ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Январь–март 2019. Том 26, № 1. С. 55–73

УДК 519.8 DOI 10.33048/daio.2019.26.625

О ТРЁХУРОВНЕВОЙ ЗАДАЧЕ КОНКУРЕНТНОГО ЦЕНООБРАЗОВАНИЯ С РАВНОМЕРНОЙ И ФАБРИЧНОЙ ЦЕНОВЫМИ СТРАТЕГИЯМИ *)

А. В. Губарева[1,a], *А. А. Панин*[1,2,b],
А. В. Плясунов[1,2,c], *Л. В. Сом*[1,d]

[1]Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

[2]Институт математики им. С. Л. Соболева СО РАН,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

E-mail: [a]a.gubareva@g.nsu.ru, [b]aapanin1988@gmail.com,
[c]apljas@math.nsc.ru, [d]milisom@mail.ru

Аннотация. Рассматривается трёхуровневая задача ценообразования, формулируемая в виде игры Штакельберга, в которой две компании — лидер и последователь — конкурируют друг с другом за спрос потребителей путём установления цен на своих предприятиях на однородную продукцию. Первым делает ход лидер. Затем, имея полную информацию о его ходе, принимает решение последователь. После чего каждый потребитель выбирает то предприятие, на котором ему выгоднее обслуживаться. Лидер и последователь используют разные стратегии ценообразования: равномерную и фабричную соответственно.

Исследуется поведение доходов компаний в зависимости от числа предприятий. Для этого предложен точный алгоритм декомпозиционного типа. Помимо этого разработан гибридный приближённый алгоритм, основанный на идеях спуска с чередующимися окрестностями и покоординатного спуска. Табл. 2, библиогр. 12.

Ключевые слова: игра Штакельберга, задача конкурентного ценообразования, трёхуровневая задача, равномерное ценообразование, фабричное ценообразование, точный алгоритм, приближённый алгоритм, спуск с чередующимися окрестностями, декомпозиция, покоординатный спуск.

---

*)Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 17–11–01021).

Введение

Данная работа продолжает исследования, начатые в [2, 3, 11]. В [2, 3] рассматривались двухуровневые модели взаимодействия производителя и потребителей, в которых на верхнем уровне единственная компания-производитель определяет цены на продукцию, основываясь на одной из трёх стратегий ценообразования: равномерной, фабричной или дискриминационной. При равномерной стратегии устанавливается единая цена на всю продукцию. В отличие от этого фабричное ценообразование характеризуется выбором своей цены на каждом предприятии. При дискриминационном ценообразовании ещё жёстче ущемляется принцип равенства: каждому потребителю на каждом предприятии назначается своя цена.

Текущее исследование посвящено трёхуровневой задаче ценообразования, которую содержательно можно описать в виде игры Штакельберга трёх игроков: лидера, последователя и клиентов(потребителей). Поведение каждого из игроков описывается некоторой оптимизационной задачей. Считается, что у лидера и последователя уже открыты предприятия, на которых они производят однородную продукцию. Лидер выбирает для всех своих предприятий одну цену на продукцию, т. е. использует равномерную стратегию ценообразования. Далее последователь, учитывая ход лидера, определяет цены на продукцию для каждого из своих предприятий (фабричное ценообразование). Затем каждый клиент выбирает предприятие, на котором ему выгоднее обслуживаться, минимизируя суммарные затраты на транспортировку и приобретение товара (в единичном количестве). Целью игры является установление цены на предприятиях лидера, при которой его доход после оптимального хода последователя будет максимальным.

Принято соглашение, что оптимальное решение является последовательным оптимумом [5]. Это значит, что сперва среди всех своих оптимальных решений потребители выбирают такое, которое оптимизирует доход последователя. В свою очередь, последователь выбирает такой оптимальный набор цен, который максимизирует доход лидера. В [11] такие решения были названы F-кооперативными. В настоящей работе мы выбираем первое определение, так как оно более точно отражает существо дела.

В [11] рассматривалась трёхуровневая задача конкурентного ценообразования с равномерной стратегией ценообразования. Для её решения был предложен точный полиномиальный алгоритм. В отличие от неё, исследуемая трёхуровневая задача ценообразования с равномерной стратегией у лидера и фабричной у последователя NP-трудна в сильном смысле. Из-за связи уровней задачи одной из наиболее сложных проблем данной постановки является разработка точного алгоритма. Существующие точные алгоритмы для одноуровневых и двухуровневых оптимизационных задач не имеют переложения для данной постановки. Поэтому в данной работе в первую очередь акцентируется внимание на разработке алгоритмов решения. Помимо этого исследуется поведение целевых функций участников игры в зависимости от числа предприятий того или иного игрока.

На данный момент практически отсутствуют исследования по взаимодействию сторон рынка. Актуальность данной работы заключается в том, что рассматриваемая модель ценообразования может использоваться в реальной практике, поскольку описывает конкурентное поведение участников рынка.

В разд. 1 речь пойдёт о математической модели. В разд. 2 исследуются связи трёхуровневой постановки с полиномиальной и аппроксимационной иерархиями. В разд. 3 предлагаются алгоритмы её решения: точный и приближённый. В разд. 4 описываются результаты численного исследования разработанных алгоритмов и исследуется поведение целевых функций производителей в зависимости от числа предприятий.

