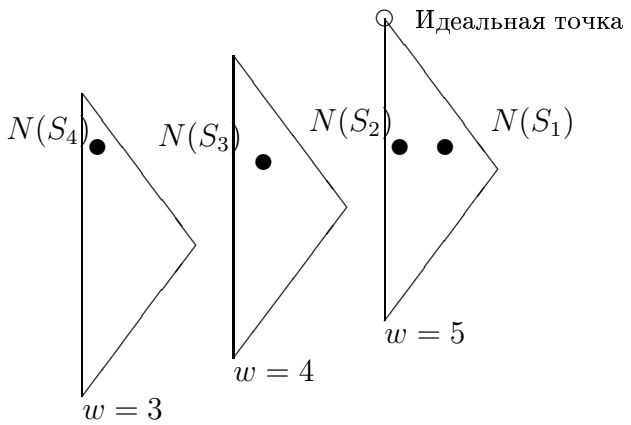


Рис. 17.19. Парето-эффективные точки

Таблица 17.28. Узкие места и акции улучшения

Составные DAs	Акции		Улучшения	
	DAs	IC	w/r	Тип
$S_1 = A_4 * B_3 * C_1$	A_2		$2 \Rightarrow 1$	2
$S_1 = A_4 * B_3 * C_1$	B_3		$2 \Rightarrow 1$	2
$S_1 = A_4 * B_3 * C_1$	C_1		$3 \Rightarrow 2$	2
$S_2 = A_2 * B_5 * C_1$	A_2		$3 \Rightarrow 2$	2
$S_2 = A_2 * B_5 * C_1$	C_1		$3 \Rightarrow 2$	2
$S_3 = A_2 * B_3 * C_4$	A_2		$3 \Rightarrow 2$	2
$S_3 = A_2 * B_3 * C_4$	B_3		$2 \Rightarrow 1$	2
$S_3 = A_2 * B_3 * C_4$		(A_2, B_3)	$4 \Rightarrow 5$	2
$S_4 = A_2 * B_5 * C_4$	A_2		$3 \Rightarrow 2$	2
$S_4 = A_2 * B_5 * C_4$		(B_5, C_4)	$3 \Rightarrow 4$	2

17.6 Выводы к главе

В данной главе представлены примеры применения составных комбинаторных 'технологических' схем в нескольких типовых приложениях в области менеджемента: комбинаторное формирование плана инвестиций, композиция продукта при электронной торговле, проектирование бригады исполнителей, схема политического маркетинга, комбинаторное планирование геологических исследований.

Очевидно, что указанные прикладные задачи могут быть использованы как базовые аналоги для различных приложений в менеджементе и экономике. Можно указать следующие направления для исследований в будущем:

1. исследование других приложений;

2. рассмотренные задач в режиме реального времени;

3. учет неопределенности параметров.

Глава 18

Приложения в преподавании

В данной главе [1] рассматриваются некоторые вопросы применения типовых комбинаторных схем поддержки модульных систем в преподавании.

18.1 Предварительные сведения

Ускорение научно-технического прогресса, наблюдаемое в последние годы, связано с рядом особенностей, которые включают следующие тенденции: (1) увеличение доли комплексных технологий, когда нововведения основываются на достижениях в нескольких областях знаний (например: биология, химия, инженерные науки, компьютерные науки, математика); (2) ускорение процессов внедрения нововведений; (3) уменьшение сроков 'макро-технологических' циклов, т.е. каждые несколько лет (или даже чаще) практически во всех отраслях происходит смена поколений технологий. Кроме того, подготовка и принятие решений повсеместно становится частью работы каждого работника. В результате, наблюдается повышение доли интеллектуальных операций переработки информации на всех этапах жизненного цикла продукции и это можно рассматривать как "третью грамотность" [19].

Уровни изобретательских задач по Г. Альтшуллеру (в рамках системы ТРИЗ) [52] можно использовать как некую порядковую шкалу при оценке уровня 'интеллектуальности'

[1] Материал главы основан на публикациях автора:
(i) Левин М.Ш. О третьей грамотности// Научно-техническая информация, сер. 2. 1995. N. 6. С. 20-30.
(ii) Левин М.Ш. О преподавании информационных технологий// Научно-техническая информация, сер. 1. 1996. N. 5. С. 14-23.
(iii) Левин М.Ш. О преподавании проектирования систем. *Информационные технологии и вычислительные системы*, 2007, Вып. 2, С. 89-94.
(iv) Levin M.Sh. Course "Design of systems: structural approach", DETC2006-99547. In: ASME Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2006), vol. 4a: 18th Int. Conf. on Design Theory and Methodology DTM, Pennsylvania, USA, Sept. 2006, pp. 475-484.
(v) Levin M.Sh., Course on system design (structural approach). Electronic preprint, 22 pp., March 20, 2011. http://arxiv.org/abs/1103.3845 [cs.SE]
(vi) Levin M.Sh., Student research projects in system design. In: Int. Conf. on Computer Supported Education CSEDU-2009, Lisbon, Portugal, vol. 2, pp. 67-72, 2009.
(vii) Levin M.Sh. Towards systems engineering education. In: 15th Eur. Meeting on Cybern. and Syst. Res. Vol. 1, Vienna, 2000, pp. 257-262.
(viii) Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Springer, 1998, chapter 10.
(ix) Levin M.Sh. Composite Systems Decisions, Springer, 2006, chapters 16, 17, 18.
(x) Levin M.Sh., Towards design of hierarchy (research survey). Electronic preprint, 36 pp., Dec. 8, 2012. Http://arxiv.org/abs/1212.1735 [math.OC].
(xi) Левин М.Ш. Иерархческий комбинаторный поход к планированию карьеры// Научно-техническая информация, сер. 2. 1996. N. 12. С. 22-27.

применительно к различным приложениям:

Уровень 1. Использование готового объекта (продукта, технологии, управленческого решения, стратегии, информации, нововведения и др.).

Уровень 2. Поиск и выбор объекта.

Уровень 3. Анализ и модификация некого исходного объекта.

Уровень 4. Создание (проектирование) нового объекта.

Уровень 5. Создание (проектирование) системы объектов.

Действительно интеллектуальными задачами являются задачи 4-го и 5-го уровней, задача 3-го уровня является промежуточной. Задача поиска и выбора (ранжирования объектов) зачастую является базовой в процессах подготовки и принятия решений. Далее будем считать задачи уровней 2 и 3 квази-интеллектуальными или квази-творческими. Разумная организация групповой работы может позволить успешно решать задачи 4-го и 5-го уровней на основе решения некоторого набора локальных квази-интеллектуальных задач.

В целом, можно указать два основных пути разрешения указанной выше ситуации, связанной с повышением доли интеллектуальных операций переработки информации:

1. Поиск творческих личностей и подготовка специалистов, способных решать интеллектуальные задачи (поиск, воспитание, обучение). Этот путь имеет определенные ограничения, связанные с предрасположенностью людей и возможностями процессов отбора, подготовки и обучения.

2. Получение глобальных творческих системных решений на основе комбинирования локальных квази-творческих задач.

Последнее является неким аналогом подхода Шеннона к построению надежных систем из ненадежных элементов, только в данном случае происходит ориентация на построение интеллектуальных решений (типа 4 и/или 5 уровней) на основе композиции решений для локальных квази-интеллектуальных задач (2-3 уровней).

18.2 Учебная среда и проектные задачи

Можно рассматривать следующую структуру современной 'среды' по некому учебному курсу (например, в области информационных технологий):

1. Лекционные материалы.

2. Упражнения (задания, проекты).

3. Базовые программное обеспечение (программы для поддержки учебного процесса, программы-прототипы, на основе которых студенты разрабатывают версии своих программ).

4. Дополнительные материалы: (4.1) учебники, (4.2) базовые научные статьи, (4.3) примеры выполненных дипломов и диссертаций, ()4.4) списки литературных источников и ссылок на сайты в Интернете, (4.5) специальные прикладные примеры (case studies).

Рис. 18.1 иллюстрирует взаимосвязь между материалами по курсу и базовыми задачами проектирования (для лектора и для студента(ов)). Эта схема соответствует методу ИММП и курсу автора “Проектирование систем: структурный подход” [22, 344, 345, 350, 353].

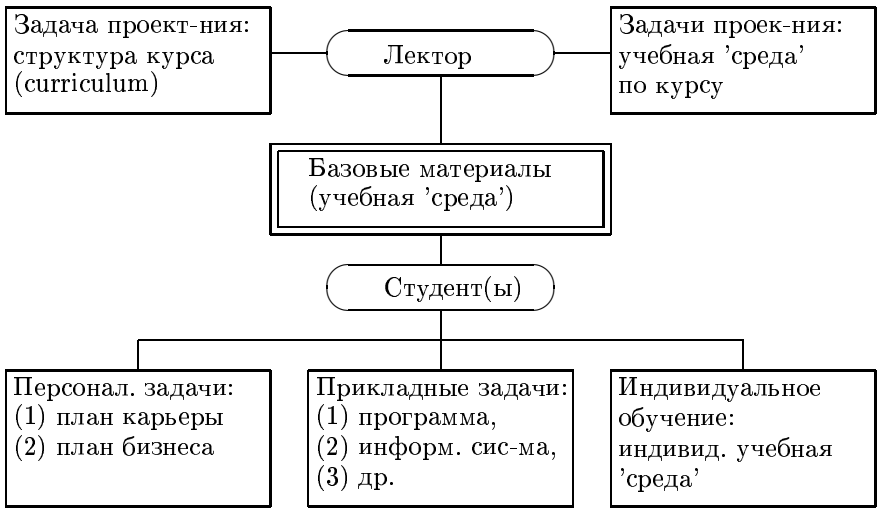


Рис. 18.1. Курс и задачи проектирования

18.3 Проектирование учебного курса

В данном разделе рассматривается пример модульного проектирования некого учебного курса на основе материалов по ИММП [336]. Данная задача является типовой (curriculum design) и может быть использована как преподавателем при формировании нового учебного курса, так и студентом при проектировании индивидуального плана обучения.

Для каждого компонента указанного курса определяется общий список возможных вариантов реализации (т.е., альтернатив DAs): без использования в курсе (O_{p0}); краткое рассмотрение (O_{p1}); изучение на некотором 'среднем' уровне (O_{p2}); тщательное (глубокое) изучение (O_{p3}); глубокое изучение включая специальную дополнительную работу (например, исследование прикладных задач, разработка алгоритмов и программы, подготовка курсовой работы и/или статьи) (O_{p4}).

Используются следующие критерии для оценивания указанных проектных альтернатив (DAs): (1) базовая профессиональная ориентация студента/студенческой группы; (2) связь с будущей трудовой деятельностью; (3) имеющаяся подготовка студентов; (4) возможности выполнения дополнительных работ. В рассматриваемом примере оценки (включая приоритеты) основаны на экспертных суждениях. Структура исходного учебного материала и DAs (в скобках указаны приоритеты для DAs):

0. Курс по ИММП $S = H \star R \star B \star X$:
1. Описание ИММП $H = D \star F \star P \star M \star C \star A$:
  1.1. Схемы проектирования D: $D_1(1)$, $D_2(3)$, $D_4(2)$.
  1.2. Основы ИММП F: $F_2(1)$, $F_3(2)$.
  1.3. Компоненты ИММП P: $P_2(2)$, $P_3(1)$.
  1.4. Задача композиции (синтеза) M: $M_2(2)$, $M_3(1)$, $M_4(3)$.
  1.5. Задача многокритериального выбора C: $C_1(1)$, $C_3(2)$, $C_4(2)$.
  1.6. Анализ и улучшение решений A: $A_2(3)$, $A_3(1)$, $A_4(1)$.
2. Связанные области $R = E \star N \star Q \star L$:
  2.1. Инженерная информация E: $E_0(1)$, $E_1(3)$, $E_4(2)$.
  2.2. Конфликтные ситуации N: $N_1(2)$, $N_4(1)$.
  2.3. Анализ качества Q: $Q_1(1)$, $Q_4(2)$.
  2.4. Образование и частная жизнь L: $L_1(2)$, $L_2(3)$, $L_4(2)$.
3. Комбинаторная оптимизация $B = K \star T \star W \star U$:

3.1. Задача о рюкзаке K: $K_2(2)$, $K_3(2)$, $K_4(1)$.

3.2. Задача коммивояжера T: $T_1(2)$, $T_4(1)$.

3.3. Задача о размещении W: $W_1(1)$, $W_4(3)$.

3.4. Задача маршрутизации U: $U_0(2)$, $U_1(1)$, $U_4(3)$.

4. Примеры прикладных задач $X = O \star I \star J \star V \star G \star Y \star Z$:

4.1. Интеграция программ O: $O_0(3)$, $O_1(1)$, $O_4(2)$.

4.2. Проектирование чел.-маш. интерфейса I: $I_0(2)$, $I_1(1)$, $I_4(3)$.

4.3. Проектирование информационного центра J: $J_0(12)$, $J_1(2)$, $J_4(2)$.

4.4. Проектирование конвейера V: $V_0(3)$, $V_1(1)$, $V_4(2)$.

4.5. Планирование исследования нефти и газа G: $G_0(2)$, $G_1(2)$, $G_2(1)$.

4.6. Планирование карьеры Y: $Y_0(3)$, $Y_2(1)$, $Y_4(2)$.

4.7. Формирование портфеля инвестиций Z: $Z_0(2)$, $Z_1(2)$, $Z_3(1)$, $Z_4(3)$.

Оценки совместимости между DAs представлены в Таблицах 18.1, 18.2, 18.3, 18.4, 18.5. Таблица 18.6 содержит составные DAs.

Таблица 18.1. Совместимость

	F_2	F_3	P_2	P_3	M_2	M_3	M_4	C_1	C_3	C_4	A_2	A_3	A_4
D_1	1	2	1	2	2	3	2	1	2	2	5	5	5
D_2	4	4	4	4	4	4	5	4	4	4	5	5	5
D_4	1	5	2	5	1	5	4	3	4	4	5	5	5
F_2			4	5	4	4	5	2	4	4	4	3	3
F_3			5	5	4	5	5	3	5	5	4	5	5
P_2					4	4	5	5	4	4	5	3	3
P_3					4	5	5	4	5	5	4	5	5
M_2								5	4	4	5	3	3
M_3								3	5	4	3	5	5
M_4								3	5	4	3	5	5
C_1											5	5	5
C_3											5	5	5
C_4											5	5	5

Таблица 18.2. Совместимость

	R_1	R_2	R_3	R_4	R_5	R_6	B_1	B_2	B_3	X_1	X_2
H_1	4	5	2	3	1	1	4	2	2	3	2
H_2	5	5	3	4	1	1	5	4	4	5	3
H_3	4	5	3	4	1	1	4	2	2	3	2
H_4	5	5	3	4	1	1	5	4	4	5	3
H_5	4	5	2	3	1	1	4	2	2	3	2
H_6	4	5	2	3	1	1	5	4	4	5	3
H_7	2	3	1	2	1	1	4	2	2	3	2
H_8	2	3	1	2	1	1	5	4	4	5	3
R_1							5	2	2	4	2
R_2							5	4	2	5	3
R_3							3	2	2	2	2
R_4							3	4	2	4	2
R_5							3	2	2	2	1
R_6							3	2	2	4	2
B_1										5	3
B_2										4	3
B_3										3	5

Таблица 18.3. Совместимость

	F_2	F_3	P_2	P_3	M_2	M_3	M_4
E_0	1	1	2	2	5	1	1
E_1	4	4	2	3	3	5	4
E_4	4	5	3	5	4	5	5
N_1			4	5	5	4	4
N_4			5	4	4	5	5
Q_1					5	5	5
Q_4					5	5	5

Таблица 18.4. Совместимость

	T_1	T_4	W_1	W_4	U_0	U_1	U_4
K_2	3	4	3	4	3	5	5
K_3	4	5	4	4	3	5	5
K_4	4	5	4	5	3	5	5
T_1			5	5	2	4	3
T_4			5	5	2	3	5
W_1					5	5	5
W_4					5	5	5

Таблица 18.5. Совместимость

	I_0	I_1	I_4	J_0	J_4	V_0	V_1	V_4	G_0	G_2	Y_0	Y_2	Y_4	Z_0	Z_1	Z_4	Z_4
O_0	5	2	2	5	4	5	5	5	5	5	5	5	5	5	5	5	5
O_1	4	4	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5
O_4	2	5	5	4	5	5	5	4	5	5	5	5	5	5	5	5	5
I_0				5	1	5	5	5	5	5	5	5	5	5	5	5	5
I_1				5	3	5	5	5	5	5	5	5	5	5	5	5	5
I_4				3	5	5	5	5	5	5	5	5	5	5	5	5	5
J_0						5	5	5	5	5	5	5	5	5	4	3	3
J_4						5	5	5	5	5	5	5	5	2	3	5	5
V_0									5	5	5	5	5	5	5	5	5
V_1									5	5	5	5	5	5	5	5	5
V_4									5	5	5	5	5	5	5	5	5
G_0											5	5	5	5	3	3	2
G_2											5	5	5	4	4	5	5
Y_0														4	4	3	3
Y_2														4	4	5	5
Y_4														4	4	5	5

Таблица 18.6. Составные DAs

DAs	N
$H_1 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_3$	$(4;3,3,0)$
$H_2 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_4$	$(4;3,3,0)$
$H_3 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_3$	$(4;3,3,0)$
$H_4 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_4$	$(4;3,3,0)$
$H_5 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_3$	$(3;4,2,0)$
$H_6 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_4$	$(3;4,2,0)$
$H_7 = D_1 \star F_2 \star P_3 \star M_3 \star C_1 \star A_3$	$(1;6,0,0)$
$H_8 = D_1 \star F_2 \star P_3 \star M_3 \star C_1 \star A_4$	$(1;6,0,0)$
$X_1 = V_1 \star O_1 \star I_4 \star J_4 \star Z_3 \star Y_2 \star G_2$	$(5;5,1,1)$
$X_2 = V_1 \star O_1 \star I_1 \star J_0 \star Z_3 \star Y_2 \star G_2$	$(4;7,0,0)$
$R_1 = N_4 \star Q_4 \star L_1 \star E_4$	$(4;1,3,0)$
$R_2 = N_4 \star Q_4 \star L_4 \star E_4$	$(4;1,3,0)$
$R_3 = N_4 \star Q_1 \star L_1 \star E_4$	$(3;2,2,0)$
$R_4 = N_4 \star Q_1 \star L_4 \star E_4$	$(3;2,2,0)$
$R_5 = N_4 \star Q_1 \star L_1 \star E_0$	$(1;3,1,0)$
$R_6 = N_4 \star Q_1 \star L_4 \star E_0$	$(1;3,1,0)$
$B_1 = K_4 \star T_4 \star W_4 \star U_4$	$(5;2,0,2)$
$B_2 = K_4 \star T_4 \star W_1 \star U_4$	$(4;3,0,1)$
$B_3 = K_4 \star T_4 \star W_1 \star U_1$	$(3;4,0,0)$
$S_1 = H_2 \star R_2 \star B_1 \star X_1$	$(5;4,0,0)$

Получаются следующие результирующая композиция курса: $S_1 = H_2 \star R_2 \star B_1 \star X_1 = (D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_4) \star (N_4 \star Q_4 \star L_4 \star E_4) \star (K_4 \star T_4 \star W_4 \star U_4) \star (V_1 \star O_1 \star I_4 \star J_4 \star Z_3 \star Y_2 \star G_2)$.

Таблица 18.7 содержит узкие места и акции улучшения.

Таблица 18.7. Узкие места и акции улучшения

Составные DAs	Узк. места		Акции (w/r)
	DA	IC	
$H_1 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_3$	D_4		$2 \Rightarrow 1$
$H_1 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_3$	F_3		$2 \Rightarrow 1$
$H_1 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_3$	C_3		$2 \Rightarrow 1$
$H_2 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_4$	D_4		$2 \Rightarrow 1$
$H_2 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_4$	F_3		$2 \Rightarrow 1$
$H_2 = D_4 \star F_3 \star P_3 \star M_3 \star C_3 \star A_4$	C_3		$2 \Rightarrow 1$
$H_3 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_3$	D_4		$2 \Rightarrow 1$
$H_3 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_3$	F_3		$2 \Rightarrow 1$
$H_3 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_3$	C_4		$2 \Rightarrow 1$
$H_4 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_4$	D_4		$2 \Rightarrow 1$
$H_4 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_4$	F_3		$2 \Rightarrow 1$
$H_4 = D_4 \star F_3 \star P_3 \star M_3 \star C_4 \star A_4$	C_4		$2 \Rightarrow 1$
$H_5 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_3$	D_4		$2 \Rightarrow 1$
$H_5 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_3$	F_3		$2 \Rightarrow 1$
$H_6 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_4$	D_4		$2 \Rightarrow 1$
$H_6 = D_4 \star F_3 \star P_3 \star M_3 \star C_1 \star A_4$	F_3		$2 \Rightarrow 1$
$R_1 = N_4 \star Q_4 \star L_1 \star E_4$	Q_4		$2 \Rightarrow 1$
$R_1 = N_4 \star Q_4 \star L_1 \star E_4$	L_1		$2 \Rightarrow 1$
$R_1 = N_4 \star Q_4 \star L_1 \star E_4$	E_4		$2 \Rightarrow 1$
$R_2 = N_4 \star Q_4 \star L_4 \star E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 \star Q_4 \star L_4 \star E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 \star Q_4 \star L_4 \star E_4$	L_4		$2 \Rightarrow 1$
$R_2 = N_4 \star Q_4 \star L_4 \star E_4$		(N_4, Q_4)	$4 \Rightarrow 5$
$R_3 = N_4 \star Q_1 \star L_4 \star E_4$	L_1		$2 \Rightarrow 1$
$R_3 = N_4 \star Q_1 \star L_1 \star E_4$	E_4		$2 \Rightarrow 1$
$R_3 = N_4 \star Q_1 \star L_4 \star E_4$		(Q_1, E_4)	$3 \Rightarrow 4$
$R_4 = N_4 \star Q_1 \star L_4 \star E_4$	L_4		$2 \Rightarrow 1$
$R_4 = N_4 \star Q_1 \star L_4 \star E_4$	E_4		$2 \Rightarrow 1$
$R_4 = N_4 \star Q_1 \star L_4 \star E_4$		(Q_1, E_4)	$3 \Rightarrow 4$
$R_5 = N_4 \star Q_1 \star L_1 \star E_0$	L_1		$2 \Rightarrow 1$
$R_5 = N_4 \star Q_1 \star L_1 \star E_0$		(N_4, E_0)	$1 \Rightarrow 2$
$R_6 = N_4 \star Q_1 \star L_4 \star E_0$	L_1		$2 \Rightarrow 1$
$B_1 = K_4 \star T_4 \star W_4 \star U_4$	W_4		$3 \Rightarrow 2$
$B_1 = K_4 \star T_4 \star W_4 \star U_4$	U_4		$3 \Rightarrow 2$
$B_2 = K_4 \star T_4 \star W_1 \star U_4$	U_4		$3 \Rightarrow 2$
$B_2 = K_4 \star T_4 \star W_1 \star U_4$		(K_4, W_1)	$4 \Rightarrow 5$
$B_3 = K_4 \star T_4 \star W_1 \star U_1$		(T_4, U_1)	$3 \Rightarrow 4$
$X_1 = V_1 \star O_1 \star I_4 \star J_4 \star Z_3 \star Y_2 \star G_2$	J_4		$2 \Rightarrow 1$
$X_1 = V_1 \star O_1 \star I_4 \star J_4 \star Z_3 \star Y_2 \star G_2$	I_4		$3 \Rightarrow 2$

18.4 Агрегированный курс по проектированию

В данном разделе рассматривается пример агрегации трех курсов (трех соответствующих морфологических структур): (1) курс по системному инженирингу (структура Λ^1) [336, 337] (Рис. 18.2); (2) курс по информационным технологиям course on information engineering (структура Λ^2) [20, 332, 336] (Рис. 18.3); (3) курс по иерархическому проектированию (структура Λ^3) [336, 344, 345, 22, 350] (Рис. 18.4). Для каждой висячей вершины в иерархических моделях курсов используются следуюшие общие типы DAs (т.е., соответствующие типовые образовательные модули) (Рис. 18.2, Рис. 18.3, Рис. 18.4): "отсутствие" образовательного модуля X_0, сокращенный (сжатый) вариант образовательного модуля X_1, средний уровень (по содержанию) образовательного модуля X_2, глубокий уровень (по содержанию) образовательного модуля X_3, глубокий уровень (по содержанию) образовательного модуля с специальной студенческой работой (проектом) X_4. Оценки совмести-

мости DAs для исследуемых морфологических структур представлены в таблицах (экспертные суждения): (i) структура Λ^1: Таблица 18.8; (ii) структура Λ^2: Таблица 18.9; (iii) структура Λ^3: Таблица 18.10.

