ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Ноябрь—декабрь 2012. Том 19, № 6. С. 56–71

УДК 519.87+519.854

ЗАДАЧА ЦЕНООБРАЗОВАНИЯ. ЧАСТЬ 2. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ *)

А. В. Плясунов, А. А. Панин

Аннотация. Показано, что исследуемая задача принадлежит классу Log-APX, не может быть аппроксимируема с абсолютной погрешностью, ограниченной константой, и связанная с ней задача оценивания нетривиальна в классе Δ_2^p. Приведены два полиномиально разрешимых случая задачи.

Ключевые слова: вычислительная сложность, аппроксимируемость, двухуровневая задача, задача ценообразования, приближённый алгоритм, класс аппроксимируемости, NP-трудность в сильном смысле, полиномиальная иерархия.

Введение

В статье продолжены исследования задачи ценообразования, начатые в [1], где для её решения разработаны точные и приближённые алгоритмы и установлена её NP-трудность в сильном смысле. Напомним содержательную постановку задачи в виде игры Штакельберга [6]. На верхнем уровне производитель выбирает цены на каждом предприятии, выпускающем однородную продукцию. На нижнем уровне каждый потребитель выбирает одно из предприятий, на котором транспортные затраты и затраты на приобретение продукции в сумме минимальны. Покупка совершается в случае, когда это позволяет бюджет потребителя. Требуется на каждом предприятии определить такие цены, при которых доход производителя максимален. Предлагаемая стратегия ценообразования называется *фабричной* (mill pricing), также рассматриваются стратегии равномерного и дискриминационного ценообразования [7]. В [1] эта игра записана в виде задачи двухуровневого квадратичного программирования со смешанными переменными. Последнее представление задачи ценообразования и используется при анализе её сложности. Постановки, основанные на равномерном ценообразовании или на дискриминационном ценообразовании, оказываются полиномиально разрешимыми. Этот

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 12–01–00079 и 11–07–00474).

факт использован при построении приближённых алгоритмов решения задачи.

Одна из основных проблем, возникающих в процессе исследования любой оптимизационной задачи, — понять, насколько эффективно задача может быть решена численно. В идеале это понимание достигается с помощью определения верхней границы (на основе разработки эффективного алгоритма решения), а также определением нижней границы. Если верхняя и нижняя границы совпадают, то проблема решена. Таким образом, теоретически оптимальный алгоритм найден. В то время как в исследовании верхних границ достигнуты значительные успехи, о нижних границах сложности известно мало [4]. В связи с отсутствием эффективных методов для получения точных нижних границ развит очень полезный и мощный инструмент, связанный с понятиями сводимости, полноты и сложности классов [4]. С использованием этого аппарата устанавливаются связи исследуемой задачи с полиномиальной иерархией и её положение в аппроксимационной иерархии оптимизационных задач.

В разд. 1 приводится математическая постановка задачи. В разд. 2 содержатся два полиномиально разрешимых случая задачи. В разд. 3 показано, что задача принадлежит классу Log-APX и неаппроксимируема с абсолютной погрешностью, ограниченной константой.

1. Постановка задачи

Для полноты изложения приводим постановку задачи [1]. Введём обозначения:

$I = \{1, \ldots, n\}$ — множество пунктов производства (предприятий);
$J = \{1, \ldots, m\}$ — множество потребителей;
$b_j \geqslant 0$ — ценовой порог (бюджет) j-го потребителя;
$c_{ij} \geqslant 0$ — матрица транспортных затрат потребителей;
$p_i \geqslant 0$ — цена товара на i-м предприятии;

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-е предприятие обслуживает } j\text{-го потребителя,} \\ 0 & \text{в противном случае.} \end{cases}$$

Задача ценообразования (MP):

$$f(p, x) = \sum_{i \in I} p_i \sum_{j \in J} x_{ij} \to \max_{p,x} \tag{1}$$

при ограничениях

$$\sum_{i \in I} (b_j - c_{ij} - p_i) x_{ij} \geqslant 0, \quad j \in J, \tag{2}$$

$$\sum_{i\in I}(c_{ij}+p_i)x_{ij} \leqslant c_{kj}+p_k, \quad k\in I,\ j\in J, \tag{3}$$

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j\in J, \tag{4}$$

$$p_i \geqslant 0, \quad x_{ij}\in\{0,1\}, \quad i\in I,\ j\in J. \tag{5}$$

Целевая функция (1) определяет доход производителя. Ограничения (2) гарантируют, что потребитель не выйдет за рамки своего бюджета. Выполнение условий (3) приводит к тому, что транспортные затраты потребителя и его затраты на приобретение продукции в сумме минимальны. Неравенства (4) означают, что каждый потребитель может быть обслужен не более чем в одном предприятии. В дальнейшем предполагаем, что b_j и c_{ij} — рациональные числа.

2. Полиномиально разрешимые случаи задачи фабричного ценообразования

В разделе приведены полиномиально разрешимые случаи задачи MP, основанные на двух точных переборных алгоритмах решения. Введены необходимые условия оптимальности, используемые при анализе сложности задачи и разработке одного из точных алгоритмов её решения.

