

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 2 2026

Новосибирск
Издательство Института математики

ВЕРХНИЕ И НИЖНИЕ ГРАНИЦЫ И АЛГОРИТМЫ СПУСКА
С ЧЕРЕДУЮЩИМИСЯ ОКРЕСТНОСТЯМИ ДЛЯ ЗАДАЧИ
КОНКУРЕНТНОГО ЦЕНООБРАЗОВАНИЯ

Р. Р. Драчев^a, А. А. Панин^b, А. В. Плясунов^c

Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

E-mail: ^arrdrachev@gmail.com,
^baapanin1988@gmail.com, ^capljas@math.nsc.ru

Аннотация. Рассматривается трёхуровневая задача конкурентного ценообразования. Две компании, производящие однородный продукт, конкурируют за потребительский спрос. На первом уровне компания-лидер, а на втором уровне компания-последователь устанавливают цены на каждом из своих предприятий с целью максимизировать собственный доход. На третьем уровне каждый потребитель выбирает предприятие с наименьшими общими расходами на покупку и транспортировку товара и совершает покупку в том случае, если расходы не превышают его бюджета. Цель игры — найти такие цены лидера, при которых его доход максимален. Известно, что проблема Σ_2^P -трудна. Предложены нижние и верхние границы оптимума, и разработаны приближённые алгоритмы решения, основанные на спуске с чередующимися окрестностями. Проведён вычислительный эксперимент с использованием примеров различной размерности и сложности. Табл. 7, ил. 2, библиогр. 20.

Ключевые слова: трёхуровневое программирование, игра Штакельберга, конкурентное ценообразование, локальный поиск, спуск с чередующимися окрестностями.

Введение

В работе исследуется обобщение двухуровневой задачи ценообразования, впервые рассмотренной в [1, 2]. Эта задача характеризуется единственным объёмом спроса, отдельными ценами на каждом пункте обслуживания одного производителя и ненулевыми затратами на транспортировку продукции из пункта обслуживания до потребителя, которые полностью

оплачиваются самим потребителем. В [3] рассмотрена близкая постановка, в которой транспортные расходы нулевые, а спрос не равен единице. Другие вариации задачи также можно найти в литературе [4–12].

Для моделирования рыночных процессов требуется учитывать интересы других производителей (конкурентов), что достаточно хорошо описывается игрой Штакельберга или многоуровневой постановкой, когда задача решается от лица производителя-лидера, принимающего решение первым, а вторым шагом на его решение реагируют производители-последователи. Основная проблема таких постановок — сложность или даже невозможность построения одноуровневых моделей с полиномиальным числом ограничений, что приводит к невозможности применения классических методов решения оптимизационных задач и требует разработки новых алгоритмических подходов для построения как точного, так и приближённого решений.

Содержательно рассматриваемую в данной работе задачу можно описать следующей игрой Штакельберга. Две компании, производящие однородный продукт, конкурируют за потребительский спрос. На первом уровне лидер устанавливает цены на своих предприятиях, следуя некоторой стратегии ценообразования. На втором уровне последователь устанавливает цены на каждом из своих предприятий с учётом действий лидера на первом ходу и с целью максимизировать свой доход. На третьем уровне каждый клиент выбирает предприятие с наименьшими общими расходами на покупку и транспортировку и совершает покупку только в том случае, если эти расходы не превышают его бюджета.

Различают несколько стратегий ценообразования [13]: равномерная стратегия, когда на всех предприятиях устанавливается единая цена; фабричная стратегия, когда на каждом предприятии устанавливается своя цена; дискриминационная стратегия, когда на каждом предприятии устанавливается своя цена для каждого клиента.

Для трёхуровневых задач необходимо уточнить характер взаимодействия (кооперации) уровней [14]. Если некоторый уровень имеет несколько оптимальных решений, то уровни выше должны точно знать, какое именно решение выберет игрок, иначе задачу можно считать некорректной. Так, последователь может либо кооперироваться с лидером, т. е. из нескольких своих оптимальных решений выбирать то, которое максимизирует доход лидера, либо не кооперироваться, т. е. из нескольких своих оптимальных решений выбирать то, которое минимизирует доход лидера. В свою очередь, клиенты могут кооперироваться либо с лидером, либо с последователем, т. е. из нескольких своих оптимальных решений выбирать то, которое максимизирует доход лидера или последователя соответственно.

В работе рассматривается задача с фабричным ценообразованием как у лидера, так и у последователя и с кооперацией всех нижних уровней с лидером. Известно, что задача принадлежит классу $\Sigma_2^P O$ и Σ_2^P -трудна [15]. Для её решения будем использовать подход, впервые предложенный в [16] для двухуровневой задачи размещения производства и ценообразования, в котором для оценки эффективности фиксированного размещения подзадача ценообразования решается приближённо. В нашем случае для оценки качества выбранных цен лидера используем приближённый эвристический алгоритм решения задачи последователя.

В разд. 1 приводится математическая модель задачи и обсуждаются её свойства. В разд. 2 представлены верхние и нижние границы оптимумов, используемые для оценки эффективности алгоритмов. В разд. 3 представлены алгоритмы решения исследуемой задачи, а в разд. 4 приведены результаты численных экспериментов.

1. Математическая модель

Введём следующие обозначения:

- $I_L = \{1, \dots, n_L\}$ — множество предприятий лидера;
- $I_F = \{n_L + 1, \dots, n_L + n_F\}$ — множество предприятий последователя;
- $I = I_L \cup I_F$;
- $J = \{1, \dots, m\}$ — множество клиентов;
- $b_j \in \mathbb{Z}_{\geq}$ — бюджет j -го клиента;
- $c_{ij} \in \mathbb{Z}_{\geq}$ — транспортные затраты j -го клиента в i -м предприятии.

Введём следующие переменные:

- $x_{ij} = \begin{cases} 1, & \text{если } j\text{-й клиент обслуживается на } i\text{-м предприятии,} \\ 0 & \text{иначе;} \end{cases}$
- $p_i \in \mathbb{Z}_{\geq}$ — цена продукта лидера на i -м предприятии;
- $q_i \in \mathbb{Z}_{\geq}$ — цена продукта последователя на i -м предприятии.

Используя введённые обозначения, запишем трёхуровневую задачу фабричного ценообразования в виде трёхуровневой модели квадратичного программирования с целочисленными и булевыми переменными:

$$\sum_{i \in I_L} \sum_{j \in J} p_i x_{ij} \rightarrow \max_{p, q, x}, \quad (1)$$

$$p_i \in \mathbb{Z}_{\geq}, \quad (2)$$

$$(q, x) \in \mathcal{F}_{F_l}^*(p), \quad (3)$$

где $\mathcal{F}_{F_l}^*(p)$ — множество оптимальных решений задачи последователя при кооперации с лидером:

$$\sum_{i \in I_F} \sum_{j \in J} q_i x_{ij} \rightarrow \max_{q, x}, \quad (4)$$

$$q_i \in \mathbb{Z}_{\geq}, \quad (5)$$

$$(x) \in \mathcal{F}_{C_i}^*(p, q), \quad (6)$$

где $\mathcal{F}_{C_i}^*(p, q)$ — множество оптимальных решений задачи клиентов при кооперации с лидером:

$$\sum_{j \in J} \left(\sum_{i \in I_L} (b_j - c_{ij} - p_i) x_{ij} + \sum_{i \in I_F} (b_j - c_{ij} - q_i) x_{ij} \right) \rightarrow \max_x, \quad (7)$$

$$\sum_{i \in I} x_{ij} \leq 1, \quad j \in J, \quad (8)$$

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

Целевая функция верхнего уровня (1) определяет доход лидера, целевая функция второго уровня (4) — доход последователя. Целевая функция третьего уровня (7) отражает стремление каждого клиента минимизировать свои расходы или, другими словами, максимизировать остаток бюджета. Ограничения (2) и (5) устанавливают ненулевые цены на продукцию лидера и последователя, а (8) гарантирует, что каждый клиент обслуживается не более чем на одном предприятии. Ограничения (3) и (6) делают модель трёхуровневой.