После использования ИММП, получаются следующие составные DAs:

I. Для структуры Λ^1 (Рис. 18.2):

$D_1 = L_3 \star G_2 \star C_2 \star M_2,\ N(D_1) = (2;3,0,1),$

$D_2 = L_4 \star G_2 \star C_1 \star M_2,\ N(D_2) = (1;2,2,0);$

$Q_1 = A_3 \star B_2,\ N(Q_1) = (3;2,0,0),\ Q_2 = A_4 \star B_4,\ N(Q_2) = (3;2,0,0);$

$S_1^1 = D_1 \star Q_1 = (L_3 \star G_2 \star C_2 \star M_2) \star (A_3 \star B_3),$

$S_2^1 = D_1 \star Q_2 = (L_3 \star G_2 \star C_2 \star M_2) \star (A_4 \star B_4),$

$S_3^1 = D_2 \star Q_1 = (L_4 \star G_2 \star C_1 \star M_2) \star (A_3 \star B_3),$

$S_4^1 = D_2 \star Q_2 = (L_4 \star G_2 \star C_1 \star M_2) \star (A_4 \star B_4).$

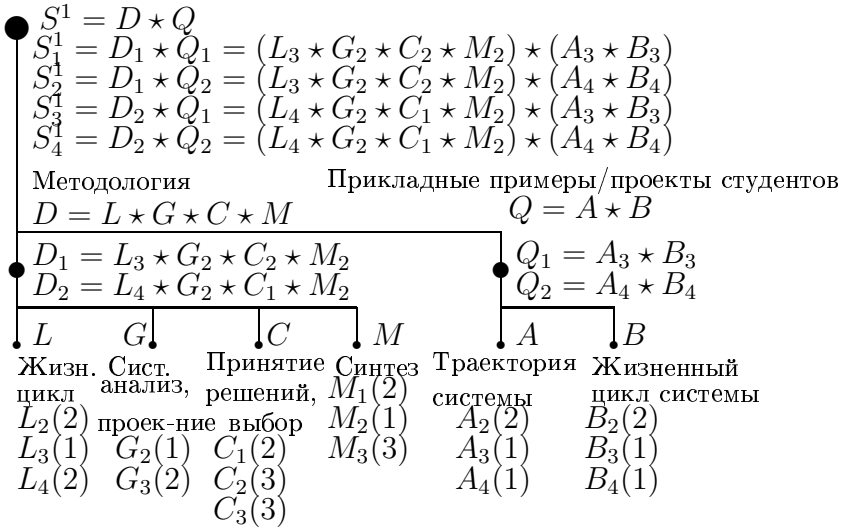


Рис. 18.2. Курс по системному инженирингу (структура Λ^1)

II. Для структуры Λ^2 (Рис. 18.3):

$I_1 = L_3 \star G_2 \star C_2 \star E_2,\ N(I_1) = (3;2,1,0);$

$O_1 = H_3 \star W_2 \star M_2,\ N(O_1) = (2;3,0,0),\ O_2 = H_2 \star W_2 \star M_2,\ N(O_2) = (3;2,1,0);$

$Q_1 = V_3 \star J_3 \star Y_2 \star R_3,\ N(Q_1) = (3;4,0,0);$

$S_1^2 = I_1 \star O_1 \star Q_1 = (G_2 \star C_2 \star E_2) \star (H_3 \star W_2 \star M_2) \star (V_3 \star J_3 \star Y_2 \star R_3),$

$S_2^2 = I_1 \star O_2 \star Q_1 = (G_2 \star C_2 \star E_2) \star (H_2 \star W_2 \star M_2) \star (V_3 \star J_3 \star Y_2 \star R_3).$

III. Для структуры Λ^3 (Рис. 18.4):

$I_1 = G_2 \star C_2,\ N(I_1) = (3;2,0,0);$

$O_1 = H_2 \star W_2 \star M_2 \star U_1,\ N(O_1) = (3;3,1,0),$

$O_2 = H_3 \star W_2 \star M_2 \star U_1,\ N(O_2) = (2;4,0,0);$

$Q_1 = F_4 \star J_1 \star Y_2 \star Z_3,\ N(Q_1) = (3;3,1,0);$

$S_1^3 = I_1 \star O_1 \star Q_1 = (G_2 \star C_2) \star (H_2 \star W_2 \star M_2 \star U_1) \star (F_4 \star J_1 \star Y_2 \star Z_3),$

$S_2^3 = I_1 \star O_2 \star Q_1 = (G_2 \star C_2) \star (H_3 \star W_2 \star M_2 \star U_1) \star (F_4 \star J_1 \star Y_2 \star Z_3).$

Очевидно, что можно агрегировать полученные составные решения (Рис. 18.5). С другой стороны, можно рассмотреть расширенную стратегию агрегирования: *стратегия IV*. Общая структура (Λ) включает следующие части: (i) древовидную модель системы $\mathbf{T}$, (ii) множество висячих вершин $\mathbf{P}$, (iii) множества проектных альтернатив DAs для каждой висячей вершины $\mathbf{D}$, (iv) ранжирование проектных альтернатив DAs (т.е., порядковые приоритеты) $\mathbf{R}$, (v) оценки совместимости между DAs $\mathbf{I}$. Можно рассматривать следующий вектор близости между двумя структурами $\Lambda^\alpha, \Lambda^\beta$:

$\overline{\rho}\ (\Lambda^\alpha, \Lambda^\beta) = (\rho^t(\mathbf{T}^\alpha,\mathbf{T}^\beta),\ \rho^t(\mathbf{P}^\alpha,\mathbf{P}^\beta),\ \rho^t(\mathbf{D}^\alpha,\mathbf{D}^\beta),\ \rho^t(\mathbf{R}^\alpha,\mathbf{R}^\beta),\ \rho^t(\mathbf{I}^\alpha,\mathbf{I}^\beta)).$

В результате, Λ^{agg} будет искаться как Парето-эффективное решение по векторам

$$\overline{\rho}\,(\Lambda^{agg}, \Lambda^i) \;\; \forall i \in \{i\},$$

где индекс i соответствует исходному решению. Рассмотрим упрощенную схему решения:

Фаза 1. Агрегация базовых исходных данных для исходных структур:
1.1. агрегация морфологических структур, включая следующее:
(1.1.1.) агрегация множеств висячих вершин,
(1.1.2.) агрегация множеств DAs для каждой висячей вершины,
(1.1.3.) агрегация ранжирований DAs,
(1.1.4.) агрегация оценок совместимости для множеств DAs;
1.2. агрегация древовидных структур.
Фаза 2. Новое иерархическое проектирование.

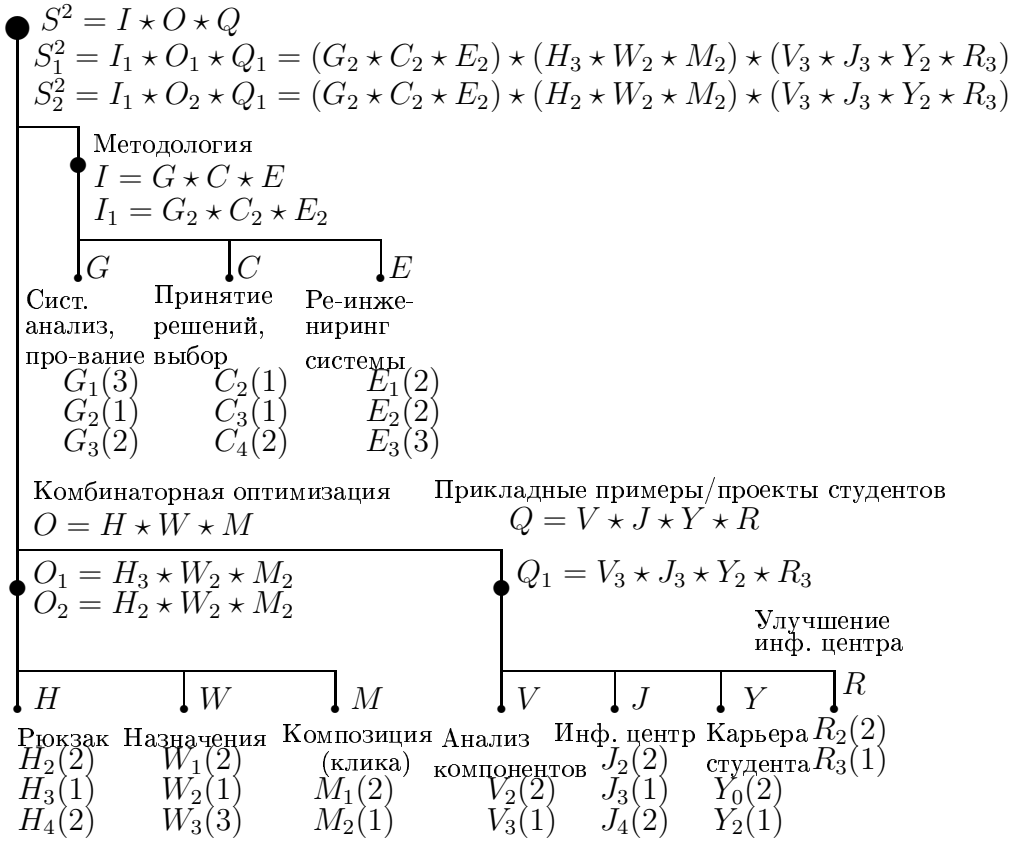


Рис. 18.3. Курс по информационным системам (структура Λ^2)

Для агрегируемых структур (Λ^1, Λ^2, Λ^3) рассматриваются следующие стадии:

Стадия 1. Агрегация множеств висячих вершин исходных структур (Рис. 18.6);
Стадия 2. Агрегация морфологических структур:
2.1. агрегация множеств DAs для каждой висячей вершины (Рис. 18.7, Рис. 18.8, Рис. 18.9, Рис. 18.10, Рис. 18.11, Рис. 18.12, Рис. 18.13, Рис. 18.14, Рис. 18.15, Рис. 18.16, Рис. 18.17, Рис. 18.18, Рис. 18.19, Рис. 18.20, Рис. 18.21, Рис. 18.22);
2.2. агрегация взаимосвязей (оценок совместимости) для множеств DAs (Таблица 18.11, Таблица 18.12, Таблица 18.13; используется правило: “выбор наибольшего значения” или экспертное суждение);
Стадия 3. Построение агрегированной древовидной структуры (Рис. 18.23);

Стадия 4. Иерархическое проектирование интегрированного курса (агрегированное решение) (Рис. 18.23).

Здесь, агрегированная суперструктура (надструктура) $\overline{\Lambda}$ (Рис. 18.23) получена на основе экспертных суждений.

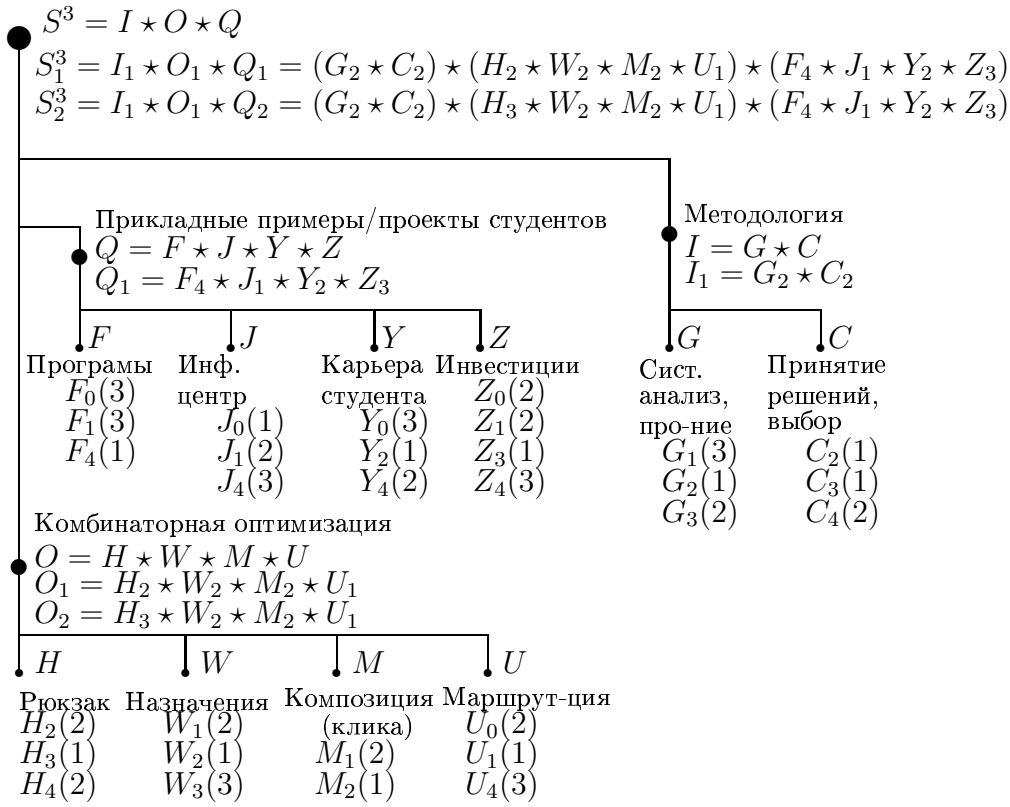


Рис. 18.4. Курс по иерархическому проектированию (структура Λ^3)

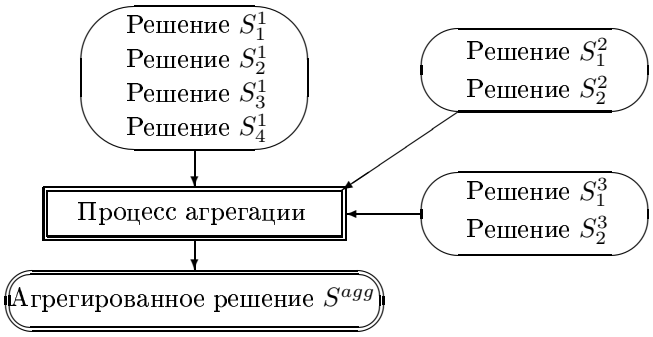


Рис. 18.5. Агрегация решений

Таблица 18.8. Оценки совместимости

	G_2	G_3	C_1	C_2	C_3	M_1	M_2	M_3
L_2	2	0	3	2	0	3	2	0
L_3	2	3	0	2	2	1	3	2
L_4	1	3	0	3	3	0	2	3
G_2			1	3	2	1	3	2
G_3			0	2	3	0	2	3
C_1						3	1	0
C_2						1	3	1
C_3						0	2	3

	B_2	B_3	B_4
A_2	3	2	1
A_3	2	3	2
A_4	1	2	3

Таблица 18.9. Оценки совместимости

	C_2	C_3	C_4	E_1	E_2	E_3
G_1	2	1	0	3	1	0
G_2	3	0	1	1	3	2
G_3	0	3	3	0	2	3
C_2				3	3	0
C_3				3	3	3
C_4				0	2	3

	W_1	W_2	W_3	M_1	M_2
H_2	3	3	2	3	3
H_3	1	2	3	1	3
H_4	0	2	3	0	3
W_1				3	2
W_2				3	3
W_3				3	3

	J_2	J_3	J_4	Y_0	Y_2	R_2	R_3
V_2	3	2	2	3	3	3	3
V_3	2	3	3	3	3	3	3
J_2				3	2	3	2
J_3				3	3	2	3
J_4				3	3	2	3
Y_0						2	2
Y_2						2	3

Таблица 18.10. Оценки совместимости

	J_0	J_1	J_4	Y_0	Y_2	Y_4	Z_0	Z_1	Z_3	Z_4
F_0	3	1	1	3	1	0	3	2	2	0
F_1	0	3	3	0	3	3	1	3	3	3
F_4	0	3	3	0	3	3	0	2	3	3
J_0				3	1	0	3	2	2	0
J_1				0	3	3	1	3	3	3
J_4				0	3	3	0	2	3	3
Y_0							3	2	2	0
Y_2							1	3	3	3
Y_4							0	2	3	3

	W_1	W_2	W_3	M_1	M_2	U_0	U_1	U_4
H_2	3	3	2	3	3	3	3	3
H_3	1	2	3	1	3	3	3	3
H_4	0	2	3	0	3	3	3	3
W_1				3	2	3	2	2
W_2				3	3	2	3	0
W_3				3	3	2	2	3
M_1						1	3	3
M_2						1	3	3

	C_2	C_3	C_4
G_1	2	1	0
G_2	3	0	1
G_3	0	3	3

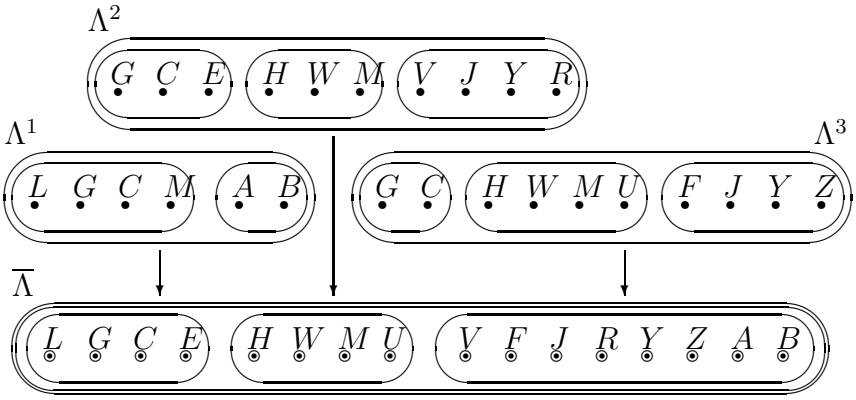


Рис. 18.6. Агрегация множеств висячих вершин

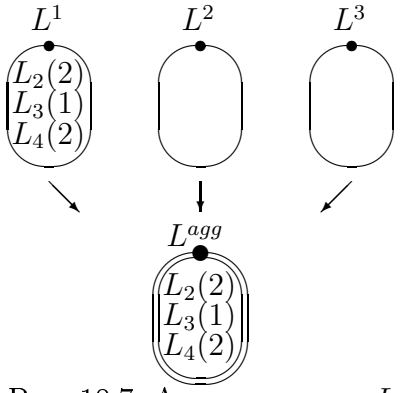


Рис. 18.7. Агрегация для L

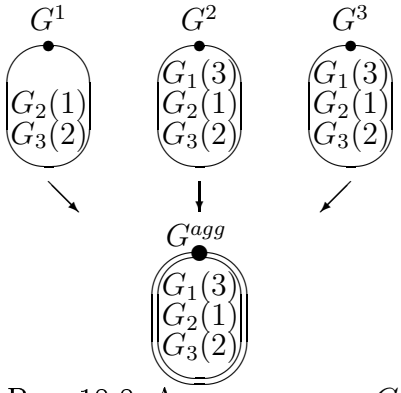


Рис. 18.8. Агрегация для G

После использования ИММП получаются следующие составные DAs для результирующей агрегированной структуры $(\overline{\Lambda})$ (Рис. 18.23):

$X_1 = V_3 \star F_4 \star J_3 \star R_3 \star Y_2 \star Z_3 \star A_3 \star B_3,\ N(X_1) = (3; 8, 0, 0),$

$O_1 = H_3 \star W_2 \star M_2 \star U_1,\ N(O_1) = (2; 4, 0, 0);$

$I_1 = L_3 \star G_2 \star C_2 \star E_2,\ N(O_1) = (2; 4, 0, 0),$

$I_2 = L_3 \star G_2 \star C_3 \star E_2,\ N(O_1) = (2; 4, 0, 0),$

$I_3 = L_4 \star G_3 \star C_3 \star E_2,\ N(O_1) = (3; 2, 2, 0);$
$\overline{S}_1 = I_1 \star O_1 \star X_1 = (L_3 \star G_2 \star C_2 \star E_2) \star (H_3 \star W_2 \star M_2 \star U_1) \star (V_3 \star F_4 \star J_3 \star R_3 \star Y_2 \star Z_3 \star A_3 \star B_3),$
$\overline{S}_2 = I_2 \star O_1 \star X_1 = (L_3 \star G_2 \star C_3 \star E_2) \star (H_3 \star W_2 \star M_2 \star U_1) \star (V_3 \star F_4 \star J_3 \star R_3 \star Y_2 \star Z_3 \star A_3 \star B_3),$
$\overline{S}_2 = I_3 \star O_1 \star X_1 = (L_3 \star G_3 \star C_3 \star E_2) \star (H_3 \star W_2 \star M_2 \star U_1) \star (V_3 \star F_4 \star J_3 \star R_3 \star Y_2 \star Z_3 \star A_3 \star B_3).$