1. Математическая модель

Введём следующие обозначения:
$I_L = \{1, \dots, M_l\}$ — множество предприятий лидера,
$I_F = \{1, \dots, M_f\}$ — множество предприятий последователя,
$J = \{1, \dots, N\}$ — множество потребителей,
$b_j \in \mathbb{Z}^+$ — бюджет j-го потребителя,
$c_{ij} \in \mathbb{Z}^+$ — транспортные затраты j-го потребителя, если он обслуживается на i-м предприятии,
$p \in \mathbb{Z}^+$ — цена продукции на предприятиях лидера (равномерное ценообразование),
$q_i \in \mathbb{Z}^+$ — цена продукции на предприятиях последователя (фабричное ценообразование),

$$x_{ij} = \begin{cases} 1, & \text{если потребитель } j \text{ обслуживается на предприятии } i, \\ 0 & \text{в противном случае.} \end{cases}$$

Используя данные переменные и обозначения, запишем игру Штакельберга в виде следующей трёхуровневой задачи квадратичного программирования с целочисленными и булевыми переменными.

Задача лидера:

$$\sum_{j \in J} \sum_{i \in I_L} p x_{ij} \to \max_{p,q,x} \tag{1}$$

при следующих ограничениях:

$$p \in \mathbb{Z}^+, \tag{2}$$

$$(x,q) \in \mathcal{F}_F^*(p). \tag{3}$$

Целевая функция (1) определяет доход лидера, $\mathcal{F}_F^*(p)$ является множеством оптимальных решений задачи последователя.

Задача последователя:

$$\sum_{j\in J}\sum_{i\in I_F} q_i x_{ij} \to \max_{q,x} \tag{4}$$

при следующих ограничениях:

$$q_i \in \mathbb{Z}^+, \quad i \in I_F, \tag{5}$$

$$x \in \mathcal{F}_C^*(p,q). \tag{6}$$

Целевая функция (4) определяет доход последователя, а $\mathcal{F}_C^*(p,q)$ является множеством оптимальных решений задачи клиента.

Задача клиентов:

$$\sum_{j\in J}\Big(\sum_{i\in I_L}(b_j - c_{ij} - p)x_{ij} + \sum_{i\in I_F}(b_j - c_{ij} - q_i)x_{ij}\Big) \to \max_x \tag{7}$$

при таких ограничениях:

$$\sum_{i\in I_L\cup I_F} x_{ij} \leqslant 1, \tag{8}$$

$$x_{ij} \in \{0,1\}, \quad i \in I_L \cup I_F,\ j \in J. \tag{9}$$

Целевая функция (7) определяет экономию клиентов. Ограничения (8) и (9) гарантируют, что клиент может обслуживаться не более чем одним предприятием лидера или последователя. Также ограничения задачи клиентов и вид целевой функции дают гарантию, что покупка совершается только в том случае, когда это позволяет бюджет потребителя.

Вектор (p,q,x) называется *допустимым решением трёхуровневой задачи*, если выполняются ограничения задачи лидера, т. е. $p \in \mathbb{Z}^+$, $(x,q) \in \mathcal{F}_F^*(p)$. Оптимальным решением трёхуровневой задачи является любое её допустимое решение, на котором достигается максимум целевой функции задачи лидера. В целом это подходящее определение. Но в некоторых случаях задача клиентов может иметь несколько оптимальных решений, которые будут одинаковы с точки зрения суммарной экономии. В таком случае лидер и последователь могут, выбрав приемлемые с точки зрения клиентов цены, не досчитаться прибыли. Это объясняется тем, что в случае, если хотя бы один из клиентов выбирает такой вариант поведения, при котором он обслуживается в пункте лидера (последователя), то прибыль последователя (лидера) становится меньше, чем ожидалось. Тем самым данная стратегия потребителя не станет оптимальной с точки зрения лидера или последователя. Чтобы не попасть в такую ситуацию, введём соглашение, что в случае нескольких оптимальных решений в задаче нижнего уровня каждый клиент будет действовать согласно такому решению, при котором прибыль последователя будет максимальной. Схожая ситуация может возникнуть на втором уровне. Аналогичным образом примем следующее соглашение: последователь выберет такое решение из множества его оптимальных решений, при котором доход лидера, т. е. значение целевой функции задачи первого уровня, будет максимальным. Таким образом, здесь принято соглашение, что под оптимальным решением понимается последовательный оптимум [5].

Отдельно остановимся на ограничениях целочисленности цен лидера и конкурента. Рассмотрим следующие примеры задачи. Допустим, у лидера и конкурента есть по одному предприятию, на которых могут обслуживаться два клиента. У обоих клиентов единичный бюджет и нулевые транспортные затраты на предприятии лидера. Пусть у первого клиента транспортные затраты на предприятии конкурента нулевые, а у второго единичные, т. е. второй клиент не может обслуживаться на предприятии конкурента. Пусть лидер установил цену p, тогда конкурент может взять $q_1 = p - \varepsilon$ для некоторого $\varepsilon > 0$ и перехватить первого клиента. Проблема в том, что оптимальный доход конкурента p достигается только в пределе. Аналогичный пример можно сделать и для лидера, когда лидер может обслуживать только первого клиента, а конкурент — обоих. Следовательно, принятые выше ограничения целочисленности цен необходимы для корректной постановки задачи.

