ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Май—июнь 2015. Том 22, № 3. С. 36–54

УДК 519.87+519.854 DOI: 10.17377/daio.2015.22.480

СРАВНЕНИЕ МЕТАЭВРИСТИК ДЛЯ РЕШЕНИЯ ДВУХУРОВНЕВОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРЕДПРИЯТИЙ И ФАБРИЧНОГО ЦЕНООБРАЗОВАНИЯ *)

Ю. А. Кочетов[1,2], *А. А. Панин*[1], *А. В. Плясунов*[1,2]
[1]Институт математики им. С. Л. Соболева,
пр. Коптюга, 4, 630090 Новосибирск, Россия
[2]Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия
e-mail: jkochet@math.nsc.ru, arteam1897@gmail.com, apljas@math.nsc.ru

Аннотация. Показано, что исследуемая задача принадлежит классу Poly-APX. Для её решения разработаны приближённые алгоритмы, использующие генетический локальный поиск и VND-метаэвристику. Приводятся результаты вычислительных экспериментов на исходных данных из библиотеки тестовых задач «Дискретные задачи размещения». Предлагаемые алгоритмы сравниваются с ранее известными приближёнными алгоритмами и точным методом из библиотеки CPLEX. Результаты экспериментов свидетельствуют о высокой эффективности разработанных методов и возможности решать задачи большой размерности. Табл. 2, библиогр. 30.

Ключевые слова: двухуровневая задача, размещение, ценообразование, локальное улучшение с чередующимися окрестностями, генетический локальный поиск, двухуровневая метаэвристика.

Введение

Постановки, связанные с размещением производства и/или ценообразования, образуют широкий спектр математических моделей, методов и приложений в исследовании операций [7, 13, 17, 18, 20, 21, 30]. В большей части моделей различные аспекты размещения исследуются без учёта цен. Процессы размещения производства и ценообразования обычно изучаются отдельно и независимо друг от друга, потому что они принадлежат к разным горизонтам планирования. Размещение производства

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 13–07–00016) и поддержано грантовым финансированием ИИВТ МОН РК.

представляет собой долгосрочное планирование, а ценообразование относится к краткосрочному планированию [19]. Как правило, выбирают сначала размещение, а затем цены [26]. Тем не менее такое разделение размещения и ценообразования может оказаться неприемлемым. Например, в тех случаях, когда размещение предприятий зависит от распределения потребителей и их спроса, который, в свою очередь, зависит от цен [22]. Более того, разделение размещения и ценообразования также нецелесообразно, когда нет необходимости знать точные цены на производимый продукт, а достаточно знать лишь диапазон цен в выбранной нише рынка, позволяющих противостоять конкурентам [9]. Поэтому современные подходы к выбору эффективного механизма взаимодействия процессов размещения производства и ценообразования основываются на их совместном анализе в рамках одной модели [8–10, 14, 28, 29]. Однако для оценки качества принимаемых решений необходимо уметь адекватно оценивать реакцию рынка и, в частности, потребителей на предлагаемый вариант размещения и ценообразования. С этой целью удобно моделировать весь процесс в виде задачи двухуровневого программирования [4, 15].

На основе [24] исследован ряд математических моделей размещения производства и ценообразования в условиях конкуренции (см., например, [19, 22, 26]). В данной статье рассмотрим более простой случай, когда на рынке уже есть предприятия, производящие однородный продукт, и известны цены на каждом из них [16]. Положение этих предприятий и цены предполагаются фиксированными и в дальнейшем не меняются. Сформулируем содержательную постановку предлагаемой задачи в виде игры Штакельберга. На верхнем уровне компания, входящая на рынок, размещает r своих предприятий и выбирает цены на каждом из них. На нижнем уровне каждый из потребителей выбирает то из предприятий, на котором транспортные затраты и затраты на приобретение продукции в сумме минимальны. Покупка совершается в том случае, когда это позволяет бюджет потребителя (порог, заданный уже существующими предприятиями и ценами на них). Требуется выбрать такие предприятия и определить такие цены на каждом предприятии, чтобы доход компании был максимален. В этой игре компания является лидером, клиенты образуют множество последователей. Также предполагается, что клиенты сосредоточены в конечном множестве дискретных точек. Если у потребителя есть несколько предприятий с одинаковой минимальной суммой платежей, то он выберет предприятие с минимальными транспортными затратами.

В рамках данной содержательной постановки естественно исследовать следующие три стратегии ценообразования. Равномерное ценообразование (uniform pricing) [22] — на всех пунктах обслуживания устанавливается одна и та же цена. Фабричное ценообразование (mill pricing) — стратегия ценообразования, при которой на каждом предприятии выбирается своя цена на производимый продукт [22]. Дискриминационное ценообразование (discriminator pricing) — стратегия ценообразования, при которой могут быть ущемлены интересы каких-то групп покупателей, так как на каждом предприятии могут устанавливаться разные цены для разных покупателей [22]. В статье рассматривается только вторая стратегия. Близкие по смыслу постановки с равномерным и дискриминационным ценообразованием рассмотрены в [19, 22, 26].

В разд. 2 представлена двухуровневая модель нелинейного программирования. В разд. 3 показано, что она принадлежит классу Poly-APX, а соответствующая ей задача оценивания является нетривиальной Δ_2^p-задачей, а также приводятся гибридные эвристики для её решения. Поскольку подзадача ценообразования, которая возникает при выборе любого размещения предприятий, NP-трудна в сильном смысле, предлагаемые эвристики развиваются как двухэтапные: сначала применяется локальный поиск для выбора размещения предприятий при фиксированном векторе цен, а затем — локальный поиск для выбора вектора цен при заданном размещении предприятий. Для подзадачи ценообразования используется либо генетический локальный поиск, либо VND-метаэвристика. Для поиска размещения используется локальный поиск. В разд. 4 приводятся результаты вычислительных экспериментов на исходных данных из библиотеки тестовых задач «Дискретные задачи размещения». Предлагаемые алгоритмы сравниваются с ранее известными приближёнными алгоритмами и точным методом из библиотеки CPLEX.

