

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СССР
ПО НАУКЕ И ТЕХНИКЕ

АКАДЕМИЯ НАУК СССР

ВСЕСОЮЗНЫЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
СИСТЕМНЫХ ИССЛЕДОВАНИЙ

На правах рукописи

ЛЕВИН Марк Шмуилович

НЕКОТОРЫЕ ДИСКРЕТНЫЕ ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ
РАСПРЕДЕЛЕНИЯ РЕСУРСОВ
В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

05.13.02 — теория систем,
теория автоматического регулирования
и управления, системный анализ

АВТОРЕФЕРАТ

*диссертации на соискание ученой степени
кандидата технических наук*

МОСКВА — 1981

Работа выполнена на кафедре математических методов анализа экономики МГУ им. М. В. Ломоносова

Научный руководитель:

доктор технических наук профессор Юдин Д. Б.

Официальные оппоненты:

доктор технических наук Пропой А. И.

кандидат физико-математических наук Мартынюк В. В.

Ведущая организация: — Центральный научно-исследовательский институт комплексной автоматизации

Защита состоится « » _____ 198 г. в _____ часов
на заседании Специализированного Совета К 038.02.01 Всесоюзного научно-исследовательского института системных исследований: Москва, ул. Кржижановского, д. 14, кор. 2. Телефон Совета: 124-36-90.

С диссертацией можно ознакомиться в библиотеке Всесоюзного научно-исследовательского института системных исследований.

Автореферат разослан « » _____ 198 г.

Ученый секретарь
Специализированного Совета
кандидат техн. наук

С. А. Пегов

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

А к т у а л ь н о с т ь . Одной из важнейших задач, возникающих в условиях широкого применения вычислительной техники, является проблема ее рационального использования. Сложность определения эффекта, связанного с применением вычислительной техники, обуславливает необходимость исследования различных частных задач, решение которых направлено на повышение эффективности использования ЭВМ. Одной из таких задач является проблема снижения затрат при эксплуатации вычислительного комплекса. Для анализа задач распределения основных ресурсов вычислительных систем — производительности и памяти — используются дискретные оптимизационные модели. Это относится к задачам составления расписаний при ограничении вычислительного процесса, к задачам организации данных.

Дискретные оптимизационные модели, например, графовые, являются наглядными и простыми для понимания. Построение таких моделей позволяет в значительной степени структуризовать стоящие перед специалистами по разработке и эксплуатации вычислительных комплексов проблемы. Большинство практических дискретных задач распределения производительности и памяти ЭВМ представляют собой сложные комбинаторные проблемы. Поэтому является актуальным построение эффективных точных и приближенных методов решения возникающих проблем; выявление классов задач, допускающих эффективные точные алгоритмы решения, эффективные приближенные методы с гарантированной погрешностью.

Ц е л ь р а б о т ы . Целью диссертационной работы является исследование проблемы эффективного использования вычислительной техники, построение дискретных оптимизационных моделей распределения ресурсов в вычислительных системах и разработка эффективных методов решения возникающих задач. В работе сформулированы следующие направления исследования:

- анализ детерминированных задач составления расписаний при организации вычислительного процесса в случае информационных связей между программами и построение эффективных методов решения;
- разработка моделей организации данных, направленных на уменьшение среднего числа обменов между оперативной памятью

и внешней памятью ЭВМ, и построение эффективных алгоритмов решения возникающих задач компоновки данных.

М е т о д ы и с с л е д о в а н и я . В работе использованы методы теории сложных систем, модели теории расписаний, теории графов. Анализ задач основывается на результатах теории сложности дискретных оптимизационных проблем. При решении задач используются методы комбинаторного программирования, в том числе приложенной оптимизации.

Н а у ч н а я н о в и з н а и п р а к т и ч е с к а я ц е н н о с т ь . В работе проведен анализ проблемы эффективно-го использования вычислительной техники. Подробно рассмотрены и решены следующие оптимизационные задачи распределения производительности и памяти:

- выделен класс задач составления расписаний в случае одновременного поступления заявок и ограничения предшествования в виде совокупности деревьев или параллельно-последовательной сети, предложен общий эффективный подход, позволяющий получать оптимальное расписание;

- предложены оптимизационные модели организации данных, направленные на уменьшение среднего числа обменов между оперативной и внешней памятью ЭВМ;

- для решения сформулированной задачи организации данных в рамках некоторых моделей построены эффективные приближенные алгоритмы, гарантирующие заданную относительную погрешность как целевой функции, так и выполнения ограничения на объем оперативной памяти.

Использование предложенных моделей организации данных позволяет структуризовать процессы проектирования и организации функционирования программных комплексов и информационных систем.