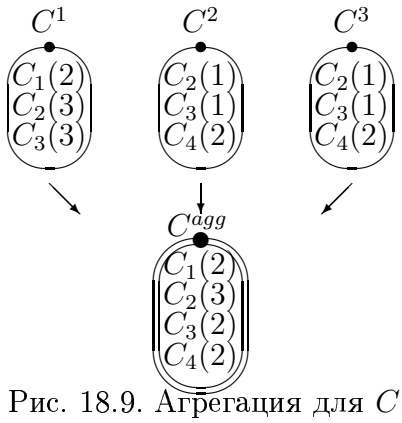


Рис. 18.9. Агрегация для C

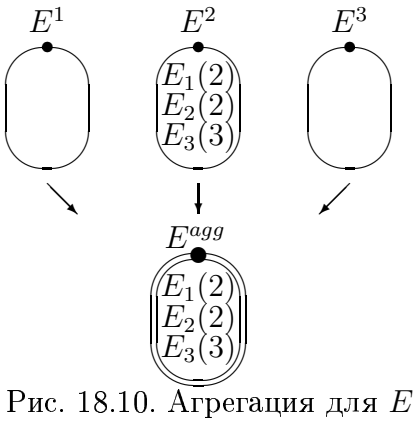


Рис. 18.10. Агрегация для E

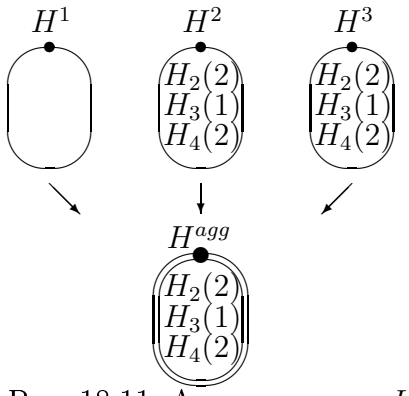


Рис. 18.11. Агрегация для H

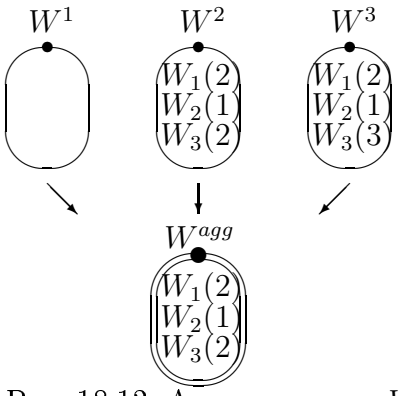


Рис. 18.12. Агрегация для W

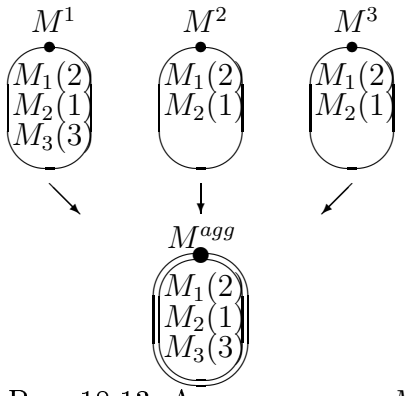


Рис. 18.13. Агрегация для M

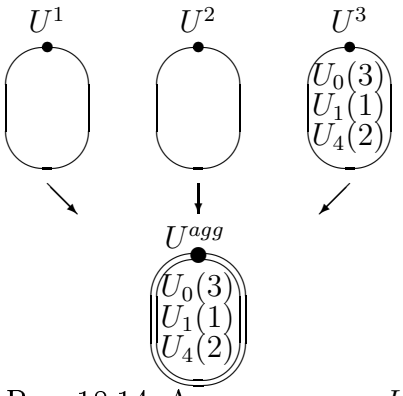


Рис. 18.14. Агрегация для U

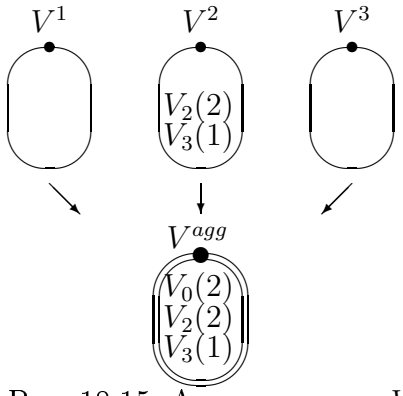


Рис. 18.15. Агрегация для V

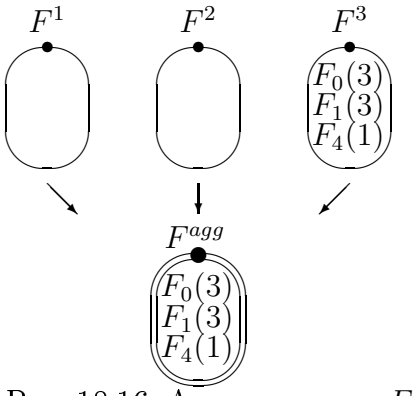


Рис. 18.16. Агрегация для F

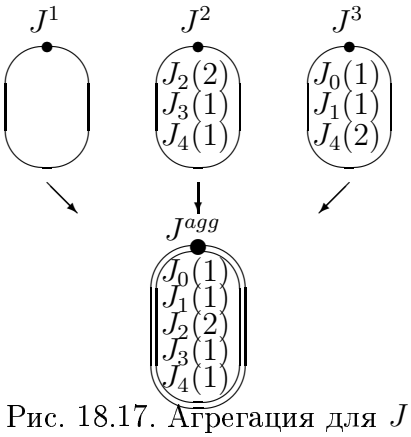


Рис. 18.17. Агрегация для J

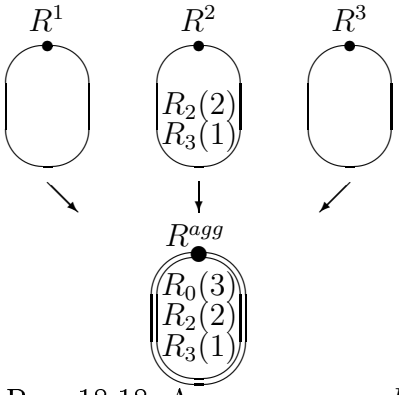


Рис. 18.18. Агрегация для R

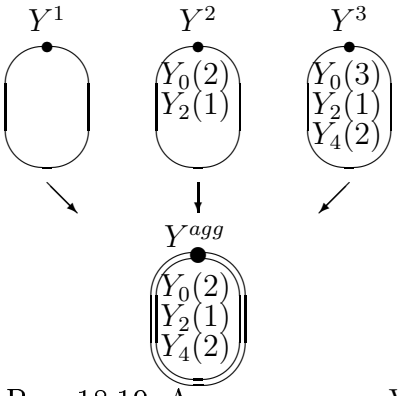


Рис. 18.19. Агрегация для Y

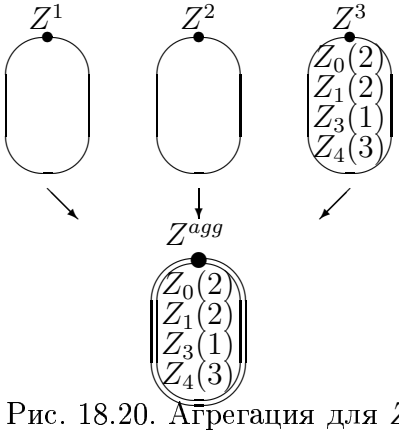


Рис. 18.20. Агрегация для Z

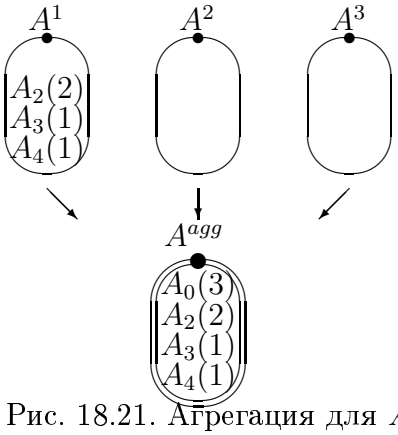


Рис. 18.21. Агрегация для A

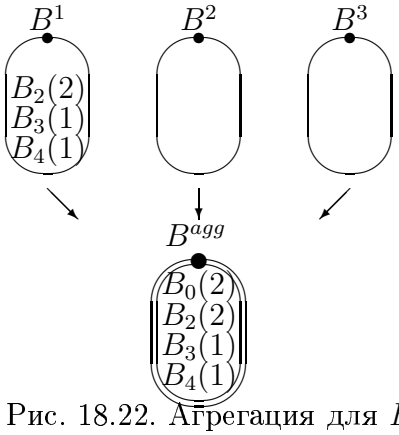


Рис. 18.22. Агрегация для B

Таблица 18.11. Оценки совместимости

	W_1	W_2	W_3	M_1	M_2	M_3	U_0	U_1	U_4
H_2	3	3	2	3	3	3	3	3	3
H_3	1	2	3	1	3	3	3	3	3
H_4	0	2	3	0	3	3	3	3	3
W_1				3	2	2	3	2	2
W_2				3	3	3	2	2	2
W_3				3	3	3	2	2	3
M_1							1	0	2
M_2							0	3	3
M_3							0	2	3

	F_0	F_1	F_4	J_0	J_1	J_2	J_3	J_4
V_0	3	1	1	3	1	1	0	0
V_2	2	3	3	1	1	3	2	2
V_3	2	2	3	1	2	2	3	3
F_0				3	1	1	1	1
F_1				1	3	3	3	3
F_4				1	3	3	3	3

Таблица 18.12. Оценки совместимости

	G_1	G_2	G_3	C_1	C_2	C_3	C_4	E_1	E_2	E_3
L_2	2	2	0	3	2	1	0	3	2	1
L_3	1	2	3	0	2	2	2	1	3	3
L_4	0	1	3	0	3	3	3	1	3	3
G_1				0	2	1	0	3	1	0
G_2				1	3	2	3	1	2	2
G_3				0	2	3	3	0	3	3
C_1								3	1	0
C_2								3	3	0
C_3								3	3	3
C_4								0	2	3

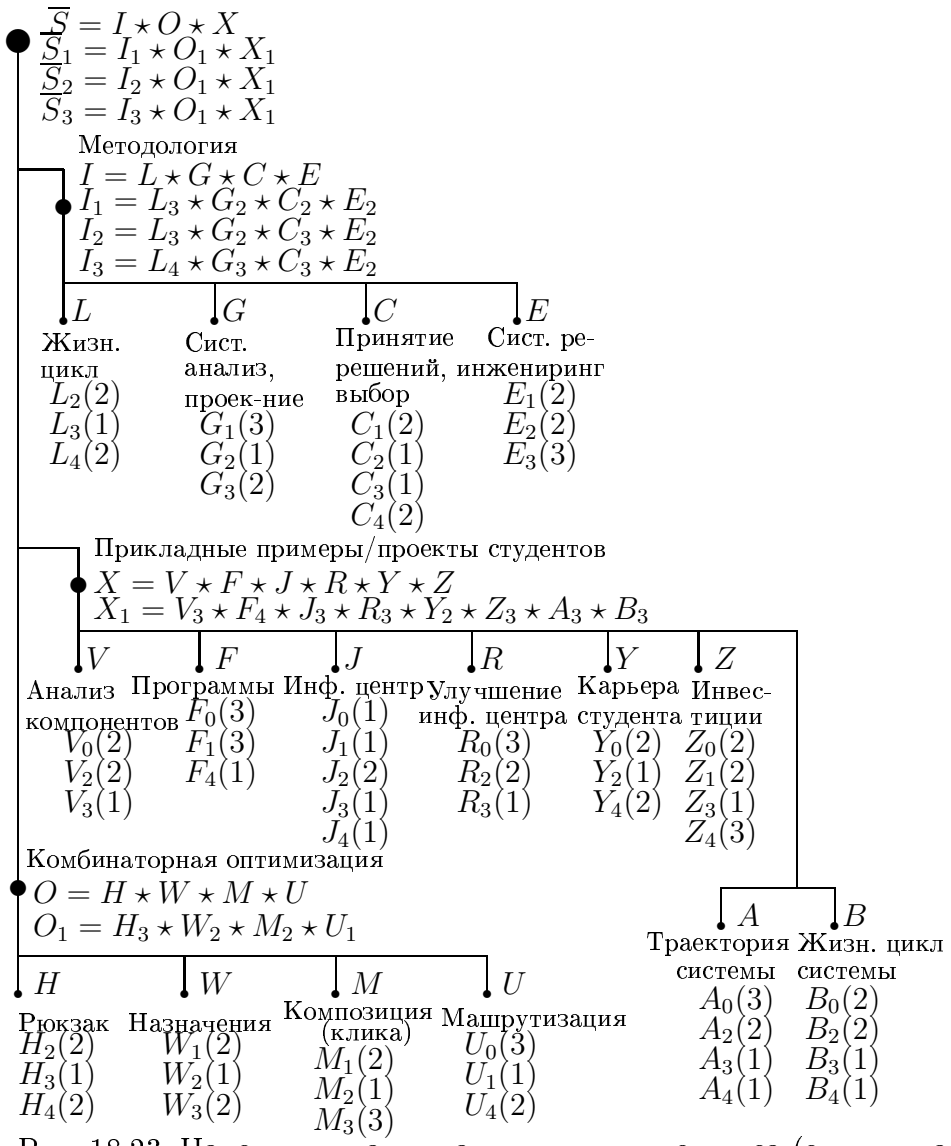


Рис. 18.23. Надструктура для агрегированного курса (структура $\overline{\Lambda}$)

Таблица 18.13. Оценки совместимости

	R_0	R_2	R_3	Y_0	Y_2	Y_4	Z_0	Z_1	Z_3	Z_4	A_0	A_2	A_3	A_4	B_0	B_2	B_3	B_4
V_0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0
V_2	1	3	3	3	3	3	1	2	3	3	0	3	2	2	0	3	2	2
V_3	1	3	3	3	3	3	1	2	3	3	0	1	3	3	0	1	3	3
F_0	0	0	1	3	1	0	3	2	2	0	0	1	1	0	0	1	1	0
F_1	0	2	2	0	3	3	1	3	3	3	0	1	2	2	0	1	2	2
F_4	1	3	3	0	3	3	0	2	3	3	1	1	3	3	1	1	3	3
J_0	0	1	0	0	1	0	3	2	2	0	0	1	1	0	0	1	1	0
J_1	0	2	2	1	1	2	1	3	3	3	0	3	2	1	0	3	2	1
J_2	0	3	2	3	3	3	1	2	3	3	0	3	2	1	0	3	2	1
J_3	0	2	3	3	3	3	1	2	3	3	1	2	3	2	1	2	3	2
J_4	0	2	3	3	3	3	0	2	3	3	1	1	2	3	1	1	2	3
R_1				2	1	1	0	0	1	1	0	1	1	0	0	1	1	0
R_2				2	3	3	1	2	3	3	0	2	3	3	0	2	3	3
R_3				2	3	3	1	2	3	3	0	1	3	3	0	1	3	3
Y_0							3	2	2	0	0	1	1	0	0	1	1	0
Y_2							1	3	3	3	1	2	3	2	1	2	3	2
Y_3							0	2	3	3	0	1	2	3	0	1	2	3
Z_0											0	1	1	0	0	1	1	0
Z_1											0	3	2	1	0	3	2	1
Z_3											1	2	3	2	1	2	3	2
Z_4											0	1	2	3	0	1	2	3
A_0															0	1	1	0
A_2															0	3	2	1
A_3															1	2	3	2
A_4															0	1	2	3

18.5 План студента

В данном разделе приводится построение некоторого базового составного плана студента (в области техники, прикладной математики, вычислительной техники) [21, 336]. Задача построения подобного индивидуального плана использовалась в качестве лабораторной работы в факультетском курсе 'Проектирование систем: структурный подход' в МФТИ (ФРТК, 2004-2008) [22, 345, 350, 353].

На Рис. 18.24 представлена общая базовая структура рассматриваемого плана. (структура, проектные альтернативы DAs, приоритеты DAs указаны в круглых в скобках). Здесь не рассматриваются дополнительные ограничения (например, финансовые, временные). Задача включает следующее:

(1) иерархическое построение плана;

(2) выявление узких мест по DAs/IC;

(3) формирование акций улучшения.

Критерии для висячих вершин рассматриваемой модели представлены в Таблице 18.14. Таблица 18.15a, Таблица 18.15b, Таблица 18.15c содержат оценки DAs. Таблицы 18.16, 18.17, 18.18, 18.19, 18.20 содержат оценки совместимости. При этом, были учтены следующие факторы совместимости:

(а) общие теоретические разделы (например, у курсов);

(б) отношение причинности или следования;

(в) возможные общие приложения;

(г) возможность того, что компоненты составят некую целостностную сущность.

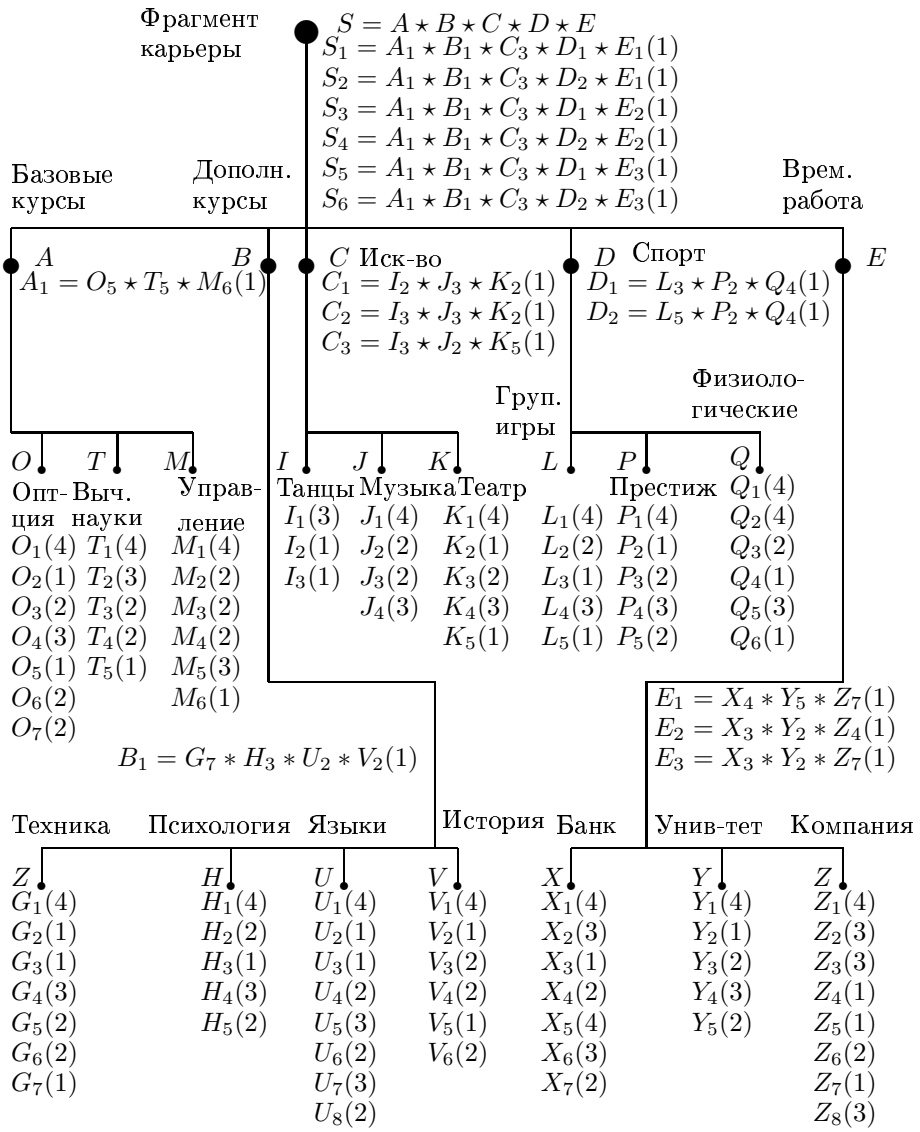


Рис. 18.24. Структура плана карьеры

Таблица 18.14. Критерии для DAs

Критерии	Веса																				
	O	T	M	G	H	U	V	I	J	K	L	P	Q	X	Y	Z	A	B	C	D	E
1.Ст-ть(-)/Зарплата	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2.Возм. встречи полезного лица														3	3	3	4	4	4	4	4
3.Возм. встречи друзей	2	2	2	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
4.Возм. встречи друга/подруги	2	2	2	3	3	3	3	4	4	4		3		1	1	1	2	2	2	2	2
5.Соответствие склонностям	2	2	2	2	2	2	2	2	2	2	4	4	4	3	3	3	3	3	3	3	3
6.Польза для карьеры	3	3	3	4	4	4	4	2	2	2		2	2	4	4	4	4	4	4	4	4
7.Польза для здоровья								2	2	2	4	4	4				3	3	3	3	3
8.Польза для частной жизни				1	1	1	1	3	3	3	3	3	3	4	4	4	2	2	2	2	2

Таблица 18.15а. Оценки для DAs

DAs	Критерии							
	1	2	3	4	5	6	7	8
O_1Нет	0		0	0	0	0		
O_2Вероятность и статистика	1		1	1	4	4		
O_3Анализ решений	1		1	1	2	4		
O_4Имитационное моделирование	1		1	1	1	2		
O_5Многокритер-ное принятие решений	1		1	1	3	4		
O_6Математическое программирование	1		1	1	3	2		
O_7Сети и комб. оптимизация	1		1	1	4	3		
T_1 Нет	0		0	0	0	0		
T_2 Языки программирования и ОС	1		1	1	2	3		
T_3 Теория вычислений	1		1	1	2	5		
T_4 Искусственный интеллект	1		1	1	2	4		
T_5 Чел.-маш. интерфейс	1		1	1	4	4		
M_1Нет	0		0	0	0	0		
M_2Управление производством	1		1	1	3	3		
M_3Управление качеством	1		1	1	3	3		
M_4Инвестиции	1		1	1	2	4		
M_5Маркетинг	1		1	1	2	3		
M_6Управление проектами	1		1	1	4	4		
G_1Нет	0		0	0	0	0		0
G_2Химические технологии	1		1	1	1	4		3
G_3Биотехнология	1		1	1	1	5		1
G_4Механика	1		1	1	1	2		2
G_5Электротехника	1		1	1	3	2		2
G_6Строительство	1		1	1	2	2		3
G_7Программирование	1		1	1	4	3		1
H_1Нет	0		0	0	0	0		0
H_2Эргономика	1		1	1	4	3		1
H_3Когнитивные науки	1		1	3	2	4		1
H_4Теория активности	1		1	2	2	2		2
H_5Эксперименты	1		1	3	2	2		3

Таблица 18.15b. Оценки для DAs

DAs	Критерии							
	1	2	3	4	5	6	7	8
U_1 Нет	0		0	0	0	0		0
U_2 Французский	1		1	3	2	3		2
U_3 Немецкий	1		1	2	3	4		2
U_4 Испанский	1		1	2	2	2		2
U_5 Португальский	2		1	2	1	2		1
U_6 Японский	1		1	2	1	4		2
U_7 Китайский	2		1	2	1	2		1
U_8 Древние	2		1	2	3	3		2
V_1 Нет	0		0	0	0	0		0
V_2 Современная история	1		1	1	3	2		3
V_3 Античная история	1		1	2	2	1		1
V_4 Христианство	1		1	1	2	1		2
V_5 Иудаизм	1		1	2	3	1		2
V_6 Ислам	1		1	0	1	1		1
I_1 Нет	0		0	0	0	0	0	0
I_2 Бальные танцы	2		1	2	2	2	2	2
I_3 Ансамбль	1		1	3	1	1	2	2
J_1 Нет	0		0	0	0	0	0	0
J_2 Классическая	2		1	1	3	2	2	2
J_3 Джаз	1		1	2	1	2	1	2
J_4 Пение	2		1	1	1	1	1	2
K_1Нет	0		0	0	0	0	0	0
K_2Актер	2		1	3	1	4	1	5
K_3Режиссер	4		1	2	3	2	3	3
K_4Техн. работник	1		1	1	2	2	2	1
K_5Автор	0		1	1	3	3	3	3
L_1 Нет	0		0		0		0	0
L_2 Баскетбол	1		1		1		2	2
L_3 Футбол	1		1		2		2	2
L_4 Бег	2		1		1		2	1
L_5 Волейбол	1		1		2		3	2

Таблица 18.15c. Оценки для DAs

DAs	Критерии							
	1	2	3	4	5	6	7	8
P_1 Нет	0		0	0	0	0	0	0
P_2 Теннис	1		1	2	2	2	2	3
P_3 Гольф	3		1	1	1	4	3	3
P_4 Аэроплан	4		1	1	1	3	1	1
P_5 Горные лыжи	3		2	2	2	1	2	2
Q_1Нет	0		0		0	0	0	0
Q_2Карате	2		1		1	2	3	3
Q_3Бокс	2		1		1	2	1	3
Q_4Бег	1		2		3	2	3	2
Q_5Лыжи	2		1		3	1	3	1
Q_6Гимнастика	2		1		1	2	2	2
X_1Нет	0	0	0	0	0	0		0
X_2Операционная работа	1	3	1	2	1	3		2
X_3Инвестиции	1	4	1	2	2	4		4
X_4Страхование	1	3	1	2	1	3		3
X_5Базы данных	2	1	1	1	1	1		1
X_6Коммуникации	2	1	1	1	2	1		1
X_7Мат. моделирование	2	1	2	1	3	4		0
Y_1 Нет	0	0	0	0	0	0		0
Y_2 Моделирование	1	2	1	1	3	3		1
Y_3 Программирование	2	1	1	1	2	2		1
Y_4 Физические эксперименты	2	1	1	1	1	1		1
Y_5 Психологические исследования	1	1	1	3	2	2		2
Z_1 Нет	0	0	0	0	0	0		0
Z_2 Производство	2	1	1	1	1	2		1
Z_3 Сервис	2	1	1	1	1	1		1
Z_4 НИР	1	3	2	1	4	3		1
Z_5 Управление операциями	1	3	2	1	3	3		1
Z_6 Эргономика	1	1	1	2	2	2		3
Z_7 Маркетинг	1	3	1	2	1	3		3
Z_8 Техн. обслуживание	1	1	1	1	1	1		2