В работе [15] для рассматриваемой задачи, а также её модификаций, полученных заменой стратегии ценообразования лидера или последователя, введены обозначения, которые будем использовать ниже:

- ММР_L — исходная задача с фабричным ценообразованием у лидера и последователя;
- DMP_L — в задаче ММР_L стратегия ценообразования лидера заменяется на дискриминационную;
- UMP_L — в задаче ММР_L стратегия ценообразования лидера заменяется на равномерную;
- MDP_L — в задаче ММР_L стратегия ценообразования последователя заменяется на дискриминационную;
- MUP_L — в задаче ММР_L стратегия ценообразования последователя заменяется на равномерную.

1.1. Одноуровневая модель для задачи последователя. Аналогично обычной двухуровневой задаче фабричного ценообразования [1, 2] задача последователя может быть переписана в виде задачи одноуровневого целочисленного линейного программирования. Для этого введём верхние границы на оптимальные цены:

$$\bar{p}_i = \max_{j \in J} \{b_j - c_{ij}\}, \quad i \in I_L, \quad \bar{q}_i = \max_{j \in J} \{b_j - c_{ij}\}, \quad i \in I_F.$$

Целевые функции и ограничения (4)–(9) перепишем в следующем виде:

$$\sum_{i \in I_F} \sum_{j \in J} z_{ij} \rightarrow \max_{q, x, z}, \quad (10)$$

$$\sum_{i \in I} (b_j - c_{ij})x_{ij} - \sum_{i \in I} z_{ij} \geq 0, \quad j \in J, \quad (11)$$

$$c_{kj} + p_k - \sum_{i \in I} c_{ij}x_{ij} - \sum_{i \in I} z_{ij} \geq 0, \quad k \in I_L, j \in J, \quad (12)$$

$$c_{kj} + q_k - \sum_{i \in I} c_{ij}x_{ij} - \sum_{i \in I} z_{ij} \geq 0, \quad k \in I_F, j \in J, \quad (13)$$

$$(1 - x_{ij})\bar{p}_i - z_{ij} + p_i \geq 0, \quad i \in I_L, j \in J, \quad (14)$$

$$(1 - x_{ij})\bar{p}_i + z_{ij} - p_i \geq 0, \quad i \in I_L, j \in J, \quad (15)$$

$$(1 - x_{ij})\bar{q}_i - z_{ij} + q_i \geq 0, \quad i \in I_F, j \in J, \quad (16)$$

$$(1 - x_{ij})\bar{q}_i + z_{ij} - q_i \geq 0, \quad i \in I_F, j \in J, \quad (17)$$

$$z_{ij} \leq \bar{p}_i x_{ij}, \quad i \in I_L, j \in J, \quad (18)$$

$$z_{ij} \leq \bar{q}_i x_{ij}, \quad i \in I_F, j \in J, \quad (19)$$

$$\sum_{i \in I} x_{ij} \leq 1, \quad j \in J, \quad (20)$$

$$q_i \in Z_{\geq}, i \in I_F, \quad x_{ij} \in \{0, 1\}, z_{ij} \in Z_{\geq}, i \in I, j \in J. \quad (21)$$

Здесь переменные z_{ij} обозначают доход, полученный на i -м предприятии от j -го клиента. Ограничения (12)–(14) гарантируют, что клиенты выберут оптимальное решение, а условия (15)–(19) гарантируют, что доход на i -м предприятии от j -го клиента равен цене на i -м предприятии, если клиент обслуживается на этом предприятии.

Заметим, что при $I = I_F$ получаем двухуровневую задачу фабричного ценообразования (MP) [2].

1.2. Сравнение кооперативной и некооперативной постановок. Для поиска кооперативного решения приходится решать NP-трудную задачу последователя дважды. Сначала находим оптимум opt_F^* задачи (10)–(21). Затем решаем новую модель, содержащую ограничения (11)–(21), целевую функцию, определяющую доход лидера:

$$\sum_{i \in I_L} \sum_{j \in J} z_{ij} \rightarrow \max_{q, x, z},$$

и новое ограничение, гарантирующее, что последователь выберет своё оптимальное решение:

$$\sum_{i \in I_F} \sum_{j \in J} z_{ij} = \text{opt}_F^*.$$

Ключевая особенность многоуровневых алгоритмов локальных улучшений, описанных ниже, заключается в упрощении процедуры вычисления дохода лидера. Например, при определении лучшего в некоторой окрестности вектора цен лидера можно отказаться от принципа кооперативного взаимодействия лидера и последователя. Следующее утверждение говорит о том, что пренебрегать кооперацией неразумно.

Теорема 1. *Зафиксируем оптимальное решение задачи ММР_L . При смене решения последователя с кооперативного на произвольное оптимальное решение (цены лидера остаются прежними) доход лидера может измениться во сколько угодно раз.*

Утверждение теоремы останется справедливым даже при выполнении одного из условий:

- (1°) $|J| \leq 2$;
- (2°) $b_j \leq 2 \quad \forall j \in J$.

ДОКАЗАТЕЛЬСТВО. Рассмотрим условие (1°). Пусть у лидера и последователя есть по одному предприятию с номерами 1 и 2 соответственно. Предположим, что есть два клиента с номерами 1 и 2 соответственно, причём клиент 2 может обслуживаться только последователем:

$$\begin{aligned} c_{11} &= 0, & c_{21} &= 0, \\ c_{12} &= 2k, & c_{22} &= 0, \\ b_1 &= k, & b_2 &= 2k. \end{aligned}$$

Доход лидера при любом решении последователя не превосходит k . Максимальный доход может быть достигнут, если обслужить клиента 1, установив цену $p_1 = k$, а последователь назначит цену $q_2 = 2k$ и обслужит клиента 2, что соответствует кооперативному оптимальному решению. Однако последователь может выбрать другое оптимальное решение $q_2 = k$, при котором он обслуживает всех клиентов.

Рассмотрим условие (2°). Пусть у лидера и последователя по-прежнему есть по одному предприятию с номерами 1 и 2 соответственно. Предположим, что есть $2d$ клиентов, половина из которых могут обслуживаться только последователем:

$$\begin{aligned} c_{1j} &= 0, & c_{2j} &= 0, & j &\in \{1, \dots, k\}, \\ c_{1j} &= 2, & c_{2j} &= 0, & j &\in \{k+1, \dots, 2k\}, \\ b_j &= 1, & j &\in \{1, \dots, k\}, \\ b_j &= 2, & j &\in \{k+1, \dots, 2k\}. \end{aligned}$$

Доход лидера при любом решении последователя не превосходит d , а достигает этой верхней границы при обслуживании d первых клиентов

с ценой продукции $p_1 = 1$ и при кооперативной оптимальной цене последователя $q_2 = 2$. При этом последователь обслуживает d последних клиентов, а при другой оптимальной цене $q_2 = 1$ последователь обслуживает всех клиентов. Теорема 1 доказана.

2. Верхние и нижние границы

Простейшей верхней оценкой, которая может быть построена для любой многоуровневой задачи, является оценка, полученная релаксацией экстремального ограничения (high-point relaxation [17]). Соответствующая релаксированная задача получается из исходной постановки удалением целевой функции последователя и объединением оставшихся ограничений лидера и последователя. В рассматриваемой в данной работе задаче в результате релаксации экстремального ограничения получается задача фабричного ценообразования [1, 2]. На практике релаксация экстремального ограничения оказывается довольно грубой оценкой. Оценку, полученную релаксацией экстремального ограничения, обозначим через UB_{hp} .

В работе [2] описаны верхняя и нижняя границы для дохода лидера в задаче фабричного ценообразования, полученные путём смены стратегии ценообразования. Возникает вопрос: можно ли использовать метод смены стратегии ценообразования с целью получения оценок для задачи MMR_L ? Следующие утверждения говорят о том, что в некоторых случаях это возможно.

Теорема 2. *Оптимум задачи DMP_L является верхней границей оптимума задачи MMR_L .*

ДОКАЗАТЕЛЬСТВО. Заметим, что фабричное ценообразование вкладывается в дискриминационное. Следовательно, множество допустимых решений задачи MMR_L представляет собой подмножество множества допустимых решений задачи DMP_L . Теорема 2 доказана.

Задача DMP_L принадлежит классу NPO [15]. Стоит отметить, что для задачи DMP_L в рамках данной работы не было найдено эффективных способов решения. Это приводит к необходимости полного перебора цен лидера, коих существенно больше, чем в задаче MMR_L . В результате практическая ценность теоремы 2 в настоящий момент остаётся низкой.