Предложенные модели и эффективные методы решения соответствующих дискретных экстремальных задач могут быть применены в сложных организационно-технических системах различного назначения: при календарном планировании производства и строительства; при составлении графиков проведения текущих и капитальных ремонтов оборудования и т. п..

Выполненные исследования дают возможность повысить эффективное быстродействие вычислительных систем, и, следовательно, снизить затраты на эксплуатацию вычислительных комплексов.

Практическое использование результатов работы. Предложенные в диссертационной работе модели и методы организации данных, планирования порядка обработки информации были использованы при решении задач практического проектирования систем программного обеспечения: АСУ-Росгеология, системы разработки комплексных тем создания изделий для различных видов спорта, подсистемы организационно-экономического управления АСУ предприятиями электрических систем.

Применение результатов диссертационной работы позволило осуществить обоснованный выбор эффективных вариантов оверлейных структур программных комплексов и рациональной организации масшштабов-каталогов для поиска информации. Выбранные варианты обеспечивают экономное использование оперативной памяти и, тем самым, повышают эффективную производительность ЭВМ.

Программное обеспечение информационно-поисковой системы ИПС-Геология, разработанное под руководством и при непосредственном участии автора на основе использования предложенных в диссертации методов, принято в отраслевой фонд алгоритмов и программ ОФАП-Геология (рег. № 547/98Ip) и передано к внедрению в ГИИЦ АСУ-Геология ВИАМСа.

Модели организации программных комплексов, разработанные в работе, предполагается использовать при генерации специализированных пакетов прикладных программ из модулей библиотеки программных модулей сбора и обработки информации, моделирования и управления технологическими процессами АСУ ТП добычи, подготовки и транспорта газа во ВНИИАНефтегаз. Методы составления расписаний при ограничениях предшествования, предложенные в работе начинают использоваться для решения задач составления оптимальных графиков текущих и капитальных ремонтов оборудования подстанций и линий электропередач в рамках подсистемы организационно-экономического управления АСУ линиями электропередач.

Изложенный в диссертации подход к оценке эффективности использования вычислительной техники и оригинальные точные и приближенные эффективные алгоритмы решения задач планирования и организации данных включены в раздел технико-экономического обо-

снования АСУ учебного курса "Исследование операций" экономического факультета Московского государственного университета им. М.В.Ломоносова.

А п р о б а ц и я р а б о т ы . По теме диссертации были сделаны доклады: на международной конференции "Проблемы проектирования и применения дискретных систем в управлении", Минск, 1977 г., на Московской конференции "Экономика, организация и управление геологоразведочным производством", 1979 г., на семинаре по дискретной оптимизации в ЦАМИ АН СССР, 1977-1979 гг., на семинаре по исследованию операций на экономическом факультете МГУ им. М.В.Ломоносова, 1977-1980 гг., на Всесоюзном научном симпозиуме "Статистический и дискретный анализ нечисловой информации, экспертные оценки и дискретная оптимизация", Алма-Ата, 1981 г..

С т р у к т у р а и о б ъ е м р а б о т ы . Диссертация состоит из введения, четырех глав, заключения, приложений и описки литературы. Она содержит 79 страниц основного печатного текста, 12 страниц приложений, 17 иллюстраций, 19 таблиц. Список использованной литературы включает 74 наименования.

С О Д Е Р Ж А Н И Е Р А Б О Т Ы

В первой главе рассмотрены подходы к оценке эффективности использования вычислительной техники, основные типы задач распределения ресурсов в вычислительных системах, исследование которых приводит к дискретным оптимизационным проблемам, и современная методология построения эффективных алгоритмов.

В большинстве случаев исследование эффекта применения вычислительной техники представляет собой сложную задачу, поэтому является целесообразным, в соответствии с методами анализа сложных систем, переход к исследованию частных подзадач, согласованных с основной целью - повышением экономической эффективности использования вычислительной техники.

Одной из важнейших подзадач является проблема снижения затрат, связанных с применением ЭВМ. В этом направлении можно выделить два уровня: глобальный и локальный. К глобальному уровню, характеризующемуся централизованным использованием технических и программных средств, можно отнести промышленную

разработку программного обеспечения, создание вычислительных центров коллективного пользования и сетей ЭВМ. Локальный уровень снижения затрат заключается в решении технических проблем по экономии важнейших ресурсов вычислительных систем: производительности и памяти.

дискретные оптимизационные модели распределения ресурсов в вычислительных системах используются в основном при решении следующих задач: составление расписаний при организации вычислительного процесса; организация данных; распределение памяти.

