

УДК 519.2

ОДНА ЭКСТРЕМАЛЬНАЯ ЗАДАЧА ОРГАНИЗАЦИИ ДАННЫХ

ЛЕВИН М. Ш.

Введение. При иерархической организации данных может возникнуть следующая ситуация. Заданы n модулей данных (блоков программ, элементов многоуровневых справочников или других поисковых структур), связанных между собой переходом с возвратом. Структура связи между модулями имеет вид дерева. Каждый модуль характеризуется требуемым объемом оперативной памяти, каждая связь — средним числом переходов. Требуется так разбить модули на сегменты, чтобы максимизировать взвешенную сумму связей, локализованных внутри сегментов, при ограничении на общий объем оперативной памяти, предоставляемый для работы с данными. В предположении, что для каждого модуля переходы к связанным с ним другим модулям, находящимся на следующем уровне иерархии, независимы, рассмотренной ситуации можно поставить в соответствие следующую древовидную модель.

Пусть вершины дерева соответствуют модулям, веса вершин — требуемому объему оперативной памяти, дуги — связям между модулями, веса дуг — средним частотам переходов. Тогда разбиение модулей на сегменты заключается в выборе стягиваемых дуг во взвешенном ориентированном дереве.

В настоящей работе проводятся анализ различных случаев задачи в зависимости от структуры дерева и построение эффективных алгоритмов решения, основанных на переходе от простейших постановок ко все более сложным. Даже в простейшем случае двухуровневого дерева рассматриваемая задача не проще задачи о рюкзаке, которая принадлежит к классу универсальных переборных или NP -полных проблем [1, 2]. Поэтому предлагаются приближенные эффективные алгоритмы с гарантированной относительной погрешностью [2, 3].

1. Математическая постановка задачи. Пусть задано направленное дерево $G=(A, \Gamma)$ с ориентацией дуг от корня к висячим вершинам, где A — множество вершин, Γ — многозначное отображение A в A . Каждая вершина $a \in A$ имеет вес $v(a) \geq 0$. Дугу из a в b обозначим (a, b) , корень дерева — a_0 . Введем для каждого пути $L(a_1, a_l) = \langle a_1, \dots, a_i, \dots, a_l \rangle$ ($a_j \in$

$\in \Gamma a_{j-1}, j=2, l$) вес $v(L(a_1, a_l)) = \sum_{i=1}^l v(a_i)$. Назовем весом графа величину

$v(G) = \max_{b \in B} v(L(a_0, b))$, где $B = \{a \in A \mid \Gamma a = \emptyset\}$.

Пусть $G_a = (A_a, \Gamma)$ — поддереву с корнем в $a \in A$ (в A_a кроме a входят все вершины из A , достижимые из a). Назовем «хвостом» вершины $a \in A$ граф $(\{A_a \setminus a\}, \Gamma)$, а весом «хвоста» вершины a — величину $v(a) = v(G_a) - v(a)$. Очевидно, что $\hat{v}(a) = \max_{b \in \Gamma a} v(G_b)$. Заметим, что вершина $a \in A$ может, в свою очередь, представлять собой взвешенный граф и вес a равен весу этого графа. Будем говорить, что граф G является допустимым, если $v(G) \leq V$, где V — ограничение на вес графа.

Пусть $|A|=n$. Введем множество дуг $U=\{u_1, \dots, u_i, \dots, u_{n-1}\}$. Обозначим через $G=(A, \Gamma)$ граф, полученный из G склеиванием вершин $a, b \in A, b \in \Gamma a$, т. е. в G вместо вершин a и b входит вершина $I=I(a, b)$ с весом $v(I(a, b))=v(a)+v(b)$, $\Gamma I=\{\Gamma a \cup \Gamma b\} \setminus b$. При этом множество дуг в $G: U=U \setminus (a, b)$. Очевидно, что операция преобразования графа при склеивании вершин ассоциативна, т. е. порядок склеивания вершин (стягивания дуг) несуществен. Введем переменную x_i ($i=1, n-1$), которая равна единице, если дуга u_i стягивается, и нулю — в противном случае. Тогда множество пар склеиваемых вершин можно задать булевым вектором $x=(x_1, \dots, x_i, \dots, x_{n-1})$. Припишем каждой дуге вес $w(u_i) \geq 0$.

Рассмотрим критерий

$$W(x) = \sum_{i=1}^{n-1} x_i w(u_i). \quad (1.1)$$

С учетом стягивания дуг для любого $a \in A$ вес вершины и вес «хвоста» можно рассматривать как функцию вектора x : $v(a, x)$, $\hat{v}(a, x)$. Таким образом, задача состоит в следующем.

П о с т а н о в к а 1. Определить вектор x , который максимизирует (1.1) при выполнении условия допустимости графа:

$$v(a_0, x) + \hat{v}(a_0, x) \leq V. \quad (1.2)$$

Пусть V^- — ограничение на вес вершины a_0 , V^+ — ограничение на вес «хвоста» a_0 , $V^- + V^+ = V$.

П о с т а н о в к а 2. Максимизировать $W(x)$ при условиях

$$v(a_0, x) \leq V^-, \quad \hat{v}(a_0, x) \leq V^+. \quad (1.3)$$

Для трехуровневого дерева, например, задача в постановке 1 имеет вид:

$$\begin{aligned} & \text{максимизировать } \sum_{a \in A \setminus a_0} x_a w_a \quad \text{при условии} \\ & v(a_0) + \sum_{a \in \Gamma a_0} x_a \left[v(a) + \sum_{b \in \Gamma a} x_b v(b) \right] + \max_{a \in \Gamma a_0} \left\{ (1-x_a) \left[v(a) + \sum_{b \in \Gamma a} x_b v(b) \right] + \right. \\ & \quad \left. + \max_{b \in \Gamma a} [(1-x_b) v(b)] \right\} \leq V, \end{aligned}$$

где для каждого $a \in A$ x_a, w_a соответствуют дуге, входящей в вершину.

Определим приближенное решение задачи. Пусть x^* — точное решение задачи (1.1), (1.2) или (1.1), (1.3). Рассмотрим вектор относительной погрешности [3] с компонентами

$$\alpha(x) = \max [0, (W(x^*) - W(x)) / W(x^*)],$$

$$\beta(x) = \max [0, (v(x) - V) / V],$$

$$v(x) = v(a_0, x) + \hat{v}(a_0, x), \quad W(x^*) \neq 0.$$

Задача состоит в определении при фиксированных $\varepsilon, \delta \in [0, 1]$ такого x^0 , что

$$\alpha(x^0) \leq \varepsilon, \quad \beta(x^0) \leq \delta. \quad (1.4)$$

Отметим, что если задача (1.1), (1.2) не имеет решения или $W(x^*)=0$, то это можно определить за $O(n^2)$ действий.