Принятое соглашение, что клиенты подыгрывают последователю, является достаточным, чтобы переписать задачу клиентов в виде следующих неравенств:

$$\sum_{i \in I_L} (b_j - c_{ij} - p)x_{ij} + \sum_{i \in I_F} (b_j - c_{ij} - q_i)x_{ij} \geqslant 0, \quad j \in J, \tag{10}$$

$$\sum_{i \in I_L} (c_{ij} + p)x_{ij} + \sum_{i \in I_F} (c_{ij} + q_i)x_{ij} \leqslant c_{kj} + p, \quad j \in J,\, k \in I_L, \tag{11}$$

$$\sum_{i \in I_L} (c_{ij} + p)x_{ij} + \sum_{i \in I_F} (c_{ij} + q_i)x_{ij} \leqslant c_{kj} + q_i, \quad j \in J,\, k \in I_F, \tag{12}$$

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

$$x_{ij} \in \{0, 1\}, \quad i \in I_L \cup I_F,\, j \in J. \tag{14}$$

Ограничение (10) гарантирует, что покупка будет осуществляться только в том случае, если сумма транспортных затрат и затрат на приобретение продукции не будет превышать бюджета клиента. Заметим, что чем больше экономия одного клиента, тем больше суммарная экономия, т. е. больше целевая функция задачи клиентов. В таком случае ограничения (11), (12) гарантируют выполнение этого свойства. И, наконец, ограничения третьего уровня (13), (14) оставляем без изменения.

Задача (7)–(9) является сепарабельной по $j \in J$ и, следовательно, распадается на N независимых задач. Выберем j-го клиента. Тогда любое оптимальное решение j-й задачи удовлетворяет тем ограничениям системы (10)–(14), которые соответствуют данному j. Рассмотрим подсистему ограничений (10)–(14), соответствующих клиенту с номером j. Если среди решений нет ненулевых, то клиент j не может быть обслужен ни предприятиями лидера, ни предприятиями последователя. Если есть ненулевые решения, т. е. найдутся $i \in I_L \cup I_F$ такие, что $x_{ij} = 1$, то, как следует из ограничений (10)–(12), все эти решения являются оптимальными решениями оптимизационной задачи j-го клиента.

Получается, что множество решений системы (10)–(14) есть множество оптимальных решений задачи (7)–(9). С учётом того, что клиенты подыгрывают последователю, задача (4)–(6) эквивалентна задаче (4), (5), (10)–(14). Таким образом, трёхуровневая задача эквивалентна следующей двухуровневой задаче: (1)–(3), (4), (5), (10)–(14).

Введём новые обозначения и переменные:

$\overline{p} = \max\limits_{i \in I^L,\, j \in J} (b_j - c_{ij})$ — максимально возможная цена на предприятии лидера,

$\overline{q}_i = \max\limits_{j \in J} (b_j - c_{ij})$ — максимально возможная цена на i-м предприятии для последователя,

$w_{ij} = px_{ij}$ — доход, получаемый на i-м предприятии лидера от j-го покупателя,

$z_{ij} = q_i x_{ij}$ — доход, получаемый на i-м предприятии последователя от j-го покупателя.

Перепишем задачу (1)–(3), (4), (5), (10)–(14) в виде двухуровневой задачи линейного целочисленного программирования, используя введённые выше обозначения и переменные:

$$\sum_{j \in J} \sum_{i \in I_L} w_{ij} \to \max_{p,q,x,w,z} \tag{15}$$

при следующих ограничениях:

$$p \in \mathbb{Z}^+, \tag{16}$$

$$(q, x, w, z) \in \widetilde{\mathcal{F}}_F^*(p). \tag{17}$$

Целевая функция (15) определяет доход лидера, а $\widetilde{\mathcal{F}}_F^*(p)$ является множеством оптимальных решений задачи последователя

$$\sum_{j\in J}\sum_{i\in I_F} z_{ij} \to \max_{q,x,w,z} \tag{18}$$

при следующих ограничениях:

$$q_i \in \mathbb{Z}^+, \quad i \in I_F, \tag{19}$$

$$\sum_{i\in I_L}(b_j - c_{ij})x_{ij} - \sum_{i\in I_L} w_{ij} + \sum_{i\in I_F}(b_j - c_{ij})x_{ij} - \sum_{i\in I_F} z_{ij} \geqslant 0, \quad j \in J, \tag{20}$$

$$c_{kj} + p - \sum_{i\in I_L}(c_{ij}x_{ij} + w_{ij}) - \sum_{i\in I_F}(c_{ij}x_{ij} + z_{ij}) \geqslant 0, \quad j \in J,\, k \in I_L, \tag{21}$$

$$c^F_{kj} + q_i - \sum_{i\in I_L}(c_{ij}x_{ij} + w_{ij}) - \sum_{i\in I_F}(c_{ij}x_{ij} + z_{ij}) \geqslant 0, \quad j \in J,\, k \in I_F, \tag{22}$$

$$\sum_{i\in I_L\cup I_F} x_{ij} \leqslant 1, \tag{23}$$

$$(1 - x_{ij})\overline{p} - w_{ij} + p \geqslant 0, \quad i \in I_L,\, j \in J, \tag{24}$$

$$(1 - x_{ij})\overline{p} + w_{ij} - p \geqslant 0, \quad i \in I_L,\, j \in J, \tag{25}$$

$$0 \leqslant w_{ij} \leqslant \overline{p}x_{ij}, \quad i \in I_L,\, j \in J \tag{26}$$

$$(1 - x_{ij})\overline{q}_i - z_{ij} + q_i \geqslant 0, \quad i \in I_F,\, j \in J, \tag{27}$$

$$(1 - x_{ij})\overline{q}_i + z_{ij} - q_i \geqslant 0, \quad i \in I_F,\, j \in J, \tag{28}$$

$$0 \leqslant z_{ij} \leqslant \overline{q}_i x_{ij}, \quad i \in I_F,\, j \in J, \tag{29}$$

$$x_{ij} \in \{0, 1\}, \quad j \in J,\, i \in I_L \cup I_F. \tag{30}$$

Ограничения (20)–(23) обсуждались выше. Ограничения (24)–(26) дают гарантию, что доход лидера w_{ij} от обслуживания j-го потребителя на i-м предприятии равен p, если j-й клиент выбрал i-е предприятие лидера, и 0 в противном случае. Неравенства (27)–(29) аналогичным образом обеспечивают равенство переменных, отвечающих за доход последователя, т. е. $z_{ij} = q_i x_{ij}$ Именно ограничения (24)–(29) обеспечивают эквивалентность постановок до и после введения новых обозначений и переменных, они являются техническими ограничениями.