Таблица 18.16. Оценки совместимости

	T_1	T_2	T_3	T_4	T_5	M_1	M_2	M_3	M_4	M_5	M_6
O_1	2	2	2	2	2	2	2	2	2	2	2
O_2	2	3	4	4	3	2	3	5	5	5	4
O_3	2	3	3	4	4	2	3	4	5	4	4
O_4	2	5	3	3	5	2	5	3	4	3	4
O_5	2	3	3	5	4	2	4	4	4	4	4
O_6	2	3	4	3	3	2	5	3	5	3	4
O_7	1	2	5	4	3	2	5	3	4	4	4
T_1						2	2	2	2	2	2
T_2						2	4	3	3	3	3
T_3						2	4	3	4	3	3
T_4						2	5	4	4	5	5
T_5						2	4	3	4	4	4

Таблица 18.17. Оценки совместимости

	Y_1	Y_2	Y_3	Y_4	Y_5	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
X_1	2	2	2	2	2	2	2	2	2	2	2	2	2
X_2	2	3	3	3	5	2	3	3	3	3	4	4	3
X_3	2	5	3	3	3	2	4	4	4	3	3	5	3
X_4	2	5	3	3	5	2	4	4	3	3	4	5	5
X_5	2	3	5	3	3	2	4	4	3	4	3	4	3
X_6	2	3	5	3	3	2	4	4	3	4	3	4	3
X_7	2	3	3	3	3	2	4	4	4	4	3	4	3
Y_1						2	2	2	2	2	2	2	2
Y_2						2	5	5	5	5	4	4	3
Y_3						2	3	3	4	5	3	3	3
Y_4						2	4	3	3	3	3	3	5
Y_5						2	3	4	3	3	5	5	4

Таблица 18.18. Оценки совместимости

	P_1	P_2	P_3	P_4	P_5	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
L_1	2	2	2	2	2	2	2	2	2	2	2
L_2	2	3	3	3	3	2	4	4	3	2	2
L_3	2	3	3	3	3	2	3	3	4	4	3
L_4	2	3	3	3	3	2	3	3	3	3	3
L_5	2	3	3	3	4	2	4	4	3	2	3
P_1						2	2	2	2	2	2
P_2						2	4	3	4	3	4
P_3						2	3	3	3	3	3
P_4						2	3	3	3	3	3
P_5						2	3	3	4	3	4

Таблица 18.19. Оценки совместимости

	H_1	H_2	H_3	H_4	H_5	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	V_1	V_2	V_3	V_4	V_5	V_6
G_1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
G_2	2	4	2	2	3	2	4	5	3	3	5	4	3	2	4	3	4	3	3
G_3	2	4	2	2	3	2	3	5	3	3	5	3	3	2	5	3	3	3	3
G_4	2	4	2	2	3	2	3	5	3	3	5	3	3	2	4	4	3	3	3
G_5	2	3	2	2	3	2	3	3	3	3	5	3	3	2	5	3	3	3	3
G_6	2	5	2	2	3	2	4	3	3	4	4	3	4	2	4	4	4	3	4
G_7	2	5	4	4	3	2	4	4	3	3	4	3	4	2	5	3	3	4	3
H_1						2	2	2	2	2	2	2	2	2	2	2	2	2	2
H_2						2	3	4	3	3	5	3	3	2	5	3	3	3	3
H_3						2	5	5	3	3	4	4	5	2	4	4	4	4	4
H_4						2	3	3	3	3	4	3	3	2	4	3	4	4	3
H_5						2	4	4	3	3	4	3	3	2	5	3	4	3	3
U_1														2	2	2	2	2	2
U_2														2	5	3	5	2	2
U_3														2	5	3	5	2	2
U_4														2	5	3	5	2	2
U_5														2	4	3	5	2	2
U_6														2	5	1	1	2	1
U_7														2	4	1	1	1	1
U_8														2	2	5	5	5	5

Таблица 18.20. Оценки совместимости

	B_4	C_3	C_3	C_4	D_4	D_4	E_5	E_5	E_5
A_1	2	3	3	3	3	2	3	3	3
B_1		3	3	5	4	4	5	4	5
C_1					4	4	5	4	5
C_2					5	5	5	4	5
C_3					5	5	5	5	5
D_1							5	5	5
D_2							5	5	5

Составные DAs представлены в Таблице 18.21, а узкие места и акции улучшения представлены в Таблице 18.22. Использованы следующие обозначения по типу для акций улучшения:

(i) генерация идеального решения (1);

(ii) улучшение Парето-эффективного решения (2);

(iii) расширение множества Парето-эффективных решений (3).

Таблица 18.21. Составные DAs

DAs	N
$A_1 = O_5 \star T_5 \star M_6$	$(4;3,0,0,0)$
$B_1 = G_7 \star H_3 \star U_2 \star V_2$	$(4;4,0,0,0)$
$C_1 = I_2 \star J_3 \star K_2$	$(4;2,1,0,0)$
$C_2 = I_3 \star J_3 \star K_2$	$(4;2,1,0,0)$
$C_3 = I_3 \star J_2 \star K_5$	$(4;2,1,0,0)$
$D_1 = L_3 \star P_2 \star Q_4$	$(3;3,0,0,0)$
$D_2 = L_5 \star P_2 \star Q_4$	$(3;3,0,0,0)$
$E_1 = X_4 \star Y_5 \star Z_7$	$(5;1,2,0,0)$
$E_2 = X_3 \star Y_2 \star Z_4$	$(4;3,0,0,0)$
$E_3 = X_3 \star Y_2 \star Z_7$	$(4;3,0,0,0)$
$S_1 = A_1 \star B_1 \star C_3 \star D_1 \star E_1$	$(4;5,0,0,0)$
$S_2 = A_1 \star B_1 \star C_3 \star D_2 \star E_1$	$(4;5,0,0,0)$
$S_3 = A_1 \star B_1 \star C_3 \star D_1 \star E_2$	$(4;5,0,0,0)$
$S_4 = A_1 \star B_1 \star C_3 \star D_2 \star E_2$	$(4;5,0,0,0)$
$S_5 = A_1 \star B_1 \star C_3 \star D_1 \star E_3$	$(4;5,0,0,0)$
$S_6 = A_1 \star B_1 \star C_3 \star D_2 \star E_3$	$(4;5,0,0,0)$

Таблица 18.22. Узкие места и акции улучшения

Составные DAs	Узк. места		Акции	
	DA	IC	(w/r)	Тип
$O_2 \star T_5 \star M_6$		(T_5, O_2)	$3 \Rightarrow 4$	3
$O_4 \star T_4 \star M_6$	T_4		$2 \Rightarrow 1$	1
$G_2 \star H_3 \star U_2 \star V_2$		(H_3, G_2)	$2 \Rightarrow 4(5)$	3
$G_2 \star H_3 \star U_3 \star V_2$		(H_3, G_2)	$2 \Rightarrow 4(5)$	3
$C_1 = I_2 \star J_3 \star K_2$	J_3		$2 \Rightarrow 1$	2
$C_2 = I_3 \star J_3 \star K_2$	J_3		$2 \Rightarrow 1$	2
$C_3 = I_3 \star J_2 \star K_5$	K_5		$2 \Rightarrow 1$	2
$I_2 \star J_2 \star K_2$		(J_2, K_2)	$3 \Rightarrow 5$	3
$D_2 = L_3 \star P_2 \star Q_4$		(L_3, P_2)	$3 \Rightarrow 4$	2
$X_4 \star Y_5 \star Z_4$	X_4		$2 \Rightarrow 1$	3
$X_4 \star Y_5 \star Z_4$	Y_5		$2 \Rightarrow 1$	3
$E_2 = X_3 \star Y_2 \star Z_4$		(X_3, Z_4)	$4 \Rightarrow 5$	1
$E_3 = X_3 \star Y_2 \star Z_7$		(X_2, Z_7)	$4 \Rightarrow 5$	1
$X_3 \star Y_2 \star Z_5$		(X_3, Z_5)	$3 \Rightarrow 4$	3

18.6 Выводы к главе

В данной главе рассмотрены некоторые современные актуальные вопросы преподавания с учетом повышения требований к интеллектуальному уровню подготовки решений и сложности и многодисциплинарности современных задач. В качестве ответов на указанные "вызовы" приведено описание комплексной учебной среды с упором на задачи проектирования и для преподавателей, и для студентов. Кроме того подробно описаны три комплексные задачи из области преподавания: (1) модульное проектирование учебного курса, (2) агрегирование трех модульных учебных курсов, (3) модульное проектирование плана студента. Указанные комплексные задачи и комбинаторные схемы их решений могут быть использованы как некие базовые аналоги. В целом, представляется важным создание специальных компьютерных сред, в рамках которых и преподаватели, и студенты, и группы студентов смогут моделировать, анализировать и проектировать учебные курсы и учебные планы.

Глава 19

Заключение

В представленной работе содержится переход к новым парадигмам в двух направлениях:

1. Технология жизненного цикла продукции/систем: *Базовая парадигма:* проектирование, управление и поддержка жизненного цикла продукции/систем.

Новая парадигма: Выделение уровня поддержки модульных систем в виде набора типовых комбинаторных схем:

(i) построение иерархической модели модульной системы,

(ii) проектирование ("снизу-вверх") модульной системы,

(iii) оценивание модульных систем,

(iv) выявление узких мест в модульных системах,

(v) улучшение/расширение модульных систем,

(vi) много-стадийное проектирование модульных систем (проектирование траекторий систем),

(vii) комбинаторное моделирование эволюции модульных систем и их прогнозирование.

2. Проектирование на основе морфологического анализа:

Базовая парадигма морфологического анализа (и его модификаций): Композиция локальных вариантов (для подсистем) с учетом их бинарной совместимости.

Новая парадигма в форме иерархического морфологического проектирования: Иерархическая композиция локальных вариантов (для подсистем) с учетом многокритериального качества локальных вариантов (или интервальных оценок в виде мультимножеств) и совместимости локальных вариантов (порядковые шкалы).

Многочисленные примеры приложений предложенных новых подходов представляют собой "библиотеку" базовых аналогов (задач-прототипов), которые могут быть использованы для формулирования и решения задач в различных областях, а также в рамках учебных курсов.

Литература

[1] Айзерман М.А., Гусев Л.А., Петров С.В., Смирнова И.М. Динамический подход к анализу структурных представлений как графов (основы графодинамики). I // Автоматика и телемеханика. 1977. N 7. С. 135-151.

[2] Айзерман М.А., Гусев Л.А., Смирнова И.М., Петров С.В. Динамический подход к анализу структурных представлений как графов (основы графодинамики). II // Автоматика и телемеханика. 1977. N 9. 123-136.

[3] Артамонов Е.И. Интерактивные системы. Синтез структур. - М.: Инсвязьиздат, 2010. - 185 с.

[4] Батоврин В.К., Левин М.Ш. Использование метода морфологического проектирования в открытых информационных системах // Информационные технологии и вычислительные системы. 2005. N. 4, С. 47-54.

[5] Воронин А.А., Губко М.В., Мишин С.П., Новиков Д.А. Математические модели организаций. - М.: ЛЕНАНД, 2008. - 360 с.

[6] Гафт М.Г., Миловидов Н.Н., Серов В.И., Гусев Е.Д. Метод принятия решений для выбора рациональных конфигураций автомобиля. В сб.: Проблемы и методы принятия решения в организацилнных системах управления. М.: ВНИИСИ, 1982, С. 77-82.

[7] Глотов В.А., Павельев В.В. Векторная стратификация. - М.: Наука, 1984. - 94 С.

[8] Гордеев Е.Н., Трасцов О.Г. Задача Штейнера. Обзор. // Дискреная математика. 1993. Т. 5, N. 2, С. 3-28.

[9] Дубов Ю.А., Травкин С.И., Якимец В.Н. Многокритериальные модели построения вариантов выбора систем. - М.: Наука, 1986. - 296 С.

[10] Емельянов С.В., Озерной С.В., Ларичев О.И. и др. Выбор рациональных вариантов технлогических схем шахт на основе многих критериев // Горный журнал. 1972. Вып. 5, С. 62-71.

[11] Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. - М.: Наука, 1983. - 423 с.

[12] Камаев В.А., Никитин С.В., Залевский Ф.Я. Поисковое конструирование// Итоги науки и техники. Сер. Техническая кибернетика. - М.: ВИНИТИ. 1986. - Т. 19. - С. 142-189.

[13] Колдасов Г.Д. Оптимизационное решение посредством морфологического синтеза технических систем // Приборостроение, 1986. Вып. 29, С. 3-6.

[14] Краснощеков П.С., Морозов В.В., Федоров В.В. Декомпзиция в задачах проектирования // Изв. АН СССР, сер. Техн. кибернетика. 1979. N. 2. С. 7-17.

[15] Левин М.Ш. Некоторые дискретные оптимизационные задачи распределения ресурсов в вычислительных системах // Дисс. ... канд. техн. наук по спец.: 05.13.02/ - М., МГУ-ВНИИСИ, 1981. - 123 с.

[16] Левин М.Ш. Одна экстремальная задача организации данных // Изв. АН СССР, сер. Техн. кибернетика. 1981. N. 5. С. 103-112.

[17] Левин М.Ш. Применение оптимизационных комбинаторных моделей в автоматизированных системах // - М.: ВНИИТЭМР, 1986. - Вып. 1. - 64 с.

[18] Левин М.Ш. Типовой подход к анализу качества в машиностроении // - М.: ВНИИКИ, 1988. - Вып. 4, 64 с.

[19] Левин М.Ш. О третьей грамотности // Научно-техническая информация, сер. 2. 1995. N. 6. С. 20-30.

[20] Левин М.Ш. О преподавании информационных технологий // Научно-техническая информация, сер. 1. 1996. N. 5. С. 14-23.

[21] Левин М.Ш. Иерархческий комбинаторный поход к планированию карьеры // Научно-техническая информация, сер. 2. 1996. N. 12. С. 22-27.

[22] Левин М.Ш. О преподавании проектирования систем. Информационные технологии и вычислительные системы // Вып. 2, 2007, С. 89-94.

[23] Левин М.Ш., Комбинированная схема формирования стратегии маркетинга // Бизнес информатика, Вып. 2, 2009, С. 42-51.

[24] Левин М.Ш. Пример комбинаторной эволюции и прогнозирования требования к коммуникационных системам. “Информационные процессы”, 2017, вып. 17, ном. 2, С. 92-100.

[25] Левин М.Ш. О комбинаторных моделях поколений беспроводных систем связи. “Информационные процессы”, 2017, вып. 17, ном. 3, С. 172-187.

[26] Левин М.Ш., Михайлов А.А. Диалоговая система для стратификации. Труды ВНИИНМАШ, М.: ВНИИНИМАШ, Вып. 57, 1987. С. 53-59.

[27] Левин М.Ш., Михайлов А.А. Построение системы для группового ранжирования. В сб.: “Модели и методы оптимизации”, М.: ВНИИСИ, Вып. 11, 1987, С. 104-114.

[28] Левин М.Ш., Михайлов А.А. Фрагменты технологии стратификации множества объектов: Препринт. - М.: ВНИИСИ. 1988. - 60 с.

[29] Левин М.Ш., Михайлов А.А. Диалоговая система сравнительной оценки многокритериальных объектов/ 89.01.0045; 50890000275; 89.03. Описание применения - 77с., Специцикация - 3с.// Информационный бюллетень Алгоритмы и программы, 1990. - N. 1.

[30] Левин М.Ш., Пороскун В.И. Комбинаторное планирование разведки нефти и газа. В сб.: Геологическое изучение и использование недр, Вып. 4, Москва, Министерство природных ресурсов Российской Федерации, ЗАО “Геоинформмарк”, 1997. С. 58-63.

[31] Левин М.Ш., Сафонов А.В. Проектирование и перепроектирование конфигурации оборудования коммуникационной системы // Информационные технологии и вычислительные системы. 2006, N. 4, С. 63-73.

[32] Левин М.Ш., Фимин А.В. Комбинаторная схема анализа политических кандидатов // Информационные процессы. 2009, Т. 9, N. 2, С. 83-92.

[33] Макаров И.М., Рахманкулов В.З., Ахрем А.А. Математическая теория виртуализации процессов проектирования и трансфера технологий. - М.: Физматлит, 2012. - 352 с.

[34] Максимов С.П. (Ред.). Месторождения нефти и газа в СССР. Справочник. - М.: Недра, 1979.

[35] Норенков И.П. Основы автоматизированного проектирования. 4-е изд. - М.: Изд. МГТУ им. Н.Э. Баумана, 2009. - 430 с.

[36] Петровский А.Б. Пространства множеств и мультимножеств. - М.: УРСС, 2003.- 246 с.

[37] Попков Ю.С. Вероятностные иерархические структуры (модели формирования структур). Препринт. - М.: ВНИИСИ, 1984. - 59 с.

[38] Соболь И.М., Статников Р.Б. Выбор оптимальых парметров в задачах со многими критериями. - М.: Долфа, 2006. - 175 с.

[39] Стемпковский А.Л., Шепелев В.А., Власов А.В. Системная среда САПР СБИС. - М.: Наука, 1994. - 251 с.

[40] Трахтенгерц Э.А. Компьютерная поддержка формирования целей и стратегий. - М.: Синтег, 2005. - 216 с.

[41] Хачатуров В.Р. Аппроксимационно-комбинаторный метод декомпозиции и композиции систем и ограниченные топологические пространства, решетки, оптимизация// Ж. вычислительной математики и математической физики. 1985. Т. 25, N. 12. С. 1777-1794.

[42] Цурков В.И. Декомпозиция в задачах большой размерности. - М.: Наука, 1981. - 352 с.

[43] Цыгичко В.Н., Попович А.Ю. Синтез иерархических систем управления: Теория и практика. - М.: КРАСАНД, 2012.- 256 с.

[44] Abts D., Felderman B. A guided tour of data-center networking. Commun. of the ACM. 2012. Vol. 55, No. 6, pp. 44-51.

[45] Agrawal A., Klein P., Ravi R. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. SIAM J. on Computing. 1995. Vol. 24, No. 3, pp. 440-456.

[46] Aitkenhead M.J. A co-evolution decision tree classification method. Expert Systems with Applications. 2008. Vol. 34, No. 1, pp. 18-25.

[47] Akyildiz F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey. Computer Networks. 2002. Vol. 38, No. 4, pp. 393-422.

[48] Alekseev V.M. Deciphering algorithm of some classes of monotonic many-valued functions. USSR Computational Mathematics and Mathematical Physics. 1976. Vol. 16, No. 1, pp. 180-189.

[49] Alexandrov M.M., Hussaini M.Y. (eds) Multidisciplinary Design Optimization: State of the Art (Proceedigns in Applied Mathematics Series; No. 80). SIAM, 1997.

[50] Alon N., Awerbuch B., Azar Y., Buchbinder N., Naor J. A general approach to online network optimization problems. ACM Trans. on Algorithms. 2006. Vol. 2, No. 4, pp. 640-660.

[51] Alon N., Fomin F., Gutin G., Krivelevich M., Saurabh S. Spanning directed trees with many leaves. SIAM J. on Discr. Math. 2009. Vol. 23, No. 1, pp. 466-476.

[52] Altshuller G.S. Creativity as Exact Science. The Theory of the Solution of Inventive Problems. Gordon and Breach Science Publishers, London, 1984.

[53] Akatsu T., Halldorsson M.M. On the approximation of largest common subtrees and largest common point sets. Theoretical Computer Science. 2000. Vol. 233, No. 1-2, pp. 33-50.

[54] Amir A., Keselman D. Maximum agreement subtree of a set of evolutionary trees - metrics and efficient algorithms. SIAM J. on Comp. 1997. Vol. 26, No. 6, pp. 1656-1669.

[55] Amir A., Landau G.M., Na J.C., Park H., Park K., Sim J.S. Efficient algorithms for consensus string problems minimizing both distance sum and radius. Theoretical Computer Science. 2011. Vol. 412, No. 39, pp. 5239-5246.

[56] An L., Pung H.K., Zhou L. Design and implementation of a dynamic protocol framework. Computer Communications. 2006. Vol. 29, No. 9, pp. 1309-1315.

[57] Apostolico A. String editing and longest common subsequences. In: Rozenberg G., Salomaa A. (eds) Handbook of Formal Languages, Vol. 2, Springer, 1997. pp. 361-398.

[58] Apostolico A., Guerra C. The longest common subsequence problem revised. Algorithmica. 1987. Vol. 2, No. 1-4, pp. 315-336.

[59] Archetti C., Bertazzi L., Speranza M.G. Reoptimizing the traveling salesman problem. Networks. 2003. Vol. 42, No. 3, pp. 154-159.

[60] Arnold M., Ohlebusch E. Linear time algorithms for generalizations of the longest common substring problem. Algorithmica, 2011. Vol. 60, No. 4, pp. 806-818.

[61] Arroyo J.E.C., Vieira P.S., Vianna D.S. A GRAPS algorithm for the multi-criteria minimum spanning tree problem. Ann. of Oper. Res. 2008. Vol. 159, No. 1, pp. 125-133.

[62] Austello G., Escoffer B., Monnot J., Paschos V. Reoptimization of minimum and maximum traveling salesman's tours. J. of Discr. Algorithms. 2009. Vol. 7, No. 4, pp. 453-463.

[63] Awerbuch B., Azar Y., Bartal Y. On-line generalized Steiner problem. Theoretical Comput. Sci. 2004. Vol. 324, No. 2-3, pp. 313-324.

[64] Ayres R.U., *Technological Forecasting and Long-Time Planning*, McGraw-Hill, New York, 1969.

Версия на русском: Эйрес Р. Научно-техническое прогнозирование и долгосрочное планирование/ Пер. с англ. - М.: Мир, 1971. - 296 с.

[65] Azarm S., Narayanan S. A multiobjective interactive sequential hybrid optimization technique for design decision making. Engineering Optimization, 2000. Vol. 32, pp. 485-500.

[66] Babcock B., Babu S., Datar M., Motwani R., Widom J. Models and issues in data stream systems. In: Proc. of 21st ACM SIGMOD-SIGACT-SUGART Symp. Principles of Database Systems PDOS-2002, Madison, WI, USA, 2002. pp. 1-16.

[67] Bachoore E.H., Boldaender H.L. Convex Recoloring of Leaf-Colored Trees. TR UU-CS-2006-010, Dept. of Inform. & CS, Utrecht Univ., 2006.

[68] Bae S.W., Lee C., Choi S. On exact solutions to the Euclidean bottleneck Steiner tree problem. Inform. Proc. Lett. 2010. Vol. 110, No. 16, pp. 672-678.

[69] Baines P.R., Harris P., Lewis B.R. The political marketing planning process: improvig image and message in strategic target areas. Marketing Intelligence & Planning. 2002. Vol. 20, No. 1, pp. 6-14.

[70] Balasubramaian K., Cellatoglu A. Improvements in home automation strategies for designing apparatus for efficient smart home. IEEE Trans. Consumer Electronics. 2008, Vol. 54, No. 4, pp. 1681-1687.