Теорема 3. *Оптимум задачи UMP_L является нижней границей оптимума задачи MMR_L .*

ДОКАЗАТЕЛЬСТВО. Достаточно заметить, что равномерное ценообразование вкладывается в фабричное, поэтому множество допустимых решений задачи UMP_L представляет собой подмножество множества допустимых решений задачи MMR_L . Теорема 3 доказана.

Задача UMP_L принадлежит классу NPO [15]. Оценку из теоремы 3 обозначим через LB_{um} . Для задачи UMP_L в рамках данной работы использован полный перебор решений. Однако их число существенно меньше, чем в задаче $MMPL$. Как показывают проведённые эксперименты, теорема 3 вполне применима на практике.

Теорема 4. *Оптимум задачи MDP_L является нижней границей для оптимума задачи $MMPL$.*

ДОКАЗАТЕЛЬСТВО. Рассмотрим задачу MDP_L . В ней последователь устанавливает на каждом предприятии для каждого клиента свою цену. Таким образом, последователь для любого клиента всегда может установить такую цену, чтобы затраты на обслуживание у последователя были на единицу меньше, чем затраты на обслуживание у лидера, исключая случаи, когда для этого нужно установить нулевую или отрицательную цену. Таким образом, клиент j обслуживается на предприятии лидера i тогда и только тогда, когда $c_{ij} + p_i \leq \min_{k \in I_F} \{c_{kj} + 1, b_j\}$.

Если в задаче $MMPL$ имеем $c_{ij} + p_i \leq \min_{k \in I_F} \{c_{kj} + 1, b_j\}$, то клиент j обслуживается на предприятии лидера i .

Пусть $\bar{p}^{opt}$ — оптимальный вектор цен лидера в задаче MDP_L , а J_L^{opt} — множество клиентов, обслуживаемых лидером при указанном векторе цен $\bar{p}^{opt}$ в некотором оптимальном решении. По приведённым выше свойствам в задаче $MMPL$ при ценах $\bar{p}^{opt}$ лидер точно обслужит всех клиентов из множества J_L^{opt} . Значит, доход лидера в задаче $MMPL$ при оптимальных ценах из задачи MDP_L будет не меньше, чем оптимум MDP_L . Теорема 4 доказана.

Задача MDP_L принадлежит классу NPO [15]. Оценку из теоремы 4 будем обозначать через LB_{md} . Для данной задачи удалось построить одноуровневую модель линейного смешанного целочисленного программирования. Введём величину $d_{ij} = \min_{k \in I_F} \{c_{kj} + 1 - c_{ij}, 0\}$. Модель получается путём добавления к модели двухуровневой задачи фабричного ценообразования (МР) из п. 1.1, записанной для лидера, нового ограничения:

$$z_{ij} \leq d_{ij}, \quad i \in I_L, j \in J,$$

которое гарантирует, что лидер не сможет обслужить клиента, которого может обслужить последователь.

В задаче MUP_L мы сужаем множество решений последователя и, казалось бы, должны ожидать увеличения дохода лидера. На самом деле, оптимум MUP_L не является ни нижней, ни верхней границей.

Пример 1. Пусть у лидера есть одно предприятие с номером 1, а у последователя есть два предприятия с номерами 2, 3. Пусть имеется 4 клиента со следующими бюджетами и транспортными затратами:

$$\begin{aligned} b_1 = b_2 = 3, \quad b_3 = 4, \quad b_4 = 9, \\ c_{11} = c_{12} = 3, \quad c_{13} = 0, \quad c_{14} = 9, \\ c_{21} = c_{22} = c_{23} = c_{24} = 0, \\ c_{31} = c_{32} = c_{33} = c_{34} = 0. \end{aligned}$$

На рис. 1 дана иллюстрация этого примера. Ромбом и квадратами обозначены предприятия лидера и последователя соответственно, кругами — клиенты. Линиями обозначены нулевые транспортные затраты, отсутствие линий означает, что транспортные затраты равны бюджету клиента. Здесь лидер может обслужить только клиента 3, значит, его максимальный доход не может превышать 4. Более того, если лидер обслуживает клиента 3, то его доход равен установленной цене.

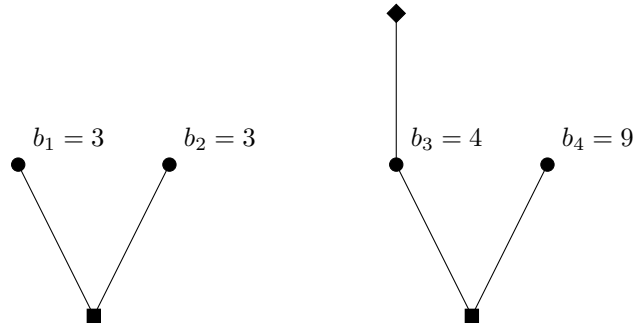


Рис. 1. Пример задачи конкурентного ценообразования

Рассмотрим исходную задачу с фабричным ценообразованием у последователя. Пусть лидер установил цену $p_1 = 4$, в таком случае последователю выгодно на предприятии 2 обслужить клиентов 1 и 2 по цене 3, а на предприятии 3 обслужить клиента 4 по цене 9, но не клиента 3 по более низкой цене. Таким образом, лидер получит доход 4.

Теперь поменяем стратегию ценообразования последователя на равнономерную. Это означает, что последователь устанавливает единую цену на обоих своих предприятиях. Доход лидера не меньше установленной цены. Следовательно, чтобы получить доход 4 (такой же, как при фабричном ценообразовании у последователя), он вынужден установить цену $p_1 = 4$. Тогда последователь установит цену $q = 3$ и обслужит при этом всех клиентов.

Таким образом, оптимальное значение целевой функции лидера при фабричном ценообразовании у последователя может быть выше, чем при равномерном.

Пример 2. Пусть у лидера есть одно предприятие с номером 1, а у последователя — два предприятия с номерами 2, 3. Пусть имеется два клиента со следующими бюджетами и транспортными затратами:

$$\begin{aligned} b_1 &= b_2 = 2, \\ c_{11} &= 1, \quad c_{12} = 2, \\ c_{21} &= 0, \quad c_{22} = 2, \\ c_{31} &= 2, \quad c_{32} = 0. \end{aligned}$$

На рис. 2 дана иллюстрация этого примера. Ромбом и квадратами обозначены предприятия лидера и последователя соответственно, кругами — клиенты. Отсутствие линий означает, что транспортные затраты равны бюджету клиента. Здесь лидер может получить доход не больше 1. Максимум достижим только при цене $p_1 = 1$.

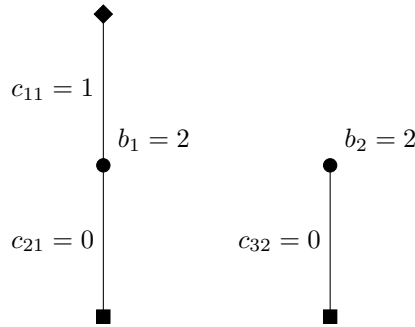


Рис. 2. Пример задачи конкурентного ценообразования.

Рассмотрим исходную задачу с фабричным ценообразованием у последователя. Пусть лидер установил цену $p_1 = 1$. Последователь назначит цены $q_2 = 1$, $q_3 = 2$ и получит максимальный доход, обслужив всех клиентов.

Теперь поменяем стратегию ценообразования последователя на равномерную. Лидер по-прежнему установит цену $p_1 = 1$. Последователь имеет два оптимальных решения $q = 1$ и $q = 2$. В силу кооперации он выберет второе решение и обслужит только второго клиента. Доход лидера составит 1.

Следовательно, оптимальное значение целевой функции лидера при фабричном ценообразовании у последователя меньше, чем при равномерном.

Эти два примера приводят нас к следующему утверждению.