1. Математическая модель

Введём следующие обозначения: $I = \{1, \dots, n\}$ — множество возможных мест открытия предприятий, $J = \{1, \dots, m\}$ — множество потребителей, r — число открываемых предприятий, $b_j \geqslant 0$ — ценовой порог (бюджет) j-го потребителя, $c_{ij} \geqslant 0$ — матрица транспортных затрат потребителей, $p_i \geqslant 0$ — цена товара на i-м предприятии, $y_i = 1$, если предприятие i открывается, и $y_i = 0$ в противном случае, $x_{ij} = 1$, если потребитель j обслуживается предприятием i, и $x_{ij} = 0$ в противном случае.

Используя данные обозначения, запишем игру Штакельберга в виде следующей задачи двухуровневого квадратичного программирования (FLMP):

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

$$\sum_{i\in I} y_i = r, \quad x \in F(y,p), \quad p_i \geqslant 0, \quad y_i \in \{0,1\}, \quad i \in I,\ j \in J,$$

где $F(y,p)$ является множеством оптимальных решений внутренней задачи

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

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

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

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

Целевая функция двухуровневой задачи определяет доход компании, а ограничение гарантирует, что будет открыто ровно r предприятий. Целевая функция нижнего уровня описывает стратегию каждого потребителя — в максимальной степени экономить свой бюджет, 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.

При описании игры Штакельберга введено предположение, 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, которое сохраняет прибыль производителя. Другими словами, рассматривается кооперативная постановка задачи. Аналогичное соглашение использовалось в [6, 7]. 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.

Этого предположения достаточно, чтобы записать двухуровневую задачу в виде следующей эквивалентной задачи квадратичного программирования со смешанными переменными:

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

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

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

$$\sum_{i\in I} y_i = r,$$

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

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

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

Целевая функция, как и в двухуровневой задаче, определяет доход компании. Первая группа ограничений гарантирует, что потребители не могут выйти за рамки своих бюджетов. Выполнение второй группы ограничений приводит к минимизации общих затрат каждого потребителя на покупку продукта и его транспортировку. Следующее ограничение гарантирует, что будет открыто ровно r предприятий. Четвёртая группа ограничений означает, что каждый потребитель может обслуживаться не более, чем на одном предприятии. Последняя группа ограничений влечёт, что потребители могут обслуживаться только открытыми предприятиями. Сохраним то же обозначение FLMP для этой переформулировки двухуровневой задачи.

Введём дополнительные соглашения и обозначения. В дальнейшем предполагаем, что все исходные данные (b_j и c_{ij}) являются рациональными числами, OPT(FLMP) — оптимальное значение целевой функции задачи FLMP, $g(y, p, x)$ — значение целевой функции задачи FLMP на допустимом решении (y, p, x).

Пусть (y, p) — вектор, удовлетворяющий ограничениям задачи лидера. Обозначим через $f(y, p)$ оптимальное значение целевой функции задачи для заданных y и p. Заметим, что величина $f(y, p)$ вычисляется за полиномиальное время [7].

2. Эвристики

Известно [7, 16], что задача FLMP NP-трудна в сильном смысле даже при заданном размещении предприятий. Покажем, что можно получить более детальную характеристику сложности задачи, уточнив положение соответствующей ей задачи оценивания $\mathrm{OPT}_{\mathrm{FLMP}}$, которая является задачей распознавания, в полиномиальной иерархии. Присоединим к входу задачи FLMP целочисленный параметр k и получим вход задачи оценивания, в которой надо определить, является ли k оптимальным значением целевой функции на множестве допустимых решений задачи FLMP или нет. Также сопоставим задаче FLMP стандартную задачу распознавания D(FLMP), в которой для каждого входа задачи FLMP надо решить, существует или нет допустимое решение со значением целевой функции, большим или равным k, где k — целочисленный параметр.

Далее используются три базовых класса P, NP и co-NP задач распознавания, образующие первый уровень полиномиальной иерархии классов сложности, и класс Δ_2^p — элемент второго уровня этой иерархии [1, 11]. Напомним определение последнего класса сложности. Задача распознавания D *принадлежит классу* Δ_2^p, если существует детерминированная оракульная машина Тьюринга, которая распознает за полиномиальное время задачу D, используя в качестве оракула некоторый язык из класса NP (NP-оракул). Класс Δ_2^p часто обозначают через P^{NP}. Задачу D назовём *нетривиальной-*Δ_2^p *задачей* [27], если $D \in \Delta_2^p,\ D \notin \Sigma_1^p \cup \Pi_1^p$.

Теорема 1. *Если* NP≠co-NP, *то задача* $\mathrm{OPT}_{\mathrm{FLMP}}$ *является нетривиальной* Δ_2^p*-задачей.*

Доказательство. Покажем, что выполняются следующие условия: $\mathrm{OPT}_{\mathrm{FLMP}} \in \Delta_2^p$ и если NP ≠co-NP, то $\mathrm{OPT}_{\mathrm{FLMP}} \notin$ NP∪ co-NP.

Нетрудно проверить, что задача распознавания D(FLMP) лежит в классе NP. Можно модифицировать сводимости из [7, 16], для того чтобы показать, что задача D(FLMP) NP-полна в сильном смысле. Из вида целевой функции задачи FLMP следует, что её оптимальное значение ограничено сверху полиномом от длины записи её исходных данных. Поэтому, используя двоичный поиск и задачу D(FLMP) в качестве оракула, можно за полиномиальное время найти решение задачи $\mathrm{OPT}_{\mathrm{FLMP}}$. По определению класса Δ_2^p это означает, что $\mathrm{OPT}_{\mathrm{FLMP}} \in \Delta_2^p$. Так как задача D(FLMP) NP-полна в сильном смысле и тривиально сводится к задаче $\mathrm{OPT}_{\mathrm{FLMP}}$ за полиномиальное время, из результатов, полученных в [27], следует, что выполняются условия $\mathrm{OPT}_{\mathrm{FLMP}} \notin \Sigma_1^p \cup \Pi_1^p$. Стало быть, при условии, что NP≠co-NP, задача $\mathrm{OPT}_{\mathrm{FLMP}}$ является нетривиальной Δ_2^p-задачей. Теорема 1 доказана.