2. Простейшие частные случаи. Рассмотрим приближенные алгоритмы решения простейших частных случаев поставленной задачи, гарантирую-

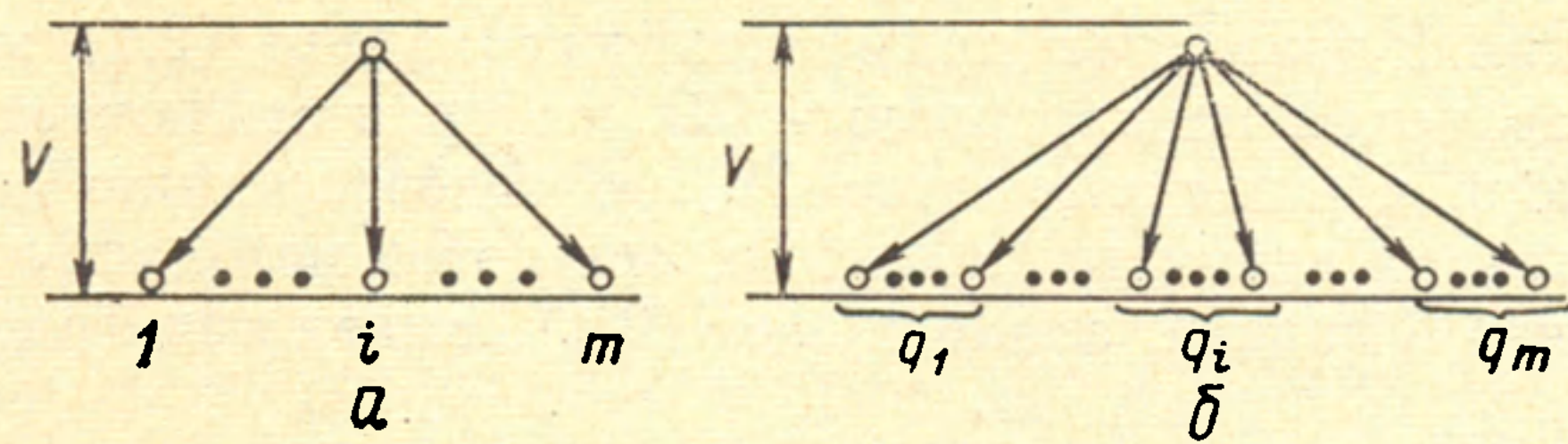


Рис. 1

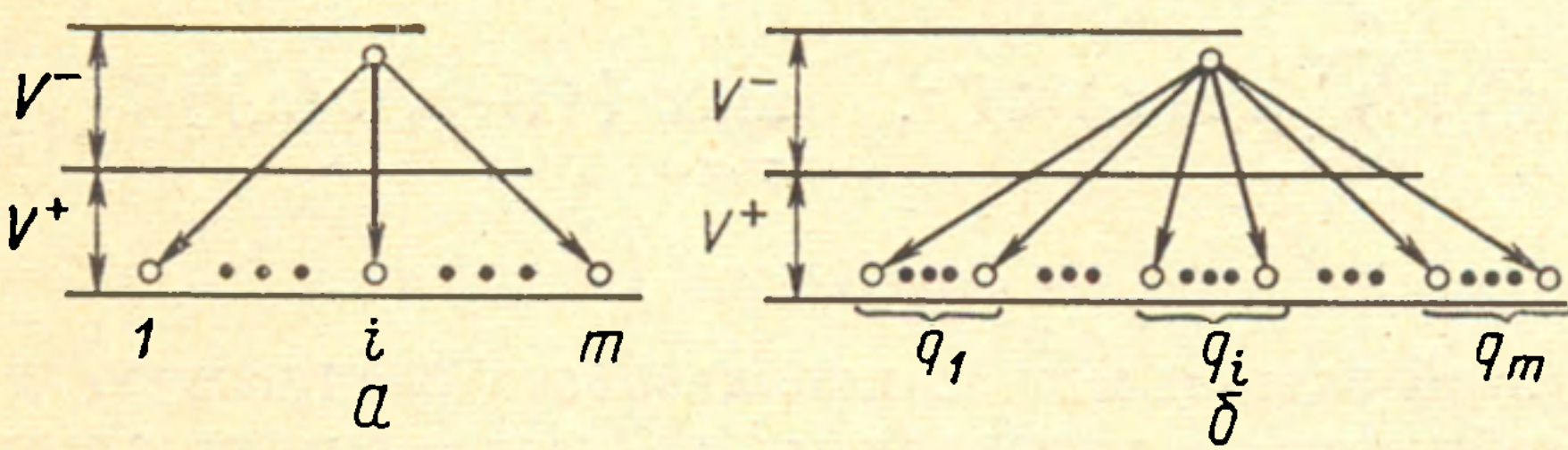


Рис. 2

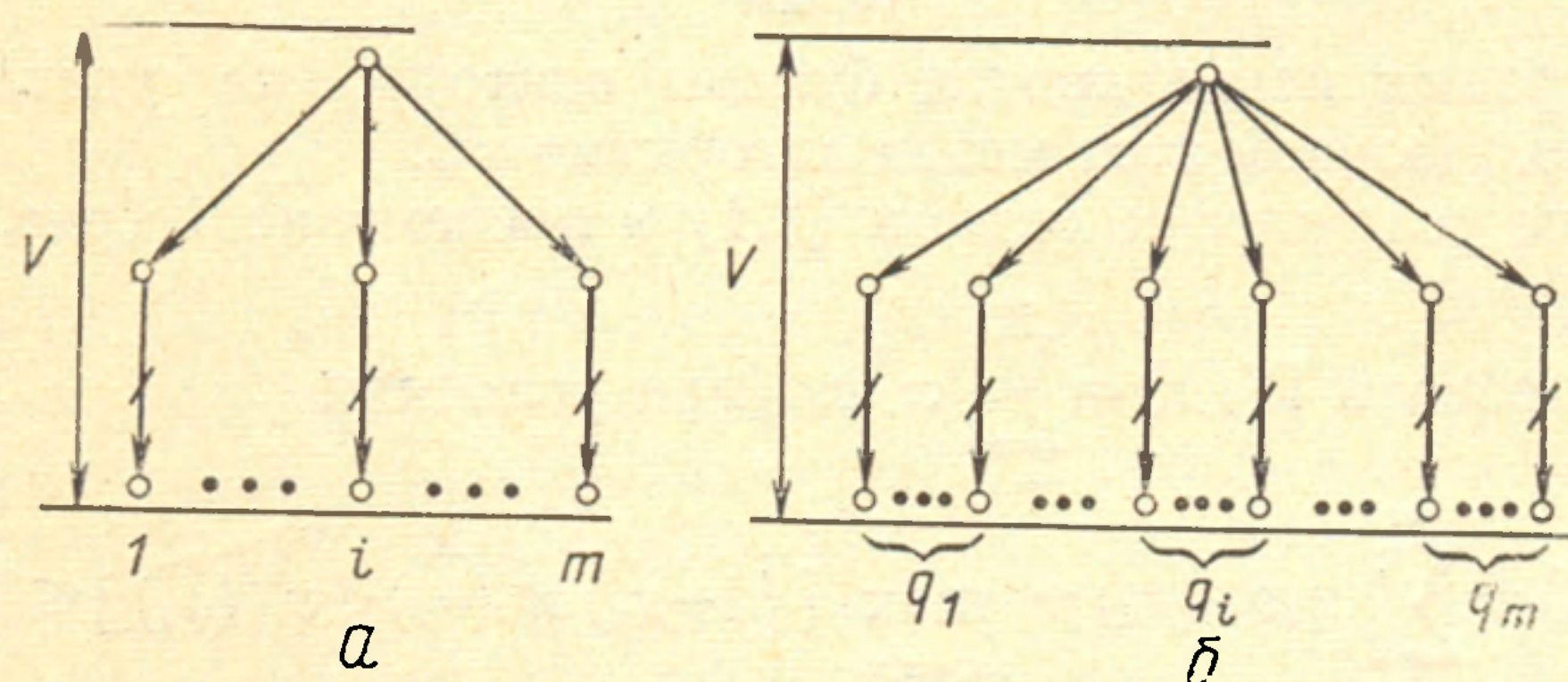


Рис. 3

щие заданную относительную ошибку целевой функции в соответствии с (1.4). Пусть G представляет собой двухуровневое дерево, $\Gamma a_0 = \{a_1, \dots, a_i, \dots, a_m\}$, u_i — дуга (a_0, a_i) , $w(u_i) = w_i$. В этом случае

$$v(a_0, x) = v(a_0) + \sum_{i=1}^m x_i v(a_i),$$

$$\hat{v}(a_0, x) = \max_{1 \leq i \leq m} [(1-x_i) v(a_i)]$$

и задача в постановке 1 будет иметь следующий вид (рис. 1, а):

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

$$v(a_0) + \sum_{i=1}^m x_i v(a_i) + \max_{1 \leq i \leq m} [(1-x_i) v(a_i)] \leq V, \quad x_i = 0 \text{ или } 1. \quad (2.1)$$

Пусть вершины множества Γa_0 перенумерованы так, что $v(a_i) \geq v(a_j)$ при $1 \leq i < j \leq m$. Тогда очевидно следующее.

У т в е р ж д е н и е 1. Для любого x

$$\hat{v}(a_0, x) = v(a_{j_0}),$$

где j_0 — наименьший индекс среди индексов нулевых компонент вектора x .

Утверждение 1 позволяет применить для задачи (2.1) модификацию «алгоритма разбиения на интервалы» [2], которая заключается в предварительном упорядочении элементов Γa_0 по невозрастанию весов и учете в процессе решения величины $v(a_{j_0})$. Назовем этот метод алгоритмом 1.

В постановке 2 задача, аналогичная (2.1), имеет вид (рис. 2, а):

$$\begin{aligned} &\text{максимизировать } W(x) = \sum_{i=1}^m x_i w_i \text{ при условиях} \\ &v(a_0) + \sum_{i=1}^m x_i v(a_i) \leq V^-, \quad \max_{1 \leq i \leq m} [(1-x_i)v(a_i)] \leq V^+, \quad (2.2) \\ &x_i = 0 \text{ или } 1. \end{aligned}$$