Теорема 5. *Оптимум задачи MUP_L не является ни верхней, ни нижней границей для оптимума задачи MMR_L .*

3. Алгоритмы решения

Для решения задачи MMR_L предлагаем подходы, основанные на приближённом алгоритме решения одноуровневой задачи размещения и фабричного ценообразования [16]. Все подходы используют принцип локальных улучшений вектора цен лидера, тогда как параметрическая задача последователя при фиксированных ценах лидера решается с помощью эвристики, также использующей локальные улучшения, что существенно сокращает время счёта. По решению последователя, построенному эвристикой, во многих случаях можно сделать довольно точное предположение об ожидаемом доходе лидера для заданного вектора цен. Другими словами, эвристический алгоритм даёт хорошую оценку значения целевой функции на допустимом решении с заданным вектором цен лидера. Чтобы найти это допустимое решение, нужно решить NP-трудную задачу последователя. Сокращение числа обращений к точному алгоритму решения задачи последователя, чего и пытаемся добиться применением эвристики, существенно ускоряет время работы алгоритма решения задачи MMR_L .

Стоит отметить, что иногда оценка, полученная эвристическим алгоритмом решения задачи последователя, оказывается слишком грубой, и локальный поиск по ценам лидера может зайти в бесперспективную окрестность и остановиться на достаточно плохом решении всего из-за одной ошибки. Тем самым для корректировки ошибок в предсказании дохода время от времени полученное эвристикой решение корректируется путём точного решения задачи последователя. Однако тут нужно соблюдать баланс: при частой корректировке решений быстро растёт время счёта, при недостаточной — алгоритм часто приводит к плохому решению.

3.1. Окрестностные структуры. Опишем используемые окрестностные структуры в алгоритмах локального поиска. Приведённые ниже окрестности хорошо известны [16].

Через $k\text{-Flip}(p)$ обозначается окрестность вектора цен лидера p , содержащая все векторы цен, отличающиеся от p не более чем в k компонентах.

Через $k\text{-Flip}(q|p)$ обозначается окрестность вектора цен последователя q , где все соседи отличаются от q не более чем в k компонентах. Вектор p в названии окрестности соответствует ценам лидера, которые используются при определении дохода последователя.

3.2. Алгоритмы решения задачи последователя. Опишем алгоритмы, используемые в данной работе для нахождения решений задачи последователя.

Для нахождения точного решения задачи последователя при фиксированном векторе цен лидера используется схема, описанная в п. 1.1, причём все возникающие задачи ЦЛП решаются MILP-решателем. Обозначим эту процедуру через $EX(p)$.

Перейдём к описанию приближённых алгоритмов. Базовым алгоритмом является локальный поиск [16], в котором, начиная с некоторого стартового решения, на каждой итерации происходит переход в соседнее (относительно некоторой заданной окрестности текущего решения) решение с лучшим значением целевой функции. В данном случае использована окрестность 1-Flip($q | p$). Для этого алгоритма различают две стратегии: первое улучшение (first improve [19]), при котором переходим в новую точку сразу, как только нашли первое решение с лучшим значением целевой функции, и наибольшее улучшение (best improve [19]), при котором переходим в лучшее решение в окрестности после её полного просмотра. В данной работе для решения задачи последователя преимущественно использовалась стратегия первого улучшения, так как в вычислительных экспериментах она занимала меньшее время при схожем качестве решений (см. табл. 7).

Алгоритм локального спуска с чередующимися окрестностями (VND [16, 18]) основан на локальном поиске и после нахождения локального оптимума позволяет выйти из окрестности локального оптимума с помощью детерминированной процедуры k -Improve($q | p$) и продолжить локальный поиск со следующей точки. Сама процедура k -Improve($q | p$) последовательно просматривает окрестности 2-Flip($q | p$), $\dots$, k -Flip($q | p$) до того момента, пока не найдёт решение лучше текущего [16].

3.3. Алгоритмы решения задачи лидера. Опишем алгоритмы решения задачи лидера. На верхнем уровне также используется простой локальный поиск или локальный спуск с чередующимися окрестностями со стратегией наибольшего улучшения. Для нахождения решения задачи

Алгоритм 1. LS | EX

Выход: Вектор цен лидера p^*

- 1: $p \leftarrow \text{random}$
 - 2: $p^* \leftarrow \text{undefined}$
 - 3: **while** $f^L(p) > f^L(p^*)$ **do**
 - 4: $p^* \leftarrow p$
 - 5: $p \leftarrow \arg \max_{p' \in 1\text{-Flip}(p)} f^L(p', EX(p'))$
-

последователя первой очевидной идеей является использование точной процедуры. Описание реализации эвристики приведено в алгоритме 1. Для удобства описания алгоритмов введём $f^L(p, q)$ — значение целевой функции лидера при ценах p и q лидера и последователя соответственно. Аргумент q опускается, если используется оптимальная цена последователя.

В алгоритме 1 необходимо решать линейную задачу целочисленного программирования для каждого просмотренного элемента окрестности, что может занимать большое вычислительное время.

Следующая идея основана на замене процедуры точного решения задачи последователя процедурой локального поиска, а точнее, процедурой VND_k : применим процедуру VND, снабжённую процедурой улучшения k -Improve с периодической корректировкой в конце каждой итерации локального поиска. Другими словами, после просмотра очередной окрестности цен лидера вычисляем доход лидера, используя точный алгоритм решения задачи последователя. При этом рассматривать стратегию первого улучшения для лидера в данном случае не имеет смысла, так как решение, полученное таким методом в конце итерации, ещё с большей вероятностью окажется хуже рекорда. Описание реализации эвристики со стратегией лучшего улучшения приведено в алгоритме 2.

Алгоритм 2. LS | VND_k

Вход: Параметр k

Выход: Вектор цен лидера p^*

- 1: $p \leftarrow \text{random}$
 - 2: $p^* \leftarrow \text{undefined}$
 - 3: **while** $f^L(p) > f^L(p^*)$ **do**
 - 4: $p^* \leftarrow p$
 - 5: $p' \leftarrow \arg \max_{p' \in 1\text{-Flip}(p)} f^L(p', VND_k(p'))$
-

Алгоритмы LS | LS, VND_k | LS определяются аналогично.

Приведённые алгоритмы исполняются существенно быстрее по времени, так как точная процедура запускается лишь в конце просмотра окрестности. Однако у них есть существенный недостаток: часто возникает ситуация, когда лучшее решение, найденное приближёнными процедурами, при проверке точным алгоритмом оказывается хуже, чем текущий рекорд, и, несмотря на то, что в окрестности есть решения лучше текущего рекорда, алгоритм заканчивает свою работу на довольно плохом решении, которое иногда имеет значение целевой функции даже меньше нижних оценок из разд. 2.

Для исправления такой ситуации разумно запоминать не только рекорд текущей итерации, полученный с помощью приближённой процедуры, но и h лучших решений на текущей итерации, проверяя их с помощью точной процедуры в порядке убывания значений целевой функции, полученных приближённым алгоритмом. Описание реализации эвристики со стратегией лучшего улучшения приведено в алгоритме 3.

Алгоритм 3. $(\text{LS} \mid \text{VND}_k)^h$

Вход: Параметры h, k

Выход: Вектор цен лидера p^*

```

1:  $p \leftarrow \text{random}$ 
2:  $p^* \leftarrow \text{undefined}$ 
3:  $p^* \leftarrow p$ 
4:  $P \leftarrow$  массив, упорядоченный по убыванию
5:  $i \leftarrow 1$ 
6: while  $i \leq h$  do
7:    $p' \leftarrow \arg \max_{p' \in 1\text{-Flip}(p) \setminus P} f^L(p', \text{VND}_k(p'))$ 
8:   поместить  $p'$  в  $P$ 
9:    $i \leftarrow i + 1$ 
10:  $i \leftarrow 1$ 
11: while  $i \leq h$  do
12:   if  $f^L(P[i]) > f^L(p^*)$  then
13:      $p \leftarrow P[i]$ 
14:     go to шаг 3
15:    $i \leftarrow i + 1$ 

```

Алгоритмы $(\text{LS} \mid \text{LS})^h$, $(\text{VND}_k \mid \text{LS})^h$ определяются аналогично.

Число проверяемых решений разумно выбирать в зависимости от размерности задачи. Заметим, что при $h = \infty$, мы в определённом порядке просматриваем все элементы окрестности, пока не найдём решение лучше текущего, т. е. это алгоритм $\text{LS} \mid \text{EX}$ со стратегией первого улучшения и приоритетами просмотра элементов окрестности. Назовём такой алгоритм $\text{LS} \mid (\text{LS} \mid \text{EX})$.