Из теоремы 1 следует, что при условии NP≠co-NP, а это более сильная гипотеза, чем классическая P≠NP [1, 11], нахождение оптимального решения FLMP с ростом размерности становится достаточно трудоёмким процессом. Тогда имеет смысл рассмотреть вопрос нахождения «неплохого» допустимого решения. Обычно в этом случае рассматривают сложность задачи с точки зрения построения эффективного алгоритма нахождения приближённого решения с гарантированной оценкой точности, т. е. положение оптимизационной задачи в иерархии аппроксимационных классов [12]

$$\mathrm{PO} \subseteq \mathrm{FPTAS} \subseteq \mathrm{PTAS} \subseteq \mathrm{APX} \subseteq \text{Log-APX} \subseteq \text{Poly-APX} \subseteq \text{Exp-APX} \subseteq \mathrm{NPO}.$$

Далее используется класс Poly-APX, состоящий из оптимизационных задач, для каждой их которых существует некоторый полиномиальный приближённый алгоритм решения с полиномиальной оценкой, зависящей от длины записи исходных данных задачи, для точности относительной погрешности полученного решения. Все необходимые формальные определения можно найти в [12]. Используя [6, 7], докажем следующее утверждение.

Теорема 2. *Задача* FLMP *принадлежит классу* Poly-APX.

Доказательство. Обозначим через FLMP^1 задачу FLMP с не более чем одним открытым предприятием $r = 1$. В [7] показано, что задача FLMP^1 полиномиально разрешима. Оптимальное решение этой задачи используется для построения допустимого решения исходной задачи. Пусть (y^1, p^1, x^1) — оптимальное решение задачи FLMP^1 и $y^1_{i_1}$ — единственная единичная компонента вектора y^1. Выберем $r-1$ различных индексов $i_2, \dots, i_r$ из множества I таких, что $i_1 \notin \{i_2, \dots, i_r\}$. Положим

$$y_i^{i_1,r} = \begin{cases} 1, & \text{если } i \in \{i_1, i_2, \dots, i_r\}, \\ 0 & \text{в противном случае,} \end{cases}$$

$$x_{ij}^{i_1,r} = \begin{cases} x^1_{ij}, & \text{если } i = i_1, \\ 0 & \text{в противном случае,} \end{cases}$$

$$p_i^{i_1,r} = \begin{cases} p^1_i, & \text{если } i = i_1, \\ 1 + \max\limits_{j \in J} b_j & \text{в противном случае.} \end{cases}$$

Очевидно, что вектор $(y^{i_1,r}, p^{i_1,r}, x^{i_1,r})$ является допустимым решением задачи FLMP и $g(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}) = g(y^1, p^1, x^1)$. Проверим, что

$$\mathrm{OPT}(\mathrm{FLMP}) \leqslant r g(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}).$$

Возьмём произвольное допустимое решение $(\widetilde{y}, \widetilde{p}, \widetilde{x})$ задачи FLMP. Пусть

$$i^* = \arg \max_{i \in I \mid \widetilde{y}_i = 1} \Big\{ \sum_j \widetilde{p}_i \widetilde{x}_{ij} \Big\},$$

$$y_i^{i^*} = \begin{cases} 1, & \text{если } i = i^*, \\ 0 & \text{в противном случае,} \end{cases} \qquad x_{ij}^{i^*} = \begin{cases} \widetilde{x}_{ij}, & \text{если } i = i^*, \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда $(y^{i^*}, \widetilde{p}, x^{i^*})$ является допустимым решением задачи FLMP[1]. Следовательно,

$$g(\widetilde{y}, \widetilde{p}, \widetilde{x}) = \sum_{i \in I} \widetilde{p}_i \sum_{j \in J} \widetilde{x}_{ij} \leqslant r \sum_{i \in I} \widetilde{p}_i \sum_{j \in J} x_{ij}^{i^*} \leqslant rg(y^{i_1,r}, p^{i_1,r}, x^{i_1,r}).$$

Поэтому OPT(FLMP) $\leqslant rg(y^{i_1,r}, p^{i_1,r}, x^{i_1,r})$. Теорема 2 доказана.

Поскольку уклонение оказывается большим, далее развиваются два гибридных алгоритма, основанных на локальном поиске: спуск по чередующимся окрестностям VND и генетический локальный поиск GLS [2, 5, 23]. В [16] для решения задачи FLMP разработаны двухэтапные эвристики: сначала применяется локальный поиск по переменным y_i для выбора размещения r предприятий при фиксированном векторе цен p_i, затем локальный поиск применяется для выбора вектора цен p_i при заданном размещении предприятий.

При таком подходе возникают две подзадачи, каждая из которых проще исходной задачи. В [16] для решения подзадачи ценообразования использовалась VNS-метаэвристика [23] с окрестностями N_k, $k = 1, \dots, K$. В окрестности N_k, $k = 1, \dots, K$, соседом вектора цен p_i является любой вектор, который получается изменением не более k компонент вектора p_i, соответствующих открытым предприятиям. Для решения подзадачи размещения предприятий применялись две метаэвристики: метод отжига и метод чередующихся окрестностей (VNS-метаэвристика). Соседние решения для текущего вектора y_i порождались с помощью двух окрестностей. Первая окрестность это окрестность k-swap, а вторая — окрестность Lin–Kernighan [25]. В окрестности k-swap любой сосед текущего размещения y_i может быть получен с помощью следующей процедуры: открываем $k' \leqslant r$ новых предприятий и закрываем ровно столько же ранее открытых. Пусть $S = \{i \mid y_i = 1\}$ и χ_S — характеристическая функция множества S. Очевидно, что $y_i = \chi_S(i)$, $i = 1, \dots, m$. Далее для упрощения изложения будем отождествлять булев вектор и соответствующую характеристическую функцию. Окрестность Lin–Kernighan для вектора y_i строится с помощью следующей итерационной процедуры.