Для решения этой задачи можно использовать следующую модификацию «алгоритма разбиения на интервалы»: на каждом шаге алгоритма дополнительно проверяется условие $\max_{1 \leq i \leq m} [(1-x_i)v(a_i)] \leq V^+$, и из дальней-

шего рассмотрения исключаются булевы векторы, не удовлетворяющие этому условию. Назовем этот метод алгоритмом 2.

Рассмотрим усложнение задачи (2.1) в постановке 1 (рис. 3, а):

$$\begin{aligned} &\text{максимизировать } W(x) = \sum_{i=1}^m x_i w_i \text{ при условиях} \\ &v(a_0) + \sum_{i=1}^m x_i v^-(a_i) + \max_{1 \leq i \leq m} [(1-x_i)v^-(a_i) + v^+(a_i)] \leq V, \quad (2.3) \\ &x_i = 0 \text{ или } 1. \end{aligned}$$

Эта задача соответствует случаю, когда для каждого $a_i \in \Gamma a_0$ вес равен $v^-(a_i)$, каждая вершина a_i имеет одного последователя с весом $v^+(a_i)$ (это вес «хвоста» вершины a_i , равный $\hat{v}(a_i)$) и возможно склеивание дуг вида (a_0, a_i) . При этом

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

Пусть вершины множества Γa_0 перенумерованы так, чтобы $v^-(a_i) + v^+(a_i) \geq v^-(a_j) + v^+(a_j)$ при $1 \leq i < j \leq m$. Тогда очевидно следующее утверждение.

У т в е р ж д е н и е 2. Для любого x

$$\hat{v}(a_0, x) = \max[v^-(a_{j_0}) + v^+(a_{j_0}), \max_{\substack{1 \leq i_0 \leq j_0 \\ i_0 \in \{i | x_i = 1\}}} v^+(a_{i_0})],$$

где j_0 определяется как и в утверждении 1.

Утверждение 2 позволяет применить для решения задачи (2.3) алгоритм, аналогичный алгоритму 1.

Задача (2.3) в постановке 2 имеет вид (рис. 4, а):

$$\begin{aligned} &\text{максимизировать } W(x) = \sum_{i=1}^m x_i w_i \text{ при условиях} \\ &v(a_0) + \sum_{i=1}^m x_i v^-(a_i) \leq V^-, \quad \max_{1 \leq i \leq m} [(1-x_i)v^-(a_i) + v^+(a_i)] \leq V^+, \quad (2.4) \\ &x_i = 0 \text{ или } 1. \end{aligned}$$

Эта задача решается аналогично задаче (2.2).