2.1. Необходимые условия оптимальности. Пусть (p,x) — произвольное допустимое решение задачи MP. Введём следующие обозначения:

$I_s := \{i\in I \mid \exists j\in J : x_{ij}=1,\ b_j-c_{ij}-p_i=0\}$;

$J_i := \{j\in J \mid x_{ij}=1\}, \quad i\in I$;

$I_j := \{i\in I \mid c_{ij}+p_i=\min\limits_{k\in I}\{c_{kj}+p_k\}\}, \quad j\in J$;

$$g_{ik} := \begin{cases} 0, & \text{если } \exists j\in J : i,k\in I_j,\\ 1 & \text{в противном случае;}\end{cases}$$

$\widetilde{I} := \big\{i\in I \mid \sum\limits_{j\in J} x_{ij}\geqslant 1\big\}$.

Будем говорить, что $i,k\in\widetilde{I}$ *принадлежат одному классу зависимости* I_d, тогда и только тогда, когда найдутся такие $\alpha_1,\dots,\alpha_r,\ r\leqslant n-2$, что $g_{i\alpha_1}=g_{\alpha_1\alpha_2}=\ldots=g_{\alpha_r k}=0$.

Теорема 1 (необходимые условия оптимальности). *Если* (p,x) — *произвольное оптимальное решение задачи* MP, *то*

(i) *для любого* $i\in\widetilde{I}$ *либо* $i\in I_s$, *либо найдётся номер* $j\in J_i$ *такой, что* $|I_j|>1$;

(ii) $I_d \cap I_s \neq \varnothing$ *для любого* I_d.

Доказательство. (i) Допустим обратное. Пусть существует $i \in \widetilde{I}$ такое, что $i \notin I_s$ и $|I_j| = 1$ для любого $j \in J_i$. Тогда для любого $j \in J_i$ выполняется неравенство $b_j - c_{ij} - p_i > 0$ и любой столбец с номером $j \in J_i$ содержит ровно один минимум, т. е. для любого $k \neq i$ имеем $c_{ij} + p_i < c_{kj} + p_k$. Отсюда следует, что можно увеличить переменную p_i так, что не нарушатся ограничения (2) и (3) и значение целевой функции увеличится. Действительно, пусть

$$\varepsilon_1 = \min_{j \in J_i}\{b_j - c_{ij} - p_i\}, \quad \varepsilon_2 = \min_{j \in J_i}\min_{k \neq i}\{c_{kj} + p_k - c_{ij} - p_i\},$$

$\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$. Увеличим i-ю компоненту вектора p на ε, получим новый вектор цен $\widetilde{p}$. Из определения ε следует, что ограничения (2) и (3) не нарушаются, а значение целевой функции на новом допустимом решении $(\widetilde{p}, x)$ увеличится на $\varepsilon|J_i|$. Это противоречит оптимальности вектора (p, x). Таким образом, для любого $i \in \widetilde{I}$ либо $i \in I_s$, либо найдётся номер $j \in J_i$ такой, что $|I_j| > 1$.

(ii) Из определения I_d следует, что каждое из них является объединением множеств I_j и если I_d и I'_d два разных подмножества, то они не пересекаются. Таким образом, если существует множество I_d такое, что $I_d \cap I_s = \varnothing$, то, как и в первой части утверждения, можно подправить цены так, что увеличится значение целевой функции. Действительно, рассмотрим величины

$$\varepsilon_1 = \min_{I_j \subseteq I_d}\min_{k \in I \setminus I_j}\{c_{kj} + p_k - d_j\}, \quad \varepsilon_2 = \min_{i \in I_d}\min_{j \in J_i}\{b_j - c_{ij} - p_i\},$$

где $d_j = \min_{i \in I_j}\{c_{ij} + p_i\}$ (если множество $I \setminus I_j$ пусто, то считаем, что внутренний минимум равен достаточно большому положительному числу).

В силу предположения $I_d \cap I_s = \varnothing$ и определения ε_1 и ε_2 их минимум положителен, т. е. $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\} > 0$. Увеличим компоненты вектора p с номерами из I_d на ε. Получим новый вектор цен $\widetilde{p}$. Из определения ε следует, что ограничения (2) и (3) не нарушаются, а значение целевой функции на допустимом решении $(\widetilde{p}, x)$ увеличится, что противоречит оптимальности вектора (p, x). Таким образом, предположение неверно, и $I_d \cap I_s \neq \varnothing$. Теорема 1 доказана.

2.2. Точные алгоритмы и полиномиально разрешимые случаи. Известно, что задача MP NP-трудна в сильном смысле [1]. Отсюда можно сделать вывод, что при условии P≠NP не существует алгоритма, находящего оптимальное решение задачи МР за полиномиальное время. Это означает, что любой точный алгоритм её решения имеет экспоненциальную трудоёмкость. Приведём два точных алгоритма решения задачи, которые имеют полиномиальную трудоёмкость либо при фиксированном числе клиентов, либо при фиксированном количестве предприятий.

Из вида ограничений задачи МР следует, что для заданных вектора цен или матрицы назначений задача полиномиально разрешима. В первом случае необходимо вычислить 0-1 матрицу назначений, удовлетворяющую ограничениям (2)–(5). Заметим, что $x_{ij} = 1$ для предприятия $i \in I$ и потребителя $j \in J$ тогда и только тогда, когда $x_{kj} = 0$ для всех $k \neq i$, к тому же сумма затрат j-го потребителя минимальна на i-м предприятии, при этом в случае равенства затрат на предприятиях i и k выполнено $p_i \geqslant p_k$, и затраты j-го потребителя не превышают его бюджет. Значит, для поиска матрицы назначений каждому потребителю необходимо сопоставить предприятие с минимальными затратами, не превышающими бюджет, а в случае равенства затрат выбирается предприятие с максимальной ценой. Получается, что задача МР с фиксированным вектором цен решается за $O(nm)$ операций. Также нетрудно убедиться, что МР с фиксированной матрицей назначений является задачей линейного программирования и, следовательно, полиномиально разрешима [3]. Если использовать алгоритм Кармаркара, то для решения задачи потребуется $O(n^{3,5}L)$ арифметических операций при условии, что величины $b_j, c_{ij},\ i \in I,\ j \in J$, являются $O(L)$-разрядными числами [3]. Исходя из этих соображений, для нахождения оптимального решения задачи МР можно воспользоваться одним из двух следующих переборных алгоритмов.