Размерность возникающих дискретных оптимизационных задач такова, что применение методов неявного перебора может привести к значительным затратам. Например, при составлении расписания число подлежащих упорядочению программ равно 20 - 50, программные комплексы включают порядка 20 - 100 программных модулей. Это обуславливает требование к эффективности применяемых алгоритмов. В качестве эффективных рассматриваются алгоритмы, трудоемкость и память которых полиномиально зависят от размерности задачи.

Вторая глава диссертации посвящена исследованию детерминированных задач составления расписаний при организации вычислительного процесса. Предполагается, что заявки поступают одновременно. Рассматривается случай, когда программы информационно связаны, то есть выходные результаты одних программ являются входными данными для других - ограничение предшествований.

В работе выделен класс задач теории расписаний, для которого предложен общий эффективный алгоритм получения точного решения при ограничении предшествования в виде дерева или параллельно-последовательной сети. В задачах рассматриваемого класса имеется множество заданий (работ) $R = \{1, 2, \dots, n\}$ и ограничение предшествования, заданное направленным ациклическим графом $G = (R, \Gamma)$. Требуется установить оптимальный порядок S^* выполнения работ (расписание), который является допустимым то есть удовлетворяет ограничениям предшествования и на котором значение функционала минимально: $f(S^*) = \min_S f(S)$. К выделенному классу относятся следующие задачи:

Задача I. Имеется один процессор. Каждое задание $i \in R$ характеризуется временем выполнения T_i и функцией штрафа $\psi_i(t) = a_i t + b_i$. Требуется минимизировать суммарный штраф $f(S) = \sum_{i=1}^n \psi_i(C_i)$.

где C_i - момент окончания выполнения работы i .

Задача 2. Отличается от задачи 1 тем, что функция штрафа имеет вид: $\varphi_i(t) = a_i \exp(\lambda t) + b_i$.

Задача 3. Это задача Беллмана-Джонсона в случае двух процессоров. Каждое задание $i \in R$ должно выполняться сначала на первом процессоре, затем на втором с временами выполнения α_i , β_i . Требуется минимизировать суммарное время выполнения всех работ.

Задача 4. Имеется один процессор. Каждое задание $i \in R$ характеризуется временем выполнения t_i и функцией штрафа $\varphi_i(t) = a_i t + b_i$. Требуется минимизировать суммарный штраф $f(s) = \sum_{i=1}^n \varphi_i(t_i)$, где $t_i = \int_{s[i-1]}^{s[i]} 1 dt$, $s[i]$ - номер задания, находящегося в расписании s на i -м месте.

Задача 5. Отличается от задачи 4 тем, что функция штрафа имеет вид: $\varphi_i(t) = a_i \ln \lambda t + b_i$.

Задача 6. Имеется один процессор. Каждое задание $i \in R$ характеризуется стоимостью выполнения p_i и вероятностью срыва выполнения задания p_i . Требуется минимизировать среднюю стоимость выполнения всех заданий $f(s) = \sum_{i=1}^n \frac{p[s[i]]}{D[s[i]]}$, где $D[s[i]] = \prod_{k=1}^{s[i]} (1 - p_k)$.

Последняя задача заменой переменных сводится к задаче 1, но ее можно рассматривать и независимо. Для задачи 3 просто доказывается, что оптимальное расписание можно искать на множестве одномагистральных расписаний. Таким образом, расписание - это перестановка n чисел $1, 2, \dots, n$: $s = \langle s[1], s[2], \dots, s[n] \rangle$.

Таким образом, можно рассматривать некоторую обобщенную задачу, которая характеризуется следующим. Во-первых, в случае отсутствия ограничения предшествования, она может быть решена упорядочением заданий по неубыванию одностепенных функций - приоритетов $\omega(i)$, вычисляемых для каждого задания $i \in R$. Во-вторых, выполняется условие склеивания заданий, которое заключается в следующем:

Пусть $N(i, j)$ - множество допустимых расписаний, в которых два произвольных фиксированных задания $i, j \in R$ стоят рядом и задание i предшествует j . Рассмотрим произвольное расписание $s \in N(i, j)$, в котором $i = s[l], j = s[l+1]$ ($1 \leq l < n$).

Пусть имеется задание $J \in R$. Определим расписание S множества заданий $(R \setminus \{i, j\}) \cup J$ следующим образом:

$$\bar{S}[\tau] = \begin{cases} g[\tau], & \text{при } \tau = 1, \ell-1, \\ J, & \text{при } \tau = \ell, \\ S[\tau+1], & \text{при } \tau = \ell+1, n-1. \end{cases}$$

Определение 1. Задание $J = J(i, j)$ назовем эквивалентным заданиям i, j , если выполняются два условия:

- 1) параметры J могут быть в явном виде выражены через параметры заданий i и j ,
- 2) для любого расписания $s \in N(i, j)$ выполняется соотношение $f(s) = f(\bar{s}) + C_{ij}$, где C_{ij} - константа, не зависящая от s .

