ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Сентябрь—октябрь 2012. Том 19, № 5. С. 83–100

УДК 519.87+519.854

ЗАДАЧА ЦЕНООБРАЗОВАНИЯ. ЧАСТЬ I. ТОЧНЫЕ И ПРИБЛИЖЁННЫЕ АЛГОРИТМЫ РЕШЕНИЯ [*)]

А. В. Плясунов, А. А. Панин

Аннотация. Для исследуемой задачи ценообразования показано, что она NP-трудна в сильном смысле. Для её решения разработаны точные и приближённые алгоритмы, использующие декомпозицию, генетический локальный поиск и поиск с запретами. Приводятся результаты вычислительных экспериментов.

Ключевые слова: NP-трудность в сильном смысле, задача ценообразования, двухуровневая задача, минимаксная задача, декомпозиция, локальный поиск, поиск с запретами, генетический алгоритм, гибридный алгоритм.

Введение

Статья является первой частью работы, посвящённой задаче ценообразования. Доказано, что эта задача NP-трудна в сильном смысле, и приведены точные и приближённые алгоритмы её решения. Во второй части описаны полиномиально разрешимые частные случаи задачи, а также приведены результаты, характеризующие её аппроксимационную сложность.

В моделях исследования операций, имеющих экономическое содержание, ценовые или затратные параметры, как правило, предполагаются заданными. Основное внимание уделяется поиску наилучших в том или ином смысле объёмов производства, планов поставок, размещений пунктов производства, номенклатур изделий и т. п. Процессы ценообразования в условиях рыночных отношений выдвигают новые задачи, целью которых является нахождение цен на продукцию, наилучших для её производителя [3, 9].

Сформулируем содержательную постановку исследуемой задачи в виде игры Штакельберга [12]. На верхнем уровне производитель выбирает цены на каждом из своих предприятий, выпускающих однородную продукцию. На нижнем уровне каждый из потребителей выбирает одно из

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

предприятий, на котором транспортные затраты и затраты на приобретение продукции в сумме минимальны. Покупка совершается в том случае, когда это позволяет бюджет потребителя. Требуется на каждом предприятии определить такие цены, при которых доход производителя максимален. Предлагаемая стратегия ценообразования называется *фабричной* [15]. Помимо неё обычно рассматриваются следующие две стратегии: *равномерное ценообразование* [15], когда на всех пунктах обслуживания устанавливается одна и та же цена, и *дискриминационное ценообразование* — стратегия, при которой могут быть ущемлены интересы каких-то групп покупателей, так как на каждом пункте обслуживания могут устанавливаться разные цены для разных покупателей [15].

Рассматриваемая задача фабричного ценообразования относится, как и многие другие интересные задачи комбинаторной оптимизации, к классу труднорешаемых задач. При разработке методов их решения возникает следующая дилемма. Если сосредоточиться на построении точных методов, то, как показывает вычислительная практика, такие подходы становятся несостоятельными из-за нехватки ресурсов. С другой стороны, при современном опыте построения метаэвристик для большинства задач удаётся построить малотрудоёмкие приближённые алгоритмы решения [4, 5, 7, 8, 23]. Однако возникает проблема с оценкой качества таких решений. Рассматриваемые в работе гибридные алгоритмы — попытка совместить сильные стороны того и другого направлений решения NP-трудных задач. Последние десятилетия это направление, лежащее на стыке математического программирования, комбинаторной оптимизации, математической логики и программирования, быстро развивается [10, 11, 16–20]. Гибридизация методов глобального поиска и метаэвристик — одно из главных направлений проводимых исследований [17].

Предлагаемые в работе алгоритмы решения задачи ценообразования основываются на использовании декомпозиции и метаэвристик. Такой подход к разработке точных и приближённых алгоритмов использовался в [8]. Проведённый там эксперимент показал работоспособность разработанных алгоритмов для задачи о $(r|p)$-центроиде. В настоящей работе оценивается эффективность данных гибридных схем при поиске точных и приближённых решений для задач, имеющих «компактное» представление допустимой области, в отличие от экспоненциального по числу ограничений представления, которое использовалось при описании задачи о $(r|p)$-центроиде в виде смешанно-целочисленной задачи.

Основной недостаток декомпозиционных алгоритмов — медленная сходимость [22]. Поэтому в работе внимание также уделяется поиску методов ускорения декомпозиционных алгоритмов. Ранее подобные исследования проводились, например, в [14, 21, 22, 24]. В [6, 8] эта проблема рассматривалась с точки зрения применения метаэвристик.

В разд. 1 дана математическая постановка задачи и доказано, что задача NP-трудна в сильном смысле. В разд. 2 приведена эквивалентная формулировка задачи ценообразования в виде линейной задачи частично целочисленного программирования и на основе метода декомпозиции построено несколько точных и приближённых схем гибридного алгоритма. В разд. 3 описаны численные результаты работы различных схем гибридного алгоритма.