Алгоритм A_1.

Обозначим через X множество всех 0-1 матриц размера $n \times m$, удовлетворяющих ограничению (4), $\widetilde{X} \subseteq X$.

Шаг 0. Положим $\widetilde{X} := \varnothing$, $f^* := 0$ и перейдём на шаг 1.

Шаг 1. Выберем матрицу $\widetilde{x} \in X \setminus \widetilde{X}$. С помощью алгоритма Кармаркара найдём оптимальный вектор цен $\widetilde{p}$ в задаче МР с фиксированной матрицей назначений $\widetilde{x}$ и перейдём на шаг 2.

Шаг 2. Если $f(\widetilde{p}, \widetilde{x}) > f^*$, то положим $p^* := \widetilde{p}, x^* := \widetilde{x}, f^* := f(p^*, x^*)$, $\widetilde{X} := \widetilde{X} \cup \widetilde{x}$. Если $X \setminus \widetilde{X} \neq \varnothing$, то перейдём на шаг 1, иначе СТОП.

Очевидно, что алгоритм A_1 конечен, так как конечно множество X. Второй алгоритм основан на переборе неотрицательных векторов цен, мощность множества которых континуальна. Поэтому прежде чем описывать сам алгоритм, покажем, что для решения задачи MP достаточно обойтись перебором конечного числа векторов цен. Докажем следующие вспомогательные утверждения.

Лемма 1. *Если (p, x) — оптимальное решение задачи* MP, *то оптимальная цена p_i на любом предприятии $i \in \widetilde{I}$ представляется в виде суммы $b_{j_r} - \sum\limits_{s=2}^{r}(c_{i_{s-1}j_s} - c_{i_{s-1}j_{s-1}}) - c_{ij_1}$ для некоторых r, $1 \leqslant r \leqslant n$, наборов из r потребителей и $r-1$ попарно различных предприятий, отличных от предприятия i.*

Доказательство. Пусть известно оптимальное решение (p, x). По определению множества I_s если $i \in I_s$, то найдётся такой потребитель $j \in J$, что $b_j - c_{ij} = p_i$. Из теоремы 1 следует, что при условии $i \in I \setminus I_s$ найдётся множество I_d, содержащее предприятие i, причём $I_d \cap I_s \neq \varnothing$ или, другими словами, существует набор $(i = i_1, \ldots, i_{r-1})$ из I_d такой, что $g_{ii_1} = g_{i_1i_2} = g_{i_2i_3} = \ldots = g_{i_{r-1}i_r} = 0$, $i_r \in I_d \cap I_s$. По определению существует обслуживаемый потребитель j_r, который тратит в i_r-м предприятии весь свой бюджет, а также существуют такие клиенты $j_1, \ldots, j_{r-1}$, что $i, i_1 \in I_{j_1},\ i_1, i_2 \in I_{j_2},\ \ldots,\ i_{r-2}, i_{r-1} \in I_{j_{r-1}}$. Тогда оптимальная цена на предприятии i равна $b_{j_r} - \sum\limits_{s=2}^{r}(c_{i_sj_s} - c_{i_sj_{s-1}}) - c_{i_1j_1}$. Лемма 1 доказана.

В дополнение к лемме 1 следует отметить, что $p_i \in [0, \max\limits_{j \in J}(b_j - c_{ij})]$ на любом предприятии $i \in \widetilde{I}$. Определим для каждого предприятия i множество цен $\mathrm{Price}(i)$ следующим образом. Будем считать, что любая цена p_i из отрезка $[0, \max\limits_{j \in J} b_j - c_{ij}]$ принадлежит $\mathrm{Price}(i)$ тогда и только тогда, когда $p_i = b_{j_r} - \sum\limits_{s=2}^{r}(c_{i_{s-1}j_s} - c_{i_{s-1}j_{s-1}}) - c_{ij_1}$ для некоторых r, $1 \leqslant r \leqslant n$, наборов из r потребителей и $r-1$ попарно различных предприятий, отличных от предприятия i. По лемме 1 множество $\mathrm{Price}(i)$ включает все возможные значения i-й координаты оптимальных векторов цен при условии $i \in \widetilde{I}$. Пусть $\mathrm{Price} = \prod\limits_{i \in I} \mathrm{Price}(i)$ — декартово произведение множеств $\mathrm{Price}(i)$.