4. Вычислительные эксперименты

В этом разделе приведены результаты численных экспериментов, проведённых на персональном компьютере, оснащённом процессором AMD Ryzen 2200 3,4 ГГц с 4 ядрами и оперативной памятью DDR4 3200 МГц 32 ГБ.

В качестве MILP-решателя применён Gurobi. В табл. 1 приведено сравнение времени работы решателей Gurobi 12.0.1 и SCIP 9.2.4 на некоторых примерах.

Таблица 1

Сравнение времени работы
решателей Gurobi и SCIP

m	n^L	n^F	LS (LS EX)			
			Gurobi		SCIP	
			gap	время	gap	время
30	2	2	0,00	0,49	0,00	4,10
		5	0,00	6,02	0,00	33,87
		10	0,02	33,44	0,02	307,88
	5	2	0,01	0,85	0,01	4,03
		5	0,03	52,42	0,03	230,05
		10	0,03	134,87	0,03	1109,88
	10	2	0,04	3,05	0,04	8,73
		5	0,05	35,36	0,05	151,25
		10	0,03	93,46	0,03	450,17
	20	2	0,05	8,35	0,05	29,22
		5	0,03	47,61	0,03	148,39
		10	0,02	127,27	0,02	485,56
60	50	2	0,04	118,83	0,04	365,98
		5	0,00	1399,64	0,02	2894,86
		10	0,03	2937,12	0,14	3600,00
Среднее			0,03	333,25	0,03	654,93

Основная сложность проведения вычислительного эксперимента заключается в невозможности сравнения полученных приближённых решений с точными ввиду того, что для задачи ММР_L неизвестно точных алгоритмов, способных решать задачу за приемлемое время, поэтому для оценки эффективности алгоритмов подготовлены две группы тестов и вычислены нижние и верхние оценки из разд. 2. Время во всех таблицах статьи указано в секундах.

4.1. Библиотека тестов. Для исследования алгоритмов используются две группы тестовых примеров. Первая группа основана на открытой библиотеке примеров задач размещения и ценообразования [20]. В этой библиотеке предприятиям и клиентам соответствуют точки плоскости, транспортные затраты j -го клиента до i -го предприятия берутся равными евклидову расстоянию между точками, соответствующими j -му клиенту и i -му предприятию. Координаты точек берутся из равномерного распределения на квадрате $[0, 100] \times [0, 100]$, бюджеты клиентов берутся из равномерного распределения на отрезке $[0, 100]$. На основе каждого примера этой библиотеки составлены примеры различной

Таблица 2

**Результаты вычислений верхних и нижних оценок
для первой группы тестов**

m	n^L	n^F	$\frac{UB_{hp}}{LB}$	$\frac{LB_{um}}{LB_{md}}$	gap_{UB}	gap_{LB}	Время		
							UB_{hp}	LB_{um}	LB_{md}
30	2	2	1,86	1,38	0,39	0,09	0,05	5,37	0,02
		5	3,24	2,73	0,63	0,07	0,05	38,21	0,01
		10	3,93	3,00	0,69	0,08	0,05	147,98	0,02
	5	2	1,80	1,24	0,34	0,15	0,45	5,35	0,11
		5	3,15	1,45	0,52	0,21	0,46	32,69	0,07
		10	3,93	1,66	0,60	0,24	0,45	134,18	0,04
	10	2	1,91	1,04	0,35	0,19	1,19	7,01	0,54
		5	3,05	1,23	0,55	0,21	1,21	37,07	0,27
		10	4,07	1,37	0,66	0,24	1,24	128,37	0,08
	20	2	1,88	0,92	0,35	0,18	7,55	9,62	1,13
		5	2,94	1,06	0,52	0,26	7,54	44,19	0,40
		10	3,81	1,21	0,62	0,27	7,66	131,75	0,23
60	50	2	—	0,89	—	0,12	—	84,50	121,44
		5	—	0,92	—	0,18	—	294,23	20,04
		10	—	1,02	—	0,16	—	1994,43	7,73
Среднее			2,92	1,41	0,51	0,18	133,89	206,33	10,14

размерности для рассматриваемой задачи. Каждый пример размерности $m \times n^L \times n^F$ состоит из m первых клиентов, n^L первых предприятий (лидера) и n^F последних предприятий (последователя). Примеры больших

Таблица 3

**Результаты вычислений верхних и нижних оценок
для второй группы тестов**

m	n^L	n^F	$\frac{UB_{hp}}{LB}$	$\frac{LB_{um}}{LB_{md}}$	gap_{UB}	gap_{LB}	Время		
							UB_{hp}	LB_{um}	LB_{md}
30	2	2	2,53	2,34	0,54	0,05	0,12	0,80	0,01
		5	4,08	4,53	0,72	0,10	0,11	7,58	0,01
		10	6,10	6,00	0,83	0,03	0,12	21,89	0,01
	5	2	2,45	1,55	0,48	0,16	1,30	0,75	0,04
		5	3,84	2,81	0,68	0,16	1,33	5,63	0,02
		10	4,84	3,58	0,77	0,10	1,37	13,90	0,01
	10	2	2,40	1,19	0,46	0,20	7,60	0,84	0,11
		Среднее		3,75	2,73	0,64	0,11	1,71	7,34

размеров формировались из примеров меньших размеров путём добавления новых предприятий к уже существующим, что позволяет пронаблюдать изменения в поведении игроков при увеличении возможностей лидера или последователя.

Вторая группа тестовых примеров была сформирована аналогично, с тем лишь отличием, что бюджеты клиентов и координаты предприятий и клиентов брались из равномерного распределения на отрезке $[0, 10]$, что позволило для небольших размерностей найти точное решение полным перебором цен.

В экспериментах использовалось по 4 примера каждой представленной ниже размерности в каждой группе тестов.

4.2. Оценки оптимумов. В табл. 2 и 3 представлены усреднённые характеристики верхних и нижних границ из разд. 2 для первой и второй групп тестов соответственно:

$$\text{gap}_{\text{UB}} = \frac{\text{UB}_{\text{hp}} - \text{best}}{\text{UB}_{\text{hp}}}, \quad \text{gap}_{\text{LB}} = \frac{\text{best} - \max\{\text{LB}_{\text{md}}, \text{LB}_{\text{um}}\}}{\text{best}},$$

где best — значение целевой функции на лучшем известном решении, $\text{LB} = \max\{\text{LB}_{\text{um}}, \text{LB}_{\text{md}}\}$. Для части примеров не удалось вычислить верхнюю оценку UB_{hp} при ограничении на время счёта 8 часов.

Из результатов следует, что оценка LB_{md} вычисляется намного быстрее оценки LB_{um} . При малом количестве предприятий лидера оценка LB_{um} даёт более точную аппроксимацию, при увеличении же количества предприятий становится более точной оценка лидера LB_{md} .

Из табл. 3 следует, что релаксация экстремального ограничения оказывается на практике довольно грубой верхней оценкой оптимума, причём с возрастанием количества предприятий лидера и количества клиентов наблюдается существенный рост затраченного на вычисление времени.

4.3. Разработанные алгоритмы. Для оценки эффективности алгоритмов сравнивались средние по размерностям время, указанное в таблицах в секундах, и величина

$$\text{gap} = \frac{\text{best} - \text{val}}{\text{best}},$$

где best — лучшая нижняя оценка, достигнутая представленными в статье алгоритмами, а val — оценка, полученная рассматриваемым алгоритмом.

В табл. 4 и 5 представлены результаты алгоритмов со стратегией первого улучшения у последователя для первой и второй группы тестов. В табл. 6 представлены абсолютные значения для некоторых примеров.