1. Постановка задачи

Введём обозначения:

$I = \{1, \dots, n\}$ — множество пунктов производства (предприятий);

$J = \{1, \dots, m\}$ — 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;

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

Используя данные обозначения, запишем игру Штакельберга в виде задачи двухуровневого квадратичного программирования:

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

$$p_i \geqslant 0, \quad i \in I,$$

где вектор x — оптимальное решение задачи нижнего уровня

$$\sum_{j \in J} \sum_{i \in I} (b_j - c_{ij} - p_i) x_{ij} \to \max_{x},$$

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

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

Целевая функция задачи определяет доход производителя. Целевая функция нижнего уровня выражает величину сэкономленного потребителями бюджета, а ограничения гарантируют, что каждый потребитель обслуживается не более чем одним предприятием производителя. Также из этих ограничений и определения целевой функции следует, что покупка совершается в том случае, когда это позволяет бюджет потребителя. Если величины c_{ij} принимают только два значения: либо 0, либо $+\infty$, и в двухуровневой задаче требуется, чтобы каждый потребитель был удовлетворён, то получим постановку, рассмотренную в [3].

Под оптимальным решением двухуровневой задачи можно понимать любое её допустимое решение, на котором достигается максимум целевой функции задачи. В целом это удовлетворительное определение. Однако возникают проблемы, когда задача нижнего уровня имеет несколько оптимальных решений, которые с точки зрения потребителей равнозначны. В этом случае производитель, выбрав наилучшим образом размещение пунктов обслуживания и приемлемые с точки зрения потребителей цены, может не досчитаться прибыли. Это связано с тем, что в данной ситуации потребители могут выбрать такой вариант поведения, который является оптимальным решением внутренней задачи, т. е. каждый из потребителей максимально экономит свой бюджет, но при этом хотя бы один из них оказывается в пункте обслуживания с меньшей ценой, чем ожидал производитель. Таким образом, данный выбор не оказывается оптимальным с точки зрения производителя. Чтобы избежать подобной ситуации, предположим, что в случае нескольких оптимальных решений в задаче нижнего уровня каждый потребитель выбирает тот пункт обслуживания (из числа доступных), который ближе к нему. Содержательно это означает, что потребители выбирают оптимальное решение, сохраняющее прибыль производителя. Другими словами, рассматривается кооперативная постановка задачи. Отметим, что в постановке, рассмотренной в [3], такой проблемы не возникает. В силу специфичного выбора транспортных затрат нет разницы между кооперативной и некооперативной постановками задачи.

Предположения о том, что потребители подыгрывают производителю, оказывается также достаточно, чтобы записать двухуровневую задачу в виде следующей задачи квадратичного программирования со смешанными переменными (MP):

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

при ограничениях

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

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

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

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

Как и ранее, 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. Ограничения (4) означают, что каждый потребитель может быть обслужен не более чем в одном предприятии. В дальнейшем предполагаем, что все исходные данные (b_j и c_{ij}) являются рациональными числами.

Теорема 1. *Задача* MP NP-*трудна в сильном смысле.*

Доказательство. Рассмотрим NP-трудную в сильном смысле задачу о минимальном покрытии [2]. Пусть заданы множество $M=\{1,\dots,k\}$ и набор его подмножеств $M_1,\dots,M_n$ таких, что $\bigcup\limits_{i=1}^{n} M_i = M$. Совокупность подмножеств $M_i,\ i\in N_0\subseteq N=\{1,\dots,n\}$, называется *покрытием множества* M, если $\bigcup\limits_{i\in N_0} M_i = M$. Каждому M_i приписан единичный вес. Требуется найти покрытие минимального веса.

Сведём задачу о минимальном покрытии к задаче MP. Для этого рассмотрим частный случай задачи MP. Пусть множество предприятий I равно N, а $J=J_1\cup J_2$, где $|J_1|=|I|$ и $J_2=M$. Определим бюджет каждого потребителя следующим образом: $b_j=3,\ j\in J_1$, и $b_j=2,\ j\in J_2$. Считаем, что $J_1=\{j_i,\ i\in I\}$. Положим $c_{ij_i}=0,\ j_i\in J_1$, и $c_{ij}=0,\ j\in M_i$. Все остальные компоненты матрицы транспортных затрат положим равными 4, т. е. в соответствии с условием (3) сделаем эти транспортные пути закрытыми. Можно считать, что в оптимальном решении (p^*,x^*) задачи MP с вышеуказанными начальными данными компоненты p_i^* вектора цен принимают значения 2 или 3. Действительно, рассмотрим вектор цен $\widetilde{p}$, компоненты которого определяются следующим образом: если $p_i^*\leqslant 2$ для некоторого $i\in I$, то положим $\widetilde{p}_i=2$, и $\widetilde{p}_i=3$, если $p_i^*>2$. Тогда решение $(\widetilde{p},x^*)$ допустимо, если выполняются условия (2) и (3). Предположим, что хотя бы одно из условий (2) нарушено. Пусть $\sum\limits_{i\in I}(b_j-c_{ij}-\widetilde{p}_i)x_{ij}^*<0$ для некоторого $j\in J$. Из допустимости решения (p^*,x^*) следует, что существует $i\in I$ такое, что $b_j-\widetilde{p}_i<0$, где b_j

равно либо 2, либо 3. Отсюда $b_j - p_i^* < 0$ и (p^*, x^*) — недопустимое решение; противоречие. Аналогично рассуждаем и в случае, когда нарушается ограничение (3). Таким образом, можно ограничиться рассмотрением допустимых решений задачи MP с дополнительным ограничением $p_i \in \{2, 3\},\ i \in I$. В этом случае матрица x_{ij} определяется следующим образом: $x_{ij_i} = 1,\ j_i \in J_1$, и $x_{ij} = 1,\ j \in M_i$, если $p_i = 2$. Все остальные компоненты матрицы x_{ij} равны 0. Тем самым целевая функция имеет вид

$$f(p, x) = 2|\widetilde{J}| + 2|\widetilde{N}| + 3(|I| - |\widetilde{N}|) \to \max_{\widetilde{N}},$$