Теорема 2. *Существует оптимальное решение (p, x) задачи* MP *такое, что $p \in \mathrm{Price}$.*

Доказательство. Пусть (p, x) — оптимальное решение. По лемме 1 $p_i \in \mathrm{Price}(i)$ для всех $i \in \widetilde{I}$. Определим вектор цен $\widetilde{p}$ следующим образом. Пусть $\widetilde{p}_i = p_i$ для всех $i \in \widetilde{I}$, а для всех $i \notin \widetilde{I}$ через $\widetilde{p}_i$ обозначим максимальный элемент Price(i). Убедимся, что решение $(\widetilde{p}, x)$ также оптимально. По построению нового вектора цен решение $(\widetilde{p}, x)$ допустимо, если выполнено ограничение (3). Пусть $\sum\limits_{i\in I}(c_{ij} + \widetilde{p}_i)x_{ij} > c_{kj} + \widetilde{p}_k$ для некоторых $k \in I$ и $j \in J$. По построению вектора $\widetilde{p}$ это возможно только в том случае, когда $k \in I \setminus \widetilde{I}$. Из определения множества Price(i) вытекает, что $\widetilde{p}_i + c_{ij} - c_{kj} \in \mathrm{Price}(k)$ для любого $i \in I$. Получается, что $\widetilde{p}_k < \sum\limits_{i\in I}(c_{ij} + \widetilde{p}_i)x_{ij} - c_{kj} \in \mathrm{Price}(k)$; противоречие. Теорема 2 доказана.

Из теоремы 2 следует, что множество Price содержит оптимальный вектор цен. Тогда для поиска оптимального решения задачи MP можно использовать следующий конечный переборный

Алгоритм A_2.

Пусть $P \subseteq \mathrm{Price}$.

Шаг 0. Положим $P := \varnothing$, $f^* := 0$ и перейдём на шаг 1.

Шаг 1. Выберем вектор $\widetilde{p} \in \mathrm{Price} \setminus P$. Найдём оптимальную матрицу назначений $\widetilde{x}$ в задаче MP с фиксированным вектором цен $\widetilde{p}$ и перейдём на шаг 2.

Шаг 2. Если $f(\widetilde{p}, \widetilde{x}) > f^*$, то положим $p^* := \widetilde{p}$, $x^* := \widetilde{x}$, $f^* := f(p^*, x^*)$, $P := P \cup \widetilde{p}$. Если $\mathrm{Price} \setminus P \neq \varnothing$, то перейдём на шаг 1, иначе СТОП.

Приведём оценки трудоёмкости алгоритмов A_1 и A_2.

Теорема 3. *Имеют место следующие утверждения.*

(i) *Время работы алгоритма A_1 оценивается величиной $O(n^{m+3,5}L)$ при условии, что параметры задачи b_j, c_{ij}, $i \in I$, $j \in J$, $O(L)$-разрядные числа.*

(ii) *Время работы алгоритма A_2 имеет порядок*

$$O\left(nm\left(\sum_{k=0}^{n-1}\frac{(n-1)!}{(n-k-1)!}m^k\right)^n\right).$$

Доказательство. Утверждение (i) следует из того, что на каждом шаге алгоритма решается задача линейного программирования. Если использовать алгоритм Кармаркара, то для решения задачи потребуется $O(n^{3,5}L)$ арифметических операций при условии, что b_j, c_{ij}, $i \in I$, $j \in J$, — $O(L)$-разрядные числа [3]. Так как 0-1 матриц размера $n\times m$, удовлетворяющих ограничению (4), не более $O(n^m)$, умножив эту величину на трудоёмкость алгоритма Кармаркара, получим требуемую оценку.

(ii) По определению каждый элемент множества $\mathrm{Price}(i)$ определяется упорядоченным набором из k предприятий, $0 \leqslant k \leqslant n-1$, и предприятием i, при этом каждому из них сопоставляется некоторый клиент. Число упорядоченных выборок размера k из $(n-1)$-элементного множества равно $A_{n-1}^k = \frac{(n-1)!}{(n-k-1)!}$. Получаем $|\mathrm{Price}(i)| = \sum\limits_{k=0}^{n-1} \frac{(n-1)!}{(n-k-1)!} m^k$. Тогда $|\mathrm{Price}| = O\left(\left(\sum\limits_{k=0}^{n-1} \frac{(n-1)!}{(n-k-1)!} m^k\right)^n\right)$. Так как задача MP с фиксированным вектором цен решается за $O(nm)$ операций, время работы алгоритма A_2 оценивается величиной $O\left(nm\left(\sum\limits_{k=0}^{n-1} \frac{(n-1)!}{(n-k-1)!} m^k\right)^n\right)$. Теорема 3 доказана.

Теорема 4. *Задача* MP *полиномиально разрешима при выполнении любого из условий* (i) $|J| = m = \mathrm{const}$ *и* (ii) $|I| = n = \mathrm{const}$.

Доказательство. Полиномиальная разрешимость задачи при условии $|J| = m = \mathrm{const}$ следует из теоремы 3(i). Аналогично из теоремы 3(ii) вытекает, что с помощью алгоритма A_2 можно решить задачу MP за время

$$T_2(n,m) = O\left(nm\left(\sum_{k=0}^{n-1} \frac{(n-1)!}{(n-k-1)!} m^k\right)^n\right) \leqslant O\big(m^{n^2}\big),$$

откуда получаем полиномиальную разрешимость задачи при условии $|I| = n = \mathrm{const}$. Теорема 4 доказана.

3. Вычислительная сложность задачи фабричного ценообразования

Раздел посвящён анализу сложностного статуса задачи MP. Устанавливается положение задачи в иерархии аппроксимационных классов. Уточняется её взаимосвязь с задачами распознавания из полиномиальной иерархии.

3.1. Сложность задачи оценивания. В [1] установлена NP-трудность задачи MP в сильном смысле. Однако можно получить более детальную характеристику сложности задачи, уточнив положение соответствующей ей задачи оценивания OPT(MP), которая является задачей распознавания, в полиномиальной иерархии. Если к входу задачи MP присоединить целочисленный параметр k, то получим вход задачи оценивания, в которой надо определить, является ли k оптимальным значением целевой функции на множестве допустимых решений задачи MP или нет.

