

УДК 519.854

ПРИБЛИЖЁННОЕ РЕШЕНИЕ  
ЗАДАЧИ УПРАВЛЕНИЯ ПОСТАВКАМИ<sup>\*)</sup>

*А. В. Еремеев, А. А. Романова, В. В. Сервах, С. С. Чаухан*

В статье рассматривается задача оптимизации доставки продукции от поставщиков потребителям. Размер каждой открытой поставки ограничен снизу и сверху, размер потребления для каждого потребителя ограничен снизу, функции стоимости поставки линейны при ненулевых объёмах поставки. Предложена вполне полиномиальная аппроксимационная схема для этой задачи в случае одного потребителя и исследуется сложность задачи в общем случае.

**Введение**

В статье исследуется возможность нахождения решения с гарантированной оценкой точности для следующей задачи управления поставками. Поставщики доставляют некоторый продукт потребителям, причём количество продукции, которое может быть доставлено по открытой поставке, заключено в границах между минимальным и максимальным значениями, а стоимость, предложенная каждым поставщиком, есть линейная функция от объёма поставки. Формально задача может быть записана в следующем виде: найти минимум функции

$$\sum_{i=1}^n \sum_{j=1}^m k_{ij}(x_{ij}) \quad (1)$$

при условиях

$$\sum_{i=1}^n x_{ij} \geq A_j, \quad 1 \leq j \leq m, \quad (2)$$

$$\sum_{j=1}^m x_{ij} \leq M_i, \quad 1 \leq i \leq n, \quad (3)$$

$$x_{ij} \in \{0\} \cup [m_{ij}, M_i], \quad 1 \leq i \leq n; \quad 1 \leq j \leq m, \quad (4)$$

---

<sup>\*)</sup>Исследование выполнено при финансовой поддержке INTAS (проект 03-51-5501) и Российского гуманитарного научного фонда (проект 04-02-00238а).

где  $n$  — число поставщиков;  $m$  — число потребителей;  $A_j$  — минимальное значение продукта, требуемое потребителю  $j$ ;  $m_{ij}$  — минимальное значение продукта, которое поставщик  $i$  готов доставить потребителю  $j$ ;  $M_i$  — максимальное количество продукта, которое поставщик  $i$  может доставить. Переменная  $x_{ij}$  обозначает размер поставки поставщиком  $i$  потребителю  $j$ .

Стоимость доставки

$$k_{ij}(x_{ij}) = \begin{cases} 0 & \text{при } x_{ij} = 0; \\ a_{ij} + b_{ij}x_{ij} & \text{при } x_{ij} > 0. \end{cases}$$

Параметры  $a_{ij}, b_{ij}, m_{ij}, M_i, A_j$  предполагаются целочисленными и неотрицательными при всех  $i, j$ .

Стоимость решения  $x = (x_{ij})$ , заданную выражением (1), будем обозначать через  $f(x)$ .

Эта задача является ослабленной версией задачи управления поставками с вогнутой целевой функцией [3, 5], в которой (2) является равенством, а функции стоимости неотрицательны и вогнуты. Нахождение любого допустимого решения в постановке [5] является NP-трудной задачей, хотя существует псевдополиномиальный точный алгоритм её решения [3]. Как будет показано ниже, ослабленная версия задачи представляет больше возможностей для приближения.

Задача (1)–(4) NP-трудна, так как задача о рюкзаке является её частным случаем. Кроме того, сведение задачи разбиения показывает, что задача нахождения допустимого решения, удовлетворяющего ограничениям (2)–(4), NP-трудна даже в случае  $m = 2$ . Действительно, рассмотрим задачу разбиения как задачу распознавания: можно ли множество натуральных чисел  $\{p_1, p_2, \dots, p_N\}$  разбить на два подмножества с равными суммами? Если сумма  $\sum_{i=1}^N p_i$  нечётна, то ответ, очевидно, отрицательный. В противном случае, положим  $m = 2$ ,  $n = N$ ,  $m_{i1} = m_{i2} = M_i = p_i$ ,  $1 \leq i \leq N$  и  $A_1 = A_2 = \frac{1}{2} \sum_{i=1}^N p_i$ . При этом система (2)–(4) разрешима тогда и только тогда, когда требуемое разбиение существует.

Под  $\rho$ -приближённым алгоритмом будем понимать алгоритм, находящий приближённое решение, стоимость которого не более чем в  $\rho$  раз превышает стоимость оптимального решения (если задача разрешима). Соответствующее решение будем называть  $\rho$ -приближённым решением.