Определение 2. Расписание $\bar{s}$, полученное из $s \in N(i, j)$, называется эквивалентным расписанием s , J - задание, эквивалентное i и j .

Параметры эквивалентного задания $J(i, j)$ и константа C_{ij} выражаются через параметры заданий i и j .

Определение 3. Задания $i, j \in R$ называются независимыми, если допустимость произвольного расписания не зависит от порядка следования элементов i, j .

Определение 4. Будем говорить, что задача имеет нормальный вид, если для любых $i, j \in R$, таких, что $j \in \Gamma_i$, выполняется $\omega(i) < \omega(j)$.

Очевидно, что если задача имеет нормальный вид, то для получения оптимального расписания достаточно упорядочить множество заданий по неубыванию их приоритетов, так как полученное упорядочение будет допустимым.

Определение 5. Множество $K' \subseteq K$ назовем модулем, если для любого $i \in R$ выполняется одно из трех условий:

- 1) любое задание $j \in K'$ образует с i независимую пару;
- 2) задание i должно предшествовать любому $j \in K'$;
- 3) любое задание $j \in K'$ должно предшествовать i .

По аналогии с определением 4 можно определить модуль нормального вида.

Определение 6. Поддерево T_x с корнем x будем называть нормальным, если

- 1) в множестве вершин поддерева T_x кроме x входят все вершины из R , которым предшествует x ,

2) для любых вершин y, z из T_x таких, что $z \in \Gamma y$, выполняется условие $\omega(y) < \omega(z)$.

Обозначим через Φ отображение $N(i, j) \rightarrow N(i, j)$. В случае выполнения условия "склеивания" задание просто доказывається следующая лемма.

Лемма 1. Для того, чтобы расписание $s_0 \in N(i, j)$ было оптимальным на $N(i, j)$, то есть для любого $\bar{s} \in N(i, j)$ выполнялось $f(s_0) \leq f(\bar{s})$ необходимо и достаточно, чтобы эквивалентное расписание $\bar{s}_0 = \Phi(s_0) \in \bar{N}(i, j)$ было оптимально на $\bar{N}(i, j)$, то есть для любого $\bar{s} \in \bar{N}(i, j)$ выполнялось $f(\bar{s}_0) \leq f(\bar{s})$.

За счет "склеивания" задания и перехода к эквивалентным задачам возможно приведение задач к нормальному виду.

Лемма 2. Пусть $G=(R, \Gamma)$ - дерево, $x \in R$, M - множество последователей x , причем для любого $y \in M$ поддерева с корнем y является нормальным. Пусть $y_0 \in M$ такое, что для любого $y \in M$ $\omega(y_0) \leq \omega(y)$ и $\omega(x) \geq \omega(y_0)$. Тогда от любого допустимого расписания задания множества R можно без ухудшения значения критерия перейти к некоторому допустимому расписанию, в котором задания x и y_0 находятся рядом.

Лемма 2 позволяет любую исходную задачу на дереве с помощью "склеивания" работ приводить к эквивалентной задаче нормального вида.

Алгоритм I.

I. Приведение исходной задачи к эквивалентной задаче нормального вида.

0 шаг. Выделение всех висячих вершин (множество M^0).

I шаг. Выделение множества M^1 (множество вершин, последователи которых принадлежат M^0). Для каждой вершины $x \in M^1$ сравнение $\omega(x)$ и $\omega(i_0)$, где i_0 такое, что $\omega(i_0) \leq \omega(i)$ для всех $i \in \Gamma x$. Если $\omega(x) \geq \omega(i_0)$, то "склеивание" задания x и i_0 и переход к эквивалентной задаче. (Для эквивалентной задачи проводится аналогичное сравнение эквивалентного задания J с его последователями.)

k шаг. Выделение множества M^k (множество вершин, все последователи которых принадлежат множеству $\bigcup_{j=1}^{k-1} M^j$). Для каждой вершины $x \in M^k$ сравнение $\omega(x)$ и $\omega(i_0)$, где i_0 такое, что $\omega(i_0) \leq \omega(i)$ для всех $i \in \Gamma x$. Если $\omega(x) \geq \omega(i_0)$, то

"склеивание" заданий x и i , и переход к эквивалентной задаче. (Для эквивалентной задачи проводится аналогичное сравнение эквивалентной работы J с ее последователями.)

П. Решение эквивалентной нормальной задачи (упорядочение работ по неубыванию приоритетов $\omega(i)$).

Ш. Замена эквивалентных заданий в решении эквивалентной нормальной задачи на задания исходного множества R .

Оценка трудоемкости алгоритма I - $O(n \log n)$.