Напомним обозначения, используемые в теории сложности при описании полиномиальной иерархии классов сложности. Первые два базовых класса задач распознавания P и NP определяются с помощью детерминированных и недетерминированных машин Тьюринга [2, 4]. Класс P образован задачами, которые распознаются за полиномиальное время на детерминированных машинах Тьюринга. Соответственно класс NP определяется как класс задач, которые распознаются за полиномиальное время на недетерминированных машинах Тьюринга. Третий базовый класс *co*-NP состоит из задач распознавания, чьи дополнения лежат в классе NP. Данные классы образуют первый уровень полиномиальной иерархии, и их обозначают через Δ_1^p, Σ_1^p и Π_1^p соответственно.

$$\begin{tikzcd}
\Sigma_2^p & & \Pi_2^p \\
& \Delta_2^p \arrow[ul] \arrow[ur] & \\
\Sigma_1^p \arrow[uu] \arrow[ur] & & \Pi_2^p \arrow[uu] \arrow[ul] \\
& \Delta_1^p \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

$$\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}, \quad \Pi_2^p = co - \mathrm{NP}^{\mathrm{NP}};$$

$$\Delta_2^p = \mathrm{P}^{\mathrm{NP}} = \mathrm{P}^{co-\mathrm{NP}};$$

$$\Sigma_1^p = \mathrm{P}^{\mathrm{NP}}, \quad \Pi_1^p = co - \mathrm{NP};$$

$$\Delta_1^p = \Sigma_1^p = \Pi_0^p = \mathrm{P}.$$

Диаграмма 1. Связь между классами уровней 1 и 2

Второй уровень полиномиальной иерархии определяется с помощью детерминированных и недетерминированных оракульных машин Тьюринга [2, 4]. Задача распознавания L принадлежит классу Δ_2^p, если существует детерминированная оракульная машина Тьюринга, которая распознает за полиномиальное время задачу L, используя в качестве оракула некоторый язык из класса NP (NP-оракул). Класс Δ_2^p часто обозначают через P^{NP}. Аналогично определяются классы Σ_2^p и Π_2^p. На диаграмме 1 приводятся все включения между классами иерархии уровней 1 и 2. Известно, что если NP $\neq$ *co*-NP, то приведённые включения строгие [4]. Задачу распознавания A назовём *нетривиальной* Δ_2^p-задачей [8], если $A \in \Delta_2^p,\ A \notin \Sigma_1^p \cup \Pi_1^p$.

Теорема 5. *Если* NP$\neq$ *co*-NP, *то* OPT(MP) — *нетривиальная* Δ_2^p *задача.*

Доказательство. Покажем, что выполняются следующие три условия:

(i) OPT(MP) $\in \Delta_2^p$,
(ii) OPT(MP) $\notin$ NP при NP$\neq$*co*-NP и
(iii) OPT(MP) $\notin$ *co*-NP при NP$\neq$*co*-NP.

Сопоставим задаче MP стандартную задачу распознавания D(MP), в которой для каждого входа задачи MP надо решить, существует или нет допустимое решение со значением целевой функции, большим или равным k. Очевидно, что задача распознавания D(MP) лежит в классе NP. Сводимость теоремы 1 из [1] можно модифицировать для получения полиномиальной сводимости задачи распознавания, соответствующей задаче о минимальном покрытии, к задаче D(MP). Это означает, что задача D(MP) NP-полна в сильном смысле. Очевидно, что, используя двоичный поиск и задачу D(MP) в качестве оракула, можно за полиномиальное время найти решение задачи OPT(MP) [5]. Тем самым OPT(MP)$\in \Delta_2^p$ по определению класса Δ_2^p. Задача D(MP) NP-полна в сильном смысле и полиномиально сводится к задаче OPT(MP). Тогда из результатов, полученных в [8], следует, что выполнены условия (ii) и (iii). Стало быть, при условии, что NP$\neq$ *co*-NP, задача OPT(MP) нетривиальна. Теорема 5 доказана.

Известно, что если задача распознавания D(MP) NP-полна, то задачи OPT(MP), D(MP) и MP эквиваленты в том смысле, что, зная полиномиальный алгоритм решения одной из них, можно построить полиномиальный алгоритм для решения оставшихся двух [5]. Так как в настоящий момент общепринята гипотеза, что P$\neq$ NP, существование полиномиального алгоритма для решения любой из этих задач маловероятно. Другими словами, из эквивалентности следует, что алгоритм решения любой из этих задач должен иметь экспоненциальную трудоёмкость при выполнении гипотезы P$\neq$ NP. Ввиду нетривиальности задачи OPT(MP) в классе Δ_2^p, если NP$\neq$ *co*-NP, то детерминированный полиномиальный оракульный алгоритм её решения не может ограничиться обращением к NP-оракулу только с «да»-вопросами или только с «нет»-вопросами [8]. На содержательном уровне это означает следующее. Предположим, что в качестве NP-оракула используется задача D(MP). Тогда «да»-вопрос в задаче D(MP) формулируется как «существует допустимое решение величины k или больше?», «нет»-вопрос формулируется как «не существует допустимого решения величины k или больше?». В этих терминах гипотетический детерминированный полиномиальный оракульный алгоритм решения задачи OPT(MP) в силу её нетривиальности потребует не только информацию о существовании допустимых решений со значениями не меньше заданных величин для некоторых конкретных исходных данных задачи MP, но и информацию о несуществовании допустимых решений со значениями не меньше заданных величин для некоторых других конкретных исходных данных задачи MP.