[71] Balakrishnan A., Magnanti T., Mirchandani P. A dual-based algorithm for multi-level network design. Manag. Sci. 1994. Vol. 40, No. 5, pp. 567-581.

[72] Baldwin C.Y., Clark K.B. Design Rules: The Power of Modularity. Cambridge, MA: MIT Press, 2000.

[73] Baligh H.H. Organization Structures: Theory and Design, Analysis and Prescription. Springer-Verlag, New York, 2006.

[74] Bansal M.S., Fernandez-Baca D. Computing distances between partial rankings. Inform. Proc. Lett. 2009. Vol. 109, No. 4, pp. 238-241.

[75] Bar-Shalom Y., Fortmann T. Tracking and Data Association. Academic Press, New York, 1988.

[76] Bar-Yehuda R., Feldman I., Rawitz D. Improved approximation algorithm for convex recoloring of trees. Theory of Computing Systems. 2008. Vol. 43, No. 1, pp. 3-18.

[77] Bard J.F., Feinberg A. Two-phase methodology for technology selection and system design. IEEE Trans. Eng. Manag. 1989. Vol. 36, No. 1, pp. 28-36.

[78] Barnhart R.K., Hottman S.B., Marshall D.M., Shappee E. (eds) Introduction to Unmanned Aircraft Systems. CRC Press, 2011.

[79] Baronti P., Pillai P., Chook V.W.C., Chessa S., Cotta A., Hu Y.F. Wireless sensor network: A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications. 2007. Vol. 30, No. 7, pp. 1655-1695.

[80] Barthelemy J.-P., Guenoche A., Hudey O. Median linear orders: Heuristics and a branch and bound algorithm. EJOR. 1989. Vol. 42, No. 3, pp. 313-325.

[81] Barton B.A. Searching a bitstream in linear time for the longest substring of any given density. Algorithmica. 2011. Vol. 61, No. 3, pp. 555-579.

[82] Beck M.P., Lin B.W. Some heuristics for the consensus ranking problem. Comp. and Oper. Res. 1983. Vol. 10, No. 1, pp. 1-7.

[83] Belaziz M., Bouras A., Brun J.M. Morphological analysis for product design. Computer-Aided Design, Vol. 32, No. 5, 2000, pp. 377-388.

[84] Belotserkovskii D.L., Vishnevskii V.M. New algorithm for generating a biconnected spanning subgraphs for topological optimization of data networks. Autom. & Remote Control. 1997. Vol. 58, No. 1 (part 2), pp. 88-97.

[85] Bergroth L., Hakonen H., Raita T. A survey of longest common subsequence algorithms. In: Proc. of the Seventh Int. Symp. on String Processing and Information Retrieval SPIRE'00, 2000. pp. 39-48.

[86] Berry V., Nicolas F. Maximum agreement and compatible supertrees. In: Proc. of the 15th Annual Symp. on Combinatorial Pattern Matching CPM-2004, LNCS 3109, Springer, 2004. pp. 205-219.

[87] Bille P. A survey on edit distance and related problems. Theoretical Computer Science. 2005. Vol. 337, No. 1-3, pp. 217-239.

[88] Bilo D., Bockenhauer H.-J., Hromkovic J., Kralovic R., Momke T., Widmayer P., Zych A. Reoptimization of Steiner trees. In: Gudmundsson J. (ed) Proc. of Scandinavian Workshop on Algorithm Theory SWAT'08, LNCS 5124, Springer, 2008. pp. 258-269.

[89] Bilo D., Widmayer P., Zych A. Reoptimization of weighted graph and covering problems. In: Bampis E., Skutella M. (eds) Proc. of 6th Int. Workshop on Approximation and Online Algorithms WAOA'08, LNCS 5426, Springer, 2008. pp. 201-213.

[90] Bilo D., Bockenhauer H.-J., Komm D., Kralovic R., Momke T., Seibert S., Zych A. Reoptimization of the shortest common superstring problem. Algorithmica. 2011. Vol. 61, No. 2, pp. 227-251.

[91] Blackman S., Popoli R. Design and Analysis of Modern Tracking Systems. Artech House Publishers, Boston, MA, 1999.

[92] Blum C., Blesa M.J., Ibanez M.L. Beam search for the longest common subsequence problem. Comp. and Oper. Res. 2009. Vol. 36, No. 12, pp. 3178-3186.

[93] Blum J., Ding M., Thaeler A., Cheng X. Connected dominating set in sensor networks and MANETs. In: Du D.-Z., Pardalos P. (eds) Handbook of Combinatorial Optimization, Kluwer Academic Publishers, 2005. pp. 329-369.

[94] Bockenhauer H.-J., Hromkovic J., Momke T., Widmayer P. On the hardness of reoptimization. In: Geffert V., Karhumaki J., Bertoni A., Preneel B., Navrat P., Bielikova M. (eds) 34th International Conf. on Current Trends in Theory and Practice of Computer Science SOFSEM 2008, LNCS 4910, Springer, 2008. pp. 50–65

[95] Bogart K.P. Preference structures I: distance between transitive preference relations. J. of Math. Soc. 1973. Vol. 3, pp. 49-67.

[96] Booch G. Object Oriented Development. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1991.

[97] Boon B., Park Y. A systematic approach for identifying technology opportunities: keyword-based morphology analysis. Technol. Forecasting & Social Change. 2005. Vol. 72, No. 2, pp. 145-160.

[98] Borg I., Groenen P. Modern Multidimensional Scaling: theory and applications. 2nd ed., Springer, New York, 2005.

[99] Boria N., Paschos V.Ph. Fast reoptimization for the minimum spanning tree problem. J. of Discr. Algorithms. 2010. Vol. 8, No. 3, pp. 296-310.

[100] Borie R.B. Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs. Algorithmica. 1995. Vol. 14, No. 2, pp. 123-137.

[101] Bose P., Czyzowicz J., Gasieniec L., Kranakis E., Krizanc D., Pelc A., Martin M.V. Strategies for hotlink assignments. In: Int. Symp. on Algorithms and Computation (ISAAC'00), LNCS 1969, Springer, 2001. pp. 23-34.

[102] Botafogo R.A., Rivlin E., Shneiderman B. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. ACM Trans. on Information Systems. 1992. Vol. 10, No. 2, pp. 83-110.

[103] Boukerche A., Turgut B., Aydin N., Ahmad M.Z., Boloni L., Turgut D. Routing protocols in ad hoc networks: survey. Computer Networks. 2011. Vol. 55, No. 13, pp. 3032-3080.

[104] Braha D., Maimon O.A. Mathematical Theory of Design: Foundations, Algorithms and Applications. Kluwer, Dordrecht, 1998.

[105] Brandeau M.L., Chiu S.S. An overview of representative problems in location research. Management Science. 1989. Vol. 35, No. 6, pp. 645-674.

[106] Brans J.P., Mareschal B., Vincke Ph. PROMETHEE: A new family of outranking methods in multicriteria analysis. In: Brans J.P. (ed) Operations Research '84, North-Holland, New York, 1984. pp. 477-490.

[107] Brin S., Page L., Motwani R., Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-0120, Computer Science Dept., Stanford Univ., Stanford, CA, 1999.

[108] Brooks R.R., Iyengar S.S. Multi-Sensor Fusion: Fundamentals and Applications with Software. Prentice Hall, Englewood Cliffs, NJ, 1998.

[109] Brooks R.R., Ramanathan P., Sayeed A.M. Distributed target classification and tracking in sensor networks. Proc. of the IEEE. 2002. Vol. 91, No. 8, pp. 1163-1171.

[110] Browning T.P. Applying the design structure matrix to system decomposition and integration problems. IEEE Trans. on Engineering Management. 2001. Vol. 48, No. 3, pp. 292-306.

[111] Buede D.M. Software review. Overview of MCDA software market. J. of Multi-Criteria Decision Analysis. 1992. Vol. 1, No. 1, pp. 59-61.

[112] Buede D.M. The Engineering Design of Systems: Models and Methods, 2nd ed., J.Wiley & Sons, New York, 2009.

[113] Bunke H. On a relation between graph edit distance and maximum common subgraph. Pattern Recogn. Lett. 1997. Vol. 18, No. 8, pp. 689-694.

[114] Bunke H., Shearer K. A graph distance metric based on the maximal common subgraph. Pattern Recogn. Lett. 1998. Vol. 19, No. 3-4, pp. 255-259.

[115] Butenko S., Cheng X., Oliveira C., Pardalos P.M. A new heuristic for the minimum connected dominating set problem on ad hoc wirless networks. In: Recent Developments in Cooperative Control and Optimization. Kluwer Academic Publishers, 2004. pp. 61-73.

[116] Campbell M.I., Cagan J., Kotovsky K. The A-design approach to managing automated design synthesis. Res. in Eng. Des. 2003. Vol. 14, No. 1, pp. 12-24.

[117] Canuto S.A., Resende M.G.C., Ribeiro C.C. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. Networks. 2011. Vol. 38, No. 1, pp. 50-58.

[118] Carden F., Henry R., Jedlicka R. Telemetry Systems Engineering. Artech House, 2002.

[119] Caro Y., West D.B., Yuster R. Connected domination and spanning trees with many leafs. SIAM J. on Discr. Math. 2000. Vol. 13, No. 2, pp. 202-211.

[120] Cela E., The Quadratic Assignment Problem. Kluwer Academic Publishers, Dordrecht, 1998.

[121] Chang R.-S., Chen W.-Y., Wen Y.-F., Hybrid wireless network protocols. IEEE Trans. on Vehicular Technology. 2003. Vol. 52, No. 4, pp. 1099-1109.

[122] Chartier T.P., Krentzer E., Langville A.N., Pedings K.E. Sensitivity and stability of ranking vectors. SIAM J. on Scientific Computing. 2011. Vol. 33, No. 3, pp. 1077-1102.

[123] Chataigner F. Approximating the maximum agreement forest on k trees. Inform. Proc. Lett. 2005. Vol. 93, No. 5, pp. 239-244.

[124] Chekuri C., Even G., Kortsarz G. A greedy approximation algorithm for the group Steiner problem. Discr. Appl. Math. 2006. Vol. 154, No. 1, pp. 15-34.

[125] Chen D., Du D.-Z., Hu X.-D., Lin G.-H., Wang L., Xue G. Approximations for Steiner trees with minimum number of Steiner points. J. of Global Optimization. 2000. Vol. 18, No. 1, pp. 17-33.

[126] Chen W. New algorithm for ordered tree-to-tree correction problem. J. of Algorithms. Vol. 40, No. 2, pp. 135-158.

[127] Chen H., Lynch K.J., Basu K., Hg N.T. Integrating, and activating thesauri for concept-based document retrieval. IEEE Expert. 1993. Vol. 8, No. 2, pp. 25-34.

[128] Chen G., Chen S., Guo W., Chen W. The multicriteria minimum spanning tree problem based genetic algorithm. Inform. Sci. 2007. Vol. 177, No. 22, pp. 5050-5063.

[129] Chen J.-L., Pan S.-W., Chen M.-C., Huang Y.-M. Communication networks: Adaptive radio resource management in an integrated GPRS/UMTS service network. Eur. Trans. on Telecommunications. 2008. Vol. 19, No. 8, pp. 837–845.

[130] Chen H., Zhou C., Huang X., Qing Y., Shi Y. Management of the reconfigurable protocol stack based on SDL for networked control systems. Information Technology Journal. 2010. Vol. 9, No. 5, pp. 849-863.

[131] Cheng X., Huang X., Li D., Wu W., Du D.-Z. A polynomial-time approaximation scheme for the minimum-connected dominating set in ad hoc wireless networks. Networks. 2003. Vol. 42, No. 4, pp. 202-208.

[132] Chiariglione L. The development of an integrated audiovisual coding standard: MPEG. Proc. of the IEEE. 1995. Vol. 83, No. 2, pp. 151-157.

[133] Chiariglione L. MPEG and multimedia communications. IEEE Trans. Circuits Syst. Video Technol. 1997. Vol. 7, No. 2, pp. 5-18.

[134] Chorpa S., Gorres E.R., Rao M.R. Solving the Steiner tree problem on a graph using branch and cut. ORSA J. Comput. 1992. Vol. 4, No. 3, pp. 320-355.

[135] Chorpa S., Tsai C. A branch-and-cut approach for mimimum cost multi-level network design. Discr. Math. 2002. Vol. 242, No. 1-3, pp. 65-92.

[136] Clune J., Mouret J.-B., Lipson H. The evolutionary origins of modularity. Electronic preprint, 17 pp., July 11, 2012. http://arxiv.org/abs/1207.2743 [q-bio.PE]

[137] Coble J., Cook D.J., Holder L.B. Structure discovery in sequentially-connected data streams. Int. J. on Artif. Intell. Tools. 2006. Vol. 15, No. 6, pp. 917-944.

[138] Coello Coello C.A., Christiansen A.D. MOSES: a multiobjective optimization tool for engineering design. Engineering Optimization. 1999. Vol. 31, No. 3, pp.1337-1368.

[139] Cohen D.M., Dalal S.R., Parelius J., Patton G.C. The combinatorial design approach to automatic test generation. IEEE Software. 1996. Vol. 13, No. 5, pp. 83-87.

[140] Conant R.C. Information flows in hierarchical systems. Int. J. of General Systems. 1974. Vol. 1, No. 1, pp. 9-18.

[141] Connor R., Simeoni F., Iakovos M., Moss R. A bounded distance metric for comparing tree structure. Information Systems. 2011. Vol. 36, No. 4, pp. 748-764.

[142] Conradi R., Westfechtel B. Versions models for software configuration management. ACM Comput. Surv. 1998. Vol. 30, No. 2, pp. 232-282.

[143] Cook W.D., Seiford L.M. Priority ranking and consensus formation. Manag. Sci. 1978. Vol. 24, No. 16, pp. 1721-1732.

[144] Cook W.P., Kress M. Ordinal Information and Preference Structures: Decision Models and Applications. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[145] Cook W.D., Seiford L.M., Kress M. A general framework for distance-based consensus in ordinal ranking models. EJOR. 1996. Vol. 96, No. 2, pp. 392-397.

[146] Corcho O., Fernandez-Lopez M., Gomez-Perez A. Methodologies, tools and languages for building ontologies: Where is their meeting point? Data & Knowledge Engineering. 2003. Vol. 46, No. 1, pp. 41-64.

[147] Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 2nd ed., MIT Press and McGraw-Hill, Boston, 2001.

[148] Costa A.M., Cordeau J.-F., Laporte G. Steiner tree problems with profits. INFOR. 2006. Vol. 44, No. 2, pp. 99-115.

[149] Costa A.M., Cordeau J.-F., Laporte G. Fast heuristics for the Steiner tree problem with revenues, budget and hop contraints. EJOR. 2008. Vol. 190, No. 1, pp. 68-78.

[150] Crnkovic I., Larson M. Challenges of component-based development. J. of Syst. and Soft. 2002. Vol. 61, No. 3, pp. 201-212.

[151] Cross N. Engineering Design Methods. 3rd ed., J. Wiley & Sons, London, 2000.

[152] Culler D., Estrin D., Srivastava M., Overview of sensor networks. IEEE Computer. 2004. Vol. 8, pp. 41-49.

[153] Current J., ReVelle C., Cohon J. The hierarchical network design problem. EJOR. 1986. Vol. 27, No. 1, pp. 57-66.

[154] Czyzowicz J., Kranakis E., Krizanc D., Pelc A., Vargas Martin M. Evaluation of hotlink structure for improving web performance. J. Web Eng. 2003. Vol. 1, No. 2, pp. 93-127.

[155] Da Costa P.C.G., Buede D.M. Dynamic decision making: A comparison of approaches. J. of Multi-Criterion Decision Analysis. 2000. Vol. 9, No. 6, pp. 243-262.

[156] Da Silveira G.J.C. Improvement trajectories in operations and technology management: concept, process and content issues. Technology Analysis & Strategic Management. 2002. Vol. 14, No. 2, pp. 227-240.

[157] Dahmus J.B., Gonzalez-Zugasti J.P., Otto K.N. Modular product architecture. Design Studies. 2001. Vol. 22, No. 5, pp. 409-424.

[158] Dai F., Wu J. An expected localized algorithm for connected dominating set formation in ad hoc wireless networks. IEEE Trans. Parallel. Distrib. Syst. 2004. Vol. 15, No. 10, pp. 908-920.

[159] Daskin M.S. Networks and Discrete Location. Models, Algorithms, and Applications. J.Wiley & Sons, New York, 1995.

[160] Davis C.C., Smolyaninov I.I., Milner S.D. Flexible optical wireless links and networks. IEEE Communications Magazine. 2003. Vol. 41, No. 3, pp. 51-57.

[161] Dean J.W., Yoon S.J., Susman G.I. Advanced manufacturing technology and organization structure: Empowerment or subordination? Organization Science. 1992. Vol. 3, No. 2, pp. 203-229.

[162] Dell'Amico M., Maffioli F. Combining linear and non-linear objectives in spanning tree problems. J. of Combinatorial Optimization. 2000. Vol. 4, No. 2, pp. 253-269.

[163] Dinu L.P., Manea F. An efficient approach for the rank aggregation problem. Theoretical Computer Science. 2006. Vol. 359, No. 1-3, pp. 455-461.

[164] Dixon J.R. Design Engineering: Inventiveness, Analysis and Decisions. McGraw-Hill, New York, 1966.

[165] Douieb K., Langerman S. Near-entropy hotlink assignments. Algorithmica. 2010. Vol. 58, No. 2, pp. 221-344.

[166] Douieb K., Langerman S. Dynamic hotlinks. Algorithmica. 2008. Vol. 50, No. 2, pp. 208-222.

[167] Dovier A., Policriti E.G., Rossi G. A uniform axiomatic view of lists, multisets, and sets, and the relevant unification algorithms. Fundamenta Informaticae. 1998. Vol. 36, No. 2/3, pp. 201-234.

[168] Drake D.E., Hougardy S. On approximation algorithms for the terminal Steiner tree problem. Inform. Proc. Lett. 2004. Vol. 89, No. 1, pp. 15-18.

[169] Drescher M., Vetta A. An approximation algorithm for the maximum leaf spanning arborescence problem. ACM Trans. on Algorihtms. 2010. Vol. 6, No. 3, art. 46.

[170] Dror M., Haouari M. Generalized Steiner problems and other variants. J. of Combinatorial Optimization. 2000. Vol. 4, No. 4, pp. 415-436.

[171] Dror M., Haouari M., Chaouachi J. Generalized spanning trees. EJOR. 2000. Vol. 120, No. 3, pp. 583-592.

[172] Du D.-Z., Wang L., Xu B. The Euclidean bottleneck Steiner tree and Steiner tree with minimum number of Steiner points. In: LNCS 2108, Springer, 2011. pp. 509-518.

[173] Du D.-Z., Hu X. *Steiner Tree Problems in Computer Communication Networks*, World Scientific Publishing Co., Singapore, 2008.

[174] Duin C.W., Volgenant A., Voss S. Solving group Steiner problems as Steiner problem. EJOR. 2004. Vol. 154, No. 1, pp. 323-329.

[175] Dulucq S., Touzet H. Analysis of tree edit distance algorithms. In: Baeza-Yates R., Chavez E., Crochemore M. (eds) Proc. 14th Annual Symp. on Combinatorial Pattern Matching CPM 2003, LNCS 2676, Springer, 2003. pp. 83-95.

[176] Easton T., Singireddy A. A large neighborhood search heuristic for the longest common subsequence problem. J. of Heuristics. 2008. Vol. 14, No. 3, pp. 271-283.

[177] Eghe L., Michel C. String similarity measures for ordered sets of documents in information retrieval. Information Processing and Management. 2003. Vol. 38, No. 6, pp. 823-848.

[178] Elemento O., Gascuet O. An exact and polynomial distance-based algorithm to reconstruct single copy tandem duplication trees. In: Baeza-Yates R., Chavez E., Crochemore M. (eds) Proc. 14th Annual Symp. on Combinatorial Pattern Matching CPM 2003, LNCS 2676, Springer, 2003. pp. 96-108.

[179] Elzinga C., Wang H., Lin Z., Kumar Y. Concordance and consensus. Information Sciences. 2011. Vol. 181, No. 12, pp. 2529-2549.

[180] England J.L. Hierarchies, theories, and methodologies. In: Trappl R. (ed) Cybernetics and Systems'88, Kluwer Academic Publishers, 1988. pp. 271-278.

[181] Escoffer B., Milanic M., Paschos V.Th. Simple and fast reoptimizations for Steiner tree problem. Algorithmic Operations Research. 2009. Vol. 4, No. 2, pp. 86-94.

[182] Eswaran K.P., Tarjan R.E. Augmentation problems. SIAM J. on Computing. 1976. Vol. 5, No. 4, pp. 653-665.

[183] Fagin R., Kumar R., Mahdian M., Sivakumar D., Vee E. Comparing partial rankings. SIAM J. Discrete Math. 2006. Vol. 20, No. 3, pp. 628-648.

[184] Farach M., Przytycka T., Thorup M. On the agreement of many trees. Inf. Process. Lett. 1995. Vol. 55, No. 6, pp. 297-301.

[185] Feldman M., Kortsarz G., Nutov Z. Improved approximating algorithms for the directed Steiner forest. In: Proc. of the 20th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), SIAM, Philadelphia, 2009. pp. 922-931.

[186] Fernandez A., Gomez S. Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. J. of Classification. 2008. Vol 25, No. 1, pp. 43-65.

[187] Fernau H., Kneis J., Kratsch D., Langer A., Liedloff M., Raible D., Rossmanith P. An exact algoriothm for the Maximum Leaf Spanning Tree problem. Theor. Computer Science. 2011. Vol. 412, No. 45, pp. 6290-6302.

[188] Ferrer M., Valveny E., Serratosa F., Riesen K., Bunke H. Generalized median graph computation for means of graph embedding in vector space. Pattern Recognition. 2010. Vol. 43, No. 4, pp. 1642-1655.

[189] Finden C.R., Gordon A.D. Obtaining common pruned trees. J. Classif. 1985. Vol. 2, No. 1, pp. 255-276.

[190] Fishburn P.C. Utility Theory for Decision Making. J. Wiley & Sons, New York, 1970.

[191] Floudas C.A. Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications (Topics in Chemical Engineering). Oxford University Press, New York, 1995.

[192] Floudas C.A. Deterministic Global Optimization. Kluwer, Boston, 1999.

[193] Floudas C.A., Pardalos P. (eds) State of the Art in Global Optimization: Computational Methods and Applications. Kluwer, Boston, 1996.

[194] Fogg C.E., Le Gall D.J., Mitchell, J.L., Pennebaker W.B. (eds) MPEG Video. Compression Standard. Kluwer, Boston, 2002.

[195] French M. Invention and Evolution. Design in Nature and Engineering. 2nd ed., Cambridge University Press, Cambridge, 1994.

[196] Fuchs B. A note on the terminal Steiner tree problem. Inform. Proc. Lett. 2003. Vol. 87, No. 4, pp. 219-220.

[197] Fuchs B., Kern W., Molle D., Richter S., Rossmanith P., Wang X. Dynamic programming for minimum Steiner trees. Theory Comput. Syst. 2007. Vol. 41, No. 3, pp. 493-500.

[198] Fuhrmann S., Krumke S.O., Wirth H.-C. Multiple hotlink assignment. In: Brandstadt A., Le V.B. (eds) WG 2001, LNCS 2204, Springer, 2001. pp. 189-200.

[199] Fujie T. An exact algorithm for the maximum leaf spanning tree problem. Computers & Oper. Res. 2003. Vol. 30, No. 13, pp. 1931-1944.