Под вполне полиномиальной аппроксимационной схемой (FPTAS) будем понимать семейство  $(1 + \varepsilon)$ -приближённых алгоритмов при всевоз-

можных  $\varepsilon > 0$  с временной сложностью, полиномиально зависящей от длины входа задачи и  $1/\varepsilon$ .

Основными результатами статьи являются теоремы 1 и 2.

**Теорема 1.** В случае  $m = 1$  существует вполне аппроксимационная полиномиальная схема для задачи (1)–(4) с временной сложностью  $O(n^3/\varepsilon + n^2)$ .

**Теорема 2.** Задачи (1)–(4) и нахождения  $p(n, m)$ -приближённого решения при любом полиноме  $p(n, m)$  с положительными коэффициентами NP-трудны в сильном смысле.

Здесь и далее в оценках временной сложности предложенных алгоритмов подразумевается вычислительная сложность машины RAM с произвольным доступом к памяти, когда стандартные арифметические операции имеют константную длительность (см., например, [1]). Все арифметические операции выполняются над числами с длиной записи, ограниченной сверху полиномом от длины входа в модели машины Тьюринга. Поэтому предложенные алгоритмы имеют полиномиальную временную сложность.

Для упрощения обозначений запишем задачу (1)–(4) в случае  $m = 1$  в более компактном виде: найти минимум функции

$$\sum_{i=1}^n k_i(x_i) \quad (5)$$

при условиях

$$\sum_{i=1}^n x_i \geq A, \quad (6)$$

$$x_i \in \{0\} \cup [m_i, M_i], \quad 1 \leq i \leq n. \quad (7)$$

Стоимость доставки

$$k_i(x_i) = \begin{cases} 0 & \text{при } x_i = 0; \\ a_i + b_i x_i & \text{при } x_i > 0, \end{cases} \quad (8)$$

где  $a_i \geq 0$  и  $b_i \geq 0$ ,  $1 \leq i \leq n$ .

Чтобы получить FPTAS для задачи (5)–(8), сначала построим эффективный жадный 2-приближённый алгоритм, а затем — псевдополиномиальный алгоритм, основанный на динамическом программировании. Этот подход аналогичен подходу, применённому для решения задачи о

рюкзаке на минимум [2]. В последнем разделе даётся доказательство теоремы 2.

В [4] для данной задачи была также построена вполне полиномиальная аппроксимационная схема, которая годится для более общего класса целевых функций и основана на подходе, подробно изложенном в [7]. В применении к постановке (5)–(8) эта схема имеет временную сложность  $O((n/\varepsilon)^2 \log_2(nk_{\max}) \log_2(k_{\max}))$ , где  $k_{\max} = \max_{i=1, \dots, n} k_i(M_i)$ . Единственное дополнительное условие то, что сумма  $a_i + b_i$ ,  $1 \leq i \leq n$  отделена от нуля положительной константой общей для всех индивидуальных задач. Как видно, по величине  $1/\varepsilon$  схема [4] имеет большую временную сложность по сравнению со схемой, рассмотренной в теореме 1, и время её работы зависит от числового входного параметра  $k_{\max}$  даже в модели вычислений RAM.

### 1. 2-приближённый жадный алгоритм

В этом разделе описывается жадный алгоритм для более общей задачи вида (5)–(7), предполагая, что стоимости доставки  $k_i(x_i)$ ,  $1 \leq i \leq n$ , заданы функциями

$$k_i(x_i) = \begin{cases} 0 & \text{при } x_i = 0; \\ a_i + g_i(x_i) & \text{при } x_i > 0, \end{cases} \quad (9)$$

где  $a_i \geq 0$  при любом  $i$ , а  $g_i(x_i)$  — вогнутая неотрицательная неубывающая функция при  $x_i \in [0, M_i]$ . Через  $y = (y_j)$  обозначим приближённое решение, построенное жадным алгоритмом;  $L$  — текущее количество продукта, которое необходимо доставить потребителю.

*Жадный алгоритм*

*Шаг 1.* Положить  $y_k := 0$ ,  $k = 1, 2, \dots, n$ .

*Шаг 2.* Положить  $L = A - \sum_{k=1}^n y_k$  и выбрать такое  $j \in \{1, 2, \dots, n\}$ , что  $y_j = 0$  и

$$\frac{k_j(\min\{\max\{L, m_j\}, M_j\})}{\min\{L, M_j\}} \leq \frac{k_s(\min\{\max\{L, m_s\}, M_s\})}{\min\{L, M_s\}} \quad (10)$$

для всех  $s$  таких, что  $y_s = 0$ . Если  $y_j > 0$  для всех  $j$ , то *Конец*.