Шаг 1. Выбираем пару элементов $i_1 \in S$, $i_2 \in \overline{S}$ таких, что значение функции $f(\chi_{S \cup \{i_2\} \setminus \{i_1\}}, p)$ является максимальным среди всех таких пар. Если таких пар несколько, то берём любую из них.

Шаг 2. Выбрасываем из множества S элемент i_1 и добавляем элемент i_2.

Шаг 3. Повторяем шаги 1 и 2 k раз. При этом на каждой итерации в качестве индексов $i_1 \in S$, $i_2 \in \overline{S}$ можно использовать только не возникавшие ранее на предыдущих итерациях.

Эта процедура определяет k соседей текущего размещения y_i. Параметр k выбирается из интервала $[\min(r, m-r), \max(r, m-r)]$. Локальный поиск относительно этой окрестности используется в методе отжига каждый раз, когда температура уменьшается, а в VNS-метаэвристике — когда рекордное решение не может быть улучшено в течении некоторого количества итераций.

Так как отыскание наилучшего соседа текущего размещения относительно любой из этих окрестностей оказывается трудоёмкой процедурой для больших k, для VNS-метаэвристики использовались окрестности с небольшими k ($k \leqslant 3$) и с $k = 1$ — для метода отжига [16].

2.1. Окрестностные структуры. Приведём описание окрестностных структур, которые используются в гибридных алгоритмах.

Обозначим через $\mathrm{Flip}(p \mid y)$ окрестность вектора цен p для заданного размещения y. Эта окрестность состоит из всех векторов, каждый из которых отличается от p ровно одной компонентой. Если $\overline{p} \in \mathrm{Flip}(p \mid y)$ и для некоторого k имеем $\overline{p}_i = p_i$, $i \neq k$, то для вычисления значения целевой функции $f(y, \overline{p})$ и значения компоненты $\overline{p}_k$ решается задача FLMP с заданным размещением y, ценами $\overline{p}_i$, $i \neq k$, и одной переменной p_k. Эта задача разрешима за полиномиальное время [7].

Окрестность $\mathrm{Swap}(y \mid p)$ определяется для заданного вектора цен p. Она содержит все векторы $\overline{y}$, где $\sum \overline{y}_i = r$, с расстоянием Хэмминга от $\overline{y}$ до y, равным 2. Рассмотрим такой вектор $\overline{y}$. Тогда для некоторых i_0 и i_1 таких, что $y_{i_0} = 0$, $y_{i_1} = 1$, выполняются равенства $\overline{y}_i = y_i$, $i \neq i_0, i_1$, $\overline{y}_{i_0} = 1$, $\overline{y}_{i_1} = 0$. Так как одно предприятие закрылось и появилось новое, необходимо пересчитать цены с индексами i_0 и i_1. Пусть $\overline{p}_i = p_i$, $i \neq i_0$, i_1, $\overline{p}_{i_1} = 0$, а значение компоненты $\overline{p}_{i_0}$ получится при решении задачи FLMP с заданным размещением предприятий $\overline{y}$, ценами $\overline{p}_i, i \neq i_0$, и одной переменной p_{i_0}.

Обозначим через $LS^{\mathrm{pr}}(p \mid y)$ алгоритм локального поиска относительно окрестности $\mathrm{Flip}(p \mid y)$, а алгоритм локального поиска относительно окрестности $\mathrm{Swap}(y \mid p)$ — через $LS^{\mathrm{loc}}(y \mid p)$.

2.2. Метаэвристики Gen и VND. Приведём описание двух метаэвристик с алгоритмом локального поиска $LS^{\mathrm{pr}}(p \mid y)$. Для решения исследуемой задачи воспользуемся генетическим алгоритмом локального поиска GLS, который интересен с теоретической и практической точек зрения. Он может рассматриваться как вариант Memetic алгоритма, который использует разные жадные стратегии и операторы кроссовера [5]. Генетический алгоритм локального поиска является итерационным методом. На каждой итерации имеется набор локальных оптимумов относительно используемой окрестности. Этот набор называется популяцией. Он развивается в течение ряда итераций, пока не выполнится критерий остановки. Схема предлагаемой метаэвристики может быть представлена следующим образом.

Алгоритм Gen

Вход: y — булев вектор, задающий размещение предприятий, $I_{\max}$ — максимальное число итераций.

Выход: наилучший среди элементов финальной популяции вектор цен p.

Шаг 0: $i \leftarrow 0$. Случайно порождаем начальную популяцию векторов цен. Применяем алгоритм $LS^{\mathrm{pr}}(p \mid y)$ к каждому вектору p из популяции.

Шаг 1. Случайно выбираем два элемента популяции в качестве родителей. С помощью равномерного кроссовера создаём потомка p' для выбранных родителей. Применяем локальный поиск к вектору цен p' и находим локальный оптимум $p^* := LS^{\mathrm{pr}}(p' \mid y)$.

Шаг 2. Модифицировать популяцию $i \leftarrow i + 1$. Если $i \leqslant I_{\max}$, то перейти на шаг 1, иначе Стоп.