2. Вычислительная сложность

Этот раздел посвящён анализу сложностного статуса задачи, уточнению её взаимосвязи с задачами распознавания из полиномиальной иерархии. Глубокое и всестороннее рассмотрение различных аспектов теории вычислительной сложности содержится в книге [1], использующей следующие определения и понятия.

Рассмотрим оптимизационную задачу A. Возьмём целочисленный параметр k. Тогда вход задачи оценивания $\mathrm{OPT}(A)$ есть вход исследуемой задачи и k. При этом требуется определить, является или нет k оптимальным значением целевой функции задачи на множестве допустимых решений. Если же изменить формулировку на «существует ли допустимое решение со значением не меньше k?», то получим стандартную задачу распознавания $D(A)$.

Напомним определение класса NPO, который можно определить как класс оптимизационных задач, стандартные задачи распознавания которых лежат в классе NP [4]. Также напомним определения второго уровня полиномиальной и аппроксимационной иерархий, а именно, классы Δ_2^P, Σ_2^P и $\Sigma_2^P O$ [4]. Говорят, что задача распознавания *принадлежит классу* Δ_2^P, если существует детерминированная оракульная машина Тьюринга, которая распознает её за полиномиальное время, используя в качестве оракула некоторый язык из класса NP. Задачи из класса Σ_2^P распознаются за полиномиальное время недетерминированными оракульными машинами, использующими в качестве оракула язык из класса NP. По аналогии с классом NPO через стандартную задачу распознавания определяется и класс $\Sigma_2^P O$.

Понятия NP-полной, NP-трудной и NP-трудной в сильном смысле задач общеприняты [1, 4]. Имеют место следующие утверждения.

Теорема 1. *Задача ценообразования* (1)–(9) NP-*трудна в сильном смысле.*

Доказательство. В [2] показано, что задача фабричного ценообразования, в которой есть лишь один производитель, использующий фабричную стратегию ценообразования, NP-трудна в сильном смысле. Покажем, что эта задача полиномиально сводится к задаче (1)–(9). По исходным данным c_{ij}, b_j, $i \in I$, $j \in J$, задачи фабричного ценообразования построим исходные данные исследуемой задачи: в качестве клиентов выберем множество J с бюджетами b_j, положим $I_L = I_F = I$, $c_{ij} = \max\limits_j b_j + 1$, $i \in I_L$, $j \in J$. Для $i \in I_F$, $j \in J$ в качестве транспортных затрат выберем величины c_{ij}.

При таком определении величин c_{ij} клиенты не будут обслуживаться на предприятиях лидера. Поэтому любое допустимое решение (p, q, x)

задачи (1)–(9) будет генерировать оптимальное решение (q, x) задачи последователя, которое в силу предложенной сводимости будет оптимальным решением задачи фабричного ценообразования. Теорема 1 доказана.

Лемма 1. *Параметрическая задача оценивания для последователя принадлежит классу* Δ_2^P *для любого* p.

Доказательство. Возьмём в качестве оракула стандартную задачу распознавания, соответствующую параметрической задаче последователя. Очевидно, что эта задача при любом фиксированном p принадлежит классу NP. Используя сводимость теоремы 1, можно показать, что данная задача распознавания NP-полна. С учётом того, что максимальное значение целевой функции параметрической задачи последователя не превосходит экспоненты от длины записи исходных данных этой задачи, при использовании двоичного поиска число обращений к оракулу будет ограничено полиномом от длины записи исходных данных. Таким образом, за детерминированное полиномиальное время можно найти оптимальное значение для любой цены лидера. Это означает, что параметрическая задача оценивания для последователя принадлежит классу Δ_2^P. Лемма 1 доказана.

Лемма 2. *Для любого* p *параметрическая задача последователя принадлежит классу* NPO.

Доказательство. Из доказательства леммы 1 следует, что стандартная задача распознавания принадлежит классу NP и является в нём полной относительно полиномиальной сводимости. Следовательно, параметрическая задача последователя принадлежит классу NPO для любого p. Лемма 2 доказана.

Теорема 2. *Задача ценообразования* (1)–(9) *принадлежит классу* $\Sigma_2^P O$.

Доказательство. Покажем, что стандартная задача распознавания трёхуровневой задачи (1)–(9) принадлежит классу Σ_2^P. Чтобы отгадать требуемое допустимое решение (p, q, x) трёхуровневой задачи, необходимо недетерминированное время. Из доказательства предыдущих лемм следует, что за полиномиальное время можно определить, является ли вектор (q, x) оптимальным решением параметрической задачи последователя. Для этого достаточно использовать в качестве оракула стандартную задачу распознавания параметрической задачи последователя. За полиномиальное время находим оптимальное значение целевой функции параметрической задачи последователя. Сравниваем значение целевой функции последователя на векторе (q, x) с найденным оптимальным значением. Если совпали, то вычисляем значение целевой функции задачи (1)–(9) на векторе (p, q, x) и сравниваем с k. Таким образом, для решения стандартной задачи распознавания трёхуровневой задачи (1)–(9) требуется недетерминированная машина Тьюринга, которая за полиномиальное время, обращаясь к оракулу из класса NP, находит ответ. Теорема 2 доказана.