В заключение отметим, что задача оценивания OPT(MP) полна в классе Δ_2^p относительно полиномиальной сводимости по Тьюрингу. Это следует из определений класса Δ_2^p и полиномиальной сводимости по Тьюрингу и NP-полноты стандартной задачи распознавания D(MP).

3.2. Аппроксимационная сложность задачи MP. В этом подразделе предполагается, что b_j, c_{ij}, $i \in I$, $j \in J$, — целые числа. Пусть $a_{ij} := b_j - c_{ij}$, $i \in I$, $j \in J$. Тогда задачу MP можно переписать в виде следующей задачи квадратичного программирования (F):

$$\sum_{i\in I} p_i \sum_{j\in J} x_{ij} \to \max_{p,x}, \tag{6}$$

$$\sum_{i\in I} (a_{ij} - p_i)x_{ij} \geqslant 0, \quad j \in J, \tag{7}$$

$$\sum_{i\in I} (a_{ij} - p_i)x_{ij} \geqslant a_{kj} - p_k, \quad k \in I,\ j \in J, \tag{8}$$

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j \in J, \tag{9}$$

$$p_i \geqslant 0, \quad x_{ij} \in \{0, 1\}, \quad i \in I,\ j \in J. \tag{10}$$

Без ограничения общности считаем, что для любого $j \in J$ существует $i \in I$ такое, что $a_{ij} > 0$. Пусть $a_1 < \ldots < a_\omega$ — строго упорядоченная по возрастанию последовательность всех $a_{w(j)} := \max\limits_{i\in I} a_{ij}$, $j \in J$. Положим $S_k := \{j \in J \mid a_k = \max\limits_{i\in I} a_{ij}\}$, $s_k := |S_k|$. Обозначим через g^* сумму $\sum\limits_{1\leqslant k\leqslant\omega} a_k s_k$, равную сумме бюджетов всех потребителей за вычетом их минимально возможных транспортных затрат. Величина g^* является оптимальным значением целевой функции задачи F с переменными p_{ij} вместо переменных p_i, т. е. задачи с дискриминационным ценообразованием, в которой можно назначать разные цены на продукцию на каждом из предприятий для каждого потребителя. Очевидно, что g^* — верхняя граница задачи F. Пусть $g_k := \sum\limits_{0\leqslant t\leqslant k} a_{\omega-t} s_{\omega-t}$. Под матрицами $\left(a_{ij}^k\right)$, $1 \leqslant k \leqslant \omega$, будем понимать матрицы размера $m \times n$ с элементами $a_{ij}^k := a_{ij}$, если $a_{ij} \geqslant a_{\omega-k}$, и 0 в противном случае. Через F_k обозначим задачу F с матрицей $\left(a_{ij}^k\right)$ и равномерным ценообразованием, т. е. одинаковыми ценами на всех предприятиях. Другими словами, к задаче F с матрицей $(a_{ij}) := \left(a_{ij}^k\right)$ добавляется ограничение $p_\alpha = p_\beta\ \forall \alpha, \beta \in I$. Пусть $\mathrm{Opt}(k)$ — оптимальное значение целевой функции задачи F_k.

Лемма 2. $\mathrm{Opt}(k) = a_{\omega-k+e}(s_\omega + \ldots + s_{\omega-k+e})$, $0 \leqslant e \leqslant k$.

Доказательство. В теореме 1 для задачи MP с фабричным ценообразованием показано, что в оптимальном решении существует хотя бы одно насыщенное предприятие. Нетрудно показать, что этот факт верен и для задачи F с равномерным ценообразованием, воспользовавшись идеей доказательства теоремы 1. Тогда для оптимального решения (p^*, x^*) задачи F_k существуют $i \in I$, $j \in J$ такие, что $x^*_{ij} = 1$ и $a_{ij} - p^* = 0$. Согласно ограничению (8) для задачи F_k оптимальная цена p^* равна $\max\limits_{i \in I} a_{ij} = a_{\omega-k+e}$ для некоторого $0 \leqslant e \leqslant k$. Из неравенств (7) и (8) следует, что потребитель j обслуживается и приносит доход p^*, если соответствующая этому потребителю величина $a_{w(j)}$ на меньше этой цены. Получаем $\mathrm{Opt}(k) = a_{\omega-k+e}(s_\omega + \ldots + s_{\omega-k+e})$ для некоторого $0 \leqslant e \leqslant k$. Лемма 2 доказана.

Следствие 1. *Функция* $\mathrm{Opt}(k)$ *монотонно не убывает.*

Доказательство следует из равенства

$$\mathrm{Opt}(k+1) = \max\{\mathrm{Opt}(k), a_{\omega-k-1}(s_\omega + \ldots + s_{\omega-k-1})\},$$

которое вытекает из леммы 2.

Следствие 2. *Оптимальные значения целевых функций задач* F_k *и* F_{k+1}, $1 \leqslant k \leqslant \omega - 1$, *совпадают тогда и только тогда, когда*

$$\mathrm{Opt}(k) \geqslant a_{\omega-k-1}(s_\omega + \ldots + s_{\omega-k-1}).$$

Пусть $H_k := 1 + 1/2 + 1/3 + \ldots + 1/\big(\sum\limits_{0 \leqslant t \leqslant k} s_{\omega-t}\big)$.

Лемма 3. *Для любого* k *выполняется неравенство* $g_k/\mathrm{Opt}(k) \leqslant H_k$.