Метод спуска с чередующимися окрестностями VND выполняет некоторое количество итераций с разными окрестностями до тех пор, пока не будет получен локальный оптимум относительно всех используемых окрестностей. Пусть $N_1, N_2, \ldots, N_K$ — множество из K окрестностных структур. Начиная с первой структуры N_1, VND-алгоритм выполняет шаги локального поиска, пока это возможно. Далее локальный поиск продолжается из уже найденного локального оптимума относительно окрестностной структуры N_2. Если в процессе просмотра элементов данной структуры удаётся найти лучшее решение, чем исходный оптимум, то VND-алгоритм возвращается к использованию структуры N_1, иначе алгоритм продолжает локальный поиск относительно окрестностной структуры N_3, и т. д. Таким образом, если алгортитм добрался до окрестностной структуры N_K и в процессе просмотра её элементов не удаётся улучшить текущее решение, то оно является локальным оптимумом относительно всех использованных окрестностных структур. В этом случае алгоритм прекращает свою работу. Окрестности обычно просматриваются в определённом порядке. Например, от самой маленькой по мощности окрестности, в которой процесс просмотра элементов является самым быстрым, до самой большой и самой трудоёмкой при реализации локального поиска.

Далее приводится описание нашей реализации этой эвристики.

Алгоритм VND (Improved)

Вход: y — булев вектор, задающий размещение предприятий, $I_{\max}$ — максимальное число итераций алгоритма, d — число улучшений текущего решения, допускаемых в процессе работы процедуры, k — параметр окрестности k-Flip$(p \mid y)$.

Выход: наилучший вектор p среди найденных в процессе работы алгоритма векторов цен.

Шаг 0: $i \leftarrow 0$. Случайно порождаем начальный вектор цен p.

Шаг 1. Применяем локальный поиск к вектору цен p и находим локальный оптимум $p^* \leftarrow LS^{\mathrm{pr}}(p \mid y)$.

Шаг 2: $i \leftarrow i+1$. Если $i > I_{\max}$, тогда стоп, иначе $p \leftarrow \mathrm{Improved}(k, p^* \mid y)$. Если $f(y,p) > f(y,p^*)$, то переходим на шаг 1, иначе стоп.

В предлагаемой реализации процедуры Improved используются окрестностные структуры 2-Flip$(p\,|\,y)$ и 3-Flip$(p\,|\,y)$. Для ускорения локального поиска по каждой из этих окрестностных структур используется d-улучшенная стратегия поиска, идея которой заключается в том, что как только найдено ровно d улучшений в данной окрестности, поиск прекращается [23].

2.3. Гибридные алгоритмы LS+Gen и LS+VND. В предлагаемых алгоритмах на каждой итерации к текущему решению (y,p) вначале применяется локальный поиск относительно окрестности Flip$(p \mid y)$, а затем — локальный поиск относительно окрестности Swap$(y \mid \overline{p})$, где вектор цен $\overline{p}$ — результат поиска по предыдущей окрестности. В качестве критерия остановки используется достижение заранее заданного числа итераций. Результатом работы алгоритма является наилучшее найденное решение. Процедура первого шага основывается на генетическом локальном поиске в первом алгоритме и на VND-эвристике — во втором алгоритме. Используя алгоритмы из п. 2.2 и алгоритм локального поиска $LS^{\mathrm{loc}}(y \mid p)$, перейдём к описанию гибридных метаэвристик. Пусть Y — множество всех векторов $\overline{y}$ таких, что $\sum \overline{y}_i = r$, а через L обозначим табу-лист.

Алгоритм LS+Gen

Вход: $I_{\max}$ — максимальное число итераций алгоритма.

Выход: наилучшее найденное решение.

Шаг 0. Положим $i \leftarrow 0$, $L \leftarrow \varnothing$. Случайно порождаем начальное размещение y.

Шаг 1. Если $i > I_{\max}$, то стоп, иначе $L := L \cup \{y\}$, $i \leftarrow i+1$, применяем метаэвристику Gen и найдём локальный оптимум p.

Шаг 2. Найдём локальный оптимум $y^* := LS^{\mathrm{loc}}(y \mid p)$. Пусть p^* — вектор цен, соответствующий размещению y^*. Если $f(y^*, p^*) \leqslant f(y, p)$, то выбираем любое размещение из множества $y \in Y \setminus L$, иначе $y := y^*$, и переходим на шаг 1.

Чтобы получить алгоритм LS+VND, надо заменить алгоритм Gen алгоритмом VND.

3. Вычислительные результаты

Тестирование гибридных алгоритмов проводилось на персональном компьютере с процессором Intel Core i7-3612QM и 4Gb оперативной памяти. Алгоритмы LS+Gen и LS+VND сравнивались с уже известными алгоритмами — двухуровневой эвристикой VNS+VNS [16] и методом ветвей и границ из известной библиотеки CPLEX. При тестировании использовались входы размерности $n = 40$, 100, $m = 100$, $r = 5$ из библиотеки «Discrete Location Problems» (табл. 1).

Отдельно приводятся результаты сравнения алгоритмов LS+Gen и LS+VND (табл. 2) на этих же входах, но другой размерности: $n = 40$, $m = 100$, $r = 10, 15$. В [16] помимо двухуровневой эвристики VNS+VNS исследовалась эвристика SA+VNS. В связи с тем, что этот алгоритм показывает результаты не лучше, чем алгоритм VNS+VNS, при этом требуя существенно большего вычислительного времени, здесь соответствующие результаты не воспроизводятся.