[200] Furnas G.W., Zacks J. Multitrees: Enriching and reusing hierarchical structure. In: Proc. of Int. Conf. Human Factors in Computing Systems CHI'94, Boston, 1994. pp. 330-336.

[201] Gabow H.N. Two algorithms for generating weighted spanning trees in order. SIAM J. on Computing. 1977. Vol. 6, No. 1, pp. 139-150.

[202] Gabow H.N., Galil Z., Spencer T., Tarjan R.E. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica. 1986. Vol. 6, No. 2, pp. 109-122.

[203] Gabriel K., Ehrig H. Modelling evolution of communication platforms and scenarios based on transformations of high-level nets and processes. Theoretical Computer Science. 2012. Vol. 429, pp. 87-97.

[204] Gaeta M., Orciuoli F., Paolozzi S., Salemo S. Ontology extraction for knowledge reuse: The e-learning perspective. IEEE Trans. SMC, Part C. 2011. Vol. 41, No. 4, pp. 798-809.

[205] Gallant J., Maier D., Storer J.A. On finding minimal length superstrings. J. Comput. Syst. Sci. 1980. Vol. 20, No. 1, pp. 50-58.

[206] Gangemi A. Ontology design patterns for semantic Web content. In: Gal Y. et al. (eds) ISWC 2005, LNCS 3729, Springer, 2005. pp. 262-276.

[207] Garey M.R., Johnson D.S. Computers and intractability. The Guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco, 1979.

Русская версия
Гэри М., Джонсон Д., Вычислительные машины и трудно-решаемые задачи/ Пер. с англ. - М.: Мир, 1982. - 416 с.

[208] Garlan D., Krueger C.W., Lerner B.S. TransformGen: Automating the maintenance of structured environments. ACM Trans. on Programming Languages and Systems. 1994. Vol. 16, No. 3, pp. 727-774.

[209] Garud R., Kumaraswamy A., Langlois R. (eds) Managing in the Age of Modularity: Architecture, Networks, and Organizations. J.Wiley& Sons, New York, 2009.

[210] Gavish B. Topological design of telecommunication networks - the overall design problem. EJOR. 1992. Vol. 58, No. 2, pp. 149-172.

[211] Gazis V., Alanistioti N., Merakos L. Metadata design for reconfigurable protocol stacks in system beyond 3G. Wireless Personal Communications. 2006. Vol. 36, No. 1, pp. 1-28.

[212] Gehrke J. Scalable decision tree construction. In: Liu L., Ozsu M.T. (eds) Encyclopedia of Database Systems 2009, Springer, new York, 2009. pp. 2400-2469.

[213] Georgiadis L., Tarjan R.T. Dominator tree certification and independent spanning trees. Electronic preprint, 44 pp., Oct. 31, 2012. http://arxiv.org/abs/1210.8303 [cs.DS]

[214] Gfeller B. Faster swap edge computation in minimum diameter spanning trees. Algorithmica. 2012. Vol. 62, No. 1-2, pp. 169-191.

[215] Ghribi B., Logrippo L. Understanding GPRS: The GSM packet radio service. Computer Networks. 2000. Vol. 34, No. 5, pp. 763-779.

[216] Godor I., Magyar G. Cost-optimal topology planning of hierarchical access networks. Comp. and Oper. Res. 2005. Vol. 32, No. 1, pp. 59-86.

[217] Goemans M.X., Myung Y.-s. A catalog of Steiner tree formulations. Networks. 1993. Vol. 23, No. 1, pp. 19-28.

[218] Goodman R.M., Smyth P. Decision tree design from a communication theory standpoint. IEEE Trans. Information Theory. 1988. Vol. 34, No. 5, pp. 979-994.

[219] Goodman R.M., Smyth P.A. Decision tree design using information theory. Knowledge Acquisition. 1990. Vol. 2, No. 1, pp. 1-19.

[220] Gordon A.D. Constructing dissimilarity measures. J. of Classification. 1990. Vol. 7, No. 2, pp. 257-269.

[221] Gordon A.D. Classification. 2nd ed., Chapman & Hall/CRC, London/Boca Raton, FL, 1999.

[222] Gower J.C. Measures of similarity, dissimilarity, and distance. In: Kotz S., Johnson N.L. (eds) Encyclopedia of Statistical Science, Vol. 5. J. Wiley & Sons, New York, 1985. pp. 397-405.

[223] Griffith W. On consecutive k-out-of-n failure systems and their generalizaitons. In A.P. Basu (ed.), Reliability and Quality Control, Elsevier, 1986. pp. 157-165.

[224] Grossmann I.E. Mixed-integer non-linear programming techniques for the synthesis of engineering systems. Res. in Eng. Design. 1990. Vol. 1, No. 2/3, pp. 205-228.

[225] Gruber T.R. A translation approach to portable ontology specification. Knowledge Acquisition. 1993. Vol. 5, No. 2, pp. 199-220.

[226] Gu P., Sosale S. Product modularalization for life-cycle engineering. Robotics and Computer-Integrated Manufacturing. 1999. Vol. 15, No. 5, pp. 387-401.

[227] Gubko M.V. The search for optimal organizational hierarchies with homogeneous manager cost functions. Autom.&Remote Control. 2006. Vol. 69, No. 1, pp. 89-104.

[228] Guha S., Khuller S. Approximation algorithms for connected dominating sets. Algorithmica. 1998. Vol. 20, No. 4, pp. 179-193.

[229] Guha S., Khuller S. Improved methods for approximating node weighted Steiner trees and connected dominating sets. Information & Computation. 1999. Vol. 150, No. 1, pp. 57-74.

[230] Guillemot S., Nicolas F. Solving the maximum agreement subtree and the maximum compatible tree problems on many bounded degree trees. In: Lewenstein M., Valiente G. (eds) Proc. of the Int. Conf. on Combinatorial Pattern Matching CPM 2006, LNCS 4009, Springer, 2006. pp. 165-176.

[231] Gupta A.P., Birmingham W.P., Siewiorek D.P. Automating the design of computer systems. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. 1993. Vol. 12, No. 4, pp. 473-487.

[232] Gupta A., Pal M. Stochastic Steiner tree without root. In: Caires L., Italiano G.F., Monteiro L., Palamidessi C., Yung M. (eds) Proc. of 32nd Int. Colloquium on Automata, Languages and Programming ICALP 2005. LNCS 3580, Springer, 2005. pp. 1051-1063.

[233] Gupta A., Hajiaghayi M., Kumar A. Stochastic Steiner tree with non-unifirm inflation. In: Charikar M., Jansen K., Reingold O., Rolim J.D.P. (eds) Approx., Randomization, and Comb. Optim.: Algorithms and Techniques, APPROX-RANDOM 2007, LNCS 4627, Springer, 2007. pp. 134-148.

[234] Gusfield D. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, 1999.

[235] Gutner S. Elementary approximation algorithms for prize collecting Steiner tree problems. In: Yang B., Du D.-Z., Wang C.A. (eds) 2nd Annu. Int. Conf. on Combinatorial Optimization and Applications COCOA 2008, LNCS 5165, Springer, 2008. pp. 246-254.

[236] Hack S., Goos G. Copy coalescing by graph recoloring. ACM SIGPLAN Notices. 2008. Vol. 43, No. 6, pp. 227-237.

[237] Hady L., Wozny G. Computer-aided web-based application to modular plant design. Computer Aided Chemical Engineering. 2010. Vol. 28, pp. 685-690.

[238] Haimes Y.Y. Hierarchical holographic modeling. IEEE Trans. SMC. 1981. Vol. 11, No. 9, pp. 606-617.

[239] Haimes Y.Y., Tarvainen K., Shima T., Thadathil J. Hierarchical Multiobjective Analysis of Large-Scale Systems. Hemisphere Pub. Corp., New York, 1990.

[240] Haine G. GSM Networks: Protocols, Terminology and Implementation. Artech House, London, 1999.

[241] Hallett M.T., McCartin C. A faster FPT algorithm for the maximum agreement forest problem. Theory of Computing Systems. 2007. Vol. 41, No. 3, pp. 539-550.

[242] Hamaher H.W., Ruhe G. On spanning tree problem with multiple objectives. Ann. of Oper. Res. 1995. Vol. 52, No. 4, pp. 209-230.

[243] Hamel A.M., Steel M.A. Finding a maximum compatible tree is NP-hard for sequences and trees. Applied Math. Lett. 1996. Vol. 9, No. 2, pp. 55-59.

[244] Hannenhalli S., Pevzner P. Transforming men into mice (polynomial algorithm for genomic distance problem). In: 36th Ann. Symp. on Foundations of Computer Science FOCS 1995, IEEE Computer Society, Milwaukee, Wisconsin, 1995. pp. 581-592.

[245] Harhalakis G., Lin C.P., Nagi R., Proth J.M. Hierarchical decision making in computer integrated manufacturing systems. In: Proc. of the Third Int. Conf. on CIM. CS Society Press. 1992. pp. 15-24.

[246] Harel D. STATECHARTS: A visual formalism for complex systems. Science of Computer Programming. 1987. Vol. 8, No. 3, pp. 231-274.

[247] Harper R. (ed) Inside the smart home. Springer, London, 2003.

[248] Heineman G.T., Councill W.T. Component Based Software Engineering: Putting the Pieces Together. Addison-Wesley, 2001,

[249] Henderson R.M., Clark K. Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. Administrative Science Quarterly. 1990. Vol. 35, pp. 9-30.

[250] Henneberg S.C., O'Shaughnessy N.J., Theory and concept development in political marketing: Issues and an agenda. J. of Political Marketing. 2007, Vol. 6, No. 2-3, pp. 69-90.

[251] Herrera F., Herrera-Viedma E., Verdegay J.L. A rational consensus model in group decision making using linguistic assessments. Fuzzy Sets and Systems. Vol. 88, No. 1, pp. 31-49.

[252] Herrera-Viedma E., Herrera F., Chiclana F. A consensus model for multiperson decision making with different preference structures. IEEE Trans. SMC, Part A. 2002. Vol. 32, No. 3, pp. 394-402.

[253] Hirschberg D.S. Algorithms for the longest common subsequence problem. J. of the ACM. 1977. Vol. 24, No. 4, pp. 664-675.

[254] Hoang D.T., Vitter J.S. Efficient Algorithms for MPEG Video Compression. J. Wiley & Sons, New York, 2002.

[255] Hoang V.T., Sung W.-K. Improved algorithms for maximum agreement and compatible supertrees. Algorithmica. 2011. Vol. 59, No. 2, pp. 195-214.

[256] Holsapple C.W., Joshi K.D. A collaborative approach to ontology design. Commun. of the ACM. 2002. Vol. 45, No. 2, pp. 42-47.

[257] Hong X., Nugent C., Mulvenna M., McClean S., Scotney B., Devlin S. Evidential fusion of sensor data for activity recognition in smart homes. Pervasive and Mobile Computing. 2009. Vol. 5, No. 3, pp. 236-252.

[258] Hoppner F., Klawonn F., Kruse R., Runkler T. Fuzzy Cluster Analysis. J.Wiley & Sons, New York, 1999.

[259] Huang C.C., Kusiak A. Modularity in design of products and systems. IEEE Trans. on Systems, Man, and Cybernetics, Part A. 1998. Vol. 28, No. 1, pp. 66-77.

[260] Hutter F., Hoos H.H., Leyton-Brown K., Stutzle T. ParamILS: an automatic algorithm configuration framework. J. of Artificial Intelligence Research. 2009. Vol. 36, No. 1, pp. 267-306.

[261] Hwang F.K., Richards D.S., Winter P. The Steiner Tree Problem. Elsevier, Amsterdam, 1992.

[262] Imase M., Waxman B.M. Dynamic Steiner tree problem. SIAM J. on Discr. Math. 1991. Vol. 4, No. 3, pp. 369-384.

[263] Inokuchi A., Washio T., Motoda H. Complete mining of frequent patterns from data: Mining graph data. Machine Learning. 2003. Vol. 50, No. 3, pp. 321-354.

[264] Intille S.S., Designing a home of the future. IEEE Pervasive Computing. 2002. Vol. 1, No. 2, pp. 80-86.

[265] Ishikawa K. What is Total Quality Control? The Japannese Way. Prentice Hall, Englewood Cliffs, NJ, 1988.

[266] Itai A., Rodeh M. The multitree approach to reliability in distributed networks. Information and Computation. 1989. Vol. 79, No. 1, pp. 43-59.

[267] Jacobs T. On the complexity of optimal hotlink assignment. Algorithmica. 2012. Vol. 62, No. 3-4, pp. 982-1005.

[268] Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review. ACM Comp. Surveys. 1999, Vol. 31, No. 3, pp. 264-323.

[269] Jansson J., Ng J.H.-K., Sadakane K., Sung W.-K. Rooted maximum agreement supertrees. Algorithmica. 2005. Vol. 43, No. 4, pp. 293-307.

[270] Jayant N. (Ed.). Broadband Last Mile: Access Technologies for Multimedia Communications. CRC Press/Taylor&Francis, 2005.

[271] Jech T. Set Theory. Springer, Berlin, 2002.

[272] Jiang D., Pei J. Mining frequent cross-graph quasi-cliques. ACM Trans. on Knowledge Discovery from Data. 2009. Vol. 2, No. 4, pp. 1-42.

[273] Jiang T., Li M. On the approximation of shortest common supersequences and longest common subsequences. SIAM J. on Computing. 1995. Vol. 24, No. 5, pp. 1122-1139.

[274] Jiang T., Timkovsky V.G. Shortest consistent superstring computable in polynomial time. Theor. Comput. Sci. 1995. Vol. 143, No. 1, pp. 113-122.

[275] Jiang X., Munger A., Bunke H. On median graphs: properties, algorithms, and applications. IEEE Trans. PAMI. 2001. Vol. 23, No. 10, pp. 1144-1151.

[276] Jiang X., Bunke H., Csirik J. Median strings: A review. In: Last M., Kandel A.A., Bunke H. (eds) Data Mining in Time Series Databases. World Scientific, 2004. pp. 173-192.

[277] Johnson J., Iravani P. The multilevel hypernetwork dynamics of complex systems of robot soccer agents. ACM Trans. on Autonomous and Adaptive Systems. 2007. Vol. 2, No. 2, Art. No. 5.

[278] Jones J.C. Design Methods. 2nd ed., J.Wiley & Sons, New York, 1992.

[279] Jones C.E., Sivalingam K.M., Agarwal P., Chen J.C., A survey of energy efficient network protocols for wireless networks. Wireless Networks. 2001, Vol. 7, No. 4, pp. 343-358.

[280] Jonyer I., Cook D.J., Holder L.B. Graph-based hierarchical conceptual clustering. J. of Machine Learning Res. 2011. Vol. 2, pp. 19-43.

[281] Joobbani R., Siewiorek D.P. WEAVER: A knowledge-based routing expert. IEEE Design and Test of Computers. 1986. Vol. 3, No. 1, pp. 12-23.

[282] Jose A., Tollenaere M. Modular and platform methods for product family design: literature analysis. J. of Intelligent Manufacturing. 2005. Vol. 16, No. 3, pp. 371-390.

[283] Kamensky E., Hadar O. Multiparameter method for analysis and selection of motion estimation algorithms for video compression. Multimedia Tools and Applications. 2008. Vol. 38, No. 1, pp. 119-146.

[284] Kaner C., Falk J., Nguyen H.Q. Testing Computer Software. 2nd ed., J.Wiley & Sons, New York, 1999.

[285] Karl H., Willig A. Protocols and Architecture for Wireless Sensor Networks. J.Wiley & Sons, New York, 2007.

[286] Kaplan H., Sharir N. The greedy algorithm for shortest superstrings. Inf. Process. Lett. 2005. Vol. 93, No. 1, pp. 13-17.

[287] Kapsalis A., Rayward-Smith V.J., Smith G.D. Solving the graphical Steiner tree problem using genetic algorithm. J. of the ORS. 1993. Vol. 44, No. 4, pp. 397-406.

[288] Kastner W., Neuschwndtner G., Soucek S., Newman H.M. Communication systems for building automation and control. Proc. of the IEEE. 2005. Vol. 93, No. 6, pp. 1178-1203.

[289] Katz R.H. Toward a unified framework for version modeling in engineering databases. ACM Computing Surveys. 1990. Vol. 22, No. 4, pp. 375-408.

[290] Keeney R.L., Raiffa H. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. J. Wiley & Sons, New York, 1976.

[291] Kellerer H., Pferschy U., Pisinger D. Knapsack problems. Springer, Berlin, 2004.

[292] Kemeny J. Mathematics without numbers. Daedelus. 1959. Vol. 88, pp. 577-591.

[293] Kemeny J.G., Snell J.L. Mathematical Models in the Social Sciences. Reprint, The MIT Press, Cambridge, Mass., 1960.

[294] Kemp C., Tenenbaum J.B. The discovery of structural form. PNAS. 2008. Vol. 105, No. 31, pp. 10687-10692.

[295] Kendall M. Rank Correlation Methods, 3rd ed., Hafner, New York, 1962.

[296] Khuller S. Approximation algoritms for finding highly connected subgraphs. In: Hochbaum D.S. (ed) Approximation Algorithms for NP-hard Problems. PWS Publishing Company, Boston, 1997. pp. 236-265.

[297] Khuller S., Zhu A. The general Steiner star problem. Inf. Proc. Lett. 2002. Vol. 84, No. 4, pp. 215-220.

[298] Kickert W.J.M. Fuzzy Theories on Decision Making. Nijhoff, London. 1978.

[299] Kirci E., Kuban E.K., Cicekli N.K., Automation composition of Web services with the abductive event calculus. Information Sciences. 2010, Vol. 180, No. 19, pp. 3589-3613.

[300] Klein P., Ravi R. A nearly best-possible approximation algorithm for node-weighted Steiner trees. J. of Algorithms. 1995. Vol. 19, No. 1, pp. 104-115.

[301] Kleitman D., West D. Spanning trees with many leafs. SIAM J. on Discr. Math. 1991. Vol. 4, No. 1, pp. 99-106.

[302] Knudsen P.V., Madsen J., Integrating communication protocol selection with hardware/software codesign. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems. 1999. Vol. 18, No. 8, pp. 1077-1095.

[303] Knuth D.E., The Art of Computer Programming. Vol. 2, Seminumerical Algorithms. Addison Wesley, Reading, 1998.

[304] Knuth D., Ratghunathan A. The problem of compatible representatives. SIAM J. on Discr. Math. 1992. Vol. 5, No. 3, pp. 422-427.

[305] Kohonen T. Median strings. Pattern Recogn. Lett. 1985. Vol. 3, No. 5, pp. 309-313.

[306] Kotler P., Overview of political candidate marketing. Advances in Consumer Research. 1975. Vol. 12, pp. 7-18.

[307] Kranakis E., Krizanc D., Shende S. Approximate hotlink assignment. In: Proc. of 12th Ann. Int. Symp. on Algorithms and Computation, LNCS 2223, Springer, 2001. pp. 756-767.

[308] Kranakis E., Krizanc D., Shende S. Approximate hotlink assignment. Inf. Process. Lett. 2004. Vol. 90, No. 3, pp. 121-128.

[309] Kroshnamachari R., Papalambros P. Optimal hierarchical decomposition synthesis using integer programming. Trans. ASME J. Mech. Design. 1997. Vol. 119, No. 4, pp. 440-447.

[310] Kruger H.A., Kearney W.D. Consensus ranking - an ICT security awareness case study. Computers & Security. 2008. Vol. 27, No. 7-8, pp. 254-259.

[311] Kuhn H.W. The Hungarian method for the assignment problems. Nav. Res. Log. 2005. Vol. 52, No. 1, pp. 7-21.

[312] Kuhn P. Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation. Kluwer, Boston, 1999.

[313] Kuppuraju N., Ittimakin P., Mistree F. Design through selection: a method that works. Design Studies. 1985. Vol. 6, No. 2, pp. 91-106.

[314] Kuppuraju N., Ganesan S., Mistree F., Sobieski J.S. Hierarchical decision making in system design. Engineering Optimization. 1985. Vol. 8, No. 3, pp. 223-252.

[315] Kuroshev M.Y., Rugachevskaya E.S., Kalinichenko L.A., Zadorozhny V. SYNTHESIS implementation architecture design by commutative object model mapping. Proc. of the First Int. Workshop on Advances in Databases and Information Systems ADBIS'1994, Moscow, 1994. pp. 187-196.

[316] Kusiak A. Engineering Design: Products, Processes, and Systems. Academic Press, New York, 1999.

[317] Kuznetsov N.A., Levin M.Sh., Vishnevsky V.M. Some combinatorial optimization schemes for multi-layer network topology. In: Electronic Proc. of 17th IMACS World Congress 2005, Paris, France, 2005. Paper T4-I-42-0486.

[318] Laber E. Hotlink assignment on the Web. In: Ahlswede R. et al. (eds) Information Transfer and Combinatorics, LNCS 4123, Springer, pp. 1088-1092, 2006.

[319] Lai Y.-J., Liu T.-Y., Hwang C.-L. TOPSIS for MODM. EJOR. 1994. Vol. 76, No. 3, pp. 486-500.

[320] Langlois R.N., Robertson P.L. Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries. Research Policy. 1992. Vol. 21, pp. 297-313.

[321] Larichev O.I., Moshkovich H.M., Furems E.M. Decision support system 'Class'. In: Brehmer B., Jungerman H., Lourens P., Sevon G. (eds) New Direction in Research on Decision Making. North Holland, Elsevier, Amsterdam, 1986. pp. 305-315.

[322] Le Gall D., MPEG: A video compression standard for multimedia applications. Comm. of the ACM. 1991, Vol. 34, No. 4, pp. 47-58.

[323] Leavens, G.T., Sitaraman, M. (eds) Foundations of Component-Based Systems. Cambridge Univ. Press, 2000.

[324] Lees-Marsment J. The marriage of politics and marketing. *Political Studies*, 2001. Vol. 49, No. 4, pp. 692-713.

[325] Lehman M.M., Ramil J.F. Rules and tools for software evolution planning and management. Annals of Software Engineering. 2001. Vol. 11, pp. 15-44.

[326] Levachkine S., Guzman-Arenas A. Hierarchy as a new data type for quantitative variables. Expert Systems with Applicaitons. 2007. Vol. 32, No. 3, pp. 899-910.

[327] Levenshtein V.I. Binary codes capable of correcting deletions, insertions and reversals. Cybernetics and Control Theory. 1966. Vol. 10, No. 8, pp. 707-710.

[328] Levin M.Sh. A hierarchical hypertext system. Automatic Documentation and Math. Linguistics. 1989. Vol. 23, No. 3, pp. 52-59.

[329] Levin M.Sh. Hierarchical components of human-computer systems. Human Computer Interaction. LNCS, Vol. 753, Berlin: Springer, 1993. pp. 37-52.

[330] Levin M.Sh. Hierarchical design of user interfaces. In: Human Computer Interaction. LNCS, Vol.876, Springer, Berlin, 1994. pp. 140-151.

[331] Levin M.Sh. Typical decision making problems in preparation of business plans. Scientific and Technical Information Processing. 1995. Vol. 22, No. 5, pp. 24-34.

[332] Levin M.Sh. Hierarchical decision making for education in information engineering. In: *Proc. of 7th Ann. Eur. Conf. of EAEEIE*, Oulu, 1996. pp. 301-307.

[333] Levin M.Sh. Hierarchical morphological multicriteria design of decomposable systems. Concurrent Eng.: Res. and Appl. 1996. Vol. 4, No. 2, pp. 111-117.

[334] Levin M.Sh. Improvement of decomposable systems. In: Advances in Concurrent Engineering, Techpub., Toronto, 1996. pp. 319-325.