*Шаг 3.* Положить  $y_j := \min\{\max\{L, m_j\}, M_j\}$ .

*Шаг 4.* Если  $L - y_j > 0$ , перейти на шаг 2. *Конец*.

Если после остановки алгоритма выполняется неравенство  $\sum_{k=1}^n y_k < A$ , то задача не имеет допустимых решений. Пусть  $T_g$  — верхняя граница временной сложности вычисления функции  $g_i(x_i)$  при всех

$i$  и  $x_i$ . Тогда для выполнения алгоритма, очевидно, требуется  $O(n^2) \cdot T_g$  арифметических операций.

**Теорема 3.** Жадный алгоритм является 2-приближённым алгоритмом для задачи (5)–(7), (9).

Доказательство. Заметим, что на шаге 3 переменная  $y_j$  принимает значение  $M_j$  на всех итерациях, кроме последней, когда она равна  $L$  или  $m_j$  (здесь и ниже под  $L$  будем понимать значение переменной  $L$ , которое она принимает по окончании выполнения алгоритма). Не ограничивая общности, после завершения работы алгоритма перенумеруем координаты решения так, что для некоторого  $i$  выполнено  $y_j = M_j$  при всех  $j \leq i$ ,  $y_{i+1} < M_{i+1}$  и  $y_j = 0$  при всех  $j > i + 1$ .

Пусть  $f^*$  — стоимость оптимального решения задачи (5)–(7), (9). Положим  $h(x) = \sum_{j=1}^n \frac{(a_j + g_j(M_j))x_j}{M_j}$  и рассмотрим задачу минимизации функции  $h(x)$  на множестве, задаваемом ограничениями (6)–(7). Рассматривая оптимум  $\hat{x}$  линейной релаксации данной задачи, т. е. задачи, в которой (7) заменено на  $0 \leq x_j \leq M_j$ ,  $j = 1, 2, \dots, n$ , мы заключаем, что  $h(\hat{x}) \geq \sum_{j=1}^i a_j + g_j(M_j) = \sum_{j=1}^i k_j(M_j)$ . По определению  $h(x)$  имеем  $h(x) \leq f(x)$  при всех  $x$ , удовлетворяющих (6)–(7). Таким образом,  $f^* \geq \sum_{j=1}^i k_j(M_j)$ . Следовательно, если  $y_{i+1} = 0$  (т. е.  $L = M_i$  на последней итерации жадного алгоритма), то  $y$  — оптимальное решение.

Теперь докажем, что если  $y_{i+1} > 0$ , то  $f^* \geq k_{i+1}(y_{i+1})$ . Действительно, предположим, что существует допустимое решение  $z$  такое, что

$$\sum_{j=1}^n k_j(z_j) < k_{i+1}(y_{i+1}). \quad (11)$$

Так как  $L = A - \sum_{k=1}^i M_k$ , то

$$\sum_{s>i} z_s \geq A - \sum_{j \leq i} z_j \geq L. \quad (12)$$

Умножая (11) на  $L/k_{i+1}(y_{i+1})$  и убирая  $i$  первых слагаемых, получаем

$$\sum_{s>i} \frac{Lk_s(z_s)}{k_{i+1}(y_{i+1})} < L. \quad (13)$$

Жадное правило (10), применённое на последней итерации, при любом  $s > i$  обеспечивает неравенство

$$\frac{k_{i+1}(y_{i+1})}{L} = \frac{k_{i+1}(y_{i+1})}{\min\{L, M_{i+1}\}} \leq \frac{k_s(\min\{\max\{L, m_s\}, M_s\})}{\min\{L, M_s\}}. \quad (14)$$

Поэтому из (13) следует, что

$$\sum_{s>i} \frac{\min\{L, M_s\}k_s(z_s)}{k_s(\min\{\max\{L, m_s\}, M_s\})} < L. \quad (15)$$

Рассмотрим слагаемые этой суммы с  $z_s > 0$ . Если  $L < m_s$ , то из (14) имеем  $k_s(m_s)/L \geq k_{i+1}(y_{i+1})/L$ , что противоречит неравенству  $k_s(z_s) < k_{i+1}(y_{i+1})$ , полученному из (11). Таким образом, слагаемые с  $L < m_s$  отсутствуют в (15).