Несложно показать, что оценки трудоемкости и памяти методов решения приведенных в данном разделе задач аналогичны оценкам «алгоритма разбиения на интервалы» и составляют $O(m^2/\epsilon)$.

3. Простейшие задачи с дополнительным выбором реализации. Рассмотрим задачи, являющиеся очевидным обобщением задач, приведенных в предыдущем разделе. Предположим, что каждая вершина $a_i \in \Gamma a_0$ представлена q_i вариантами реализации, отличающимися один от другого весами (вершины — $v(a_{ij})$ или $v^-(a_{ij})$; «хвоста» — $v^+(a_{ij})$; дуги, входящей в вершину, — $w(a_{ij})$) и числом $\bar{w}(a_{ij})$, характеризующим взвешенную сумму дуг,

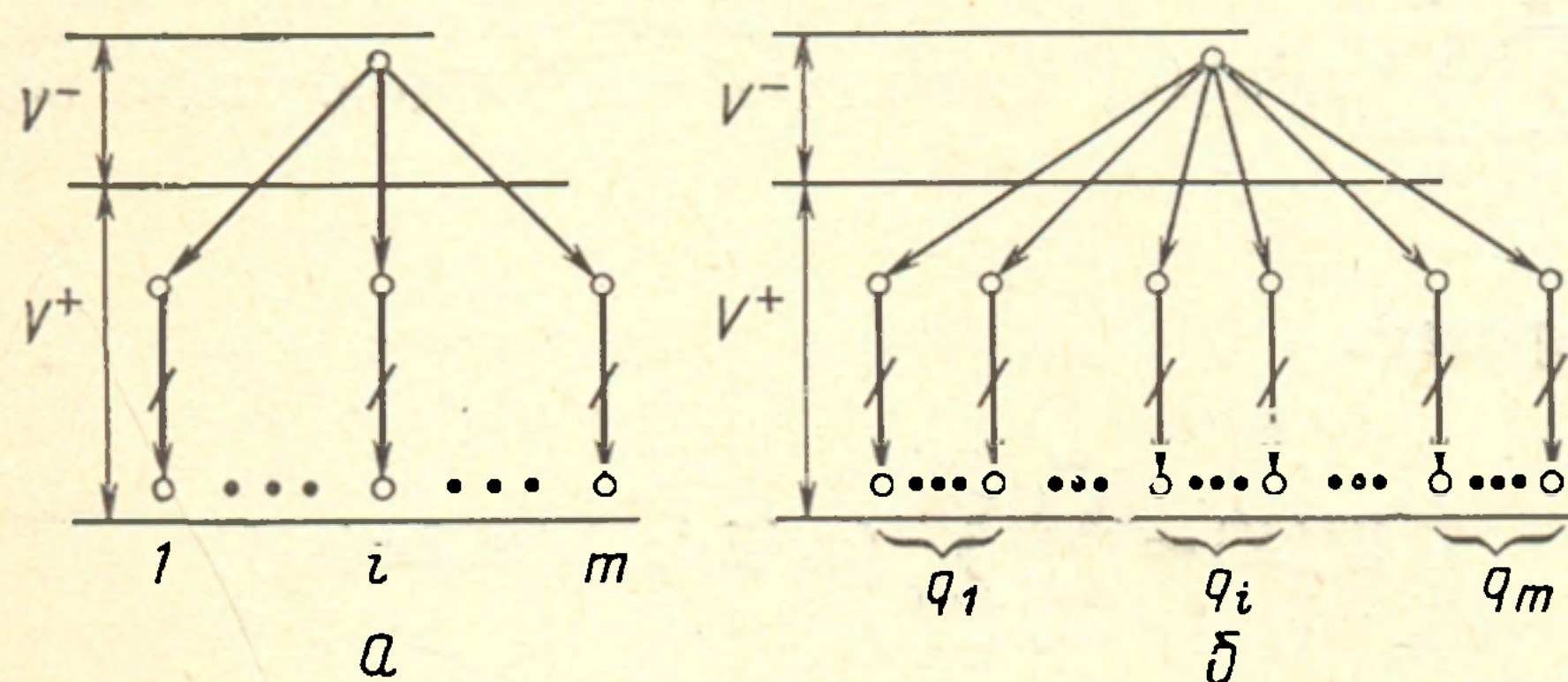


Рис. 4

стянутых в графе, соответствующем данной вершине. Вариант имеет индекс j ($j=1, q_i \forall i$). Задача заключается не только в выборе объединяемых с корнем вершин, но и в предварительном выборе варианта реализации.

Введем для задач, рассматриваемых здесь, набор булевых векторов $X = \{x_{ij} = (x_{ij}^1, x_{ij}^2) \mid x_{ij}^1, x_{ij}^2 = 0 \text{ или } 1\}$, где x_{ij}^1 соответствует выбору варианта реализации j для вершины i , x_{ij}^2 — объединению выбранного варианта с корнем. При этом в постановке задач необходимо учитывать следующие

общие ограничения: $\sum_{j=1}^{q_i} x_{ij}^1 = 1 \forall i$; $x_{ij}^2 \leq x_{ij}^1 \forall i, j$. Задача, получающаяся

таким обобщением из (2.1), имеет вид (рис. 1, б):
максимизировать

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

при условии

$$v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v(a_{ij}) + \max_{i,j} [(x_{ij}^1 - x_{ij}^2) v(a_{ij})] \leq V. \quad (3.1)$$

Задача с дополнительным выбором реализаций, соответствующая (2.2), формулируется аналогично (рис. 2, б):
максимизировать

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

при условиях

$$v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v(a_{ij}) \leq V^-,$$

$$\max_{i,j} [(x_{ij}^1 - x_{ij}^2) v(a_{ij})] \leq V^+. \quad (3.2)$$

В следующем разделе потребуются другие две задачи рассматриваемого типа, соответствующие (2.3), (2.4). Эти задачи имеют вид соответственно (рис. 3, б, 4, б):

максимизировать

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

при условии

$$v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v^-(a_{ij}) + \max_{i,j} [(x_{ij}^1 - x_{ij}^2) v^-(a_{ij}) + x_{ij}^1 v^+(a_{ij})] \leq V; \quad (3.3)$$