Таблица 4

Результаты вычислений на первой группе тестов

m	n ^L	n ^F	LS EX		LS LS		LS VND ₂		VND ₂ LS	
			гар	время	гар	время	гар	время	гар	время
30	2	2	0,00	33,8	0,00	0,5	0,00	0,8	0,00	1,3
		5	0,06	220,8	0,00	3,6	0,15	12,7	0,00	9,9
		10	0,02	734,6	0,31	8,6	0,03	90,5	0,29	35,0
	5	2	0,01	138,6	0,04	0,5	0,01	2,1	0,03	8,8
		5	0,03	727,8	0,13	3,0	0,08	30,2	0,10	44,9
		10	0,02	2602,4	0,36	7,9	0,15	275,8	0,26	260,6
	10	2	0,04	489,5	0,15	0,6	0,04	5,6	0,02	46,8
		5	0,05	2325,3	0,12	4,6	0,08	74,4	0,07	160,4
		10	0,11	3571,3	0,34	7,6	0,27	356,8	0,24	1109,1
	20	2	0,03	1826,7	0,17	1,6	0,04	16,7	0,06	260,9
		5	0,05	3600,0	0,26	2,7	0,14	239,3	0,11	781,4
		10	0,18	3600,0	0,41	8,7	0,20	1474,8	0,33	2707,9
60	50	2	0,79	3600,0	0,15	19,9	0,04	176,1	0,06	2899,5
		5	1,00	3600,0	0,21	15,5	0,16	1158,6	0,21	3600,0
		10	1,00	3600,0	0,35	79,1	0,29	3469,6	0,35	3600,0
Среднее			0,23	2044,7	0,20	11,0	0,11	492,3	0,14	1035,1

m	n ^L	n ^F	(LS (LS EX))		(LS LS) ^{1,5n^L}		(LS VND ₂) ^{1,5n^L}		(VND ₂ LS) ^{1,5n^L}	
			гар	время	гар	время	гар	время	гар	время
30	2	2	0,00	0,5	0,00	0,5	0,00	0,8	0,00	1,3
		5	0,00	6,0	0,10	4,3	0,15	13,4	0,10	11,9
		10	0,02	33,4	0,27	13,5	0,02	94,9	0,01	55,6
	5	2	0,01	0,9	0,01	0,8	0,01	2,1	0,00	9,7
		5	0,03	52,4	0,04	10,6	0,02	42,6	0,04	45,9
		10	0,03	134,9	0,03	56,3	0,05	404,1	0,02	289,7
	10	2	0,04	3,0	0,04	3,0	0,04	5,5	0,00	36,1
		5	0,05	35,4	0,08	11,6	0,05	91,3	0,07	153,5
		10	0,03	93,5	0,04	64,7	0,08	782,9	0,04	787,9
	20	2	0,05	8,4	0,05	4,5	0,04	16,7	0,02	261,2
		5	0,03	47,6	0,03	47,4	0,00	336,7	0,02	472,5
		10	0,02	127,3	0,02	125,6	0,05	2578,7	0,02	2504,0
60	50	2	0,04	118,8	0,04	161,9	0,04	182,9	0,00	3600,0
		5	0,00	1399,6	0,00	1184,6	0,01	3235,3	0,00	3600,0
		10	0,03	2937,1	0,06	2591,0	0,28	3600,0	0,06	3600,0
Среднее			0,03	333,3	0,05	285,4	0,06	759,2	0,03	1028,6

Таблица 5

Результаты вычислений на второй группе тестов

m	n^L	n^F	LS EX		LS LS		LS VND ₂		VND ₂ LS	
			гар	время	гар	время	гар	время	гар	время
30	2	2	0,00	3,2	0,10	0,1	0,07	0,2	0,07	0,2
		5	0,00	20,6	0,04	1,2	0,04	1,3	0,04	1,5
		10	0,00	72,3	0,03	2,8	0,03	3,0	0,03	3,0
30	5	2	0,00	5,9	0,07	0,2	0,05	0,2	0,06	0,3
		5	0,09	40,6	0,09	1,2	0,09	1,3	0,00	1,5
		10	0,00	83,1	0,00	2,2	0,00	2,9	0,00	2,7
30	10	2	0,01	19,1	0,03	0,2	0,03	0,2	0,01	0,4
Среднее			0,01	35,0	0,05	1,1	0,04	1,3	0,03	1,4
m	n^L	n^F	(LS (LS EX))		(LS LS) ^{1,5n^L}		(LS VND ₂) ^{1,5n^L}		(VND ₂ LS) ^{1,5n^L}	
			гар	время	гар	время	гар	время	гар	время
30	2	2	0,10	0,1	0,10	0,4	0,07	0,1	0,07	0,2
		5	0,04	1,2	0,04	1,3	0,04	1,1	0,04	1,5
		10	0,03	2,8	0,03	2,9	0,03	3,2	0,03	3,2
30	5	2	0,07	0,2	0,07	0,2	0,05	0,2	0,06	0,2
		5	0,09	1,0	0,09	1,2	0,09	1,2	0,00	1,6
		10	0,00	2,0	0,00	2,3	0,00	2,7	0,00	2,7
30	10	2	0,03	0,3	0,04	0,3	0,04	0,2	0,00	0,4
Среднее			0,05	1,1	0,05	1,2	0,04	1,3	0,03	1,4

На основе результатов, полученных из экспериментов, можно заключить, что алгоритмы, использующие большее количество окрестностных структур, в среднем работают лучше. Когда VND-метаэвристика используется на верхнем уровне, просматривается большее количество потенциально оптимальных решений, чем когда используется LS-процедура. Когда же VND-метаэвристика используется на нижнем уровне, делается более точное предположение о решении последователя, следовательно, верхний уровень принимает более разумное решение, чем при использовании LS-процедуры. Алгоритмы, проверяющие большее количество решений последователя точной процедурой, дают в среднем лучшие решения. Отдельно можно выделить алгоритм LS | (LS | EX), который имеет низкий гар и сравнительно небольшое время счёта, что делает его предпочтительным в приложениях.

При анализе результатов экспериментов установлено, что на используемых примерах алгоритм LS | LS со стратегией первого улучшения доминируется алгоритмами VND₂ | LS, LS | (LS | EX) и (VND₂ | LS)^{1,5n^L} со стратегией первого улучшения, алгоритм (LS | LS)^{1,5n^L} доминируется

Таблица 6

Абсолютные значения на первой группе тестов

m	n^L	n^F	LS	EX	LS	LS	LS	VND ₂	VND ₂	LS
			знач.	время	знач.	время	знач.	время	знач.	время
30	2	2	202	43,0	202	0,8	202	0,9	202	1,1
		5	87	197,8	87	4,6	87	16,4	87	11,7
		10	72	549,5	61	10,0	72	133,7	61	45,9
	5	2	319	178,5	314	0,6	319	2,0	314	3,3
		5	186	700,2	165	4,0	185	29,1	184	85,4
		10	183	1640,1	118	7,5	165	307,7	118	148,5
	10	2	498	828,9	472	1,1	498	6,2	492	25,3
		5	266	2219,0	262	6,0	266	63,6	262	131,6
		10	240	3485,2	145	5,6	198	372,3	177	967,3
	20	2	537	1810,1	479	1,2	537	12,0	479	54,6
		5	348	3600,0	285	3,6	202	41,7	351	971,7
		10	243	3600,0	102	6,3	283	1766,8	197	3600,0
60	50	2	0	3600,0	775	9,6	1044	205,0	1122	3600,0
		5	0	3600,0	601	20,9	651	1176,9	601	3600,0
		10	0	3600,0	381	133,7	385	3600,0	381	3600,0
m	n^L	n^F	(LS (LS EX)		(LS LS) ^{1,5n^L}		(LS VND ₂) ^{1,5n^L}		(VND ₂ LS) ^{1,5n^L}	
			знач.	время	знач.	время	знач.	время	знач.	время
30	2	2	202	0,8	202	0,8	202	0,9	202	1,1
		5	87	5,2	87	5,4	87	16,4	87	13,7
		10	72	21,3	72	20,8	72	130,1	72	62,6
	5	2	319	1,3	319	1,4	319	2,0	319	5,3
		5	186	14,0	186	14,1	186	35,0	186	59,3
		10	189	37,9	189	39,6	183	469,9	189	219,8
	10	2	498	7,6	498	7,5	498	5,8	498	23,9
		5	266	12,0	266	11,6	266	62,0	266	149,2
		10	233	124,3	224	53,1	228	596,4	224	587,8
	20	2	537	21,9	534	6,4	537	12,0	534	82,5
		5	335	25,0	335	24,0	357	245,8	335	337,2
		10	276	174,0	276	175,2	293	2461,5	276	1747,9
60	50	2	1045	27,6	1049	30,7	1049	213,7	1120	3600,0
		5	843	587,8	843	630,9	817	3520,4	843	3600,0
		10	486	3600,0	486	3600,0	385	3600,0	486	3600,0

алгоритмом $(VND_2 | LS)^{1,5n^L}$ со стратегией первого улучшения, а алгоритм $LS | LS$ со стратегией наибольшего улучшения доминируется алгоритмом $(VND_2 | LS)^{1,5n^L}$ со стратегией наибольшего улучшения. Других доминируемых алгоритмов нет.