[335] Levin M.Sh. Towards comparison of decomposable systems. In: Data Science, Classification, and Related Methods, Tokyo, Springer, 1998. pp. 154-161.

[336] Levin M.Sh. Combinatorial Engineering of Decomposable Systems. Spinger, 1998. - 372 pp.

[337] Levin M.Sh. Towards systems engineering education. In: 15th Eur. Meeting on Cybern. and Syst. Res. Vol. 1, Vienna, 2000. pp. 257-262.

[338] Levin M.Sh. System synthesis with morphological clique problem: fusion of subsystem evaluation decisions. Inform. Fusion. 2001. Vol. 2, No. 3, pp. 225-237.

[339] Levin M.Sh. Hierarchical approach for engineering skills acquisition. In: Engineering of Intelligent Systems, LNCS 2070, Berlin: Springer, 2001. pp. 92-100.

[340] Levin M.Sh. Combinatorial evolution of composite systems. In: 16th Eur. Meeting on Cybern. and Syst. Res. Vol. 1, Vienna, 2002. pp. 275-280.

[341] Levin M.Sh. Towards combinatorial analysis, adaptation, and planning of human-computer systems. Applied Intelligence. 2002. Vol. 16, No. 3, pp. 235-247.

[342] Levin M.Sh. Common part of preference relations. Foundations of Computing & Dec. Sciences. 2003. Vol. 28, No. 4, pp. 223-246.

[343] Levin M.Sh. Modular system synthesis: Example for composite packaged software. IEEE Trans. on SMC, Part C. 2005. Vol. 35, No. 4, pp. 544-553.

[344] Levin M.Sh. Composite Systems Decisions. Springer, London, 2006. - 254 pp.

[345] Levin M.Sh. Course 'Design of systems: structural approach', DETC2006-99547. In: ASME Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE 2006), vol. 4a: 18th Int. Conf. on Design Theory and Methodology DTM, Pennsylvania, USA, Sept. 2006. pp. 475-484.

[346] Levin M.Sh. Combinatorial technological systems problems (examples for communication system). In: Proc. of Int. Conf. on Systems Engineering and Modeling ICSEM-2007, Israel, 2007. pp. 24-32.

[347] Levin M.Sh. Towards hierarchical clustering. In: V. Diekert, M. Volkov, A. Voronkov (Eds.), CSR-2007, LNCS 4649, Springer, 2007. pp. 205-215.

[348] M.Sh. Levin Morphological approach to electronic shopping. In: IEEE Int. Conf. Sibircon 2008, Novosibirsk, 2008. pp. 280-285.

[349] Levin M.Sh. Combinatorial optimization in system configuration design. Autom. and Remote Control. 2009. Vol. 70, No. 3, pp. 519-561.

[350] Levin M.Sh. Student research projects in system design. In: Int. Conf. on Computer Supported Education CSEDU-2009, Lisbon, Portugal, 2009. Vol. 2, pp. 67-72.

[351] Levin M.Sh. Towards communication network development (structural systems issues, combinatorial problems). In: IEEE Region 8 Int. Conf. Sibircon 2010, 2010. Vol. 1, pp. 204-208.

[352] Levin M.Sh. Restructuring in combinatorial optimization. Electronic preprint. 11 pp., Febr. 8, 2011. http://arxiv.org/abs/1102.1745 [cs.DS]

[353] Levin M.Sh. Course on system design (structural approach). Electronic preprint, 22 pp., Febr. 19, 2011. http://arxiv.org/abs/1103.3845 [cs.SE]

[354] Levin M.Sh. Towards configuration of applied Web-based information system. Electronic preprint, 13 pp., Aug. 31, 2011. http://arxiv.org/abs/3108.3865 [cs.SE]

[355] Levin M.Sh. Four-layer framework for combinatorial optimization problems domain. Advances in Engineering Software. 2011. Vol. 42, No. 12, pp. 1089-1098.

[356] Levin M.Sh. Selection of user's connection in last mile problem. IEEE Trans. SMC - Part A. 2011. Vol. 41, No. 2, pp. 370-374.

[357] Levin M.Sh. Aggregation of composite solutions: strategies, models, examples. Electronic preprint. 72 pp., Nov. 29, 2011. http://arxiv.org/abs/1111.6983 [cs.SE]

[358] Levin M.Sh. Morphological methods for design of modular systems (a survey). Electronic preprint. 20 pp., Jan. 9, 2012. http://arxiv.org/abs/1201.1712 [cs.SE]

[359] Levin M.Sh. Towards electronic shopping of composite product. Electronic Preprint. 10 pp., March 3, 2012. http://arxiv.org/abs/1203.0648 [cs.SE]

[360] Levin M.Sh. Multiset estimates and combinatorial synthesis. Electronic preprint. 30 pp., May 9, 2012. http://arxiv.org/abs/1205.2046 [cs.SY]

[361] Levin M.Sh. Composition of modular telemetry system with interval multiset estimates. Electronic preprint. 9 pp., July 25, 2012. http://arxiv.org/abs/1207.6651 [cs.SY]

[362] Levin M.Sh. Towards clique-based fusion of graph streams in multi-function system testing. Informatica (Lith.). 2012. Vol. 23, No. 3, pp. 391-404.

[363] Levin M.Sh. Combinatorial synthesis of communication protocol ZigBee with interval multiset estimates. In: 4nd Int. Congress on Ultra Modern Telecomm. & Control Systems ICUMT-2012, St. Petersburg, Russia, 2012. pp. 29-34.

[364] Levin M.Sh. Composite strategy for multicriteria ranking/sorting (methodological issues, examples) Electronic preprint. 24 pp., Nov. 9, 2012. http://arxiv.org/abs/1211.2245 [math.OC]

[365] Levin M.Sh. Towards design of system hierarchy (research survey). Electronic preprint. 36 pp., Dec. 7, 2012. http://arxiv.org/abs/1212.1735 [math.OC]

[366] Levin M.Sh. Towards combinatorial evolution of composite systems. Expert Systems with Applications. 2013. Vol. 40, No. 4, pp. 1342-1351.

[367] Levin M.Sh. Improvement/extension of modular systems as combinatorial reengineering (survey). Electronic preprint, 24 pp., Apr. 17, 2013. Http://arxiv.org/abs/1304.4965 [cs.AI].

[368] Levin M.Sh. Synthesis of MPEG-like standard with interval multiset estimates. In: The Eight Int. Conf. on Digital Telecommunications ICDT 2013, Italy, 2013. pp. 14-19.

[369] Levin M.Sh. Modular design and improvement of the management system in the smart home with the use of interval multiset estimates. J. of Communications Technology and Electronics. 2013. Vol. 58, No. 6, pp. 584-593.

[370] Levin M.Sh. A modular approach to the communication protocol and standard for multimedia information: A review. J. of Communications Technology and Electronics. 2013. Vol. 58, No. 6, pp. 594-601.

[371] Levin M.Sh. Note on combinatorial engineering frameworks for hierarchical modular systems. Electronic preprint. 11 pp., Mar. 29, 2013. http://arxiv.org/abs/1304.0030 [math.OC]

[372] Levin M.Sh. Note on evaluation of hierarchical modular systems. Electronic preprint, 15 pp., May 21, 2013. Http://arxiv.org/abs/1305.4917 [cs.AI].

[373] Levin M.Sh. Towards detection of bottlenecks in modular systems. Electronic preprint, 12 pp., June 1, 2013. Http://arxiv.org/abs/1306.0128 [cs.AI].

[374] Levin M.Sh. Towards multistage design of modular systems. Electronic preprint, 13 pp., June 19, 2013. Http://arxiv.org/abs/1306.4635 [cs.AI].

[375] Levin M.Sh. Towards integrated glance to restructuring in combinatorial optimization. Electr. prepr., 31 pp., Dec. 20, 2015, http://arxiv.org/abs/1512.06427 [cs.AI]

[376] Levin M.Sh. Note on evolution and forecasting of requirements: communication example. Electr. prepr., 8 pp., May 22, 2016, http://arxiv.org/abs/1705.07558 [cs.NI]

[377] Levin M.Sh. Towards combinatorial modeling of wireless technology generations. Electr. prepr., 20 pp., Aug. 29, 2017, http://arxiv.org/abs/1708.08996 [cs.NI]

[378] Levin M.Sh. An example of combinatorial evolution and forecasting of requirements to communication systems. J. of Commun. Technol. Electr., 2017, Vol. 62, No. 12, pp. 1499–1505.

[379] Levin M.Sh. Combinatorial framework for planning of geological exploration. Electr. prepr., 14 pp., Jan. 22, 2018, http://arxiv.org/abs/1801.07229 [cs.AI]

[380] Levin M.Sh. On combinatorial models of generations of wireless communication systems. J. of Commun. Technol. Electr., 2018, Vol. 63, No. 6, pp. 667–679.

[381] Levin M.Sh., Kaganov Yu.T. Hierarchical design of vibration conveyor. In: Proc. of Intl. Conf. on "Information Technology in Design", Moscow, 1996. pp. 164-169.

[382] Levin M.Sh., Nisnevich M.L. Combinatorial scheme for management of life cycle: Example for concrete macrotechnology. J. of Intell. Manuf. 2001. Vol. 12, No. 4, pp. 393-401.

[383] Levin M.Sh., Sokolova L.V. Hierarchical combinatorial planning of medical treatment. Comp. Meth. and Progr. in Biomedicine. 2004. Vol. 73, No. 1, pp. 3-11.

[384] Levin M.Sh., Firer M. Hierarchical morphological design of immunoassay technology. Computers in Biology and Medicine. 2005. Vol. 35, No. 3, pp. 229-245.

[385] Levin M.Sh., Danieli M.A. Hierarchical decision making framework for evaluation and improvement of composite systems (example for building). Informatica. 2005. Vol. 16, No. 2, pp. 213-240.

[386] Levin M.Sh., Last M. Design of test inputs and their sequences in multi-function system testing. Applied Intelligence. 2006. Vol. 25, No. 1, pp. 105-124.

[387] Levin M.Sh., Khodakovskii I.A., Structural composition of the telemetry system. Automation and Remote Control. 2007. Vol. 68, No. 9, pp. 1654-1661.

[388] Levin M.Sh., Vishnitskiy R.O. Towards morphological design of GSM network. Inform. Processes. 2007. Vol. 7, No. 2, pp. 183-190. http://www.jip.ru/2007/183-190-2007.pdf

[389] Levin M.Sh., Merzlyakov A.O. Composite combinatorial scheme of test planning (example for microprocessor systems). In: IEEE Region 8 Int. Conf. "Sibircon-2008", Novosibirsk, 2008, pp. 291-295.

[390] Levin M.Sh., Kruchkov O., Hadar O., Kaminsky E. Combinatorial systems evolution: Example of Standard for multimedia information. Informatica. 2009. Vol. 20, No. 4, pp. 519-538.

[391] Levin M.Sh., Sharov S.Yu. Hierarchical morphological design of Web-hosting system. Int. J. of Integrated Design and Process Science. 2009. Vol. 13, No. 1, pp. 1-14.

[392] Levin M.Sh., Leus A.V. Configuration of integrated security system. In: 7th IEEE Int. Conf. INDIN 2009, Cardiff, UK, 2009. pp. 101-105.

[393] Levin M.Sh., A. Andrushevich, R. Kistler, A. Klapproth Combinatorial evolution of ZigBee protocol. In: IEEE Region 8 Int. Conf. Sibircon 2010, 2010. Vol. 1, pp. 314-319.

[394] Levin M.Sh., Fimin A.V. Configuration of alarm wireless sensor element. 2nd Int. Congress ICUMT-2010, Moscow, 2010. pp. 924-928.

[395] Levin M.Sh., Petukhov M.V. Multicriteria assignment problem (selection of access points). In: Proc. of 23rd Int. Conf. IEA/AIE 2010, "Trends in Applied Intelligent Systems", LNCS 6097, part II, Springer, Cordoba, 2010. pp. 277-287.

[396] Levin M.Sh., Petukhov M.V. Connection of users with a telecommunications network: multicriteria assignment problem. J. of Communications Technology and Electronics. 2010. Vol. 55, No. 12, pp. 1532-1542.

[397] Levin M.Sh., Safonov A.V. Towards modular redesign of networked system. 2nd Int. Congress ICUMT-2010, Moscow, 2010. pp. 109-114.

[398] Levin M.Sh., Andrushevich A., Klapproth A. Improvement of building automation system. In: LNCS 6704, Part II, Springer, 2011. pp. 459-468.

[399] Levin M.Sh., Nuriakhmetov R.I. Multicriteria Steiner tree problem for communication network. Electronic preprint, 11 pp., Febr. 12, 2011. http://arxiv.org/abs/1102.2524 [cs.DS]

[400] Levin M.Sh., Safonov A.V. Improvement of regional telecommunications networks. J. of Communications Technology and Electronics. 2011. Vol. 56, No. 6, pp. 770-778.

[401] Levin M.Sh., Zamkovoy A.A. Multicriteria Steier tree with the cost of Steiner vertices. J. of Communications Technology and Electronics. 2011. Vol. 56, No. 12, pp. 1527-1542.

[402] Levin M.Sh., Fimin A.V. Design of modular wireless sensor. Electronic preprint. 7 pp., March 9, 2012. http://arxiv.org/abs/1203.2031 [cs.SE]

[403] Levitin G. Linear multi-state sliding window systems. IEEE Trans. Relaibility. 2003. Vol. 52, No. 2, pp. 263-269.

[404] Liang W. Constructing minimum-energy broadcast trees in ad hoc wireless networks. In: Proc. of 3rd MOBIHOC, ACM, New York, 2002. pp. 112-122.

[405] Lin G.-H., Xue G. Steiner tree problem with minimum number of Steiner points and bounded edge-length. Inform. Proc. Lett. 1999. Vol. 69. No. 2, pp. 53-57.

[406] Lin G.-H., Xue G. On the terminal Steiner tree problem. Inform. Proc. Lett. 2000. Vol. 84, No. 2, pp. 103-107.

[407] Lipson H., Pollack J.B., Suh N.P. On the origin of modular evolution. Evolution. 2002. Vol. 56, No. 8, pp. 1549-1556.

[408] Ljubic I., Weiskircher R., Pferschy U., Klau G., Mutzel P., Fischetti M. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. Math. Programming. Ser. B. 2006. Vol. 105, No. 2-3, pp. 427-449.

[409] Lopez M.F., Gomez-Perez A., Sierra J.P., Sierra A.P. Building a chemical ontology using methodology and the ontology design environment. IEEE Intelligent Systems. 1999. Vol. 14, No. 1, pp. 37-46.

[410] Lu H., R Ravi R. The power of local optimization: Approximation algorithms for maximum leaf spanning tree. In: Proc. of the Annual Allerton Conf. on Communicaiton Control and Computing, USA, Univ. of Illinois, 1992. Vol. 30, pp. 533-542.

[411] Lu H., Ravi R. Approximating maximum leaf spanning trees in almost linear time. J. of Algorithms. 1998. Vol. 29, No. 1, pp. 132-141.

[412] Lust T., Teghem J. The multiobjective multidimensional knapsack problem: a survey and a new approach. Electronic preprint, 2011. http: //arxiv.org/abs/1007.4063.v1

[413] Magnanti T.L., Wolsey L.A. Optimal trees. In: Ball M.O., et al. (eds) Handbooks in OR & MS, North-Holland, Amsterdam, 1995. Vol. 7, pp. 503-615.

[414] Magnani M., Rossi L. The ml-model for multi-layer social networks. In: ASONAM, IEEE Computer Society, 2011. pp. 5-12.

[415] Magnani M., Rossi L. Multi-stratum networks: toward a unified model of on-line identities. Electronic preprint, 18 pp., Nov. 1, 2012. http://arxiv.org/abs/1211.0169 [cs.SI]

[416] Maier D. The complexity of some problems on subsequences and supersequences. J. of the ACM. 1978. Vol. 25, No. 2, pp. 322-336.

[417] March J.G., Simon H.A. Organizations. 2nd ed., Wiley-Blackwell, 1993.

[418] Marianov V., Serra D. Hierarchical location-allocation models for congested systems. EJOR. 2001. Vol. 135, No. 1, pp. 196-209.

[419] Martello S., Toth P. Knapsack Problem: Algorithms and Computer Implementation. J. Wiley & Sons, New York, 1990.

[420] Marzal A., Vidal E. Computation of normalized edit distance and applications. IEEE Trans. PAMI. 1993. Vol. 15, No. 9, pp. 926-932.

[421] Matichin R., Peleg D. Approximation algorithm for hotlink assignment in the greedy model. Theor. Comput. Sci. 2004. Vol. 383, No. 1, pp. 102-110.

[422] Mauser G.A. Political Marketing: An Approach to Compaign Strategy. Praeger, New York, 1983.

[423] McDermott J. R1: A rule-based configurer of computer systems. Artificial Intelligence. 1982. Vol. 19, No. 2, pp. 39-88.

[424] McKeenly P.K., Sadjadi S.M., Kasten E.P., Cheng B.H.C. Composing adaptive software. IEEE Computer. 2004. Vol. 37, No. 7, pp. 56-64.

[425] Mesarovic M.D., Macko D., Takahara Y. Theory of Hierarchical, Multilevel Systems. Academic Press, New York and London, 1970.

[426] Miettinen K. Nonlinear Multiobjective Optimization. Kluwer, Boston, 1999.

[427] Minieka E. Optimization Algorithms for Networks and Graphs. Marcel Dekker, New York, 1978.

[428] Mirchandani P.B., Francis R.L. (eds) Discrete Location Theory. J.Wiley & Sons, New York, 1990.

[429] Mirkin B.G. Group Choice. Halsted Press, 1979.

[430] Mirkin B.G., Clustering for Data Mining: A Data Discovery Appoach. Chapman&Hill/CRC, NJ. 2005.

[431] Mirkin B.G., Chernyi L.B. On measurement of distance between partition of a finite set of units. Automat. and Remote Control. 1970, Vol. 31, pp. 786-792.

[432] Mirkin B., McMorris F.R., Robets F.S., Rzhetsky A. (eds) Mathematical Hierarchies and Biology, DIMACS Ser. in Discrete Math. and Theoretical Comput. Sci., AMS, Providence, 1996.

[433] Mishin S. Optimal organizational hierarchies in firms. J. of Business Economics and Management. 2007, Vol. 8, No. 2, pp. 79-99.

[434] Mistree F., Muster D. The decision support problem technique for design. In: ASSE Annual Conference Proceedings, Cincinnati, 1986. pp. 117-126.

[435] Mitchell J.L., Pennebaker W.B., Fogg C.E., D.J. Legall (eds.), MPEG Video Compression Standard. Chapman&Hall, London, 1996.

[436] Molnar M., Bellabas A., Lahoud S. The cost optimal solution of the multi-constrained multicast routing problem. Computer Networks. 2012, Vol. 56, No. 13, pp. 3136-3149.

[437] Moran S., Snir S. Efficient approximation of convex recoloring. J. of Comput. and System Sci. 2007. Vol. 73, No. 7, pp. 1078-1089.

[438] Moran S., Snir S. Convex recoloring of strings and trees: Definitions, hardness results and algorithms. J. of Comput. and System Sci. 2008. Vol. 74, No. 5, pp. 850-869.

[439] Moses J. Architecting engineering systems. In: van de Poel I., D Goldberg D.E. (eds) Philosophy and Engineering, Springer, 2010. pp. 275-284.

[440] Mousavi S.R., Tabataba F. An improved algorithm for the longest common subsequence problem. Comp. and Oper. Res. 2012. Vol. 39, No. 3, pp. 512-520.

[441] Muhanna W.A., Pick R.A. Meta-modeling concepts and tools for model management: A systems approach. Management Science. 1994. Vol. 40, No. 9, pp. 1093-1123.

[442] Muja M., Lowe D.G. Fast approximate nearest neighbors with automatic algorithm configuration. In: Int. Conf. on Computer Vision Theory and Applications VISSAPP'09, 2009. pp. 331-340.

[443] Murhammer M.W., Lee K.-K., Motallebi P., Borghi P., Wozabal K. IP Network Design Guide, IBM Red Book, 1999.

[444] Mussen M.A. Dimensions of knowledge sharing and reuse. Computers and Biomedical Research. 1992. Vol. 25, pp. 435-467.

[445] Morgan C.L. Emergent Evolution. Northgate and Williams, London, 1927.

[446] Narayanan S., Azarm S., On improving multiobjective genetic algorithms for design optimization. Structural Optimization. 1999. Vol. 18, pp. 146-155.

[447] Nederlof J. Fast polynomial-space algorithms using inclusion-exclusion (improving on Steiner tree and related problems). Algorithmica. 2013, Vol. 65, No. 4, pp. 868-884.

[448] Nemhauser G.L., Wolsey L.A. Integer and Combinatorial Optimization. J.Wiley & Sons, New York, 1988.

[449] Newman B.I. The role of marketing in politics. J. of Political Marketing. 2002. Vol. 1, pp. 1-5.

[450] Niamanesh M., Jalili R. DRAPS: A framework for dynamic reconfigurable protocol stacks. J. of Information Science and Engineering. 2009. Vol. 25, pp. 827-841.

[451] Nicolas F., Rivals E. Complexities of centre and median string problems. In: Baeza-Yates R., Chavez E., Crochemore M.(eds) Proc. 14th Annual Symp. on Combinatorial Pattern Matching CPM 2003, LNCS 2676, Springer, Mexico, 2003. pp. 315-327.

[452] Noltmeier H., Wirth H.-C., Krumke S.O. Network design and improvement. ACM Computing Surveys. 1999. Vol. 32(3es), Art. No. 2.

[453] Norris M., Davis R.E., Pengelly A. Component-Based Network Systems Engineering. Artech House Publishers, Boston, MA, 2000.

[454] Noy N.F., Hafner C. The state of the art in ontology design: A survey and comparative review. AI Magazine. 1997. Vol. 18, No. 3, pp. 53-74.

[455] Noy N., Musen M. PROMPT: Algorithm and tool for automated ontology merging and alignment. In: Proc. AAAI 2000, 2000. pp. 450-455.

[456] Noy N., Doan A.H., Halevy A.Y. Semantic integration. AI Magazine. 2005. Vol. 26, No. 1, pp. 7-10.

[457] Obreque C., Marianov V., Rios M. Optimal design of hierarchical network with free main path extremes. Oper. Res. Lett. 2008. Vol. 36, No. 3, pp. 366-371.

[458] Obreque C., Donoso M., Gutierrez G., Marianov V. A branch and cut algorithm for the hierarchical network design problem. EJOR. 2010. Vol. 200, No. 1, pp. 28-35.

[459] Oh H., Chae K. An energy-efficient sensor routing with low latency, stability for smart home network. Int. J. of Smart Home. 2007, Vol. 1, No. 2, pp. 71-82.

[460] Osborne M.J., Rubinstein A. A Course in Game Theory. MA: MIT Press, 1994.

[461] Osger Z.J. Reasoning with quantitativd preferences to develop optimal component-based systems. In: Proc. of the 2013 Int. Conf. on Softwae Engineering. IEEE Press, 2013. pp. 1456-1458.

[462] Osteen R.E., Tou J.T. A clique-detection algorithm based on neighbourhoods in graphs. Int. J. of Comput. and Inform. Sciences. 1973. Vol. 2, No. 4, pp. 257-268.

[463] Osuagwu L. Political marketing: conceptualization, dimensions and research agenga. Marketing Intelligence & Planning. 2008. Vol. 26, No. 7, pp. 793-810.

[464] Otto K.N., Wood K.L. Product evolution: a reverse engineering and redesign methodology. Res. in Eng. Des. 1998. Vol. 10, No. 4, pp. 226-243.