максимизировать

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

при условиях

$$\begin{aligned} v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v^-(a_{ij}) &\leq V^-, \\ \max_{i,j} [(x_{ij}^1 - x_{ij}^2) v^-(a_{ij}) + x_{ij}^1 v^+(a_{ij})] &\leq V^+. \end{aligned} \quad (3.4)$$

Для решения задач (3.2), (3.4) можно использовать модификацию приближенного метода решения «лестничной задачи о рюкзаке» [2]. При этом на каждом шаге получаем большее число вариантов по сравнению с методом из [2] за счет рассмотрения векторов x_{ij} , равных (1,0), (1,1), (0,0), вместо булевых переменных, принимающих значения 0 и 1, как в [2]. Кроме того, на каждом шаге исключаются варианты, не удовлетворяющие второму условию. Оценка трудоемкости и памяти такого алгоритма

$O\left(m\epsilon^{-1} \sum_{i=1}^m q_i\right)$ и $O(m^2\epsilon^{-1} \max_{1 \leq i \leq m} q_i)$ соответственно. Назовем та-

кой метод решения задачи (3.4) алгоритмом 3.

Задачи (3.1) и (3.3) требуют особого подхода. Рассмотрим его на примере (3.3).

В качестве решения (3.3) будем искать такое X^0 , чтобы выполнялись соотношения типа (1.4). Введем $\delta_0 = \delta/2$, $r = \lceil 1/\delta_0 \rceil + 1$. Для решения задачи (3.3) рассмотрим аналогичную задачу (3.4) при следующих правых частях ограничений на ресурс: вместо V^- поставим $V\delta_0 l$, вместо V^+ поставим $V\delta_0(r-l)$, где $l=0, r$.

Лемма 1. Если $r+1$ раз решить задачу (3.4) с помощью алгоритма 3 при $l=0, r$ с точностью ϵ и выбрать из полученного набора решений $\{X_l\}$ решение $\bar{X}$ с наибольшим значением $W(X)$, то это решение будет искомым X^0 для задачи (3.3).

Доказательство. Пусть X^* — оптимальное решение задачи (3.3). Существует такое l_0 , что для X^* выполняется соотношение

$$V\delta_0(l_0-1) \leq v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v^-(a_{ij}) \leq V\delta_0 l_0.$$

Среди наборов $\{X_l\}$ имеется решение X_{l_0} , для которого $V^- = V\delta_0 l_0$, $V^+ = V\delta_0(r-l_0)$. Для решения X^* рассмотрим задачу (3.4) с

$$V^- = v(a_0) + \sum_{i=1}^m \sum_{j=1}^{q_i} x_{ij}^2 v^-(a_{ij}),$$

$$x_{ij} = (x_{ij}^1, x_{ij}^2) \in X^*, \quad V^+ = V - V^-$$

и назовем ее задачей 1.

В качестве задачи 2 рассмотрим задачу (3.4), соответствующую X_{l_0} . Очевидно, что X^* является оптимальным решением задачи 1. Пусть Y^* — оптимальное решение задачи 2. Так как в задаче 2 значение ограничений не меньше, чем в задаче 1, то

$$W(Y^*) \geq W(X^*). \quad (3.5)$$

Поскольку X_{l_0} получено при решении задачи 2 алгоритмом 3, то $W(Y^*) \geq W(X_{l_0})$ и $W(X_{l_0}) \geq W(Y^*)(1-\epsilon)$, т. е. с учетом (3.5) $W(X_{l_0}) \geq W(X^*)(1-\epsilon)$. Так как для решения X верно по условию леммы 1 соотношение $W(X) \geq W(X_l) \forall l$, то $\alpha(X) \leq \epsilon$. Очевидно, что $\beta(\bar{X}) \leq \delta$.

Назовем подход к решению задачи (3.3), содержащийся в лемме 1, алгоритмом 4. Оценки его трудоемкости и памяти соответственно

$$O\left(m(\epsilon\delta)^{-1} \sum_{i=1}^m q_i\right) \text{ и } O(m^2\epsilon^{-1} \max_{1 \leq i \leq m} q_i).$$

4. Решение задачи для многоуровневого дерева. Пусть G является $(k+1)$ -уровневым деревом. Введем для каждого $a \in A$: $m(a) = |\Gamma a|$, $\eta(a) = |A_a \setminus \{b \in A_0 \mid \Gamma b = \emptyset\}|$. Выделим из A множества $M_0 = \{a \in A \mid \Gamma a = \emptyset\}$, $M_1 = \{a \in A \mid \Gamma a \leq M_0\}$, $M_2 = \{a \in A \mid \Gamma a \subseteq \{M_0 \cup M_1\}\}$ и т. д.

Введем

$$\delta_0 = \frac{\delta}{2\eta(a_0)}, \quad r = \left\lceil \frac{1}{\delta_0} \right\rceil + 1, \quad \epsilon_0 = \frac{\epsilon}{\eta(a_0)}, \quad V_l = V\delta_0 l, \quad l = \overline{0, r}.$$

Алгоритм решения задачи имеет вид.

1-й шаг. Сформулируем для каждой вершины $a_i \in \bar{M}$, как для корневой, задачи (2.2) при $V^- = V_{l_1}$, $V^+ = V_{l_2}$ ($0 \leq l_1 \leq r+2$, $0 \leq l_2 \leq r+2-l_1 \forall l_1$) и решим с помощью алгоритма 2 с точностью ϵ_0 . Перенумеруем полученные для каждой вершины a_i решения с помощью индекса j и введем следующие обозначения характеристик:

$$v^-(a_{ij}) = v(a_i, x), \quad v^+(a_{ij}) = \hat{v}(a_i, x),$$

$$\bar{w}(a_{ij}) = W(x), \quad w(a_{ij}) = w(b, a_i),$$

$$b = \Gamma^{-1} a_i, \quad q_i = (r+3)^2/2 \text{ для всех } i.$$

μ -й шаг ($\mu = \overline{2, k-1}$). Воспользовавшись результатами предыдущего шага, сформулируем для каждой вершины $a_i \in \bar{M}_\mu$, как для корневой, задачу (3.4) при $V^- = V_{l_1}$, $V^+ = V_{l_2}$ ($0 \leq l_1 \leq r+2\eta(a_i)$, $0 \leq l_2 \leq r+2\eta(a_i)-l_1 \forall l_1$) и решим с помощью алгоритма 3 с точностью ϵ_0 . Перенумеруем полученные для каждой вершины решения индексом j и введем следующие обозначения характеристик:

$$v^-(a_{ij}) = v(a_i, x), \quad v^+(a_{ij}) = \hat{v}(a_i, x),$$

$$\bar{w}(a_{ij}) = W(X), \quad w(a_{ij}) = w(b, a_i),$$

$$b = \Gamma a_i, \quad q_i \leq \frac{1}{2}(r+2\eta(a_i)+1)^2 \text{ для всех } i.$$

k -й шаг. Воспользовавшись результатами предыдущего шага, сформулируем для вершины a_0 задачу (3.3) при ограничении в условии, равном

$V(1+2\delta_0(\eta(a_0)+1))$, и решим ее с помощью алгоритма 4 с точностью ε_0 и $2\delta_0$.