где $\widetilde{N} \subset I$, а $\widetilde{J} = \bigcup\limits_{i \in \widetilde{N}} M_i$. Покажем, что максимум достигается на покрытии минимального веса N_0^*. Допустим, что это не так. Тогда

$$\begin{aligned}(2|\widetilde{J}| + 2|\widetilde{N}| + 3(|I| - |\widetilde{N}|)) - (2|J| + 2|N_0^*| + 3(|I| - |N_0^*|))\\ = (2|\widetilde{J}| - |\widetilde{N}|) - (2|J| - |N_0^*|) > 0.\end{aligned}$$

Если $|\widetilde{J}| = |J|$, то $\widetilde{N}$ — покрытие, вес которого меньше веса оптимального покрытия; противоречие. Пусть $2(|J| - |\widetilde{J}|) < |N_0^*| - |\widetilde{N}|$ и $|\widetilde{J}| < |J|$. Так как N_0^* — покрытие, существует не более $|J| - |\widetilde{J}|$ множеств M_i, где $i \in V \subset N_0^* \setminus \widetilde{N},\ |V| \leqslant |J| - |\widetilde{J}|$, таких, что $\widetilde{N} \cup V$ — покрытие. Поскольку $|V| \leqslant |J| - |\widetilde{J}|$, покрытие N_0^* не оптимально. Теорема 1 доказана.

Из теоремы 1 следует, что при условии P≠NP не существует алгоритма, находящего оптимальное решение задачи MP за полиномиальное или псевдополиномиальное время. Это означает, что любой точный алгоритм решения такой задачи будет иметь экспоненциальную трудоёмкость в худшем случае. В связи с этим возникает проблема разработки приближённых и точных алгоритмов решения, трудоёмкость которых достаточно мала по крайней мере для задач реальной размерности.

2. Гибридные алгоритмы для задачи ценообразования

Для решения задачи фабричного ценообразования предлагается базовый гибридный точный алгоритм, использующий метаэвристики и декомпозицию. Этот алгоритм — основа ряда приближённых и точных алгоритмов, вычислительное исследование которых проводилось в рамках данной работы. Интерес к подобного рода итерационным алгоритмам связан с их полезным свойством порождать верхние и нижние оценки на значение искомого оптимума.

2.1. Декомпозиция и максиминные задачи. Сформулируем в общем виде метод декомпозиции применительно к максиминным задачам. Аналогичные построения и рассуждения верны и для минимаксных задач. С этой точки зрения декомпозицию Бендерса можно рассматривать как сведéние исследуемой задачи к эквивалентной максиминной задаче (координирующая задача или master problem в классической формулировке [13]).

Рассмотрим максиминную задачу

$$z = \max_{x \in X} \min_{y \in Y} R(x, y). \tag{6}$$

В дальнейшем будем предполагать, что хотя бы одно из множеств X или Y конечно и задача разрешима. Перепишем её в эквивалентном виде, который обычно используется при описании декомпозиции Бендерса:

$$\gamma \to \max_{\gamma, x \in X},$$

$$\gamma \leqslant R(x, y), \quad y \in Y.$$

Сформулируем метод декомпозиции для задачи (6). Пусть $\overline{Y} \subseteq Y$.

Шаг 1. Решить релаксированную задачу

$$\overline{z} = \max_{\gamma, x \in X} \gamma,$$

$$\gamma \leqslant R(x, y), \quad y \in \overline{Y}.$$

Пусть x^* — её оптимальное решение.

Шаг 2. Решить подзадачу

$$\underline{z} = \min_{y \in Y} R(x^*, y).$$

Пусть y^* — её оптимальное решение.

Шаг 3. Если $\overline{z} = \underline{z}$, то стоп, иначе $\overline{Y} := \overline{Y} \cup y^*$ и вернуться на шаг 1.

Если на шагах 1 и 2 существуют оптимальные решения x^* и y^*, то $\underline{z} \leqslant z \leqslant \overline{z}$. При этом генерируемая данным методом последовательность $\overline{z}$ невозрастающая. Если на шаге 3 выполняется условие $\overline{z} = \underline{z}$, то $z = \overline{z} = \underline{z}$ — искомое оптимальное значение задачи (6), а (x^*, y^*) — её оптимальное решение. Для упрощения анализа метода будем предполагать, что любая релаксированная задача (шаг 1) и подзадача (шаг 2) разрешимы. Тогда очевидно, что описываемый метод конечен, если на шагах

1 или 2 повторяются оптимальные решения. По-видимому, впервые это отмечено в [13], где доказано, что если хотя бы одно из множеств X или Y конечно, то конечен и метод. В этом случае можно отказаться от требования, что всегда разрешимы задачи, возникающие на шагах 1 и 2. Чтобы гарантировать корректную работу алгоритма, будем считать, что в случае неразрешимости задачи на шаге 1 генерируется вектор $x^* \in X$, который не встречался на предыдущих итерациях. Аналогично поступаем, если неразрешима подзадача на шаге 2.