Если  $m_s \leq L \leq M_s$ , то соответствующие слагаемые в (15) принимают вид  $Lk_s(z_s)/k_s(L)$ . Возможны два случая:

(а)  $z_s \geq L$ . Так как  $k_s(x)$  неубывающая функция, то  $k_s(z_s) \geq k_s(L)$ . Из (14) следует, что  $k_s(L) \geq k_{i+1}(y_{i+1})$ . Поэтому  $k_s(z_s) \geq k_{i+1}(y_{i+1})$ , что противоречит (11). Следовательно, слагаемые с  $z_s \geq L$  также отсутствуют в (15).

(б)  $z_s < L$ . Из вогнутости функции  $k_s$  следует, что  $Lk_s(z_s)/k_s(L) \geq z_s$ .

Наконец, если  $L > M_s$ , то слагаемые в (15) равны  $M_s k_s(z_s)/k_s(M_s)$  и в силу вогнутости функции  $k_s$  имеем  $M_s k_s(z_s)/k_s(M_s) \geq z_s$ . Следовательно,

$$\sum_{s>i} z_s \leq \sum_{s>i} \frac{\min\{L, M_s\}k_s(z_s)}{k_s(\min\{\max\{L, m_s\}, M_s\})}, \quad (16)$$

и из (15) и (16) получаем неравенство  $\sum_{s>i} z_s < L$ , что противоречит (12).

Таким образом,  $f^* \geq k_{i+1}(y_{i+1})$  и  $2f^* \geq \sum_{j=1}^i k_j(M_j) + k_{i+1}(y_{i+1})$ . Теорема 3 доказана.

## 2. Псевдополиномиальный точный алгоритм

Пусть  $\mathbb{Z}$  — множество целых чисел.

**Лемма 1.** Разрешимая задача (5)–(8) имеет оптимальное решение  $z \in \mathbb{Z}^n$ , где  $z_i \in \{0, m_i, M_i\}$  при любых  $i$ ,  $1 \leq i \leq n$ , кроме может быть одного.

**Доказательство.** Сначала установим существование решения  $y$  со свойствами, указанными в формулировке леммы, кроме свойства целочисленности. Предположим,  $x$  — некоторое оптимальное решение задачи

(5)–(8) и существует пара  $i, j \in \{1, 2, \dots, n\}$  такая, что  $m_i < x_i < M_i$ ,  $m_j < x_j < M_j$  и  $b_i \leq b_j$ . Пусть  $\delta = \min\{M_i - x_i, x_j - m_j\}$  и положим  $y_i = x_i + \delta$ ,  $y_j = x_j - \delta$ , остальные же координаты у решений  $y$  и  $x$  совпадают. Легко видеть, что  $y$  является также оптимальным решением задачи (5)–(8) и либо  $y_i = M_i$ , либо  $y_j = m_j$ . Так как все параметры задачи целочисленные, то существует по крайней мере одна нецелочисленная координата в решении  $y$ . Пусть это будет  $y_i$ . Тогда, полагая  $z = (y_1, \dots, y_{i-1}, \lfloor y_i \rfloor, y_{i+1}, \dots, y_n)$ , снова получаем оптимальное решение, так как  $A \in \mathbb{Z}$  и  $k_i(x_i)$  — неубывающая функция. Лемма 1 доказана.

Рассмотрим задачу (5)–(8) с дополнительным условием:

$$x_i \in \mathbb{Z}, \quad 1 \leq i \leq n. \quad (17)$$

В силу леммы 1 условие (17) не исключает оптимального решения в (5)–(8). В следующем разделе для построения FPTAS будет удобно использовать следующую переформулировку задачи (5)–(8), (17): найти минимум функции

$$\sum_{i=1}^n (\alpha_i u_i + \beta_i v_i) \quad (18)$$

при условиях

$$\sum_{i=1}^n (m_i u_i + (M_i - m_i) v_i) \geq A, \quad (19)$$

$$v_i \leq u_i, \quad 1 \leq i \leq n, \quad (20)$$

$$u_i \in \{0, 1\}, v_i \in \left\{0, \frac{1}{M_i - m_i}, \frac{2}{M_i - m_i}, \dots, 1\right\}, \quad 1 \leq i \leq n. \quad (21)$$

Параметры задачи предполагаются целочисленными и неотрицательными. Заметим, что теперь значения всех переменных заключены между 0 и 1 и исходная задача (5)–(8), (17) принимает форму (18)–(21) при

$$\alpha_i = a_i + b_i m_i, \quad \beta_i = b_i (M_i - m_i), \quad x_i = u_i m_i + (M_i - m_i) v_i, \quad 1 \leq i \leq n.$$