В табл. 1 и 2 первый столбец обозначает код каждого входа. Столбец $n(r)$ задаёт количество возможных мест открытия предприятий и число открываемых предприятий. Столбцы доход, время, итерация задают наилучший найденный доход, требуемое вычислительное время и номер итерации, на которой было найдено наилучшее решение. Для метода ветвей границ из пакета CPLEX приводятся результаты только для входов наименьшей размерности. Процесс счёта обрывался по истечении 24 часов, и наилучшее найденное решение предъявлялось как результат работы алгоритма. Для входов большей размерности по истечении указанного времени алгоритм не успевал найти хотя бы одно допустимое решение. Как следует из табл. 1, алгоритмы LS+Gen и LS+VND не уступают по качеству найденных решений эвристике VNS+VNS и методу ветвей и границ, но при этом затрачивают меньшее вычислительное время. Как показывает табл. 2, метод LS+VND оказывается предпочтительней метода LS+Gen с точки зрения затрат вычислительного времени для больших r.

Таблица 1

Сравнение алгоритмов LS+Gen, LS+VND, VNS+VNS и пакета CPLEX

Instance		VNS+VNS		CPLEX	LS+Gen			LS+VND		
	$n(r)$	доход	время	доход	доход	итерация	время	доход	итерация	время
1	40(5)	2245	45m	2226	2245	5	36s	2245	20	4s
2	40(5)	2259	51m	2259	2259	14	97s	2259	72	16s
3	40(5)	2019	41m	2019	2019	8	47s	1984	12	3s
4	40(5)	1533	42m	1508	1533	63	347s	1552	22	5s
5	40(5)	2386	46m	2313	2386	11	77s	2346	34	8s
6	40(5)	1960	60m	1949	1956	9	57s	1987	17	4s
7	40(5)	2179	60m	2142	2179	55	415s	2178	58	14s
8	40(5)	2139	51m	2139	2139	30	224s	2140	31	7s
9	40(5)	1895	59m	1877	1904	17	115s	1900	45	10s
10	40(5)	2209	37m	2209	2209	4	32s	2252	4	1s
11	100(5)	2235	1h 9m		2230	7	48s	2235	24	14s
12	100(5)	2240	3h 12m		2240	295	2015s	2233	31	18s
13	100(5)	1923	1h 19m		1923	16	107s	1957	13	8s
14	100(5)	2133	1h 48m		2133	466	3194s	2118	15	9s
15	100(5)	2099	1h 58m		2099	27	197s	2153	25	15s
16	100(5)	2237	1h 10m		2237	108	806s	2182	126	75s
17	100(5)	1888	1h 15m		1893	31	202s	1921	108	62s
18	100(5)	1825	2h 51m		1825	48	312s	1871	4	2s
19	100(5)	1767	1h 43m		1767	8	48s	1767	4	3s
20	100(5)	2363	1h 2m		2368	55	369s	2368	139	84s

4. Заключение

В статье предлагаются две двухуровневые метаэвристики для решения двухуровневой задачи размещения и фабричного ценообразования. Обе метаэвристики основаны на локальном поиске: одна в форме метода подъёма с чередующимися окрестностями, другая в форме генетического локального поиска. Приводится экспериментальное сравнение этих эвристик с двухуровневой эвристикой VNS+VNS [16], а также с методом ветвей и границ из библиотеки CPLEX.

В дальнейшем предполагается использовать другие окрестности, например, окрестность Lin–Kernighan, а также математические эвристики.

Таблица 2

Сравнение алгоритмов LS+Gen и LS+VND

Instance		LS+Gen			LS+VND		
	$n(r)$	доход	итерация	время	доход	итерация	время
21	40(10)	2650	43	2157s	2626	421	635s
22	40(10)	2617	39	2266s	2621	182	282s
23	40(10)	2351	31	1601s	2327	73	106s
24	40(10)	1888	55	2305s	1895	78	112s
25	40(10)	2827	24	942s	2793	88	141s
26	40(10)	2261	53	2902s	2289	349	538s
27	40(10)	2481	15	498s	2480	528	855s
28	40(10)	2447	56	1842s	2443	192	311s
29	40(10)	2245	47	2629s	2246	758	1250s
30	40(10)	2497	58	3054s	2596	112	180s
31	40(15)	2804	26	3162s	2796	794	3413s
32	40(15)	2758	26	3635s	2797	346	1616s
33	40(15)	2515	13	1908s	2477	146	620s
34	40(15)	1952	29	3161s	1983	453	2046s
35	40(15)	2908	11	1881s	2870	9	45s
36	40(15)	2286	23	3129s	2436	25	114s
37	40(15)	2529	16	2144s	2571	288	1369s
38	40(15)	2532	8	545s	2558	302	1510s
39	40(15)	2355	6	630s	2372	659	3091s
40	40(15)	2424	24	3431s	2698	52	261s

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
2. **Давыдов И. А., Кочетов Ю. А., Младенович Н., Уросевич Д.** Быстрые метфэвристики для дисретной задачи о $(r|p)$-центроиде // Автоматика и телемеханика. 2014. Вып. 4. С. 677–687.
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. **Кочетов Ю. А., Плясунов А. В.** Полиномиально разрешимый класс задач двухуровневого линейного программирования // Дискрет. анализ и исслед. операций. Сер. 2. 1997. Т. 1, № 2. С. 23–33.
5. **Кочетов Ю. А., Плясунов А. В.** Генетическтй локальный поиск для задачм о размещении графа на доли ограниченной мощности // Журн. вычисл. математики и мат. физики. 2012. Т. 52, № 1. С. 157–167.
6. **Панин А. А., Плясунов А. В.** О сложности двухуровневых задач размещения и ценообразования // Дискрет. анализ и исслед. операций. 2014. Т. 21, № 5. С. 54–66.