Важным при реализации этого метода является выбор начального семейства решений $\overline{Y}$. Из описания метода следует, что если множество X конечно, то вне зависимости от того, конечно или бесконечно Y, основная доля ограничений, генерируемая с помощью решений из него, несущественна. Поэтому возникает проблема отыскания таких начальных множеств $\overline{Y}$, которые позволяют быстро найти оптимальное решение исследуемой задачи. Именно здесь открывается простор для использования метаэвристик.

2.2. Гибридные алгоритмы. Идея использовать декомпозицию для решения оптимизационных задач восходит к работам Данцига, Вольфа и Бендерса [13, 22, 25]. Существенный прогресс достигнут при разработке декомпозиционных алгоритмов для задач размещения производства [22]. В области двухуровневых комбинаторных задач размещения первый точный алгоритм декомпозиционного типа предложен в [7] для решения двухуровневой задачи о конкурентной p-медиане (задача о $(r|p)$-центроиде). Этот алгоритм и его модификации — в настоящее время наиболее эффективные точные алгоритмы решения указанной задачи. Идея модифицированных алгоритмов заключается в более глубокой интеграции локального поиска, метаэвристик и декомпозиции.

В [6] для задачи ценообразования предложен гибридный точный алгоритм. Однако он оказался не очень удобным в реализации, так как в релаксированной координирующей задаче целевая функция получилась кусочно-линейной, что существенно ограничило применимость библиотек типа CPLEX при практической реализации декомпозиции. В этом разделе предлагается алгоритм, в котором релаксированная координирующая задача является линейной задачей булевого программирования с одной непрерывной переменной, а подзадача, решаемая на каждой итерации алгоритма, является задачей линейного программирования, что существенно упрощает программную реализацию и вычислительный эксперимент.

Перепишем модель из разд. 1 в виде линейной задачи частично целочисленного программирования. Обозначим через $\overline{p}_i = \max_j(b_j - c_{ij})$ максимально возможную цену на i-м предприятии, а через $z_{ij} \geqslant 0$ — доход, получаемый производителем на i-м предприятии от j-го потребителя. Тогда

$$\max_{p,x,z} \sum_{i \in I} \sum_{j \in J} z_{ij}, \tag{7}$$

$$\sum_{i \in I} (b_j - c_{ij}) x_{ij} - \sum_{i \in I} z_{ij} \geqslant 0, \quad j \in J, \tag{8}$$

$$c_{kj} + p_k - \sum_{i \in I} c_{ij} x_{ij} - \sum_{i \in I} z_{ij} \geqslant 0, \quad k \in I,\ j \in J, \tag{9}$$

$$(1 - x_{ij})\overline{p}_i - z_{ij} + p_i \geqslant 0, \quad i \in I,\ j \in J, \tag{10}$$

$$(1 - x_{ij})\overline{p}_i + z_{ij} - p_i \geqslant 0, \quad i \in I,\ j \in J, \tag{11}$$

$$z_{ij} \leqslant \overline{p}_i x_{ij}, \quad i \in I,\ j \in J, \tag{12}$$

$$\sum_{i \in I} x_{ij} \leqslant 1, \quad j \in J, \tag{13}$$

$$p_i \geqslant 0, \quad x_{ij} \in \{0, 1\}, \quad z_{ij} \geqslant 0, \quad i \in I,\ j \in J. \tag{14}$$

Ограничения (10)–(12) гарантируют, что доход z_{ij} производителя от обслуживания j-го потребителя на i-м предприятии равен p_i, если j-й потребитель выбрал i-е предприятие, и 0 в противном случае.

Применим к задаче подход, реализованный в [6], и получим точный гибридный декомпозиционный алгоритм решения. Задача (7)–(14) при фиксированном x является задачей линейного программирования. Введём двойственные переменные $t,\ \mu,\ \lambda_1,\ \lambda_2,\ \lambda_3 \geqslant 0$ для ограничений (8)–(12). Обозначим через $\delta(x,\ t, \mu,\ \lambda_1, \lambda_2, \lambda_3)$ целевую функцию задачи $D(x)$, двойственной задаче (7)–(14) при фиксированном x, а через $\delta^1_{ij}(x, t, \mu, \lambda_1, \lambda_2, \lambda_3)$, $\delta^2_i(x, t, \mu, \lambda_1, \lambda_2, \lambda_3)$ — ограничения задачи $D(x)$, соответствующие прямым переменным z_{ij} и p_i. Так как задача (7)–(14) разрешима, двойственная задача $D(x)$ при любом x имеет допустимые решения. Следовательно, она может быть неразрешима только при наличии в допустимой области бесконечного ребра, вдоль которого неограниченно убывает целевая функция [1]. В дальнейшем этот факт использован в декомпозиционном алгоритме при порождении отсечений, названных в соответствии со сложившейся традицией *отсечениями допустимости*. В соответствии с этой традицией отсечения, порождаемые с помощью оптимальных вершин, которые получены при решении задачи $D(x)$, называются *отсечениями оптимальности.*

Приведём описание точного гибридного алгоритма решения задачи (7)–(14).