Предложенный подход распространяется на случай, когда ограничение предшествования представляет собой параллельно-последовательную сеть.

Определение 7. Пусть вершина x предшествует вершинам модуля R' , вершина y следует за ними. Назовем модуль $R' = \{\tau_1, \dots, \tau_i, \dots, \tau_m\}$ последовательным, если $\tau_{i+1} = \tau_i$ ($i = 1, m-1$). Назовем модуль параллельным, если он состоит из двух последовательных $R'_1 = \{\tau_1, \dots, \tau_i, \dots, \tau_m\}$ и $R'_2 = \{q_1, \dots, q_j, \dots, q_k\}$ таких, что для любых i и j ($i = 1, m$, $j = 1, k$) τ_i, q_j независимы.

В случае, когда сеть предшествования сводится к параллельно-последовательной сети, алгоритм имеет вид:

Алгоритм 2.

1 шаг. Приведение к нормальному виду всех последовательных модулей за счет "склеивания" заданий и перехода к эквивалентным задачам.

2 шаг. Если вся сеть приведена к последовательному модулю, то переход к шагу 4.

3 шаг. Проведение всех возможных слияний последовательных модулей нормального вида в параллельных модулях. Слияние заключается в упорядочении вершин двух последовательных модулей, входящих в состав параллельного, в порядке неубывания приоритетов. Переход к шагу 1.

4 шаг. Замена эквивалентных заданий на задания исходного множества.

Предложенный подход позволяет эффективно решать задачи и в некоторых случаях, когда ограничение предшествования задано более сложным способом, чем сеть, сводящаяся к параллельно-последовательной.

В третьей главе исследуется организация данных в рамках

оверлейной структуры на примере задачи компоновки программных модулей. За счет объединения модулей возможно снижение среднего числа вызовов из внешней памяти. Предполагается, что передача по управлению осуществляется с возвратом в вызывающий модуль и связи между модулями имеют вид дерева. Рассматривается ограничение на общий объем используемой оперативной памяти, в то время, как в известных моделях имеется ограничение на объем сегмента.

Пусть задано направленное дерево $G=(R, \Gamma)$ с ориентацией дуг от корня $a_0 \in R$ к висячим вершинам, где R - множество вершин, взаимно однозначно соответствующее множеству программных модулей, Γ - многозначное отображение R в R , характеризующее отношение вызова между программными модулями. Каждая вершина $\alpha \in R$ имеет вес $v(\alpha) \geq 0$ (требуемый объем оперативной памяти). Дугу из α в β ($\alpha, \beta \in R$) обозначим через (α, β) , направление дуги указывает вершину, соответствующую вызываемому модулю. Введем для каждого пути $L(a_1, a_\ell) = \langle a_1, \dots, a_\ell \rangle$ ($a_j \in \Gamma a_{j-1}, j=2, \ell$) вес $v(L(a_1, a_\ell)) = \sum_{i=1}^{\ell} v(a_i)$.

Определение 8. Назовем весом графа G величину $v(G) = \max_{\alpha \in R} v(L(a_0, \alpha))$, где $\alpha = \{\alpha \in R \mid \Gamma \alpha = \emptyset\}$.

Пусть $G_\alpha = (R_\alpha, \Gamma)$ - поддереву о корнем в $\alpha \in R$ (в R_α кроме α входят все вершины из R , достижимые из α).

Определение 9. Назовем "хвостом" вершины $\alpha \in R$ граф $(\{R_\alpha\}, \Gamma)$, а весом "хвоста" вершины α - величину $\hat{v}(\alpha) = v(G) - v(\alpha)$. Очевидно, что $\hat{v}(\alpha) = \max_{\beta \in \Gamma \alpha} v(G_\beta)$.

Пусть среднее число вызовов модуля β модулем α ($\beta \in \Gamma \alpha$) при функционировании системы (вес дуги от α к β) равно $w(\alpha, \beta) \geq 0$. Предполагается, что при вызове модуля β модулем α последний находится в оперативной памяти, а модуль β загружается в оперативную память из внешней (адрес загрузки для любого $\beta \in R$: $\delta(\beta) = \max_{\alpha \in R} \{ \hat{v}(\alpha) + v(\alpha, \beta) \}$, $\hat{v}(a_0) = \hat{v}_0$). Вес дерева G определяет требуемый для функционирования программного комплекса объем оперативной памяти, а среднее число вызовов программных модулей из внешней памяти равно

$$W(G) = \sum_{\substack{\alpha, \beta \in R \\ \beta \in \Gamma \alpha}} w(\alpha, \beta).$$

Определение 10. Назовем граф G допустимым, если $v(G) < V$, где V - ограничение на объем оперативной памяти, предоставляемой комплексу.