3. Алгоритмы решения

Чтобы решить исследуемую задачу, можно перебрать все цены в целочисленном отрезке $[0, \max\limits_{i\in I^L,\, j\in J}\{b_j - c_{ij}\}]$ и с каждой выбранной ценой решить задачу последователя (например, пакетом программ Gurobi), которую мы предварительно линеаризовали. Так как задача последователя NP-трудна, сокращение числа её решений может значительно уменьшить время счёта переборного алгоритма. Для этого воспользуемся идеей декомпозиции.

Для подробного изучения алгоритмов декомпозиции можно обратиться к работам [3, 6, 9, 12]. В этой схеме имеются два ключевых понятия: координирующая задача (master problem) и подзадача [6]. Координирующая задача эквивалентна исходной задаче. И при подходящем преобразовании координирующей задачи возникает релаксированная задача, которая даёт верхнюю оценку на исходную задачу. Подзадача — это источник нижних оценок и дополнительных ограничений для релаксированной координирующей задачи.

Соответствующая двухуровневой задаче (15)–(30) координирующая задача получается с помощью функции возмущения задачи последователя (18)–(30):

$$\rho^F(p) = \sum_{j\in J}\sum_{i\in I_F} z_{ij} \to \max_{q,x,w,z}$$

при ограничениях (19)–(30).

Задача (15)–(30) эквивалентна следующей оптимизационной задаче MP (master problem):

$$\sum_{j\in J}\sum_{i\in I_L} w_{ij} \to \max_{p,q,x,w,z}$$

при ограничениях (16),

$$\sum_{j\in J}\sum_{i\in I_F} z_{ij} = \rho^F(p), \tag{17'}$$

(19)–(30).

Если исключить ограничение (17′) из данной координирующей задачи, то получим релаксированную координирующую задачу RMP, оптимальное значение которой является верхней границей для задачи MP. Впервые данный подход с использованием функции возмущения был использован при разработке методов решения двухуровневых задач в [10].

Задача RMP эквивалентна задаче равномерного ценообразования, в которой требуется назначить цену на предприятиях единственного производителя, которая максимизирует его доход от обслуживания клиентов. Известно, что эта задача полиномиально разрешима [3].

При построении подзадачи используется понятие проекции оптимизационной задачи на пространство переменных [6]. Рассмотрим следующую задачу P:

$$\max_{x\in X,\, y\in Y} f(x,y),$$

$$\varphi_i(x,y) \leqslant 0, \quad i=\overline{1,m}.$$

Введём функцию возмущения данной задачи:

$$\rho(x) = \sup_{y\in Y} f(x,y),$$

$$\varphi_i(x,y) \leqslant 0, \quad i=\overline{1,m}.$$

Пусть $V = \big\{x \mid \exists y\colon \varphi_i(x,y) \leqslant 0,\ i=\overline{1,m}\big\}$ — это множество проекций области допустимых решений задачи P на пространство переменных x. Заметим, что задача P эквивалентна задаче Pr_x:

$$\max_{x\in X\cap V} \rho(x),$$

которая является проекцией задачи P на пространство переменных x. Имеет место следующее утверждение [6].

Теорема 3. *Задача P недопустима или неограничена сверху тогда и только тогда, когда то же самое выполняется для её проекции. Если (x^*,y^*) — оптимальное решение задачи P, то x^* — оптимальное решение проекции Pr_x. Если x^* — оптимальное решение проекции и инфимум в $\rho(x^*)$ достигается на y^*, то (x^*,y^*) — оптимальное решение исходной задачи P.*

При проектировании задачи (15)–(30) на единственную переменную лидера p используется следующая функция возмущения:

$$\rho(p) = \max_{q,x,w,z} \sum_{j\in J}\sum_{i\in I_L} w_{ij} \tag{15′}$$

при ограничениях (17)–(30), которая и используется в качестве подзадачи далее. В декомпозиционном алгоритме на 1-м шаге решается задача

RMP. Далее зафиксируем переменную p, взяв её значение p^* из оптимального решения релаксированной задачи и решим подзадачу. Однако подзадача (15′), (17)–(30) является двухуровневой задачей, и для того чтобы найти её оптимальное значение, надо решить задачу последователя при заданном p^*, а затем заменить ограничение (17) равенством

$$\sum_{j\in J}\sum_{i\in I_F} z_{ij} = \rho^F(p^*). \qquad (17')$$

Наконец, решив задачу (15′), (17′), (19)–(30), получим оптимальное значение и оптимальное решение подзадачи.

Подобная схема решения подзадачи требует дважды обращаться к пакету Gurobi. Для уменьшения трудоёмкости решения данной подзадачи предлагается её свед́ение к решению следующей оптимизационной задачи с $p = p^*$:

$$(\overline{p}+1)N\sum_{j\in J}\sum_{i\in I_F} z_{ij} + \sum_{j\in J}\sum_{i\in I_L} w_{ij} \to \max_{q,x,w,z} \qquad (15'')$$

при ограничениях (19)–(30).