Шаг 1. Для задачи (7)–(14) с помощью алгоритма генетического локального поиска построить семейство решений $x^r,\ r = \overline{1, R}$, для каждого из которых решить задачу

$$\rho(x^r) = \min_{t,\mu,\lambda_1,\lambda_2,\lambda_3} \delta(x^r, t, \mu, \lambda_1, \lambda_2, \lambda_3),$$

$$\delta^1_{ij}(x^r, t, \mu, \lambda_1, \lambda_2, \lambda_3) \leqslant 0, \quad i \in I,\ j \in J,$$

$$\delta^2_i(x^r, t, \mu, \lambda_1, \lambda_2, \lambda_3) \leqslant 0, \quad i \in I.$$

Найти оптимальные значения двойственных переменных $\big(t^r, \mu^r, \lambda^r_1,\ \lambda^r_2,\ \lambda^r_3\big)$, если задача разрешима. Иначе в качестве вектора $\big(t^r, \mu^r, \lambda^r_1, \lambda^r_2, \lambda^r_3\big)$ взять направляющий вектор соответствующего бесконечного ребра. Предположим, что задача $D(x^r)$ разрешима для всех $r = 1, \ldots, Q$. Положим

$$\mathrm{LB} := \max\big\{\delta\big(x^r, t^r, \mu^r, \lambda^r_1, \lambda^r_2, \lambda^r_3\big),\ r = \overline{1, Q}\big\}, \quad U = R - Q.$$

Шаг 2. Решить релаксированную координирующую задачу

$$\max_{x_{ij}\in\{0,1\},\ y\geqslant 0} y$$

$$y \leqslant \delta\big(x, t^q, \mu^q, \lambda^q_1, \lambda^q_2, \lambda^q_3\big), \quad q = \overline{1, Q} \text{ (отсечения оптимальности)},$$

$$\delta\big(x, t^u, \mu^u, \lambda^u_1, \lambda^u_2, \lambda^u_3\big) \geqslant 0, \quad u = \overline{1, U} \text{ (отсечения допустимости)},$$

$$1 - \sum_{i\in I} x_{ij} \geqslant 0, \quad j \in J.$$

Пусть $(\overline{y}, \overline{x})$ — оптимальное решение, $\mathrm{UB} := \overline{y}$ — верхняя граница.

Шаг 3. Решить подзадачу $\rho(x)$ с $x = \overline{x}$.

(i) Задача разрешима. Если $\mathrm{UB} = \rho(\overline{x})$, то стоп, так как оптимальное решение задачи найдено. Иначе $Q := Q + 1,\ t^Q := t,\ \mu^Q := \mu,\ \lambda^Q_1 := \lambda_1$, $\lambda^Q_2 := \lambda_2,\ \lambda^Q_3 := \lambda_3$, где t, μ и $\lambda_1,\ \lambda_2,\ \lambda_3$ — оптимальные двойственные переменные. Если $\rho(\overline{x}) > \mathrm{LB}$, то $\mathrm{LB} := \rho(\overline{x})$ — новая нижняя граница. Вернуться на шаг 2.

(ii) Задача не разрешима. Найти направляющий вектор бесконечного ребра $(t, \mu, \lambda_1, \lambda_2, \lambda_3)$ и положить

$$U := U + 1, \quad t^U := t, \quad \mu^U := \mu, \quad \lambda^U_1 := \lambda_1, \quad \lambda^U_2 := \lambda_2, \quad \lambda^U_3 := \lambda_3.$$

Вернуться на шаг 2.

Этот алгоритм решения задачи конечен [6]. Если на шаге 3 заменить условие совпадения верхней и нижней границ неравенством $(\mathrm{UB} - \mathrm{LB})/\mathrm{UB} < \varepsilon$, где $\varepsilon > 0$, то получим приближённую версию алгоритма.

3. Вычислительный эксперимент

Из описания базового алгоритма видно, что если отбросить шаг 1, то получится классический алгоритм Бендерса [24]. Основной недостаток таких алгоритмов — медленная сходимость. Поэтому в нашей работе значительное внимание уделяется поиску методов ускорения декомпозиционных алгоритмов. В настоящее время имеется большое количество исследований на эту тему, в которых разрабатываются быстрые модификации декомпозиционных алгоритмов [22, 24]. В [6, 8] эта проблема рассматривалась с точки зрения применения метаэвристик.

Так как релаксированная координирующая задача, возникающая на шаге 2, является задачей смешанного программирования с булевыми переменными, для её решения на каждой итерации базового алгоритма требуются значительные вычислительные ресурсы. Поэтому основная доля исследований на эту тему посвящена или уменьшению числа решаемых на шаге 2 релаксированных координирующих задач, или различным процедурам, упрощающим нахождение решения текущей релаксированной координирующей задачи. В [21] впервые предложено порождать отсечения, используя решения линейной релаксации релаксированной координирующей задачи. Эта идея основана на том, что любая вершина (или любой крайний луч [1]) задачи, двойственной к подзадаче, порождает отсечение оптимальности (или допустимости) для релаксированной координирующей задачи. Таким образом, вместо сложной задачи на шаге 2 решается задача линейного программирования. Другой подход предложен в [14]. Вместо того чтобы искать оптимальное решение релаксированной координирующей задачи, авторы предложили остановить процесс решения на некотором допустимом решении, которое затем используется на шаге 3 для порождения отсечения оптимальности (или допустимости). Основной минус таких упрощений в том, что нельзя гарантировать сходимость процесса. Проще всего использовать такие идеи для генерации приближённых решений, либо, накопив с какого-то момента достаточно большое множество отсечений с небольшими затратами вычислительных ресурсов, вернуть процесс к исходной схеме и таким образом гарантировать сходимость. Далее для задачи фабричного ценообразования предлагаются два подхода — двухфазная схема и двухфазная схема с контролируемым числом ограничений. Эти схемы используются для ускорения базового алгоритма.