Таблица 7

**Результаты вычислений на первой группе тестов
со стратегией наибольшего улучшения**

m	n^L	n^F	LS LS		LS VND ₂		VND ₂ LS	
			gap	время	gap	время	gap	время
30	2	2	0,00	0,48	0,00	0,88	0,00	0,78
		5	0,00	3,27	0,15	11,00	0,00	6,61
		10	0,30	8,83	0,03	79,34	0,30	19,44
	5	2	0,04	0,48	0,01	2,00	0,03	3,25
		5	0,21	2,16	0,07	32,79	0,21	16,00
		10	0,43	5,36	0,23	217,73	0,25	175,70
	10	2	0,15	0,64	0,04	5,52	0,04	20,61
		5	0,12	4,14	0,08	75,17	0,03	101,74
		10	0,39	6,02	0,34	328,65	0,28	578,97
	20	2	0,17	1,51	0,04	16,53	0,06	156,65
		5	0,26	2,67	0,14	236,26	0,12	503,92
		10	0,41	6,45	0,31	1140,42	0,33	1334,61
	Среднее		0,21	3,50	0,12	178,86	0,14	243,19
m	n^L	n^F	$(LS LS)^{1,5n^L}$		$(LS VND_2)^{1,5n^L}$		$(VND_2 LS)^{1,5n^L}$	
			gap	время	gap	время	gap	время
30	2	2	0,00	0,42	0,00	0,77	0,00	0,80
		5	0,10	3,79	0,15	12,12	0,10	8,90
		10	0,25	16,19	0,02	88,48	0,25	32,56
	5	2	0,01	0,83	0,01	1,97	0,00	4,17
		5	0,05	12,89	0,02	40,93	0,05	32,78
		10	0,07	48,53	0,06	336,63	0,06	172,41
	10	2	0,04	2,38	0,04	5,35	0,00	18,14
		5	0,02	12,33	0,05	95,29	0,00	91,12
		10	0,08	57,86	0,08	913,87	0,08	383,02
	20	2	0,05	4,03	0,04	16,49	0,02	156,57
		5	0,03	49,76	0,00	330,36	0,03	322,49
		10	0,02	161,60	0,09	2416,76	0,01	1357,19
	Среднее		0,06	30,88	0,05	354,92	0,05	215,01

Отметим, что некоторые алгоритмы не нашли допустимых решений с ненулевым доходом лидера для части примеров (например, алгоритм LS | LS в случае размерности $m = 30$, $n^L = 2$, $n^F = 10$ первой группы тестов). В таких примерах стартовое решение содержит нулевой доход, а процедура локального поиска не может улучшить решение в силу того, что предполагаемый доход существенно отличается от реального,

поскольку решение задачи последователя приближённое. В связи с этим проведён эксперимент, в котором при фиксированной цене лидера сравниваются доходы лидера при оптимальном кооперативном решении последователя и решении последователя с отсечённым оптимальным решением. В результате в 9% случаев доходы лидера не совпали.

Также тестировались алгоритмы локального поиска для задачи последователя, использующие помимо стратегии первого улучшения стратегию наибольшего улучшения, но при схожих значениях целевой функции они тратят большее вычислительное время, что видно из результатов, представленных в табл. 7.

Заключение

Результаты экспериментов демонстрируют работоспособность предложенных алгоритмов для решения трёхуровневых задач ценообразования. Можно сделать вывод, что для прикладных задач предпочтительно использовать алгоритм $LS \mid (LS \mid EX)$, который за разумное время находит (в среднем) лучшие решения. Однако для ситуации, когда требуется найти наилучшее решение за разумное время, экспериментально выявлен класс недоминируемых алгоритмов.

Также построены алгоритмы вычисления верхних и нижних границ. Как показал численный эксперимент, на практике они дают плохое представление об оптимуме задачи.

С практической точки зрения исследование может быть продолжено, поскольку имеются возможности для существенного улучшения предложенных алгоритмов. В частности, можно разработать методы отсечения бесперспективных элементов окрестностей, рассмотреть различные способы построения стартовых решений и исследовать другие эвристические подходы для приближённого решения задачи последователя.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2026-0021). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Панин А. А., Плясунов А. В. Задача ценообразования. Часть 1. Точные и приближённые алгоритмы решения // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 5. С. 83–100.

2. **Плясунов А. В., Панин А. А.** Задача ценообразования. Часть 2. Вычислительная сложность // Дискрет. анализ и исслед. операций. 2013. Т. 19, № 6. С. 56–71.
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. **Lederes Ph. J., Thisse J.-F.** Competitive location on network under delivered pricing // Oper. Res. Lett. 1990. V. 9, No. 3. P. 147–154. DOI: 10.1016/0167-6377(90)90012-T.
5. **Aboolian R., Berman O., Krass D.** Optimizing pricing and location decisions for competitive service facilities charging uniform price // J. Oper. Res. Soc. 2008. V. 59, No. 11. P. 1506–1519. DOI: 10.1057/palgrave.jors.2602493.
6. **Luer-Villagra A., Marianov V.** A competitive hub location and pricing problem // Eur. J. Oper. Res. 2013. V. 231, No. 3. P. 734–744. DOI: 10.1016/j.ejor.2013.06.006.
7. **Serra D., ReVelle Ch.** Competitive locations and pricing on networks // Geogr. Anal. 1999. V. 31, No. 2. P. 109–129. DOI: 10.1111/gean.1999.31.1.109.
8. **Aboolian R., Berman O., Krass D.** Competitive facility location model with concave demand // Eur. J. Oper. Res. 2007. V. 181, No. 2. P. 598–619. DOI: 10.1016/j.ejor.2005.10.075.
9. **Bouhtou M., Grigoriev A., van Hoesel S., van der Kraaij A. F., Spieksma F. C. R., Uetz M.** Pricing bridges to cross a river // Nav. Res. Logist. 2007. V. 54, No. 4. P. 411–420. DOI: 10.1002/nav.20216.
10. **Diakova Z. S., Kochetov Yu. A.** A double VNS heuristic for the facility location and pricing problem // Electron. Notes Discrete Math. 2012. V. 39, No. 4. P. 29–34. DOI: 10.1016/j.endm.2012.10.005.
11. **Ahmadi-Javid A., Amire E., Meskar M.** A profit-maximization location-routing-pricing problem: A branch-and-price algorithm // Eur. J. Oper. Res. 2018. V. 271, No. 3. P. 866–881. DOI: 10.1016/j.ejor.2018.02.020.
12. **Lin Y. H., Tian Q.** Facility location and pricing problem: Discretized mill price and exact algorithms // Eur. J. Oper. Res. 2023. V. 308, No. 2. P. 568–580. DOI: 10.1016/j.ejor.2022.11.052.
13. **Hansen P., Hanjoul P., Thisse J.-F., Peeters D.** Uncapacitated plant location under alternative spatial price policies // Manag. Sci. 1990. V. 36, No. 1. P. 41–57. DOI: 10.1287/mnsc.36.1.41.
14. **Garcia-Herreros P., Florensa C., Pratik M.** Capacity planning with competitive decision-makers: Trilevel milp formulation, degeneracy and solution approaches // Eur. J. Oper. Res. 2017. V. 262, No. 2. P. 449–463. DOI: 10.1016/j.ejor.2017.04.013.
15. **Plyasunov A. V., Panin A. A.** The multilevel facility location and pricing problems: the computational complexity and the stability analysis // Optim. Lett. 2022. V. 17, No. 6. P. 1295–1315. DOI: 10.1007/s11590-022-01924-3.