Объединение (склеивание) программных модулей a и b ($b \in \Gamma_a$) позволяет заменить вызов модуля b модулем a из внешней памяти на переход к модулю b внутри оперативной памяти. Это снижает среднее число вызовов программных модулей из внешней памяти на $w(a, b)$, но увеличивает вес графа на величину $\Delta(a, b) \in [0, v(b)]$.

Пусть $|R| = n$. Введем множество дуг $U = \{u_1, \dots, u_i, \dots, u_{n-1}\}$. Обозначим через $\bar{G} = \{\bar{R}, \bar{\Gamma}\}$ граф, полученный из G склеиванием вершин $a, b \in R$, $b \in \Gamma_a$, то есть в $\bar{G}$ вместо вершин a и b входит вершина $J = J(a, b)$ с весом $v(J(a, b)) = v(a) + v(b)$. $\bar{\Gamma}J = \{\Gamma_a \cup \Gamma_b\} \setminus b$. При этом множество дуг в $\bar{G}$: $\bar{U} = U \setminus (a, b)$. Очевидно, что операция преобразования графа при склеивании вершин ассоциативна, то есть порядок склеивания дуг несущественен. Введем переменную x_i ($i = 1, n-1$), равную 1, если дуга u_i стягивается, и 0 в противном случае. Тогда множество стягиваемых дуг определяется булевым вектором $x = (x_1, \dots, x_{n-1})$. Введем критерий, характеризующий уменьшение среднего числа вызовов модуля из внешней памяти $W(x) = \sum_{i=1}^{n-1} x_i w(u_i)$. С учетом стягивания дуг для любого $a \in R$ вес вершины и вес "хвоста" можно рассматривать как функцию вектора x : $v(a, x)$, $\hat{v}(a, x)$. Таким образом, задача компоновки состоит в определении вектора x , который максимизирует критерий при выполнении условия допустимости графа $v(a_0, x) + \hat{v}(a_0, x) \leq V$. Назовем эту задачу постановкой I. Далее потребуется вспомогательная задача. Пусть V^- - ограничение на вес вершины a_0 , V^+ - ограничение на вес "хвоста" a_0 . $V^- + V^+ = V$.

Постановка 2. Максимизировать $W(x)$ при условиях

$$v(a_0, x) \leq V^-, \quad \hat{v}(a_0, x) \leq V^+.$$

Даже в простейшем случае, когда комплекс состоит из основного сегмента и вызываемых модулей, задача компоновки не проще задачи о рюкзаке, которая принадлежит к классу универсальных переборных проблем. Для решения задачи компоновки предложены эффективные приближенные алгоритмы, гарантирующие заданную относительную погрешность целевой функции и ресурса.

Пусть $\epsilon, \delta \in [0, 1]$, x^* - точное решение задачи компоновки. Введем для некоторого x функции относительной ошибки целевой функции $d(x)$ и ресурса $\beta(x)$:

$$d(x) = \max \left[0, \frac{W(x^*) - W(x)}{W(x^*)} \right],$$

$$\beta(x) = \max \left[0, \frac{v(x) - V}{V} \right],$$

где $v(x) = v(a_0, x) + \hat{v}(a_0, x)$, $W(x^*) \neq 0$. Задача состоит в определении при фиксированных $\epsilon, \delta \in [0, 1]$ такого x^0 , что $d(x^0) \leq \epsilon$, $\beta(x^0) \leq \delta$.

В работе описан эффективный метод, позволяющий выявлять случаи, когда задача компоновки не имеет решения или $W(x^*) = 0$.

В одном из простейших случаев, когда комплекс состоит из основного сегмента и вызываемых подпрограмм, G представляет собой двухуровневое дерево. Пусть $\Gamma a_0 = \{a_1, \dots, a_i, \dots, a_m\}$, $u_i = (a_0, a_i)$, $w(u_i) = w_i$ и задача компоновки имеет вид (постановка I):

максимизировать $W(x) = \sum_{i=1}^m x_i w_i$ при условии

$$v(a_0) + \sum_{i=1}^m x_i v(a_i) + \max_{1 \leq i \leq m} [(1 - x_i) v(a_i)] \leq V.$$

Для этой задачи в диссертации предложена модификация "алгоритма разбиения на интервалы" для решения задачи о рюкзаке - алгоритм 3. Оценка трудоемкости и памяти алгоритма $O(\frac{m^2}{\epsilon})$. Работа алгоритма 3 распространена на три других простейших случая задачи компоновки. Рассмотрим следующую задачу. Пусть каждая $a_i \in \Gamma a_0$ (вес $a_i = v^-(a_i)$) имеет одного последователя с весом $v^+(a_i)$, но объединение a_i с последователем запрещено. Два других случая соответствуют рассмотренным, но в постановке 2.