Перечислим все тестируемые алгоритмические схемы, после чего перейдём к их непосредственному описанию:

1) базовая схема (шаги 1–3) (general scheme);

2) двухфазная схема (2-round scheme);

3) двухфазная схема с приближённой версией алгоритма на 1-й фазе (2-round scheme with the approximate 1st round (Appr. 2-r.));

4) двухфазная схема с контролируемым числом ограничений (Box);

5) упрощённая двухфазная схема (simplified 2-round scheme).

Первые четыре схемы точные, а схема 5 даёт неплохую верхнюю оценку. Схемы 2–4 позволяют уменьшить число итераций и время счёта базовой схемы.

Идея двухфазного метода (двухфазной схемы) основывается на хорошо известном факте из теории. Если известно множество отсечений, которое определяет выпуклую оболочку множества допустимых решений целочисленной задачи, то декомпозиционный алгоритм находит оптимальное целочисленное решение за одну итерацию [22]. Множество отсечений, определяющих допустимую область линейной релаксации целочисленной задачи, с вычислительной точки зрения найти легче, и оно содержит выпуклую оболочку допустимых решений целочисленной задачи. Поэтому в алгоритмах, основанных на двухфазной схеме, сначала решается линейная релаксация задачи (7)–(14) приведённым выше базовым алгоритмом, у которого на шаге 2 решается релаксированная координирующая задача с непрерывными переменными x_{ij}. Затем при решении исходной задачи (7)–(14) к семейству отсечений шага 1 добавляется оптимальное семейство отсечений, полученное при решении линейной задачи.

Идея схемы 3 заключается в том, чтобы ограничить время работы базового алгоритма решения задачи (7)–(14) на 1-й фазе, когда решается задача с непрерывными переменными x_{ij}. Для этого берём $0 < \varepsilon < 1$, используемое для ограничения времени выполнения 1-й фазы. При выполнении неравенства $(\mathrm{UB} - \mathrm{LB})/\mathrm{UB} < \varepsilon$ осуществляется переход к второй фазе.

Идея схемы 4 заключается в том, чтобы контролировать число отсечений. Она является гибридом двухфазной схемы и поиска с запретами [23]. На каждой фазе используется три списка: два ограниченных, а третий — произвольной длины. Один ограниченный список предназначен для хранения отсечений оптимальности, а другой — для отсечений допустимости. Когда добавляется новое отсечение оптимальности, из соответствующего списка выбрасывается самое старое. Если верхняя граница, определяемая текущей релаксированной координирующей задачей, оказывается меньше предыдущей, то выброшенное ограничение возвращается и заносится в неограниченный список. Аналогичным образом выполняются операции при работе с отсечениями допустимости.

В упрощённой двухфазной схеме вычисления завершаются на шаге 2 на первой итерации второй фазы.

Т а б л и ц а 1

Результат работы схем 2 и 3

		2-round		App. 2-r.(0.02)		App. 2-r.(0.05)	
n	m	Δ	t	Δ	t	Δ	t
5	10	0	8.8"	0	8.5"	0	8.4"
5	15	0	90"	0	69"	0	110"
5	20	0.013	956"	0.009	776"	0.017	848"
5	25	0.076	1544"	0.104	1821"	0.135	1831"
5	30	0.166	1823"	0.179	1821"	0.207	1825"
		App. 2-r.(0.1)		App. 2-r.(0.2)			
n	m	Δ	t	Δ	t		
5	10	0	12.1"	0	17.7"		
5	15	0	133"	$<$0.001	315"		
5	20	0.033	1094"	0.057	1293"		
5	25	0.183	1807"	0.224	1815"		
5	30	0.244	1822"	0.327	1822"		

Т а б л и ц а 2

Результат работы схемы 4

		Box(10)		Box(15)		Box(30)	
n	m	Δ	t	Δ	t	Δ	t
5	10	0	17.5"	0	21"	0	15.9"
5	15	0	192"	0	205"	0	232"
5	20	0.025	1251"	0.025	1343"	0.031	1348"
5	25	0.133	1748"	0.129	1724"	0.139	1785"
5	30	0.187	1705"	0.197	1709"	0.199	1712"

Все эти схемы тестировались на примерах небольших размерностей до $n = 30$ предприятий и $m = 50$ потребителей, в которых значения величин b_j и c_{ij} выбирались равновероятно из интервала $[1, 99]$. Тесты проводились на двадцати примерах каждой размерности. Чтобы не ограничивать размерности решаемых задач, время работы схем 1–4 ограничивалось получасом (табл. 1 и 2), а упрощённой схемы — часом (табл. 3). Так как время работы и точность базовой схемы (в том случае, когда время работы алгоритма ограничивалось) оказались на порядок хуже остальных и зачастую этот метод даже не находил допустимого решения за ограниченное время, результат тестирования первой схемы в нижеприведённых таблицах отсутствует. Следует отметить, что время работы и точность схем 2–4, как показал эксперимент, в среднем отличаются друг от друга не более чем в 2–3 раза.