16. **Кочетов Ю. А., Панин А. А., Плясунов А. В.** Сравнение метаэвристик для решения двухуровневой задачи размещения предприятий и фабричного ценообразования // Дискрет. анализ и исслед. операций. 2015. Т. 22, № 3. С. 36–54. DOI: 10.17377/daio.2015.22.480.
17. **Sinnl M., Fischetti M., Ljubić I., Monaci M.** Intersection cuts for bilevel optimization // Integer programming and combinatorial optimization. Proc. 18th Int. Conf. (Liège, Belgium, June 1–3, 2016). Cham: Springer, 2016. P. 77–88. (Lect. Notes Comput. Sci.; V. 9682.) DOI: 10.1007/978-3-319-33461-5_7.
18. **Hansen P. N., Mladenović N.** Variable neighborhood search // Eur. J. Oper. Res. 2001. V. 130, No. 3. P. 449–467. DOI: 10.1016/S0305-0548(97)00031-2.
19. **Mladenović N., Hansen P.** First vs. best improvement: An empirical study // Discrete Appl. Math. 2006. V. 154, No. 5. P. 802–817. DOI: 10.1016/j.dam.2005.05.020.
20. Задача размещения и ценообразования // Дискретные задачи размещения. Новосибирск: ИМ СО РАН, 2025. URL: old.math.nsc.ru/AP/benchmarks/Pricing/price.html (accessed: 20.10.2025).

Драчев Руслан Романович

Панин Артём Александрович

Плясунов Александр Владимирович

Статья поступила

24 декабря 2025 г.

После доработки —

12 февраля 2026 г.

Принята к публикации

23 марта 2026 г.

UPPER AND LOWER BOUNDS
AND VARIABLE NEIGHBOURHOOD DESCENT
FOR COMPETITIVE PRICING PROBLEM*R. R. Drachev^a, A. A. Panin^b, and A. V. Plyasunov^c*¹Sobolev Institute of Mathematics,
4 Acad. Koptuyug Avenue, 630090 Novosibirsk, RussiaE-mail: ^arrdrachev@gmail.com,
^baapanin1988@gmail.com, ^capljas@math.nsc.ru

Abstract. This article considers a three-level competitive pricing problem. Two companies producing a homogeneous product compete for consumer demand. At the first level, the leader and the follower set prices at each of their facilities to maximize their own revenue. At the third level, each client chooses the facility with the lowest total purchasing and transportation costs and makes a purchase if these costs do not exceed their budget. The goal of the game is to find the leader's prices that maximize their revenue. The problem is known to be Σ_2^P -hard. Lower and upper bounds for the optimum are proposed, and approximate solution algorithms based on descent with alternating neighborhoods are developed. A computational experiment is conducted using examples of varying dimensions and complexity. Tab. 7, illustr. 2, bibliogr. 20.

Keywords: three-level programming, Stackelberg game, competitive pricing, local search, VND.

References

1. **A. V. Plyasunov** and **A. A. Panin**, The pricing problem. Part I: Exact and approximate algorithms, *Diskretn. Anal. Issled. Oper.* **19** (5), 83–100 (2012) [Russian] [*J. Appl. Ind. Math.* **7** (2), 241–251 (2013), DOI: 10.1134/S1990478913020142].
2. **A. V. Plyasunov** and **A. A. Panin**, The pricing problem. Part II: Computational complexity, *Diskretn. Anal. Issled. Oper.* **19** (6), 56–71 (2013) [Russian] [*J. Appl. Ind. Math.* **7** (3), 420–430 (2013), DOI: 10.1134/S1990478913030150].

3. **V. T. Dementyev** 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** (2), 31–40 (2002) [Russian].
4. **Ph. J. Lederes** and **J.-F. Thisse**, Competitive location on network under delivered pricing, *Oper. Res. Lett.* **9** (3), 147–154 (1990), DOI: 10.1016/0167-6377(90)90012-T.
5. **R. Aboolian**, **O. Berman**, and **D. Krass**, Optimizing pricing and location decisions for competitive service facilities charging uniform price, *J. Oper. Res. Soc.* **59** (11), 1506–1519 (2008), DOI: 10.1057/palgrave.jors.2602493.
6. **A. Luer-Villagra**, and **V. Marianov**, A competitive hub location and pricing problem, *Eur. J. Oper. Res.* **231** (3), 734–744 (2013), DOI: 10.1016/j.ejor.2013.06.006.
7. **D. Serra** and **Ch. ReVelle**, Competitive locations and pricing on networks, *Geogr. Anal.* **31** (2), 109–129 (1999), DOI: 10.1111/gean.1999.31.1.109.
8. **R. Aboolian**, **O. Berman**, and **D. Krass**, Competitive facility location model with concave demand, *Eur. J. Oper. Res.* **181** (2), 598–619 (2007), DOI: 10.1016/j.ejor.2005.10.075.
9. **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, *Nav. Res. Logist.* **54** (4), 411–420 (2007), DOI: 10.1002/nav.20216.
10. **Z. S. Diakova** and **Yu. A. Kochetov**, A double VNS heuristic for the facility location and pricing problem, *Electron. Notes Discrete Math.* **39** (4), 29–34 (2012), DOI: 10.1016/j.endm.2012.10.005.
11. **A. Ahmadi-Javid**, **E. Amire**, and **M. Meskar**, A profit-maximization location-routing-pricing problem: A branch-and-price algorithm, *Eur. J. Oper. Res.* **271** (3), 866–881 (2018), DOI: 10.1016/j.ejor.2018.02.020.
12. **Y. H. Lin** and **Q. Tian**, Facility location and pricing problem: Discretized mill price and exact algorithms, *Eur. J. Oper. Res.* **308** (2), 568–580 (2023), DOI: 10.1016/j.ejor.2022.11.052.
13. **P. Hansen**, **P. Hanjoul**, **J.-F. Thisse**, and **D. Peeters**, Uncapacitated plant location under alternative spatial price policies, *Manag. Sci.* **36** (1), 41–57 (1990), DOI: 10.1287/mnsc.36.1.41.
14. **P. Garcia-Herreros**, **C. Florensa**, and **M. Pratik**, Capacity planning with competitive decision-makers: Trilevel milp formulation, degeneracy and solution approaches, *Eur. J. Oper. Res.* **262** (2), 449–463 (2017), DOI: 10.1016/j.ejor.2017.04.013.
15. **A. V. Plyasunov** and **A. A. Panin**, The multilevel facility location and pricing problems: the computational complexity and the stability analysis, *Optim. Lett.* **17** (6), 1295–1315 (2022), DOI: 10.1007/s11590-022-01924-3.
16. **Yu. A. Kochetov**, **A. A. Panin**, and **A. V. Plyasunov**, Comparison of metaheuristics for the bilevel facility location and mill pricing problem, *Diskretn. Anal. Issled. Oper.* **22** (3), 36–54 (2015), DOI: 10.17377/daio.2015.22.480 [Russian] [*J. Appl. Ind. Math.* **9** (3), 392–401 (2015), DOI: 10.1134/S1990478915030102].

17. **M. Sinnl, M. Fischetti, I. Ljubić, and M. Monaci**, Intersection cuts for bilevel optimization, in *Integer Programming and Combinatorial Optimization*, Proc. 18th Int. Conf. (Liège, Belgium, June 1–3, 2016) (Springer, Cham, 2016), pp. 77–88 (Lect. Notes Comput. Sci., Vol. 9682.) DOI: 10.1007/978-3-319-33461-5_7
18. **P. N. Hansen and N. Mladenović**, Variable neighborhood search, *Eur. J. Oper. Res.* **130** (3), 449–467 (2001), DOI: 10.1016/S0305-0548(97)00031-2.
19. **N. Mladenović and P. Hansen**, First vs. best improvement: An empirical study, *Discrete Appl. Math.* **154** (5), 802–817 (2006), DOI: 10.1016/j.dam.2005.05.020.
20. Facility location and pricing problem, in *Discrete Location Problems* (IM SO RAN, Novosibirsk, 2025) [Russian], URL: old.math.nsc.ru/AP/benchmarks/Pricing/price.html (accessed: 20.10.2025).

Ruslan R. Drachev
Artem A. Panin
Aleksandr V. Plyasunov

Received December 24, 2025
Revised February 12, 2026
Accepted March 23, 2026