Доказательство проведём индукцией по k. При $k = 0$ очевидно, что $\mathrm{Opt}(0) = a_\omega s_\omega = g_0$.

Пусть для $k = z$ выполнено $g_z/\mathrm{Opt}(z) \leqslant H_z$. Допустим, что это неравенство не выполнено для $k = z + 1$, т. е. $g_{z+1}/\mathrm{Opt}(z+1) > H_{z+1}$.

Случай 1. Пусть оптимальные значения целевых функций задач F_z и F_{z+1} совпадают. Тогда по следствию 2 это условие можно описать неравенством $\mathrm{Opt}(z) \geqslant a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})$ или

$$a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e}) \geqslant a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})$$

для некоторого $0 \leqslant e \leqslant z$. Рассмотрим отношение g_{z+1} к $\mathrm{Opt}(z+1)$:

$$\frac{g_{z+1}}{\mathrm{Opt}(z+1)} = \frac{g_z + a_{\omega-z-1}s_{\omega-z-1}}{\mathrm{Opt}(z)} = \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{\mathrm{Opt}(z)} = \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e})}.$$

Первое равенство следует из того, что $\mathrm{Opt}(z) = \mathrm{Opt}(z+1)$, и из определения g_{z+1} и g_z, второе равенство очевидно, а третье следует из леммы 2. Тем самым

$$\frac{g_{z+1}}{\mathrm{Opt}(z+1)} = \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e})}.$$

Таким образом, исходное неравенство $g_{z+1}/\mathrm{Opt}(z+1) > H_{z+1}$ можно переписать в виде

$$\begin{aligned} H_{z+1} &< \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e})} \leqslant H_z + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e})} \\ &\leqslant H_z + \frac{a_{\omega-z-1}s_{\omega-z-1}}{\left(a_{\omega-z-1}\frac{(s_\omega+\ldots+s_{\omega-z-1})}{(s_\omega+\ldots+s_{\omega-z+e})}(s_\omega + \ldots + s_{\omega-z+e})\right)} = H_z + \frac{s_{\omega-z-1}}{(s_\omega + \ldots + s_{\omega-z-1})}. \end{aligned}$$

Здесь использован индукционный переход и следствие 2. Эти выкладки приводят к следующему результату:

$$H_{z+1} < H_z + \frac{s_{\omega-z-1}}{s_\omega + \ldots + s_{\omega-z-1}}.$$

Воспользовавшись определением величины H_z, получаем

$$\frac{1}{s_\omega + \ldots + s_{\omega-z} + 1} + \ldots + \frac{1}{s_\omega + \ldots + s_{\omega-z-1}} < \frac{s_{\omega-z-1}}{s_\omega + \ldots + s_{\omega-z-1}}.$$

В левой части последнего неравенства ровно $s_{\omega-z-1}$ слагаемых со знаменателями, не превосходящими сумму $s_\omega + \ldots + s_{\omega-z-1}$, а значит, левая часть не меньше правой; противоречие.

Случай 2. Пусть оптимальные значения целевых функций задач F_z и F_{z+1} не совпадают. Тогда в силу следствия 2 верно неравенство

$$\mathrm{Opt}(z) < a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1}).$$

Из доказательства следствия 1 вытекает, что для любого $0 \leqslant e \leqslant z$ выполняется $a_{\omega-z+e}(s_\omega + \ldots + s_{\omega-z+e}) < a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})$. Рассмотрим отношение g_{z+1} к $\mathrm{Opt}(z+1)$:

$$\frac{g_{z+1}}{\mathrm{Opt}(z+1)} = \frac{g_z}{\mathrm{Opt}(z+1)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{\mathrm{Opt}(z+1)} = \frac{g_z}{\mathrm{Opt}(z+1)}$$
$$+ \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})} < \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})}.$$

Первые два равенства очевидны. Так как $\mathrm{Opt}(z+1) > \mathrm{Opt}(z)$, последнее неравенство верно. Получаем

$$\frac{g_{z+1}}{\mathrm{Opt}(z+1)} < \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})}.$$

Стало быть, исходное неравенство $g_{z+1}/\mathrm{Opt}(z+1) > H_{z+1}$ можно записать в виде

$$H_{z+1} < \frac{g_z}{\mathrm{Opt}(z)} + \frac{a_{\omega-z-1}s_{\omega-z-1}}{a_{\omega-z-1}(s_\omega + \ldots + s_{\omega-z-1})} \leqslant H_z + \frac{s_{\omega-z-1}}{(s_\omega + \ldots + s_{\omega-z-1})}.$$

Второе неравенство следует из индукционного предположения. Эти преобразования приводят нас к следующему результату:

$$H_{z+1} < H_z + \frac{s_{\omega-z-1}}{s_\omega + \ldots + s_{\omega-z-1}},$$

который, как показано в первом случае, ведёт к противоречию. Лемма 3 доказана.

Теорема 6. *Задача* MP *принадлежит классу* Log-APX.