Для краткости положим  $\gamma(u, v) = \sum_{i=1}^n (\alpha_i u_i + \beta_i v_i)$  и  $w(u, v) = \sum_{i=1}^n (m_i u_i + (M_i - m_i) v_i)$ . Целесообразно использовать также дополнительную задачу, в которой (21) заменено на условие

$$u_i \in \{0, 1\}, v_i \in \{0, 1\}, \quad 1 \leq i \leq n, \quad (22)$$

т. е. размер поставки может принимать только крайние значения. Пусть  $U$  — верхняя граница целевой функции задачи (18)–(20), (22). Для каждого  $k = 1, 2, \dots, n$  и каждого  $c = 0, 1, \dots, U$  определим множество

$$S(k, c) = \arg \max_{u, v \in \{0, 1\}^n} \{w(u, v) \mid \gamma(u, v) = c, u_i = v_i = 0 \text{ при любом } i > k\}.$$

Для любой пары  $(k, c)$  определим функцию  $\varphi(k, c)$ . Если  $S(k, c) \neq \emptyset$ , то положим  $\varphi(k, c) = w(u, v)$ , где  $(u, v) \in S(k, c)$ . Если же  $S(k, c) = \emptyset$ , то положим  $\varphi(k, c) = -\infty$ . Тогда начальная стадия динамического программирования определяется следующим образом:

$$\varphi(1, c) = \begin{cases} 0 & \text{при } c = 0; \\ m_1 & \text{при } c = \alpha_1; \\ M_1 & \text{при } c = \alpha_1 + \beta_1; \\ -\infty, & \text{при } c \neq 0, c \neq \alpha_1 \text{ и } c \neq \alpha_1 + \beta_1. \end{cases}$$

Каждый шаг  $k = 2, 3, \dots, n$  выполняется с использованием рекуррентного соотношения

$$\varphi(k, c) = \begin{cases} \varphi(k-1, c), & \text{если } c < \alpha_k; \\ \max\{\varphi(k-1, c), m_k + \varphi(k-1, c - \alpha_k)\}, & \text{если } c \geq \alpha_k, c < \alpha_k + \beta_k; \\ \max\{\varphi(k-1, c), m_k + \varphi(k-1, c - \alpha_k), \\ M_k + \varphi(k-1, c - \alpha_k - \beta_k)\}, & \text{если } c \geq \alpha_k + \beta_k. \end{cases}$$

Следовательно, если  $\gamma'$  — оптимальное значение целевой функции задачи (18)–(20), (22) и  $\gamma' \leq U$ , то  $\gamma' = \min\{c \mid \varphi(n, c) \geq A\}$  и множество  $S(n, \gamma')$  содержит оптимальное решение (или решения) этой задачи.

Как следует из леммы 1 для решения задачи (18)–(21) в множестве  $\left\{0, \frac{1}{M_j - m_j}, \frac{2}{M_j - m_j}, \dots, 1\right\}$  с  $u_j = 1$  достаточно искать только одну переменную  $v_j$ , в то время как остальные переменные оптимизируются с помощью описанной выше схемы динамического программирования. Такая задача решается при каждом  $j = 1, 2, \dots, n$  и из полученных решений выбирается лучшее.

Предположим, что имеется верхняя граница  $UB \geq \gamma^*$ , где  $\gamma^*$  — стоимость оптимального решения задачи (18)–(21). Для этой задачи мы предлагаем следующий алгоритм.

**Алгоритм 1.**

Пусть  $\tilde{\gamma} = UB$ .

Для  $j = 1$  до  $n$ :

1. Применить метод динамического программирования с  $U = UB$  к задаче (18)–(20), (22), где  $m_j$  и  $M_j$  равны 0.
2. Запомнить значения  $\varphi_j(n, c)$ ,  $c = 0, 1, \dots, UB$ , и представителей  $(u^{(0)}, v^{(0)})$ ,  $\dots$ ,  $(u^{(UB)}, v^{(UB)})$  множеств решений  $S(n, 0), \dots, S(n, UB)$ . Если для некоторого  $c$  множество  $S(n, c) = \emptyset$ , то  $(u^{(c)}, v^{(c)})$  произвольно.
3. Для  $c = 0, 1, \dots, UB$  выполнить:

Если  $A \leq \varphi_j(n, c)$  и  $c \leq \tilde{\gamma}$ , то обновить  $\tilde{\gamma} := c$ ,  $\tilde{u} := u^{(c)}$ ,  $\tilde{v} := v^{(c)}$ .