В табл. 1 и 2 представлены результаты экспериментов, которые проводились для сравнения трудоёмкости точных схем. В табл. 1 представлены относительная погрешность (величина $\Delta = (\mathrm{UB} - \mathrm{LB})/\mathrm{UB}$) и время t работы схем 2 и 3 при выборе $\varepsilon = 0.02, 0.05, 0.1, 0.2$, в табл. 2 — относительная погрешность Δ и время t работы схемы 4, при этом эксперимент проводился при суммарном количестве отсечений оптимальности и допустимости, равном 20, 30 и 60.

Эксперимент со схемами 2 и 3 (табл. 1) показал, что на небольших размерностях ($5 \times 10 - 20$) для ускорения счёта можно часть отсечений линейной задачи выбросить. Из табл. 1 видно, что схема 2 проигрывает по времени схеме 3 при выборе $\varepsilon = 0.02, 0.05$, когда число потребителей равно 10 и 20, и $\varepsilon = 0.02$, когда число потребителей равно 15. При переходе к большим размерностям с ростом числа итераций появляется необходимость переносить во вторую фазу все отсечения линейной задачи. Здесь существенной проблемой становится плохая сходимость декомпозиционных алгоритмов. В среднем число отсечений, генерируемых для получения оптимального значения, практически не отличается для схем 2 и 3. Поэтому, убирая часть отсечений линейной задачи в схеме 3, приходится дополнять их гораздо более дорогостоящими отсечениями булевой задачи, что приводит к увеличению времени работы алгоритмов на основе данной схемы с ростом ε.

Схема 4 при сравнении со схемами 2 и 3 оказалась вполне конкурентоспособной. Из табл. 2 видно, что схема Box при выборе разного суммарного количества отсечений оптимальности и допустимости имеет практически одинаковые показатели времени счёта и точности, которые не уступают показателям схемы 3 с $\varepsilon = 0.2$ на размерностях $5 \times 10 - 15$, и с $\varepsilon = 0.1, 0.2$ на размерностях $5 \times 20 - 25$, и проигрывает только схеме 3 с $\varepsilon = 0.02$ на размерности 5×30. Стоит отметить, что на больших размерностях схема 4 уступает совсем немного схеме 2 (меньше чем в 1.3 раза по точности). Эксперимент с двухфазной схемой показал также, что с ростом размерности задачи отсечения её линейной релаксации всё хуже и хуже аппроксимируют выпуклую оболочку допустимых решений, что выражалось в стремительном росте числа итераций 2-й фазы: в среднем одна итерация для размерности 5×5, на порядок больше для размерности 5×10, на 2 порядка больше для размерности 5×15 и т. д. Тем не менее, казалось бы, естественная попытка в этой ситуации уменьшить количество отсечений, генерируемых на первой фазе, нередко приводила к росту числа итераций второй фазы и общего времени счёта (схема 3). Таким образом, с точки зрения уменьшения общей трудоёмкости вычислений очень важно качество аппроксимации на первой фазе. Как показывает следующий эксперимент с приближённой схемой, это позволяет гарантировать и хорошие верхние оценки.

Т а б л и ц а 3

Результат работы схемы 5

		Simplified 2-round		Linear	
n	m	Δ	t	Δ_1	t
5	10	0.0458	7.25"	0.125	<1"
5	15	0.0627	11.65"	0.142	<1"
5	30	0.136	199.5"	0.215	1.15"
5	50	0.192	1722.6"	0.273	4.75"
10	20	0.115	171.15"	0.179	1,65"
10	30	0.15	1032.45"	0.212	4.5"
20	20	0.111	1080.8"	0.165	5.8"
20	40	0.26	4025.6"	0.236	156.8"
30	30	0.204	3292.05"	0.201	202.1"

В упрощённой двухфазной схеме вычисления завершаются на шаге 2 одной из первых итераций второй фазы, что позволяет получить достаточно неплохую верхнюю границу задачи фабричного ценообразования. Чтобы упростить анализ результатов, в эксперименте вычисления завершались на шаге 2 первой итерации. Также время работы первой фазы ограничивалось одним часом. Для некоторых примеров размерностей 20×40 и 30×30 за отведённый час не удалось оптимально решить задачу, возникающую на первой фазе. При меньших размерностях такого не наблюдалось. В последних двух колонках табл. 3 записана величина

$$\Delta_1 = \frac{\mathrm{Lin} - \mathrm{Opt}}{\mathrm{Lin}},$$

где Lin — оптимальное значение задачи (7)–(14) с непрерывными переменными x_{ij}, и время t работы пакета CPLEX на этой задаче. Из табл. 3

видно, что относительное уклонение верхней границы превысило относительное уклонение оптимального решения задачи (7)–(14) с непрерывными переменными x_{ij} только при размерностях 20×40 и 30×30. Здесь можно улучшить результаты, если останавливаться не на первой итерации второй фазы, а просто ограничить время счёта.

Заключение

Рассмотрен частный случай задачи размещения и ценообразования — задача с фабричным ценообразованием. Показано, что при P≠NP для неё не существует точных псевдополиномиальных алгоритмов решения.