В работе выделена группа вспомогательных задач компоновки, являющихся усложнением простейших частных случаев. Формулировки вспомогательных задач соответствуют постановкам задачи компоновки в простейших частных случаях. Основная вспомогательная задача имеет следующую интерпретацию. Пусть a_0 - головной сегмент и имеются вызываемые им модули с номерами $1, \dots, i, \dots, m$. Для каждого модуля с номером i имеется множество вариантов реализации $\{a_{i1}, \dots, a_{ij}, \dots, a_{iq_i}\}$. Каждый вариант a_{ij} может представлять собой взвешенный граф, в котором осуществлено стягива-

ние дуг. Варианту α_{ij} соответствует вес $v(\alpha_{ij})$, среднее число вызовов модулем $\alpha_0 - w(\alpha_{ij})$ и число $\bar{w}(\alpha_{ij})$, равное уменьшению среднего числа обращений к внешней памяти при его работе. Требуется осуществить для каждого i выбор одного варианта реализации модуля (x_{ij}^1 равно 1, если выбирается α_{ij} и 0 в противном случае и присоединить его к α_0 ($x_{ij}^2 = 1$) или не присоединить ($x_{ij}^2 = 0$). Таким образом, искомым является набор булевых векторов $X = \{x_{ij} = (x_{ij}^1, x_{ij}^2)\}$, удовлетворяющих ограничениям: $\sum_{j=1}^m x_{ij}^1 = 1 \forall i, x_{ij}^2 \leq x_{ij}^1 \forall i, j$. Задача имеет вид:

максимизировать $W(X) = \sum_{i=1}^m \sum_{j=1}^{q_i} (x_{ij}^1 \bar{w}(\alpha_{ij}) + x_{ij}^2 w(\alpha_{ij}))$
при условии $v(\alpha_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v(\alpha_{ij}) \leq V$.

Для этой задачи предложена модификация алгоритма задачи о "лестничном" рюкзаке - алгоритм 4. Оценки трудоемкости и памяти, соответственно, $O(\frac{m}{\epsilon} \sum_{i=1}^m q_i)$ и $O(\frac{m^2}{\epsilon} \max_{1 \leq i \leq m} q_i)$.

Работа алгоритма распространена на другие вспомогательные задачи. Исключение составляет вспомогательные задачи вида:

максимизировать $W(X) = \sum_{i=1}^m \sum_{j=1}^{q_i} (x_{ij}^1 \bar{w}(\alpha_{ij}) + x_{ij}^2 w(\alpha_{ij}))$
при условии

$$v(\alpha_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v(\alpha_{ij}) + \max_{i,j} [(x_{ij}^1 - x_{ij}^2) v(\alpha_{ij})] \leq V,$$

максимизировать $W(X) = \sum_{i=1}^m \sum_{j=1}^{q_i} (x_{ij}^1 \bar{w}(\alpha_{ij}) + x_{ij}^2 w(\alpha_{ij}))$
при условии $v(\alpha_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v(\alpha_{ij}) + \max_{i,j} [(x_{ij}^1 - x_{ij}^2) v(\alpha_{ij}) + x_{ij}^1 v(\alpha_{ij})] \leq V$.

Для этих задач в качестве платы за быстроту решения дополнительно приходится мириться с гарантированной относительной погрешностью выполнения ограничения на память. Алгоритм 5, предложенный для решения этих задач, имеет оценку трудоемкости $O(\frac{m}{\epsilon} \sum_{i=1}^m q_i)$.

На основе алгоритмов решения простейших частных случаев и вспомогательных задач в диссертации построен эффективный алгоритм позволяющий решать задачу компоновки в случае, когда граф G имеет вид дерева. Этот алгоритм гарантирует относительную погрешность значения целевой функции ϵ и невязки ограничения на память δ . Алгоритм 6 основан на последовательном выделении и решении частных задач, имеющих вид частных случаев или вспомогательных.

При разработке программного обеспечения часто имеют место

олучаи, когда некоторые характеристики программных модулей практически равны. На основе предположения равенства характеристик модулей можно приближенно определять варианты компоновки на ранних этапах проектирования систем. Если G - двухуровневое дерево и объемы модулей или веса дуг, направленных к ним, равны, то оуществует точный эффективный алгоритм. В работе рассмотрена задача компоновки, когда G - трехуровневое дерево и объемы модулей равны. Эта постановка сведена к одной из вспомогательных задач и решается с трудоемкостью $O(n^3)$. Связи между данными часто имеют более сложный вид, чем дерево. В диссертации показано, что эти случаи могут быть сведены к рассмотренным.