Если  $\varphi_j(n, c) < A < m_j + \varphi_j(n, c)$  и  $c + \alpha_j \leq \tilde{\gamma}$ , то

обновить  $\tilde{\gamma} := c + \alpha_j$ ;  $\tilde{u}_i := u_i^{(c)}$ ,  $\tilde{v}_i := v_i^{(c)}$ ,  $i \neq j$ ;  $\tilde{u}_j := 1$ ,  $\tilde{v}_j := 0$ .

Если  $\varphi_j(n, c) + m_j \leq A \leq \varphi_j(n, c) + M_j$  и

$c + \alpha_j + \beta_j \frac{A - m_j - \varphi_j(n, c)}{M_j - m_j} \leq \tilde{\gamma}$ , то обновить

$$\tilde{\gamma} := c + \alpha_j + \beta_j \frac{A - m_j - \varphi_j(n, c)}{M_j - m_j}; \tilde{u}_i := u_i^{(c)}, \tilde{v}_i := v_i^{(c)}, i \neq j; \tilde{u}_j := 1,$$

$$\tilde{v}_j := \frac{A - m_j - \varphi_j(n, c)}{M_j - m_j}.$$

Пара  $(\tilde{u}, \tilde{v})$  является оптимальным решением.

**Теорема 4.** Для решения задачи (18)–(21) алгоритм 1 использует  $O(n^2 UB)$  арифметических операций.

Доказательство. Корректность алгоритма следует из леммы 1 и свойств схемы динамического программирования (заметим, что  $U = UB$  в дополнительной задаче не исключает частичного решения, соответствующего оптимальному). Оценка временной сложности очевидна. Теорема 4 доказана.

### 3. Вполне полиномиальная аппроксимационная схема

Применим подход, основанный на округлении входных данных (см., например, [2, 6]), чтобы сложность процедуры динамического программирования стала полиномиальной от длины входа.

Доказательство теоремы 1. Прежде всего, исключим из рассмотрения случай  $f^* = 0$ , так как в этом случае легко найти оптимальное решение. Поэтому предположим, что  $f^* > 0$ .

Взяв стоимость жадного 2-приближённого решения за верхнюю границу  $UB^*$  оптимума задачи (5)–(8), мы также получаем его нижнюю границу  $LB^* = UB^*/2 \leq f^*$  (заметим, что  $LB^* > 0$ , так как  $0 < f^* \leq UB^*$ ).

Зафиксируем  $\varepsilon$ ,  $0 < \varepsilon < 2$  (при  $\varepsilon \geq 2$  жадное решение обеспечивает справедливость теоремы), и положим  $\delta = \max\{\frac{\varepsilon LB^*}{2n}, 1\}$ . Применим алгоритм 1 к примеру задачи (18)–(21) со следующими значениями параметров:

$$\alpha_i = \left\lceil \frac{a_i + b_i m_i}{\delta} \right\rceil, \beta_i = \left\lceil \frac{M_i - m_i}{\delta/b_i} \right\rceil, i = 1, 2, \dots, n; UB = \frac{UB^*}{\delta} + 2n. \quad (23)$$

Верхняя граница  $UB$  справедлива, так как стоимость оптимума в (18)–(21) с  $\alpha_i, \beta_i$ , определенными в (23), не превышает  $f^*/\delta + 2n$ . Далее заметим, что если  $\frac{\varepsilon LB^*}{2n} > 1$ , то  $UB = 4n/\varepsilon + 2n$ , в противном случае всё равно  $UB \leq 4n/\varepsilon + 2n$ . Таким образом, в силу теоремы 4 вычисления в алгоритме 1 и жадном алгоритме суммарно требуют  $O(n^{3\frac{1}{\varepsilon}} + n^2)$  операций.

Заметим, что полученное решение

$$\tilde{x}_i = m_i \tilde{u}_i + (M_i - m_i) \tilde{v}_i, \quad 1 \leq i \leq n$$

является  $(1 + \varepsilon)$ -приближённым решением задачи (5)–(8).

В самом деле, если  $\delta = 1$ , то  $\tilde{x}$  — оптимальное решение. Если  $\delta > 1$ , обозначим  $\gamma(x) = \gamma(u(x), v(x))$ , где

$$u(x)_i = \text{sign } x_i, \quad v(x)_i = \frac{x_i - m_i u(x)_i}{M_i - m_i}, \quad 1 \leq i \leq n,$$

т. е.  $\gamma(x)$  — стоимость решения, соответствующего  $x$ , в задаче (18)–(21).