Шаги изложенного метода, который будем называть алгоритмом 5, иллюстрируются на рис. 5, 6 и 3, б соответственно.

В результате получим решение $x^0=(x_1^0, \dots, x_{n-1}^0)$. Обозначим компоненты вектора $x=(x_1, \dots, x_{n-1})$, относящиеся к дугам в поддереве G_a ($a \in A$), через $x(a)$.

Лемма 2. Для x^0 выполняется $v(x^0) \leq V + \delta V$.

Доказательство. Во всех постановках задач в алгоритме 5 правая часть условий, связанных с ресурсом, не больше величины $V + 2\delta_0\eta(a_0)V$, так как $\eta(a_0) \geq \eta(a) \forall (a) \in A$.

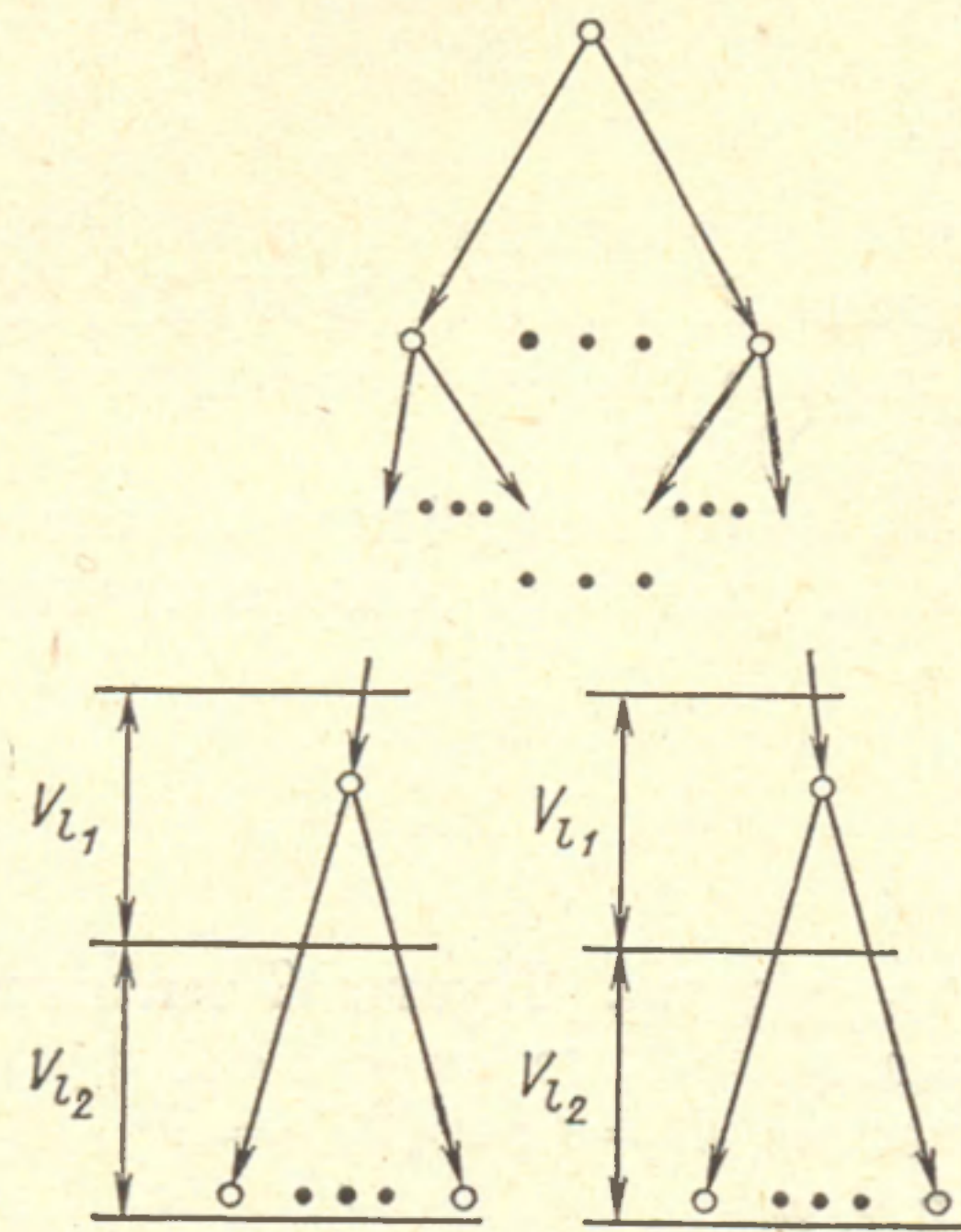


Рис. 5

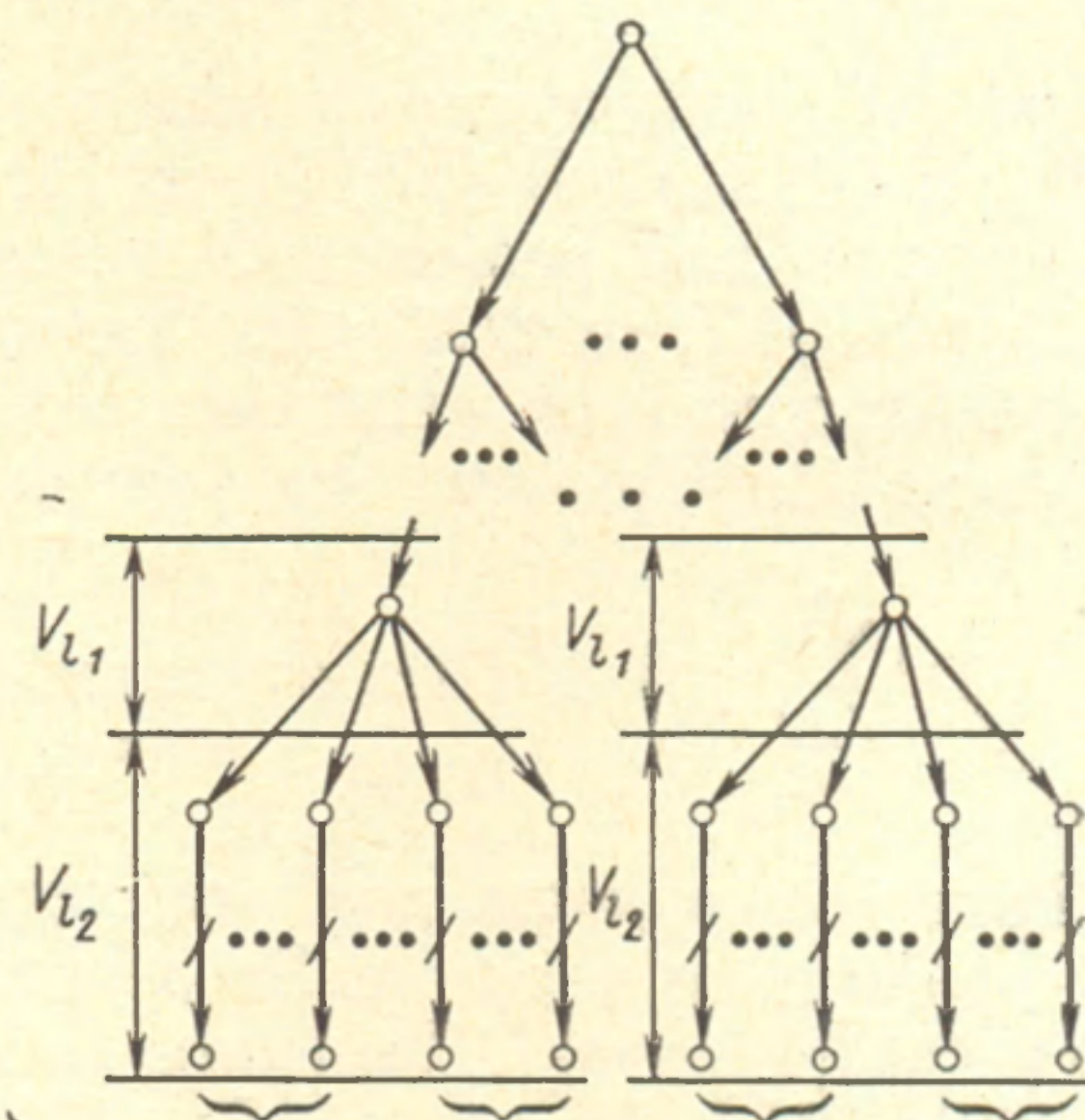


Рис. 6

Лемма 3. При решении задач для вершины $a \in M_\mu$ на μ -м шаге ($\mu = 1, k-1$) алгоритма 5 среди полученных вариантов имеется такое решение, что для соответствующего ему $\tilde{x}(a)$ $\alpha(\tilde{x}(a)) \leq \varepsilon_0 \eta(a)$.

Доказательство. Проведем его методом математической индукции.

1. За основание индукции примем $\mu=1$. Тогда нужно показать, что при решении задачи (2.2) для вершины $a \in M_1$, среди полученных вариантов имеется такое x , что для соответствующего $\tilde{x}(a)$ выполняется $\alpha(\tilde{x}(a)) \leq \varepsilon_0$, так как $\eta(a)=1$. Рассмотрим набор компонент оптимального решения $x^*(a)$. Ему соответствуют величины $W(x^*(a))$ и l_1^*, l_2^* такие, что $V_{l_1^*-1} \leq v(a, x^*(a)) \leq V_{l_1^*}$, $V_{l_2^*-1} \leq v(a, x^*(a)) \leq V_{l_2^*}$. Среди полученных вариантов имеется некоторый $\tilde{x}(a)$, являющийся решением задачи (2.2) при $V^-=V_{l_1^*}$, $V^+=V_{l_2^*}$.