Корректность свед́ения следует из следующих замечаний. Коэффициент $(\overline{p}+1)N$ выбран так, что он строго больше любого значения второго члена целевой функции (15″). Тогда в силу этого выбора и целочисленности переменных оптимальные решения задачи (15″), (19)–(30) содержатся среди допустимых решений (q,x,w,z), на которых достигается максимальная прибыль последователя $\sum\limits_{j\in J}\sum\limits_{i\in I_F} z_{ij}$, т. е. среди оптимальных решений последователя. Если среди этих оптимальных решений выбрать те, на которых достигает максимума второй член целевой функции (15″), то получим множество оптимальных решений задачи (15″), (19)–(30). Таким образом, оптимальное решение этой задачи позволяет получить оптимальное решение подзадачи.

Допустим, что уже просмотрен набор цен $\overline{P}$ и f^* — текущее рекордное значение целевой функции. Тогда если оптимум задачи RMP с ограничением $p \notin \overline{P}$ не превосходит f^*, то $\overline{P}$ содержит оптимальную цену. Отсюда получаем схему алгоритма решения.

Шаг 0. $LB = 0$, $UB = \infty$, $\overline{P} = \varnothing$, $p^* = 0$, $f^* = 0$.

Шаг 1. Решить задачу RMP с ограничением $p \notin \overline{P}$. Пусть $\overline{p}$ — оптимальная цена, а $\overline{f}$ — оптимум. Если $f^* \leqslant \overline{f}$, то стоп.

Шаг 2. Решить задачу последователя с $p = \overline{p}$ (с помощью пакета Gurobi). Пусть $\widetilde{f}$ — оптимум. Если $\widetilde{f} > f^*$, то положить $f^* = \widetilde{f}$, $p^* = \overline{p}$. $\overline{P} := \overline{P} \cup \overline{p}$. Перейти на шаг 1.

Теорема 4. *Шаг* 1 *реализуется за полиномиальное время.*

Доказательство. Задачу RMP можно решить за полиномиальное время, если отсортировать доступный бюджет каждого клиента для магазинов лидера, т. е. $\max\limits_{i \in I^L}(b_j - c_{ij})$. Эти величины будут концами ценовых отрезков. Для каждого такого отрезка хранится информация о том, какое количество клиентов готово купить по данным ценам, определяемое после сортировки тривиальным образом. Причём число обслуживаемых клиентов для каждого отрезка своё. Значит, выбирая цену из определённого отрезка, можно легко понять, какую прибыль получим. Решением подзадачи RMP является цена, при которой прибыль максимальна. Так как внутри каждого отрезка функция прибыли возрастает, просматривать нужно только правый край отрезка, а это $O(N)$ элементарных операций. После просмотра определённой цены её можно удалить из отрезка, тем самым укоротив отрезок справа. При этом задача RMP с новым набором отрезков по-прежнему будет решаться за $O(N)$ операций, так как число отрезков не увеличивается. Теорема 4 доказана.

Помимо точного алгоритма декомпозиционного типа рассмотрим гибридный приближённый алгоритм, использующий идеи VND-метаэвристик и покоординатного спуска. VND-метаэвристика (спуск с чередующимися окрестностями) [7] успешно применяется для решения сложных оптимизационных задач. Её основная идея — последовательный поиск локальных оптимумов с использованием разных окрестностных структур, чтобы выбираться из локальных экстремумов и продолжать поиск оптимального или близкого к оптимальному решения. Покоординатный спуск используется, как правило, для решения непрерывных задач безусловной оптимизации. Но эту же идею можно использовать для поиска оптимальной цены в случае равномерного ценообразования, двигаясь по допустимой области влево и вправо, меняя при этом длину шага.

Как и прежде, будем использовать пакет Gurobi для решения задачи последователя. Опишем схему гибридного алгоритма.

Шаг 0. Возьмём длины $D_1 < \dots < D_k$, которые будут определять окрестностные структуры. Пусть p_0 — стартовая цена, $\overline{p} = p_0$ — текущая цена.

Шаг 1. Двигаемся до упора влево (или вправо) с шагом D_1 в сторону возрастания значения целевой функции. Переобозначим цену через $\overline{p}$.

Шаг 2. Просмотрим все цены в целочисленном интервале $(\overline{p} - D_1, \overline{p} + D_1)$. Если есть цены лучше $\overline{p}$, то переопределяем $\overline{p}$ и на шаг 1.

Шаг 3. Изменяем длину шага (от D_2 до D_k) и проверяем решения слева и справа. Как только улучшаем цену $\overline{p}$, возвращаемся на шаг 1.

Критерием остановки такого алгоритма может быть ограничение по времени или по числу итераций (заходов на шаг 3). Основной вопрос: насколько успешно использование покоординатного спуска для определения окрестностных структур в спуске с чередующимися окрестностями?

4. Вычислительный эксперимент

Эксперимент состоит из двух частей. Сперва покажем зависимость оптимального значения (дохода лидера) от числа предприятий. Для этого рассмотрим примеры задачи следующей размерности: 40 потребителей и до 16 предприятий у лидера (Ml) и у последователя (Mf), в которых значения величин $b_j - c_{ij}^L$ и $b_j - c_{ij}^F$ (рассматривать отдельно бюджет и транспортные расходы нет смысла) выбирались случайным образом из интервала $[0, 100]$. Число предприятий лидера и последователя: 4, 8, 12 и 16. При этом примеры с разным числом предприятий связаны между собой, а именно, все они получены из примеров большей размерности сокращением входных параметров. Для поиска оптимального значения использовался разработанный точный алгоритм декомпозиционного типа. Тесты проводились на десяти группах примеров каждой размерности. В табл. 1 представлены результаты эксперимента с усреднёнными значениями. Здесь Доход Л. — доход лидера, Доход П. — доход последователя, Время, с — время работы алгоритма в секундах, Итер. — число итераций алгоритма (число обращений к пакету Gurobi для решения задачи последователя).