7. **Плясунов А. В., Панин А. А.** Задача ценообразования. Ч. 1: Точные и приближённые алгоритмы решения // Дискрет. анализ и исслед. операций. 2012. Т. 9, № 2. С. 31–40.
8. **Aboolian R., Berman O., Krass D.** Competitive facility location model with concave demand // Eur. J. Oper. Res. 2007. Vol. 181. P. 598–619.
9. **Aboolian R., Berman O., Krass D.** Optimizing pricing and location decisions for competitive service facilities charging uniform price // J. Oper. Res. Soc. 2008. Vol. 59. P. 1506–1519.
10. **Adler N., Smilowitz K.** Hub-and-spoke network alliances and mergers: price-location competition in the airline industry // Transp. Res. Part B. 2007. Vol. 41. P. 394–409.
11. **Attallah M.** Algorithms and theory of computation handbook. Boca Raton: CRC Press LLC, 1998. 1312 p.
12. **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.
13. **Bouhtou M., Grigoriev A., Van Der Kraaij A. F., Van Hoesel S., Spieksma F., Uetz M.** Pricing bridges to cross a river // Nav. Res. Logist. 2007. Vol. 54, No. 4. P. 411–420.
14. **Dasci A., Laporte G.** Location and pricing decisions of a multistore monopoly in a spatial market // J. Region Sci. 2004. Vol. 44. P. 489–515.
15. **Dempe S.** Foundations of bilevel programming. Dordrecht: Kluwer Acad. Publ., 2002. 320 p.
16. **Diakova Z., Kochetov Yu.** A double VNS heuristic for the facility location and pricing problem // Electron. Notes Discrete Math. 2012. Vol. 39. P. 29–34.
17. **Drezner Z.** Facility location: A survey of applications and methods. New York: Springer-Verl., 1995. 572 p.
18. **Drezner T., Eiselt H. A.** Consumers in competitive location models // Facility location: Applications and theory. Berlin: Springer-Verl., 2004. P. 151–178.
19. **Eiselt H. A., Laporte G., Thisse J.-F.** Competitive location models: A framework and bibliography // Transp. Sci. 1993. Vol. 27, No. 1. P. 44–54.
20. **Eiselt H. A., Marianov V.** Foundations of location analysis. New York: Springer-Verl., 2011. 510 p. (Int. Ser. Oper. Res. Manag. Sci.; Vol. 155).
21. **Grigoriev A., van Loon J., Sviridenko M., Uetz M., Vredeveld T.** Optimal bundle pricing with monotonicity constraint // Oper. Res. Lett. 2008. Vol. 36, No. 5. P. 609–614.
22. **Hanjoul P., Hansen P., Peeters D., Thisse J.-F.** Uncapacitated plant location under alternative spatial price policies // Manage Sci. 1990. Vol. 36, No. 1. P. 41–57.
23. **Hansen P., Mladenovic N.** Variable neighborhood search // Eur. J. Oper.

Res. 2001. Vol. 130, No. 3. P. 449–467.

24. **Hotelling H.** Stability in competition // Econom. J. 1929. Vol. 39, No. 153. P. 41–57.
25. **Kochetov Yu., Alekseeva E., Levanova T., Loresh M.** Large neighborhood local search for the p-median problem // Yugosl. J. Oper. Res. 2005. Vol. 15, No. 1. P. 53–63.
26. **Lederes Ph. J., Thisse J.-F.** Competitive location on network under delivered pricing // Oper. Res. Lett. 1990. Vol. 9, No. 3. P. 147–154.
27. **Leggett E. W., Moore D. J.** Optimization problems and the polynomial hierarchy // Theor. Comput. Sci. 1981. Vol. 15, No. 3. P. 279–289.
28. **Marianov V., Lüer-Villagra A.** A competitive hub location and pricing problem // Eur. J. Oper. Res. 2013. Vol. 231, No. 3. P. 734–744.
29. **Serra D., ReVelle Ch.** Competitive location and pricing on networks // Geograph. Anal. 1999. Vol. 31, No. 2. P. 109–129.
30. **Vives X.** Oligopoly pricing: old ideas and new tools. Cambridge, MA: MIT Press, 2001. 441 p.

Кочетов Юрий Андреевич,
Панин Артем Александрович,
Плясунов Александр Владимирович

Статья поступила
14 марта 2015 г.
Исправленный вариант —
6 апреля 2015 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
May–June 2015. Volume 22, No. 3. P. 36–54

UDC 519.87+519.854 DOI: 10.17377/daio.2015.22.480

COMPARISON OF METAHEURISTICS FOR THE BILEVEL FACILITY LOCATION AND MILL PRICING PROBLEM

Yu. A. Kochetov[1,2], *A. A. Panin*[1], *A. V. Plyasunov*[1,2]
[1]Sobolev Institute of Mathematics,
4 Koptyug Ave., 630090 Novosibirsk, Russia
[2]Novosibirsk State University,
2 Pirogov St., 630090 Novosibirsk, Russia
e-mail: jkochet@math.nsc.ru, arteam1897@gmail.com, apljas@math.nsc.ru

Abstract. We consider the bilevel nonlinear facility location and pricing problem. We assume that facilities can charge different prices and the objective is to maximize the total revenue. It is known that the problem is NP-hard in the strong sense even for the given facility location. We show that it belongs to class Poly-APX. We present two hybrid algorithms based on local search: variable neighborhood descent and genetic local search. Being compared with previously known algorithms and CPLEX software, these algorithms show their competitiveness. Computational experiments were conducted on instances from the benchmark library «Discrete Location Problems». Tab. 2, bibliogr. 30.

Keywords: bilevel problem, location, pricing, VND metaheuristic, genetic local search metaheuristics, two level metaheuristics.

REFERENCES

1. **M. R. Garey** and **D. S. Johnson,** *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman, San Francisco, 1979. Translated under the title *Vychislitel'nye mashiny i trudnoreshaemye zadachi,* Mir, Moscow, 1982.
2. **I. A. Davydov, Yu. A. Kochetov, N. Mladenović,** and **D. Urošević,** Fast metaheuristics for the discrete $(r|p)$-centroid problem, *Avtom. Telemekh.,* No. 4, 106–119, 2014. Translated in *Autom. Remote Control,* **75**, No. 4, 677–687, 2014.
3. **V. T. Dement'ev** and **Yu. V. Shamardin,** The problem of price selection for production under the condition of obligatory satisfaction of demand, *Diskretn. Anal. Issled. Oper., Ser. 2,* **9**, No. 2, 31–40, 2002.