Пусть  $x^*$  — оптимальное решение задачи (5)–(8). Тогда

$$\begin{aligned} \frac{f(x^*)}{\delta} &= \sum_{i=1}^n \frac{a_i + b_i m_i}{\delta} u(x^*)_i + \sum_{i=1}^n \frac{b_i (M_i - m_i)}{\delta} v(x^*)_i \\ &\geq \sum_{i=1}^n \left\lceil \frac{a_i + b_i m_i}{\delta} \right\rceil u(x^*)_i + \sum_{i=1}^n \left\lceil \frac{b_i (M_i - m_i)}{\delta} \right\rceil v(x^*)_i - 2n = \gamma(x^*) - 2n. \end{aligned}$$

Таким образом, так как  $(\tilde{u}, \tilde{v})$  — оптимум для (18)–(21), то

$$\delta\gamma(\tilde{x}) \leq \delta\gamma(x^*) \leq f^* + 2n\delta \quad \text{и} \quad \frac{|f(\tilde{x}) - f^*|}{f^*} \leq \frac{\delta\gamma(\tilde{x}) - f^*}{f^*} \leq \frac{2n\delta}{LB^*} = \varepsilon.$$

Теорема 1 доказана.

#### 4. Сложность приближённого решения в случае нескольких потребителей

Покажем, что в общем случае задача (1)–(4) для приближения с любой фиксированной точностью NP-трудна в сильном смысле. Для этого будем использовать задачу распознавания наличия вершинного покрытия требуемой мощности в графе  $G = (V, E)$  с множеством вершин  $V = \{1, 2, \dots, s\}$  и множеством рёбер  $E = \{e_1, e_2, \dots, e_r\}$ . Подмножество  $C \subseteq V$  называется *вершинным покрытием* графа  $G$ , если каждая вершина из  $V \setminus C$  смежна по крайней мере с одной вершиной из  $C$ . Задача состоит в том, чтобы для заданного  $K$ ,  $K < s$ , ответить на вопрос: существует ли в графе  $G$  вершинное покрытие мощности не более  $K$ ? Степень вершины  $j$  обозначим через  $\deg(j)$ .

Опишем сведёние задачи о вершинном покрытии графа к задаче (1)–(4). Каждый вход задачи о вершинном покрытии представляет собой некоторый граф  $G = (V, E)$ . Поставим в соответствие вершинам графа  $G$  пункты потребления, а рёбрам — пункты производства. Введём дополнительный пункт производства с номером  $r + 1$ . Таким образом, имеем  $n = r + 1$  и  $m = s$ .

Для каждого поставщика  $i$ ,  $1 \leq i \leq r$ , положим  $M_i = m_{ij} = 1$ ,  $b_{ij} = 0$  при каждом  $j$ , и пусть  $a_{ij} = 0$  для двух его «выделенных» потребителей, соответствующих вершинам, инцидентным ребру  $e_i$  в графе  $G$ . Для остальных потребителей  $j \notin e_i$  полагаем  $a_{ij} = \Omega$ , где  $\Omega > r + s^2$  — достаточно большое число. Другими словами, каждый поставщик  $i$ ,  $1 \leq i \leq r$ , может отправить одну единицу продукции по низкой цене только одному из «выделенных» покупателей. Для дополнительного поставщика  $r + 1$  положим  $m_{r+1,j} = s$ ,  $a_{r+1,j} = 1$ ,  $b_{r+1,j} = 0$  при всех  $j$ ;  $M_{r+1} = Ks$ .

Далее полученную с помощью описанной сводимости задачу о поставках будем обозначать через  $P(G)$ .

**Лемма 2.** (а) Если в  $G$  имеется вершинное покрытие мощности не более  $K$ , то оптимальное значение целевой функции задачи о поставках  $P(G)$  не превышает  $K$ . (б) Если задача о поставках  $P(G)$  имеет допустимое решение, стоимость которого меньше  $\Omega$ , то существует вершинное покрытие графа  $G$  мощности не более  $K$ .

**Доказательство.** (а) Пусть задано вершинное покрытие  $C$  графа  $G$ . При каждом  $j = 1, 2, \dots, s$  размер поставки от дополнительного поставщика положим равным  $x_{r+1,j} = s$  при  $j \in C$  и  $x_{r+1,j} = 0$  при  $j \notin C$ . Тогда для того чтобы  $x$  было допустимым решением задачи (1)–(4), достаточно для всех  $i = 1, 2, \dots, r$  и  $j = 1, 2, \dots, s$  положить  $x_{i,j} = 1$ , если  $j \in e_i$  и  $j \notin C$ ; в противном случае  $x_{i,j} = 0$ . Так как все поставки от

поставщиков  $i = 1, 2, \dots, r$  направлены «выделенным» потребителям и поставщик  $r + 1$  обслуживает не более  $K$  клиентов, то  $f(r) \leq K$ .