Таблица 1

Ml	Mf	Доход Л.	Доход П.	Время, с	Итер.
16	16	176	732,4	1577,5735	97
16	12	250,8	1084,4	788,3578	95,1
16	8	399,7	1304	245,4093	91,2
16	4	747,2	1079,6	55,0507	81,9
12	16	181,9	836,3	1870,7602	96,7
12	12	236,4	883,8	991,6571	95,3
12	8	390,8	1219	463,0386	90,8
12	4	727,7	1141,6	88,4694	81,8
8	16	210,9	1004	2669,1576	95,8
8	12	263,6	840,4	1373,4486	94,2
8	8	390,9	1428,3	537,6217	90,8
8	4	652,2	1091,4	90,4895	83,2
4	16	271,5	1064,7	8179,961	93,1
4	12	308,8	1003,1	3379,692	92,1
4	8	406,4	1427,7	1006,2417	89
4	4	618,8	1178,5	116,671	82,3

Как и ожидалось, значение целевой функции значительно уменьшается с ростом числа предприятий последователя. В первую очередь это связано с тем, что используются разные стратегии ценообразования. Удивительным оказался результат, что иногда с ростом числа предприятий лидера его доход и доход конкурента падают. Когда сокращается число предприятий, лидеру становится труднее забирать клиентов себе. Поэтому последователь в некоторых случаях начинает повышать цены с целью получения сверхприбыли с небольшого числа клиентов, отдавая лидеру иногда тех клиентов, которых он обслуживал при большем числе предприятий лидера.

Также стоит отметить разницу в доходах лидера и последователя. Конечно, у второго игрока есть некоторое преимущество. Поэтому крайне маловероятно, что спрос может разделиться в пользу лидера. Но при большом числе предприятий разница в доходах очень заметна. Здесь сказывается выбор стратегий ценообразования. Если бы последователь применял дискриминационную стратегию, то, скорее всего, разница была бы ещё заметнее. В худшем случае для решения задачи требуется перебрать все цены p от 1 до 100. Как видно из табл. 1, использование идеи декомпозиции позволяет в некоторых случаях сократить число перебираемых цен до 81.

Таблица 2

Размер		Относ. уклонение		Задача послед.		Время, с	
n	m	среднее	макс.	среднее	макс.	среднее	макс.
16	16	7,3246	25	21,9	32	1025,98	2761,4
16	12	6,2459	17,347	26,2	35	501,4051	1242,13
16	8	1,015	6,521	26,1	36	233,6785	565,396
16	4	3,7205	13,479	24,1	43	68,8452	226,362
12	16	4,49	23,529	23,2	39	1392,9597	2254,3
12	12	3,0888	12,605	24,4	37	698,5263	1891,38
12	8	3,0743	11,111	26,8	42	350,7978	951,233
12	4	3,4921	9,09	25,1	39	64,9215	199,693
8	16	11,3837	38,636	23,3	35	1813,432	3265,73
8	12	6,6696	24,545	21,4	35	839,3648	2000,03
8	8	1,9573	6,897	26,3	32	409,7409	1113,93
8	4	1,9171	9,434	25,5	40	71,1642	166,632
4	16	8,8496	31,578	27,2	37	6817,916	17383,2
4	12	3,8595	17,105	22,6	32	2059,3256	3012,5
4	8	5,294	16,129	24,8	37	777,6665	1441,01
4	4	4,1768	10,638	22,7	33	88,4661	229,285

В табл. 2 представлены результаты работы приближённого алгоритма на этом же наборе примеров. Здесь во втором столбце таблицы колонки среднее и макс. соответствуют среднему и максимальному относительному уклонению (в процентах) значения на решении, получаемом приближённым алгоритмом, от оптимального значения. В третьем столбце таблицы колонки среднее и макс. соответствуют среднему и максимальному числу решаемых задач последователя (шаги 1–3). И, наконец, в последнем столбце таблицы колонки среднее и макс. соответствуют среднему и максимальному времени работы алгоритма в секундах.

Если сравнить работу точного и приближённого алгоритмов, то спуск с чередующимися окрестностями в совокупности с покоординатным спуском позволяет незначительно сократить время работы алгоритма при значительной потере в качестве получаемых решений. Это говорит о том, что в совокупности с окрестностями, порождаемыми покоординатным спуском, требуется рассмотрение и других окрестностных структур.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М: Мир, 1982. 416 с.
2. **Плясунов А. В., Панин А. А.** Задача ценообразования. Ч. I. Точные и приближённые алгоритмы решения // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 5. С. 83–100.
3. **Плясунов А. В., Панин А. А.** Задача ценообразования. Часть 2: Вычислительная сложность // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 6. С. 56–71.
4. **Ausiello G., Crescenzi P., Gambosi G., Kann V., Marchetti-Spaccamela A., Protasi M.** Complexity and approximation: Combinatorial optimization problems and their approximability properties. Heidelberg: Springer-Verl., 1999. 524 p.
5. **Florensa C., Garcia-Herreros P., Misra P., Arslan E., Mehta S., Grossmann I. E.** Capacity planning with competitive decision-makers: Trilevel MILP formulation, degeneracy, and solution approaches // Eur. J. Oper. Res. 2017. Vol. 262, No. 2. P. 449–463.
6. **Geoffrion A. M.** Generalized Benders decomposition // J. Optim. Theory Appl. 1972. Vol. 10, No. 4. P. 237–260.
7. **Hansen P., Mladenovic N.** Variable neighborhood search // Eur. J. Oper. Res. 2001. Vol. 130, No. 3. P. 449–467.
8. **Leggette E. W., Jr., Moore D. J.** Optimization problems and the polynomial hierarchy // Theor. Comput. Sci. 1981. Vol. 15, No. 3. P. 279–289.
9. **McDaniel D., Devine M.** A modified Benders partitioning algorithm for mixed integer programming // Manage. Sci. 1977. Vol. 24, No 3. P. 312–319.
10. **Outrata J. V.** On the numerical solution of a class of Stackelberg problems // ZOR. 1990. Vol. 34, No. 4. P. 255–277.