Пусть $\tilde{x}(a)$ — оптимальное решение задачи (2.2) при аналогичных ограничениях. Тогда $W(\tilde{x}(a)) - W(\tilde{x}(a)) \leq \varepsilon_0 W(\tilde{x}(a))$, $W(x^*(a)) \leq W(\tilde{x}(a))$. Следовательно, или $W(\tilde{x}(a)) \geq W(x^*(a))$, или $W(x^*(a)) - W(\tilde{x}(a)) \leq \varepsilon_0 W(x^*(a))$, так как

$$W(\tilde{x}(a)) \geq (1 - \varepsilon_0) W(\tilde{x}(a)) \geq (1 - \varepsilon_0) W(x^*(a)).$$

2. Предположим, что лемма верна для $\mu-1$ ($\mu=2, k-2$). Покажем, что она верна при решении задачи (3.4) для вершины $a \in M_\mu$ на μ -м шаге алгоритма 5. Пусть $G_a = \{b_1, \dots, b_m\}$. Рассмотрим набор компонент оптимального решения $x^*(a) = \{x_1^*, \dots, x_m^*, x^*(b_1), \dots, x^*(b_m)\}$. Ему соответствуют величины $W(x^*(a))$ и l_1^*, l_2^* такие, что $V_{l_1^*-1} \leq v(a, x^*(a)) \leq V_{l_1^*}$, $V_{l_2^*-1} \leq v(a, x^*(a)) \leq V_{l_2^*}$. Рассмотрим набор $x(a) = \{x_1^*, \dots, x_m^*, \tilde{x}(b_1), \dots$

..., $\bar{x}(b_m)\}$, где наборы компонент $\bar{x}(b_i)$ при $i=1, m$ соответствуют предыдущему шагу. Набору $x(a)$ соответствуют такие l_1, l_2 , что $v(a, x(a)) \leq V_{l_1}$, $\hat{v}(a, x(a)) \leq V_{l_2}$. Индексам l_1, l_2 соответствует некоторый вариант $\bar{x}(a)$ решения задачи (3.4), т. е. $v(a, \bar{x}(a)) \leq V_{l_1}$, $\hat{v}(a, \bar{x}(a)) \leq V_{l_2}$. Пусть $\bar{x}(a)$ — оптимальное решение задачи (3.4) при l_1, l_2 . Тогда $W(\bar{x}(a)) \geq (1-\varepsilon_0)W(x(a))$. Пусть $W(x(a)) \geq W(x^*(a))$. Тогда $W(\bar{x}(a)) \geq (1-\varepsilon_0)W(x^*(a))$ и $\alpha(\bar{x}(a)) \leq \varepsilon_0$. Рассмотрим случай, когда $W(\bar{x}(a)) < W(x^*(a))$. Очевидно, что $W(x(a)) \leq W(\bar{x}(a))$. Согласно предположению индукции

$$|W(x^*(a)) - W(x(a))| \leq \left[\varepsilon_0 \sum_{i=1}^m \eta(b_i) \right] W(x^*(a)).$$