Доказательство. Оптимизационная задача принадлежит классу Log-APX тогда и только тогда, когда существует полиномиальный алгоритм, находящий $(c * \ln(r))$-приближённое решение задачи, где r — длина входа, а c — некоторая константа [5]. Пусть $\mathrm{Opt}_{\mathrm{F}}$ — оптимальное значение задачи F. Из определения задачи F_w следует, что любое её оптимальное решение является допустимым решением задачи F со значением целевой функции, равным $\mathrm{Opt}(w)$. Тогда из леммы 3 имеем

$$\mathrm{Opt}_{\mathrm{F}}/\mathrm{Opt}(w) \leqslant g_w/\mathrm{Opt}(w) \leqslant H_w,$$

где $\mathrm{Opt}(w)$ — оптимальное значение задачи F_w с одинаковой ценой на всех предприятиях, а $g_w = g^*$ — верхняя граница для $\mathrm{Opt}_{\mathrm{F}}$. Очевидно, что $\mathrm{Opt}(w)$ — оптимальная цена на всех предприятиях и соответствующие переменные x_{ij} вычисляются за полиномиальное время. Тогда утверждение теоремы следует из того, что значение H_w с помощью неравенства Стирлинга оценивается величиной порядка $c\ln(m)$, где $m = |J|$. Теорема 6 доказана.

Теорема 7. *При условии* P≠NP *для задачи* MP *не существует полиномиального приближённого алгоритма с абсолютной погрешностью, ограниченной константой.*

Доказательство. Допустим обратное. Пусть для некоторого полиномиального алгоритма A верна оценка $f(p^*, x^*) - f(p^A, x^A) < W$, где $W = \mathrm{const} > 0$. Без ограничения общности можно считать, что W — целое число. Покажем, что с помощью алгоритма A можно построить полиномиальный алгоритм для решения задачи MP, которая NP-трудна [1].

Поскольку за полиномиальное время можно построить оптимальное решение задачи MP при фиксированных переменных x_{ij}, будем считать, что алгоритм A устроен так, что переменные p_i^A оптимальны для заданных переменных x_{ij}^A. Пусть L — произвольный вход задачи MP, $\mathrm{Opt}(L) = f(p^*(L), x^*(L))$ и $A(L) = f(p^A(L), x^A(L))$.

Пусть исходные данные индивидуальной задачи L' получены из исходных данных задачи L умножением векторов (b_j) и (c_{ij}) на W. Заметим, что теорема 1, лемма 1 и теорема 2 верны для задачи MP с фиксированной матрицей назначений. Можно считать, что $p^* \in \mathrm{Price}$ и алгоритм A находит решение (p^A, x^A), $p^A \in \mathrm{Price}$. Тогда из определения множества Price следует, что $p^*(L)$, $p^A(L)$, $p^*(L')$, $p^A(L')$ — целые числа, причём $p^*(L')$ и $p^A(L')$ кратны W, а $\mathrm{Opt}(L') = W\,\mathrm{Opt}(L)$. Следовательно,

$$\mathrm{Opt}(L') - A(L') = W * \mathrm{Opt}(L) - A(L') < W.$$

Отсюда

$$\mathrm{Opt}(L) - A(L')/W = f(p^*(L), x^*(L)) - f(p^A(L')/W, x^A(L')) < 1,$$

где $f(p^*(L), x^*(L))$ и $f(p^A(L')/W, x^A(L'))$ — целые числа. Тогда

$$f(p^*(L), x^*(L)) = f(p^A(L')/W, x^A(L')),$$

и $(p^A(L')/W, x^A(L'))$ — оптимальное решение задачи MP на входе L. Таким образом, задача MP полиномиально разрешима, что противоречит условию P≠NP [1]. Теорема 7 доказана.

4. Заключение

В работе рассмотрен частный случай задачи размещения и ценообразования — задача с фабричным ценообразованием.

Предлагаются два полиномиально разрешимых случая исследуемой задачи. Установлен сложностной статус связанной с ней задачи оценивания. Показано, что задача MP принадлежит классу Log-APX и для неё не существует полиномиального приближённого алгоритма с абсолютной погрешностью, ограниченной константой при условии, что P≠NP.

В дальнейшем предполагается уточнить положение задачи в классе Log-APX, т. е. проверить, является ли данная задача полной в этом классе относительно некоторой сохраняющей аппроксимируемость сводимости или нет.

ЛИТЕРАТУРА

1. **Панин А. А., Плясунов А. В.** Задача ценообразования. Часть 1: точные и приближённые алгоритмы решения // Дискрет. анализ и исслед. операций. — 2012. — Т. 19, № 5. — С. 83–100.
2. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. — М.: Мир. 1982. — 416 с.
3. **Схрейвер А.** Теория линейного и целочисленного программирования. Т. 1. — М.: Мир, 1991. — 360 с.
4. **Attallah M.** Algorithms and theory of computation handbook. — Boca Raton: CRC Press LLC, 1999. — 1312 p.
5. **Ausiello G., Crescenzi P., Gambosi G., Kann V., Marchetti-Spaccamela A., Protasi M.** Complexity and approximation: combinatorial optimization problems and their aproximability properties. — Berlin: Springer-Verl., 1999. — 524 p.
6. **Dempe S. J.** Foundations of bilevel programming. — Dordrecht: Kluwer Acad. Publ., 2002. — 320 p.
7. **Hanjoul P., Hansen P., Peeters D., Thisse J.-F.** Uncapacitated plant location under alternative spatial price policies // Market Sci. — 1990. — Vol. 36. — P. 41–57.
8. **Leggette E. W., Jr., Moore D. J.** Optimization problems and the polynomial hierarchy // Theor. Comput. Sci. — 1981. — Vol. 15, N 3. — P. 279–289.

Плясунов Александр Владимирович,
e-mail: apljas@math.nsc.ru
Панин Артём Александрович,
e-mail: arteam1897@gmail.com

Статья поступила
1 июня 2011 г.

Переработанный вариант —
4 июня 2012 г.