11. Plyasunov A. V., Panin A. A. On three-level problem of competitive pricing // Proc. 2nd Int. Conf. Numerical Computations: Theory and Algorithms NUMTA–2016 (Pizzo Calabro, Italy, June 19–25, 2016). Melville, NY: AIP Publ., 2016. P. 050007-1–050007-5. (AIP Conf. Proc.; Vol. 1776).

12. Vanderbeck F., Savelsbergh M. W. P. A generic view of Dantzig–Wolfe decomposition for integer programming // Oper. Res. Lett. 2006. Vol. 34, No. 3. P. 296–306.

Губарева Анна Вячеславовна,
Панин Артём Александрович,
Плясунов Александр Владимирович,
Сом Людмила Васильевна

Статья поступила
23 июля 2018 г.

После доработки —
26 ноября 2018 г.

Принята к публикации
28 ноября 2018 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
January–March 2019. Volume 26, No. 1. P. 55–73

UDC 519.8 DOI 10.33048/daio.2019.26.625

ON A THREE-LEVEL COMPETITIVE PRICING PROBLEM WITH UNIFORM AND MILL PRICING STRATEGIES

A. V. Gubareva[1,a], *A. A. Panin*[1,2,b],
A. V. Plyasunov[1,2,c], and *L. V. Som*[1,d]

[1]Novosibirsk State University,
2 Pirogov St., 630090 Novosibirsk, Russia

[2]Sobolev Institute of Mathematics,
4 Acad. Koptyug Ave., 630090 Novosibirsk, Russia

E-mail: [a]a.gubareva@g.nsu.ru, [b]aapanin1988@gmail.com,
[c]apljas@math.nsc.ru, [d]milisom@mail.ru

Abstract. Under study is a three-level pricing problem formulated as a Stackelberg game in which the two companies, the Leader and the Follower, compete with each other for customers demand by setting prices for homogeneous products on their facilities. The first decision is made by the Leader. Then, having full information about the Leader's choice, the Follower makes his own decision. After that each customer chooses the facility with minimal service costs to be serviced from. The Leader and the Follower use different pricing strategies: uniform and mill pricing respectively. We study the behavior of company revenues depending on the number of facilities. For this, an exact decomposition type algorithm is proposed. Moreover, we developed a hybrid approximation algorithm that is based on the variable neighborhood descent and coordinate descent. Tab. 2, bibliogr. 12.

Keywords: Stackelberg game, competitive pricing problem, three-level problem, uniform and mill pricing, exact and approximate algorithm, variable neighborhood descent, coordinate descent, decomposition.

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 [Russian].

2. **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 [Russian]. Translated in *J. Appl. Ind. Math.,* **7**, No. 2, 241–251, 2013.
3. **A. V. Plyasunov** and **A. A. Panin,** The pricing problem. Part II: Computational complexity, *Diskretn. Anal. Issled. Oper.,* **19**, No. 6, 56–71, 2012 [Russian]. Translated in *J. Appl. Ind. Math.,* **7**, No. 3, 420–430, 2013.
4. **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, Heidelberg, 1999.
5. **C. Florensa, P. Garcia-Herreros, P. Misra, E. Arslan, S. Mehta,** and **I. E. Grossmann,** Capacity planning with competitive decision-makers: Trilevel MILP formulation, degeneracy, and solution approaches, *Eur. J. Oper. Res.,* **262**, No. 2, 449–463, 2017.
6. **A. M. Geoffrion,** Generalized Benders decomposition, *J. Optim. Theory Appl.,* **10**, No. 4, 237–260, 1972.
7. **P. Hansen** and **N. Mladenović,** Variable neighborhood search, *Eur. J. Oper. Res.,* **130**, No. 3, 449–467, 2001.
8. **E. W. Leggette, Jr.** and **D. J. Moore,** Optimization problems and the polynomial hierarchy, *Theor. Comput. Sci.,* **15**, No. 3, 279–289, 1981.
9. **D. McDaniel** and **M. Devine,** A modified Benders partitioning algorithm for mixed integer programming, *Manage. Sci.,* **24**, No. 3, 312–319, 1977.
10. **J. V. Outrata,** On the numerical solution of a class of Stackelberg problems, *ZOR,* **34**, No. 4, 255–277, 1990.
11. **A. V. Plyasunov** and **A. A. Panin,** On three-level problem of competitive pricing, in *Numerical Computations: Theory and Algorithms* (Proc. 2nd Int. Conf., Pizzo Calabro, Italy, June 19–25, 2016), pp. 050006-1–050006-5, AIP Publ., Melville, NY, 2016 (AIP Conf. Proc., Vol. 1776).
12. **F. Vanderbeck** and **M. W. P. Savelsbergh,** A generic view of Dantzig–Wolfe decomposition for integer programming, *Oper. Res. Lett.,* **34**, No. 3, 296–306, 2006.

Anna V. Gubareva,
Artem A. Panin,
Aleksandr V. Plyasunov,
Lyudmila V. Som

Received July 23, 2018
Revised November 26, 2018
Accepted November 28, 2018