4. **Yu. A. Kochetov** and **A. V. Plyasunov,** A polynomially solvable class of two-level linear programming problems, *Diskretn. Anal. Issled. Oper., Ser. 2,* **1**, No. 2, 23–33, 1997.
5. **Yu. A. Kochetov** and **A. V. Plyasunov,** Genetic local search the graph partitioning problem under cardinality constraints, *Zh. Vychisl. Mat. Mat. Fiz.,* **52**, No. 1, 164–176, 2012. Translated in *Comput. Math. Math. Phys.,* **52**, No. 1, 157–167, 2012.
6. **A. A. Panin** and **A. V. Plyasunov,** On complexity of the bilevel location and pricing problems, *Diskretn. Anal. Issled. Oper.,* **21**, No. 5, 54–66, 2014. Translated in *J. Appl. Ind. Math.,* **8**, No. 4, 574–581, 2014.
7. **A. V. Plyasunov** and **A. A. Panin,** The pricing problem. Part I: Exact and approximate algorithms, *Diskretn. Anal. Issled. Oper.,* **19**, No. 5, 83–100, 2012. Translated in *J. Appl. Ind. Math.,* **7**, No. 2, 241–251, 2013.
8. **R. Aboolian, O. Berman,** and **D. Krass,** Competitive facility location model with concave demand, *Eur. J. Oper. Res.,* **181**, No. 2, 598–619, 2007.
9. **R. Aboolian, O. Berman,** and **D. Krass,** Optimizing pricing and location decisions for competitive service facilities charging uniform price, *J. Oper. Res. Soc.,* **59**, No. 11, 1506–1519, 2008.
10. **N. Adler** and **K. Smilowitz,** Hub-and-spoke network alliances and mergers: Price-location competition in the airline industry, *Transp. Res., Part B,* **41**, No. 4, 394–409, 2007.
11. **M. J. Attallah,** ed., *Algorithms and Theory of Computation Handbook,* CRC Press, Boca Raton, 1998.
12. **G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela,** and **M. Protasi,** *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties,* Springer-Verlag, Berlin, 1999.
13. **M. Bouhtou, A. Grigoriev, S. van Hoesel, A. F. van der Kraaij, F. C. R. Spieksma,** and **M. Uetz,** Pricing bridges to cross a river, *Naval Res. Logist.,* **54**, No. 4, 411–420, 2007.
14. **A. Dasci** and **G. Laporte,** Location and pricing decisions of a multistore monopoly in a spatial market, *J. Reg. Sci.,* **44**, No. 3, 489–515, 2004.
15. **S. Dempe,** *Foundations of Bilevel Programming,* Kluwer Acad. Publ., Dordrecht, 2002.
16. **Z. Diakova** and **Yu. A. Kochetov,** A double VNS heuristic for the facility location and pricing problem, *Electron. Notes Discrete Math.,* **39**, No. 4, 29–34, 2012.
17. **Z. Drezner,** ed., *Facility Location: A Survey of Applications and Methods,* Springer-Verlag, New York, 1995.
18. **T. Drezner** and **H. A. Eiselt,** Consumers in competitive location models, in Z. Drezner and H. W. Hamacher, eds., *Facility Location: Applications and Theory,* pp. 151–178, Springer-Verlag, Berlin, 2004.
19. **H. A. Eiselt, G. Laporte,** and **J.-F. Thisse,** Competitive location models:

A framework and bibliography, *Transp. Sci.,* **27**, No. 1, 44–54, 1993.
20. **H. A. Eiselt** and **V. Marianov,** *Foundations of Location Analysis,* Springer, New York, 2011 (Int. Ser. Oper. Res. Manag. Sci.; Vol. 155).
21. **A. Grigoriev, J. van Loon, M. Sviridenko, M. Uetz,** and **T. Vredeveld,** Optimal bundle pricing with monotonicity constraint, *Oper. Res. Lett.,* **36**, No. 5, 609–614, 2008.
22. **P. Hanjoul, P. Hansen, D. Peeters,** and **J.-F. Thisse,** Uncapacitated plant location under alternative spatial price policies, *Manag. Sci.,* **36**, No. 1, 41–57, 1990.
23. **P. Hansen** and **N. Mladenović,** Variable neighborhood search, *Eur. J. Oper. Res.,* **130**, No. 3, 449–467, 2001.
24. **H. Hotelling,** Stability in competition, *Econ. J.,* **39**, No. 153, 41–57, 1929.
25. **Yu. A. Kochetov, E. V. Alekseeva, T. V. Levanova,** and **M. A. Loresh,** Large neighborhood local search for the p-median problem, *Yugosl. J. Oper. Res.,* **15**, No. 1, 53–63, 2005.
26. **Ph. J. Lederer** and **J.-F. Thisse,** Competitive location on network under delivered pricing, *Oper. Res. Lett.,* **9**, No. 3, 147–153, 1990.
27. **E. W. Leggett Jr.** and **D. J. Moore,** Optimization problems and the polynomial hierarchy, *Theor. Comput. Sci.,* **15**, No. 3, 279–289, 1981.
28. **A. Lüer-Villagra** and **V. Marianov,** A competitive hub location and pricing problem, *Eur. J. Oper. Res.,* **231**, No. 3, 734–744, 2013.
29. **D. Serra** and **Ch. ReVelle,** Competitive locations and pricing on networks, *Geogr. Anal.,* **31**, No. 2, 109–129, 1999.
30. **X. Vives,** *Oligopoly Pricing: Old Ideas and New Tools,* MIT Press, Cambridge, 2001.

Yury A. Kochetov
Artem A. Panin
Alexander V. Plyasunov

Received
14 March 2015
Revised
6 April 2015