[465] Ozer M. A survey of new product evolution models. The Intl. J. of Prod. Innovation Manag. 1999. Vol. 16, No. 1, pp. 77-94.

[466] Padberg M., Rijal M.P. Location, Scheduling, Design and Integer Programming. Boston: Kluwer Academic Publishers, 1996.

[467] Pahl G., Beitz W. Engineering Design. A Systematic Approach. The Design Council & Springer, London, 1988.

[468] Para J., Hossain M.A., Uribarren A., Jacob E., Saddik A.E. Flexible smart home architecture using device profile for web services: a peer-to-peer approach. Int. J. of Smart Home. 2009, Vol. 3, No. 2, pp. 39-56.

[469] Pardalos P.M., Wolkowicz H. (eds) Quadratic Assignment and Related Problems. AMS, Providence, 1994.

[470] Pareto V. Mannual of Political Economy. (English translation), A. M. Kelley Publishers, New York, 1971.

[471] Parra-Hernandez R., Dimopoulos N. A new heuristic for solving the multichoice multidimensional knapsack problem. IEEE Trans. SMC, Part A. 2002. Vol. 35, No. 5, pp. 708-717.

[472] Pei J., Jiang D., Zhang A. On mining cross-graph quasi-cliques. In: Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining, 2005. pp. 228-238.

[473] Peine A. Understanding the dynamics of technological configurations: A conceptual framework and the case of smart home. Technological Forecasting and Social Change. 2009. Vol. 76, No. 3, pp. 396-409.

[474] Pereira F. The MPEG-4 Book. Wiley, New York, 2002.

[475] Perkowitz M., Etzioni O. Towards adaptive web sites: conceptual framework and case study. Computer Networks. 1999. Vol. 31, No. 11-16, pp. 1245-1258.

[476] Pettie S., Ramachandran V. An optimal minimum spanning tree algorithm. J. of the ACM. 2002. Vol. 49, No. 1, pp. 16-34.

[477] Pettie S., Ramachandran V. A randomized time-work optimal parallel algorithm for finding a minimum spanning forest. SIAM J. on Computing. Vol. 31, No. 6, pp. 1876-1895.

[478] Phillips C., Warnow T.J. The asymmetric tree - a new model for building consensus tree. Discrete Applied Mathematics. 1996. Vol. 71, No. 1-3, pp. 311-335.

[479] Pirkul H., Current J., Nagarajan V. The hierarchical network design problem: A new formulation and solution procedures. Transportation Science. 1991. Vol. 25, No. 3, pp. 175-182.

[480] Punakivi M., Yriola H., Holmstrom J. Solving the last mile issues: reception box or delivery box? Int. J. of Physical Distribution & Logistics Management. 2001. Vol. 31, No. 6, pp. 427-439.

[481] Ramirez-Marquez J.E., Sauser B.J. System development planning via system maturity optimization. IEEE Trans. on Engineering Management. 2009. Vol. 56, pp. 533-548.

[482] Rasoul Mousavi S., Tabataba F. An improved algorithm for the longest common subsequence problem. EJOR. 2012. Vol. 39, No. 3, pp. 512-520.

[483] Rashidi P., Cook D.J., Keeping the resident in the loop: Adapting the smart home to the user. IEEE Trans. SMC, Part A. 2009, Vol. 39, No. 5, pp. 949-959.

[484] Rasmussen J. The role of hierarchical knowledge representation in decision making and system management. IEEE Trans. SMC, 1985. Vol. 15, No. 2, pp. 234-243.

[485] Raymond J.W., Gardiner E.J., Willett P. RASCAL: calculation of graph similarity using maximal common edge subgraph. The Computer J. 2002. Vol. 45, No. 6, pp. 631-644.

[486] Ritchey T. Problem strucutring using computer-aided morphological analysis. J. of the ORS. 2006. Vol. 57, No. 7, pp. 792-801.

[487] Ritchey T. Wicked Problems/Social Messes: Decision Support Modelling with Morphological Analysis, Springer, Berlin, 2012.

[488] Robles-Kelly A., Hancock E.R. String edit distance, random walks and graph matching. Int. J. of Pattern Recognition and Artificial Intelligence. 2004. Vol. 18, No. 3, pp. 315-327.

[489] Rodrigues E.M., Sagot M.F., Wakabayashi Y. The maximum agreement forest problem: Approximation algorithms and computational experiments. Theoretical Computer Science. 2007. Vol. 374, No. 1-3, pp. 91-110.

[490] Rosenwein M.B., Wong R.T. Constrained Steiner tree problem. EJOR. 1995. Vol. 81, No. 2, pp. 430-439.

[491] Rouse W.B. A theory of enterprise transformation. Systems Engineering. 2005. Vol. 8, No. 4, pp. 279-295.

[492] Roy B. Multicriteria Methodology for Decision Aiding. Kluwer Academic Publishers, Dordrecht, 1996.

[493] Rutti O., Schiper A. A predicate-based approach to dynamic protocol update in group communication. In: Int. Parallel and Distributed Processing Sypm. IPDPS 2008, 2008. pp. 1-12.

[494] Qi H., Iyengar S.S., Chakrabarty K. Distributed sensor networks - a review of recent research, J. of the Franklin Institute. 2001. Vol. 338, No. 6, pp. 655-668.

[495] Quenum J.G., Aknine S., Shehory O., Honiden S. Dynamic protocol selection in open and heterogeneous systems. In: IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology, 2006. pp. 333-341.

[496] Saaty T.L. The Analytic Hierarchy Process. MacGraw-Hill, New York, 1988.

[497] Sabin D., Weigel R. Product configuration frameworks - a survey, IEEE Intell. Syst. & Their Appl. 1998. Vol. 13, No. 4, pp. 42-49.

[498] Sahal D. Patterns of Technological Innovations. Addison-Wesley, Reading, Mass., 1981.

[499] Sahni S. Approximate algorithms for 0/1 knapsack problem. J. of the ACM. 1975. Vol. 22, No. 1, pp. 115-124.

[500] Sahni S., Horowitz E. Combinatorial problems: reducibility and approximation. Oper. Res. 1978. Vol. 26, No. 5, pp. 718-759.

[501] Sanchis L.A. Experimental analysis of heuristic algorithms for the dominating set problem. Algorithmica. 2002. Vol. 33, No. 1, pp. 3-18.

[502] Sasha D., Zhang K. Simple fast algorithms for the editing distance between trees and related problems. SIAM J. on Computing. 1989. Vol. 18, No. 6, pp. 1245-1262.

[503] Savigny H., Ontology and epistemology in political marketing: Keeping it real? J. of Political Marketing. 2007. Vol. 6, No. 2-3, pp. 33-47.

[504] Scammell M. Political marketing: Lessons for political science. Political Studies. 1999. Vol. 47, No. 4, pp. 718-739.

[505] Scarelli A., Narula S.C. A multicriteria assignment problem. J. of Multi-Criteria Dec. Anal. 2002. Vol. 11, No. 2, pp. 65-74.

[506] Schein J. An information model for building automation systems. Automation in Construction. 2007. Vol. 16, No. 2, pp. 125-139.

[507] Segev A. The node-weighted Steiner tree problem. Networks. 1987. Vol. 17, No. 1, pp. 1-17.

[508] Selkow S.M. The tree-to-tree editing problem. Information Processing Letters. 1977. Vol. 6, No. 6, pp. 184-186.

[509] Semple C., Steel M. A supertree method for rooted trees. Discrete Appl. Math. 2000. Vol. 105, No. 1-3, pp. 147-158.

[510] Sen P., Yang J.-B. Multiple Criteria Decision Support in Engineering Design. Springer, 1998.

[511] Serzhantov A.V. An optimal deciphering algorithm for some classes of monotonic functions. USSR Computational Mathematics and Mathematical Physics. 1983. Vol. 23, No. 1, pp. 144-148.

[512] Shakun M.F. Evolutionary System Design: Policy Making Under Complexity and Group Decision Support Systems. Holden Day Inc., Oakland, CA, 1988.

[513] Shama A. The marketing of political candidates. J. of the Academy of Marketing Science. 1976. Vol. 4, No. 4, pp. 764-777.

[514] Shaw M., Garlan D. Software Archicture: Perspectives on an Emerging Disicpline. Prentice Hall, New York, 1996.

[515] Shibata K., Okamura K., Araki K. Design and evaluation of dynamic protocol seleciton architecture for reliable multicast. In: Symp. on Applications and the Internet SAINT 2002, 2002. pp. 262-269.

[516] Shih H.-S., Shyur H.-J., Lee E.S. An extension of TOPSIS for group decision making. Mathematical and Computer Modelling. 2007. Vol. 45, No. 7-8, pp. 801-813.

[517] Shoham Y., Leyton-Brown K., Multiagent Systems. Cambridge Univ. Press, New York, 2009.

[518] Shoubridge P., Kraetzl M., Wallis W., Bunke H. Detection of abnormal change in time series of graphs. J. of Interconnnection Networks. 2002. Vol. 3, No. 1-2, pp. 85-101.

[519] Sim J.S., Park K. The consensus string problem for a metric is NP-complete. In: Raman R., Simpson J. (eds) Proc. of the 10th Australian Workshop on Combinatorial Algorithms, Perth, WA, Australia, 1999. pp. 107-113.

[520] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied Soft Computing. 2009. Vol. 9, No. 2, pp. 625-631.

[521] Smirnov A., Sheremetov L., Chilov N., Cortes J.R. Soft-computing technologies for configuration of cooperative supply chain. Applied Soft Computing. 2004. Vol. 4, pp. 87-107.

[522] Smirnov A., Chilov N., Levashova T., Sheremetov L., Contreras M. Ontology-driven intelligent service for configuraiton support in networked organizations. Know. Inf. Syst. 2007. Vol. 12, No. 2, pp. 229-253.

[523] Smirnov A., Shilov N., Kashevnik A., Jung T., Sinko M., Orozszi A. Ontology-driven product configuration: industrial use case. In: Proc. of Int. Conf. on Knowledge Management and Informaiton Sharing - KMIS 2011, Oct. 26-29, Paris, France, 2011. pp. 38-47.

[524] Sneath P.H.A., Sokol R.R. Numerical Taxonomy: The Principles and Practice of Numerical Classification. W.H. Freeman and Company, San Francisco, 1973.

[525] Sohraby K., Minoli D., Znati T., Wireless Sensor Networks: Technology, Protocols, and Applications. J.Wiley & Sons, New York, 2007.

[526] Solis-Oba R. 2-approximation algorithm for finding a spanning tree with maximum number of leafs. In: Bilardi G., Italiano G.F., Pietracaprina A., Pucci G. (eds) Proc. of 6th Annual Eur. Symp. on Algorihtms - ESA'98, LNCS 1461, Springer, 1998. pp. 441-452.

[527] Solnon C., Jolion J.-M. Generalized vs set median string for histogram based distances: Algorithms and classification results in image domain. In: Escolano F., Vento M. (eds) Proc. of the 6th Workshop on Graph Based Representation in Pattern Recognition GbPrP 2007, LNCS 4538, Springer, 2007. pp. 404-414.

[528] Song G., Ding F., Zang W., Song A., A wireless power outlet system for smart homes. IEEE Trans. Consumer Electronics. 2008. Vol. 54, No. 4, pp. 1688-1691.

[529] Song X., Dou W., A workflow framework for intelligent service composition. Future Generaiton Computer Systems. 2011, Vol. 27, No. 5, pp. 627-636.

[530] Sousa J.P., Poladian V., Garlan D., Schmerl B., M. Shaw M. Task-based adaptation for ubiquitous computing, IEEE Trans. on SMC, Part C. 2006. Vol. 36, No. 3, pp. 328-340.

[531] Steel M., Warnow T. Kaikoura tree theorems: Computing the maximum agreement subtree. Inf. Process. Lett. 1993. Vol. 48, No. 2, pp. 77-82.

[532] Stefik M., Introduction to Knowledge Systems. Morgan Kaufmann, San Francisco, CA, 1995.

[533] Steuer R.E. Multiple Criteria Optimization: Theory, Computation, and Application. J.Wiley & Sons, New York, 1986.

[534] Steward D.V. The design structure system: A method for managing the design of complex systems. IEEE Trans. on Eng. Management. 1981. Vol. 28, pp. 71-74,.

[535] Stewart D.B., Volpe R.A., Khosta P.K. Design of dynamicaly reconfigurable real-time software using port-based objects. IEEE Trans. on Software Engineeirng. 1997, Vol. 23, No. 12, pp. 759-776.

[536] Storer J.A. Constructing full spanning trees for cubic graphs. Infor. Proc. Lett. 1981. Vol. 13, No. 1, pp. 8-11.

[537] Suh N.P. Axiomatic Design. Advances and Applications. Oxford University Press, 2001.

[538] Sun A., Zeng D.D., Hsinchun C. Burst detection from multiple data streams: A network-based approach. IEEE Trans. SMC, Part C. 2010. Vol. 40, No. 3, pp. 258-267.

[539] Suryanegara M., Miyazaki K. Tehcnological changes in the innovation system towards 4G mobile service. Int. J. of Technology, Policy and Management. 2010. Vol. 10, No. 4, pp. 375-394.

[540] Sykes E.A., White III, C.C. Multiobjective intelligent computer-aided design. IEEE Trans. on SMC. 1989. Vol. 21, No. 6, pp. 1498-1511.

[541] Syropoulos A. Mathematics of multisets. In: Calude C.S. et al. (eds) Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View, LNCS 2235, Springer, 2001, pp. 347-358.

[542] Szekely G.J., Rizzo M.L. Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. J. of Classification. 2005. Vol. 22, No. 2, pp. 151-183.

[543] Tai K.-C. The tree-to-tree correction problem. J. of the ACM. 1979. 1979. Vol. 26, No. 3, pp. 422-433.

[544] Tanaka E. A metric between unrooted and unordered trees and its bottom-up computing method. IEEE Trans. PAMI. 1994. Vol. 16, No. 12, pp. 1233-1238.

[545] Tanenbaum A.S. Computer Networks. 4th ed., Prentice Hall PTR, NJ, 2002.

[546] Tanenbaum A.S. Structured Computer Organization, 5th ed., Prentice Hall, Upper Saddle River, NJ, 2006.

[547] Tannenbaum A.S., Kavcic B., Rosner M., Vianello M., Weisner G. Hierarchy in Organization. Jossey-Bass, Inc., San Fransisco, 1974.

[548] Tarhio J., Ukkonen E. A greedy approximation algorithm for constructing shortest common superstrings. Theor. Comput. Sci. 1988. Vol. 57, No. 1, pp. 131-145.

[549] Tarjan R.E. Edge-disjoint spanning trees and depth-first search. Acta Informatica. 1976. Vol. 6, No. 2, pp. 171-185.

[550] Tavana M. CROSS: a multicriteria group-decision-making model for evaluating and prioritizing advanced-technology projects at NASA. Interfaces. 2003. Vol. 33, No. 3, pp. 40-56.

[551] Taura T., Kubota A. A study on engineering history base. Res. in Eng. Des. 1999. Vol. 11, pp. 45-54.

[552] Thai M., Wang F., Liu D., Zhu S., Du D. Connected dominating sets in wireless networks with different transmission ranges. IEEE Trans. Mob. Comput. 2007. Vol. 6, No. 7, pp. 721-730.

[553] Thanassoulis E., Introduction to the Theory and Application of Data Envelopment Analysis: A foundation text with integrated software. Kluwer Academic Publishers, Boston, 2001.

[554] Timkovsky V.G. Complexity of common subsequence and supersequence problems. Cybernetics and Systems Analysis. 1989. Vol 25, No. 5, pp. 565-580.

[555] Torsello A., Hancock E.R. Efficiently computing weighted tree edit-distance using relaxation labeling. In: Figueiredo M., Zerubia J., Jain A.K. (eds) Energy Minimization Methods in Computer Vision and Pattern Recognition. LNCS 2134, Springer, 2001. pp. 438-453.

[556] Torsello A., Hidovic D., Pelillo M. Polynomial-time metrics for attributed trees. IEEE Trans. PAMI. 2005, Vol. 27, No. 7, pp. 1087-1099.

[557] Tsai Y.-T., Wang K.-S. The development of modular-based design in considering technology complexity. EJOR. 1999. Vol. 119, No. 3, pp. 692-703.

[558] Tseng H.-E., Chang C.-C., Li J.-D. Modular design to support green life-cycle engineering. Expert Systems with Applications. 2008. Vol. 34, No. 4, 2524-2537.

[559] Uchoa E. Reduction tests for the prize-collecting Steiner problem. Oper. Res. Lett. 2006. Vol. 34, No. 4, pp. 437-444.

[560] Ueda K. Synthesis and emergence - research overview. Artif. Intell. in Engineering. 2001. Vol. 15, No. 4, pp. 321-327.

[561] Ulrich K.T., Eppinger S.D. Product Design and Development. 2nd ed., Irwin McGraw-Hill, Boston, 2000.

[562] Umeda Y., Fukushige S., Tonoike K., Kondoh S. Product modularity for life cycle design. CIRP Annals - Manufacturing Technology. 2008. Vol. 57, No. 1, pp. 13-16.

[563] Uschold M., Gruninger M. Ontologies: Principles, methods and applications. Knowledge Eng. Rev. 1996. Vol. 11, No. 2, pp. 93-136.

[564] Valiente G. An efficient bottom-up distance between trees. In: Proc. of the 8th Int. Symp. String Processing Information Retrieval SPIRE 2001, IEEE CS Press, Laguna de San Rafael, Chile, 2001. pp. 212-219.

[565] Valiente G. Algorithms on Trees and Graphs. Springer, Berlin, 2002.

[566] Valtchev D., Frankov I. Service gateway architecture for a smart home. IEEE Communication Magazine. 2002, Vol. 40, No. 4, pp. 126-132.

[567] Vidal E., Marzal A., Aibar P. Fast computation of normalized edit distances. IEEE Trans. PAMI. 1995. Vol. 17, No. 9, pp. 899-902.

[568] Voronin A.A., Mishin S.P. Algorithms to seek the optimal structure of the organizational system. Autom.&Remote Control. 2002. Vol. 63, No. 5, pp. 803-814.

[569] Voss S. Steiner's problem in graphs: heuristic methods. Discr. Appl. Math. 1992. Vol. 40, No. 1, pp. 45-72.

[570] Voss S. The Steiner tree problem with hop constraints. Ann. of Oper. Res. 1999. Vol. 86, pp. 321-345.

[571] Voss S. (2000) Modern heuristic search methods for the Steiner tree problem in graph. In: Du D.-Z., Smith J.M., Rubinstein J.H. (eds) Advances in Steiner Trees, Kluwer Academic Publishers, 2000. pp. 283-323.

[572] Voss C.A., Hsuan J. Service architecture and modularity. Decision Sciences. 2009. Vol. 40, No. 3, pp. 541-569.

[573] Vujosevic M., Stanojevic M. A bicriteria Steiner tree problem on graph. Yugoslav J. of Operations Research. 2003. Vol. 13, No. 1, pp. 25-33.

[574] Wagner R.A., Fisher M.J. The string-to-string correction problem. J. of the ACM. 1974. Vol. 21, No. 1, pp. 168-173.

[575] Waller H.J. The synthesis of hierarchical structures: techniques and applications. Decision Sciences. 1976. Vol. 7, No. 4, pp. 659-674.

[576] Wallis W.D., Shoubridge P., Kraetz M., Ray D. Graph distances using graph union. Pattern Recognition Letters. 2001. Vol. 22, No. 6-7, pp. 701-704.

[577] Wang J.T.-L., Zhang K. Finding similar consensus between trees: an algorithm and distance hierarchy. Pattern Recognition. 2001. Vol. 34, No. 1, pp. 127-137.

[578] Wang Y.-H., Yang L., Smart home information: A framework for integration and management. Measurement and Control. 2008. Vol. 41, No. 10, pp. 300-304.

[579] Watkinson J., MPEG Handbook. Butterworth-Heinemann, UK, 2001.

[580] Westbrook J., Yan D.C.K. (1993) Greedy algorithm for the on-line Steiner tree and generalized Steiner problems. In: Dehne F.K.H.A., Sack J.-R., Santoro N., Whitesides S. (eds) Proc. of the Third Workshop on Algorithms and Data Structures WADS'93, LNCS 709, Springer, 1993. pp. 622-633.

[581] Whidden C., Beiko R.G., Zeh N. Fast FPT algorithms for computing rooted agreement forest: Theory and experiments. In: Proc. of the 9th Int. Symp. on Experimental Algorithms SEA 2010, LNCS 6049, Springer, 2010. pp. 141-153.

[582] Whidden C., Beiko R.G., Zeh N. Fixed parameter and approximation algorithms for maximum agreement forests. Electronic preprint, 12 Aug. 2011. Http://arxiv.org/abs/1108.2664v1 [q-bio.PE]

[583] Winter P. Steiner problem in networks: a survey. Networks. 1987. Vol. 17, No. 2, pp. 129-167.

[584] Wu B.Y., Chao K.-M. Spanning Trees and Optimization Problems. CRC Press, Boca Raton, FL, 2004.

[585] Yager R.R., On the theory of bags. Int. J. of General Systems. 1986. Vol. 13, No. 1, pp. 23-37.

[586] Yelowitz L. An efficient algorithm for constructing hierarchical graphs. IEEE Trans. SMC. 1976. Vol. 6, No. 4, pp. 327-329.

[587] Yerramareddy S., Lu S.C.-Y. Hierarchical and interactive decision refinement methodology for engineering design. Res. in Eng. Des. 1993. Vol. 4, No. 4, pp. 227-240.

[588] Zadorozhny V., Chrysanthis P., Krishnamurthy P., Network-Aware Wireless Sensor Management. Springer, New York, 2007.

[589] Zhang K. A constrained edit-distance between unordered labeled trees. Algorithmica. 1996. Vol. 15, No. 3, pp. 205-222.

[590] Zhang K., Sasha D. Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. 1989. Vol. 18, No. 6, pp. 1245-1262.

[591] Zhang K., Statman R., Sasha D. On the editing distance between unordered labeled trees. *Inform. Process. Letters*, Vol. 42, No. 3, 1992. pp. 133-139.

[592] Zhang K., Wang J.T.L., Sasha D. On the editing distance between undirected acyclic graphs. Int. J. Foundations of Computer Science. 1996. Vol. 7, No. 1, pp. 43-57.

[593] Zhou C.-J., Chen H., Qin Y.-Q., Shi Y.-F., Yu G.-C., Self-organization of reconfigurable protocol stack for networked control systems. Int. J. of Automation and Control. 2001. Vol. 8, No. 2, pp. 221-235.

[594] Zimmerman H. OSI Reference Model - The ISO model of architecture for open systems interconnection. IEEE Trans. on Communications. 1980. Vol. 28, No. 4, 425-432.

[595] Zopounidis C., Doumpos M. Multicriteria classification and sorting methods: a literature review. EJOR. 2002. Vol. 138, No. 2, pp. 229-246.

[596] Zopounidis C., Doumpos M. Multicriteria sorting methods. In: Floudas C.A., Pardalos P.M. (eds) Encyclopedia of Optimization. 2nd ed., Springer, 2009. pp. 2379-2396.

[597] Zou F., Li X., Kim D., Wu W. Two constant approximation algorithms for node-weighted Steiner tree in unit disk graphs. In: Yang B., Du D.-Z., Wang C.A. (eds) Proc. of Second Annu. Int. Conf. on Comb. Optim. and Appl. COCOA 2008, LNCS 5165, Springer, 2008. pp. 278-285.

[598] Zwicky F. Discovery Invention, Research Through the Morphological Approach. McMillan, New York, 1969.

View publication stats