В четвертой главе проводится анализ программного комплекса на примере варианта подсистемы АСУ-Росгеология. Подсистема ориентирована на использование моделей ЭВМ ЕС, начиная с ЕС-1020. Программное обеспечение работает под управлением ДОС ЕС и имеет модульную структуру.

В работе приведены постановка и решение задачи выбора варианта компоновки программных модулей на основе предложенного в диссертации подхода. Показано, что полученные варианты компоновки позволяют значительно уменьшить число обменов между оперативной памятью и внешней памятью. Выигрыш во временных затратах линейно зависит от частоты использования программного комплекса.

О С Н О В Н Ы Е Р Е З У Л Ь Т А Т Ы И В Ы В О Д Ы

В диссертации получены следующие основные результаты:

1. Проведен анализ проблемы эффективного использования вычислительной техники и выделены направления исследования.
2. Исследован класс детерминированных задач составления расписаний в одно- и двухпроцессорном вычислительном комплексе при одновременном поступлении заявок и ограничении предшествования в виде дерева или параллельно-последовательной сети. На основе исследования установленных свойств задач этого класса разработан общий эффективный метод нахождения оптимального расписания.
3. Проведена формализация задачи организации данных в рамках древовидной оверлейной структуры. Предложенная модель пред-

полагает минимизацию среднего числа обменов между оперативной и внешней памятью при ограничении на общий объем оперативной памяти, предоставляемой для работы комплекса.

4. Исследованы различные случаи сформулированной в работе задачи организации данных в зависимости от структуры связей между модулями и предложены эффективные методы решения: для некоторых точные алгоритмы, для других - приближенные алгоритмы с гарантированной относительно погрешностью целевой функции, а для третьих - приближенные алгоритмы с гарантированной относительной погрешностью как целевой функции, так и ограничения на память.

Показано, что в случаях, когда связи между данными имеют вид более сложный, чем дерево, задача может быть сведена к рассмотренным в диссертации случаям.

5. На основе исследования задачи организации данных предложена методика выбора вариантов структуры компоновки блоков данных и приведен пример ее реализации при выборе варианта оверлейной структуры программного комплекса.

На основе результатов, полученных в диссертации, а также на основе опыта проведенного решения практических задач, могут быть сделаны следующие выводы:

1. Построение дискретных оптимизационных моделей распределения ресурсов в вычислительных системах позволяет структуризовать процесс проектирования вычислительных комплексов.

2. Решение задач планирования при рассмотренных видах ограничений предшествования является в большом числе случаев достаточно эффективным. Поэтому практические задачи целесообразно сводить к задачам рассмотренной структуры.

3. Применение предлагаемой в диссертации методики, разработанных моделей и алгоритмов, а также рекомендаций по организации данных благодаря структуризации процесса практического проектирования систем программного обеспечения позволяет повысить эффективность использования оперативной памяти и, тем самым, эффективную производительность вычислительных систем.

ПУБЛИКАЦИИ

1. Левин М.Ш., Левнер Е.В. Об эффективном решении задачи

Беллмана-Джонсона на сети, имеющей вид дерева. - Автоматика и телемеханика, 1978, № 10, стр. 110-118.

2. Левин М.Ш., Мошкович Л.И. Задача размещения данных на магнитных дисках. - В кн.: Тр. Науч.-исслед. и проектного ин-та систем управления Госснаба СССР, вып. 7.- Тула, 1979, с. 60-65.

3. Левин М.Ш., Гильберштейн А.П. Информационно-поисковая система "ИПС-Геология". - В кн. Разработка и создание АСУ-Росгеология, М., ВГМС, 1979, Вып. 4 (18), с. 41-54.

4. Левин М.Ш. Об эффективном решении некоторых задач теории расписаний на сетях. - Кибернетика, 1980, № 1, с. 134-135.

Кроме того

5. Левин М.Ш. Одна экстремальная задача организации данных. - Изв. АН СССР, техн. киберн., 1981, № 5, с. 103-112.

6. Левин М.Ш. Приближенное решение одной задачи дискретного программирования. - Тезисы докладов I Всесоюзного совещания "Статистический и дискретный анализ нечисловой информации, экстремальные оценки и дискретная оптимизация". - Алма-Ата, 1981, с. 188-189.

7. Левин М.Ш. Детерминированные задачи планирования при идентичных процессорах и одновременном поступлении заявок. - Изв. АН СССР, техн. киберн., 1982, № 4, с. 51-57.

Подписано в печать 21.10.81 Т-28005 Формат 60х90¹/16

Печать офсетная

Усл.печ.л. 1,0

Уч.-изд.л. 0,96

Тираж 125 экз.

Зак. 6884

Производственно-издательский комбинат ВИНТИ

Люберцы, Октябрьский проспект, 403