Вычислительный эксперимент показал, что двухфазная схема и двухфазная схема с контролем ограничений позволяют получить алгоритмы декомпозиционного типа, более быстрые чем базовый алгоритм. В том числе эксперимент с упрощённой двухфазной схемой показал, что с помощью алгоритмов такого типа можно получить неплохие верхние оценки. Однако этого недостаточно для получения быстрых точных и приближённых алгоритмов гибридного типа. Поэтому в дальнейшем необходимо продолжить разработку и исследование гибридных алгоритмов, сделав упор на

(a) разработку способов формирования хорошего начального множества отсечений на шаге 1, которое способствовало бы ускорению сходимости алгоритма;

(b) сокращение количества решаемых релаксированных координирующих задач, рассмотрев альтернативные способы генерации отсечений.

В заключение авторы выражают искреннюю благодарность Ю. А. Кочетову за плодотворные дискуссии и замечания.

ЛИТЕРАТУРА

1. **Васильев Ф. П.** Методы оптимизации. — М.: Факториал Пресс, 2002. — 829 с.
2. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. — М.: Мир. 1982. — 416 с.
3. **Дементьев В. Т., Шамардин Ю. В.** Задача о выборе цен на продукцию при условии обязательного удовлетворения спроса // Дискрет. анализ и исслед. операций. Сер. 2. — 2002. — Т. 9, № 2. — С. 31–40.
4. **Панин А. А.** Генетический алгоритм для одной задачи ценообразования // Тр. ИВМиМГ СО РАН. Сер. Информатика. — 2009. — Вып. 9. — С. 190–196.

5. **Плясунов А. В.** О вычислительных возможностях метаэвристик // Мат. Росс. конф. «Дискретная оптимизация и исследование операций» (Владивосток, 7–14 сентября 2007 г.). — Новосибирск: Изд-во Ин-та математики, 2007. — С. 61–68.
6. **Плясунов А. А.** Гибридные методы решения сложных комбинаторных задач, использующие декомпозицию // Сб. докл. 8-й междунар. конф. «Интеллектуальная обработка информации» (Республика Кипр, Пафос, 17–24 октября, 2010 г.). — М.: МАКС Пресс, 2010. — С. 286–289.
7. **Alekseeva E., Kochetova N., Kochetov Yu., Plyasunov A.** A hybrid memetic algorithm for the competitive p-median problem // Preprints 13th IFAC Symp. Information Control Problems in Manufacturing. — Moscow, 2009. — P. 1516–1520.
8. **Alekseeva E. V., Kochetova N. A., Kochetov Yu. A., Plyasunov A. V.** A heuristic and exact methods for the dicrete $(r|p)$-centroid problem. — Berlin: Springer-Verl., 2010. — P. 11–22. (Lect. Notes Comput. Sci.; Vol. 6022).
9. **Bouhtou M., Grigoriev A., Van Der Kraaij A. F., Van Hoesel S., Spieksma F., Uetz M.** Pricing bridges to cross a river // Naval Res. Logistics. — 2007. — Vol. 54, N 4. — P. 411–420.
10. **Chandru V., Hooker J. N.** Optimization methods for logical inference. — New York: John Wiley & Sons, 1999. — 365 p.
11. **Dechter R.** Constraint processing. — San Francisco: Morgan Kaufmann, 2003. — 481 p.
12. **Dempe S. J.** Foundations of bilevel programming. — Dordrecht: Kluwer Acad. Publ., 2002. — 481 p.
13. **Geoffrion A. V.** Generalized Benders decomposition // J. Optimization Theory Appl. — 1972. — Vol. 10, N 4. — P. 237–260.
14. **Gote G., Laughton M. A.** Large scale mixed integer programming: Benders-type heuristics // Eur. J. Oper. Res. — 1984. — Vol. 16. — P. 327–333.
15. **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.
16. **Van Hentenryck P., Milano M.** Hybrid optimization: the ten years of CPAIOR. — New York: Springer Sci.+Business Media, 2011. — 570 p.
17. **Hooker J. N.** Integrated methods for optimization. — New York: Springer Sci.+Business Media, 2007. — 490 p.
18. **Hooker J. N.** Logic-based methods for optimization: combining optimization and constraint satisfaction. — New York: Wiley–Intersci., 2000. — 520 p.
19. **Lodi A., Milano M., Toth P.** Integration of AI and OR techniques in constraint programming for combinatorial optimization problems. — Berlin: Springer-Verl., 2010. — 369 p. (Lect. Notes Comput. Sci.; Vol. 6140).
20. **Marriott K., Stuckey P.** Programming with constraints: an introduction. —

Cambridge: The MIT Press, 1998. — 476 p.

21. McDaniel D., Devine M. A modified Benders partitioning algorithm for mixed integer programming // Manage. Sci. — 1977. — Vol. 24. — P. 312–319.

22. Mirchandani P. D., Francis R. L. Decomposition methods for facility location problems // Discrete Location Theory. — New York: Wiley & Sons, 1990. — P. 439–478.

23. Rebeiro C. C., Hansen P. Essays and surveys in metaheuristics. — Boston: Kluwer Acad. Publ., 2002. — 651 p.

24. Vanderbeck F., Wolsey L. A. Reformulation and decomposition of integer programs. — Montreal: Center Oper. Res. Econometrics, 2009. — Vol. 2188. — 71 p.

25. Vanderbeck F., Savelsbergh M. W. P. A generic view of Dantzig–Wolfe decomposition for integer programming // Oper. Res. Lett. — 2006. — Vol. 34. — P. 296–306.

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.