Если $W(\bar{x}(a)) \geq W(x(a))$, то

$$\alpha(\bar{x}(a)) \leq \varepsilon_0 \sum_{i=1}^m \eta(b_i).$$

Если $W(\bar{x}(a)) < W(x(a))$, то, так как $W(x(a)) - W(\bar{x}(a)) \leq \varepsilon_0 W(\bar{x}(a))$, имеем

$$W(x(a)) - W(\bar{x}(a)) \leq \varepsilon_0 W(\bar{x}(a)),$$

$$W(x^*(a)) - W(\bar{x}(a)) \leq \left[\varepsilon_0 \sum_{i=1}^m \eta(b_i) \right] W(x^*(a)) + \varepsilon_0 W(\bar{x}(a)) \leq$$

$$\leq \varepsilon_0 \left[1 + \sum_{i=1}^m \eta(b_i) \right] W(x^*(a)), \text{ т. е. } \alpha(\bar{x}(a)) \leq \varepsilon_0 \eta(a).$$

Лемма 4. При решении задачи (3.3) на k -м шаге алгоритма 5 получается решение X такое, что для соответствующего ему x^0 $\alpha(x^0) \leq \varepsilon_0 \eta(a_0)$.

Лемма 4 доказывается аналогично предыдущей.

Рассмотрим характеристики алгоритма 5. На первом шаге алгоритма к каждой вершине $a \in M_1$ порядка $1/\delta_0^2$ раз применяется алгоритм 2 с трудоемкостью $O(m^2(a)/\varepsilon_0)$. Общая трудоемкость шага $O\left((\varepsilon_0 \delta_0^2)^{-1} \sum_{a \in M_1} m^2(a)\right)$.

На μ -м шаге ($\mu=2, k-1$) к каждой вершине $a \in M_\mu$ применяется алгоритм 3 с трудоемкостью $O(m^2(a) \varepsilon_0^{-1} (1/\delta_0 + 2\eta(a))^4)$. Общая трудоемкость шага μ равна $O\left((\varepsilon_0 \delta_0^4)^{-1} \sum_{a \in M_\mu} m(a) (1+2\delta_0 \eta(a))^4\right)$. На k -м шаге к a_0 применяется

алгоритм 4 с трудоемкостью $O((m^2(a_0)/\varepsilon_0 \delta_0^3) (1+2\delta_0 \eta(a_0))^3)$.

Таким образом, оценка трудоемкости алгоритма 5 составляет $O(n^2 \eta^5(a_0)/\varepsilon \delta^4)$. Оценка памяти алгоритма 5 — $O(m^2 n^4/\varepsilon \delta^4)$, где $m = \max_{a \in A} m(a)$. Если G — трехуровневое дерево, алгоритм 5 имеет оценку трудоемкости $O(n^2 \eta^4(a_0)/\varepsilon \delta^3)$.

5. Дополнение. В случае, когда веса вершин множества $A \setminus a_0$ равны v , формальная постановка задачи для трехуровневого дерева, приведенная в п. 1, упрощается:

максимизировать $\sum_{a \in A \setminus a_0} x_a w_a$ при условии

$$\sum_{a \in \Gamma a_0} x_a (y_a + 1) + \max_{a \in \Gamma a_0} [(1-x_a)(y_a + 1) + \text{sign}(m(a) - y_a)] \leq \frac{V - v(a_0)}{v},$$

$$\forall a \in \Gamma a_0 \quad y_a = \sum_{b \in \Gamma a} x_b, \quad m(a) = |\Gamma a|. \quad (5.1)$$

Введем для каждого $a \in \Gamma a_0$ переменную $z(y_a)$, равную единице, если к a присоединяются y_a последователей (в порядке невозрастания весов направленных в них дуг). Таким образом, для каждого $a \in \Gamma a_0$ имеется $m(a) + 1$ вариантов, каждый из которых характеризуется числом присоединяемых последователей y_a ($0 \leq y_a \leq m(a)$) и величиной $w(y_a)$, равной сумме весов стянутых при этом дуг. Рассмотрим задачи:

$$\begin{aligned} & \text{максимизировать} \quad \sum_{a \in \Gamma a_0} \left[x_a w_a + \sum_{y_a=0}^{m(a)} z(y_a) w(y_a) \right] \\ & \text{при условиях} \quad \sum_{a \in \Gamma a_0} x_a \sum_{y_a=0}^{m(a)} z(y_a) (y_a + 1) \leq l, \\ & \max_{a \in \Gamma a_0} [(1 - x_a) z(y_a) (y_a + 1) + \text{sign}(m(a) - y_a)] \leq \left\lceil \frac{V - v(a_0)}{v} \right\rceil - l, \\ & \sum_{y_a=0}^{m(a)} z(y_a) = 1, \quad x_a \leq z(y_a) \quad \forall a \in \Gamma a_0, \\ & l = 0, \quad \lceil (V - v(a_0)) / v \rceil. \end{aligned} \quad (5.2)$$

Для решения задачи (5.2) при фиксированном l можно использовать модификацию приближенного метода решения задачи «лестничного рюкзака» [2] с оценками трудоемкости и памяти $O(m(a_0) \varepsilon^{-1} \sum_{a \in \Gamma a_0} m(a)) \leq O(n^2/\varepsilon)$. Для решения задачи (5.1) необходимо решить соответствующие (5.2) при всех l и выбрать решение с наибольшим значением целевой функции. Оценка трудоемкости и памяти этого метода — $O(n^3/\varepsilon)$.

ЛИТЕРАТУРА

1. Аха А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Левнер Е. В., Генс Г. В. Дискретные оптимизационные задачи и эффективные приближенные алгоритмы. — Препринт. М.: ЦЭМИ АН СССР, 1978.
3. Пемировский А. С., Юдин Д. Б. Сложность задач и эффективность методов оптимизации. М.: Наука, 1979.

Москва

Поступила в редакцию
8.XII.1980