(b) Пусть  $x$  — допустимое решение задачи (1)–(4),  $f(x) \leq \Omega$  и  $C(x) = \{j \mid x_{r+1,j} > 0\}$ . Здесь  $|C(x)| \leq K$ , так как  $x_{r+1,j} \geq s$  при всех  $j \in C(x)$ , и  $x_{r+1,1} + \dots + x_{r+1,n} \leq M_{r+1} = Ks$ . Первые  $r$  поставщиков снабжают только «выделенных» потребителей. Поэтому  $C(x)$  — вершинное покрытие, так как если  $x_{r+1,u} = 0$  и  $x_{r+1,v} = 0$  для какого-то ребра  $e_i = uv$ , то

$$\sum_{i=1}^n x_{iu} + \sum_{i=1}^n x_{iv} \leq \deg(u) + \deg(v) - 1 < A_u + A_v,$$

что противоречит (2). Лемма 2 доказана.

Доказательство теоремы 2. Пусть дан граф  $G$  и  $\Omega = p(n, s)s$ . При этом в задаче о поставках  $P(G)$  численные параметры ограничены некоторым полиномом от длины записи задачи о вершинном покрытии графа  $G$ .

Пусть  $x^*$  — оптимальное решение задачи (1)–(4), а  $x'$  — её  $p(r, s)$ -приближённое решение. С одной стороны, если существует вершинное покрытие мощности не более  $K$ , то по утверждению (a) леммы 2 имеем  $f(x^*) \leq K$ . Следовательно,  $f(x') \leq p(n, s)K$ . С другой стороны, если  $f(x') \leq p(n, s)K$ , то  $f(x^*) < \Omega$  и вершинное покрытие мощности не более  $K$  существует по утверждению (b) леммы 2.

Таким образом, наличие  $p(n, m)$ -приближённого решения для задачи (1)–(4) позволяет ответить на вопрос о существовании требуемого вершинного покрытия положительно, если  $f(x') < \Omega$ , и отрицательно в противном случае. Теорема 2 доказана.

Авторы благодарят А. А. Колоколова, А. В. Кононова и М. Ю. Ковалёва за полезные замечания.

## ЛИТЕРАТУРА

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Генс Г. В., Левнер Е. В. Эффективные приближённые алгоритмы для комбинаторных задач. Препринт. М.: ЦЭМИ АН СССР, 1981.
3. Chauhan S. S., Eremeev A. V., Kolokolov A. A., Servakh V. V. Concave cost supply management problem for single manufacturing unit // Supply chain optimisation. Product/process design, facility location and flow control. New York: Springer, 2005. P. 167–174. (Applied Optimization; V. 94).
4. Chauhan S. S., Eremeev A. V., Romanova A. A., Servakh V. V., Woeginger G. J. Approximation of the supply scheduling problem // Oper. Res. Lett. 2005. V. 33, N 3. P. 249–254.

5. **Chauhan S. S., Proth J.-M.** The concave cost supply problem // European J. Oper. Res. 2003. V. 148, N 2. P. 374–383.
6. **Ibarra O., Kim C. E.** Fast approximation algorithms for the knapsack and sum of subset problems // J. Assoc. Comput. Math. 1975. V. 22, N 4. P. 463–468.
7. **Woeginger G. J.** When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? // INFORMS J. on Computing. 2000. V. 12, N 1. P. 57–75.

Адреса авторов:

*А. В. Еремеев, В. В. Сервах*

Омский филиал Института математики  
им. С.Л. Соболева СО РАН,  
ул. Певцова, 13,  
644099 Омск, Россия.  
E-mail: eremeev@iitam.omsk.net.ru  
svv@iitam.omsk.net.ru

*А. А. Романова*

Омский государственный университет  
им. Ф. М. Достоевского,  
пр. Мира, 55а,  
664077 Омск, Россия.  
E-mail: romanova\_ann@bk.ru

*С. С. Чаухан*

INRIA-SAGEP, UFR Scientifique,  
Université de Metz,  
Ile du Saulcy, 57045 Metz, France.  
E-mail: Singh.Chauhan@loria.fr

Статья поступила

26 октября 2004 г.

Переработанный вариант —

13 ноября 2005 г.