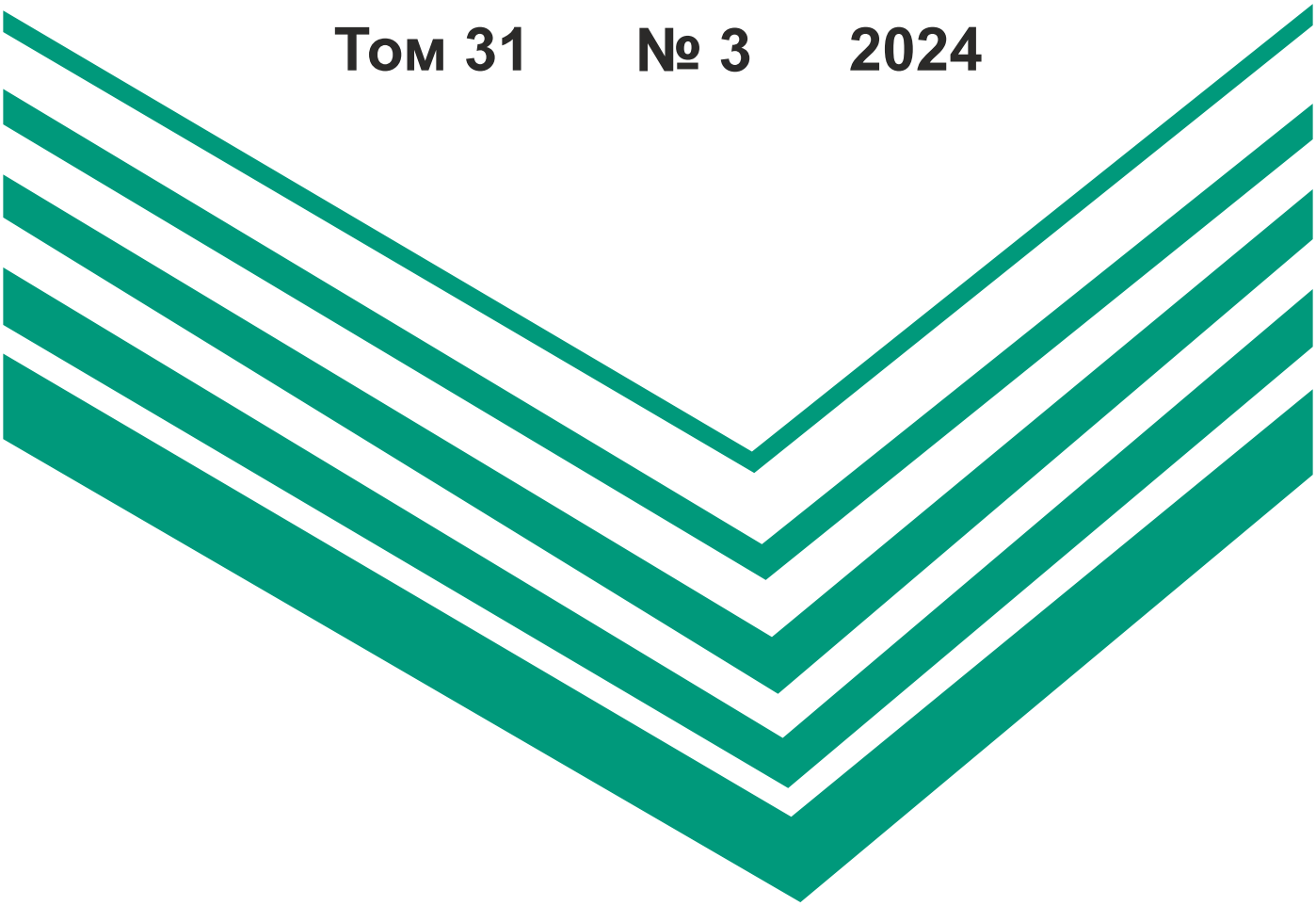
ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 31 № 3 2024



Новосибирск
Издательство Института математики

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2024. Т. 31, № 3. С. 79–104

УДК 519.8+518.25 DOI: 10.33048/daio.2024.31.788

ИССЛЕДОВАНИЕ ПОРОГОВОЙ УСТОЙЧИВОСТИ ДВУХУРОВНЕВОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ ПРОИЗВОДСТВА И ДИСКРИМИНАЦИОННОГО ЦЕНООБРАЗОВАНИЯ

М. Е. Водян[1, a], *А. А. Панин*[2, b], *А. В. Плясунов*[2, c]

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

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

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

Аннотация. Рассматривается задача пороговой устойчивости для двухуровневой задачи с медианным типом размещения предприятий и дискриминационным ценообразованием. При решении такой задачи необходимо найти радиус пороговой устойчивости и такое полудопустимое решение исходной двухуровневой задачи, для которого выручка лидера не меньше заранее заданного значения (порога) при любом отклонении бюджетов, не превышающем порогового радиуса устойчивости, и которое сохраняет свою полудопустимость. Таким образом, пороговый радиус устойчивости определяет предел возмущений бюджетов потребителей, при котором выполняются эти условия.

Разработаны два приближённых алгоритма решения задачи пороговой устойчивости на основе эвристики спуска с чередующимися окрестностями. Эти алгоритмы основываются на поиске хорошего приближённого размещения предприятий, а также на вычислении оптимального набора цен для найденного размещения предприятий. Алгоритмы отличаются способом сравнения различных размещений предприятий, что в конечном итоге приводит к различным оценкам радиуса пороговой устойчивости. Численный эксперимент показал эффективность выбранного подхода как с точки зрения времени работы алгоритмов, так и качества получаемых решений. Табл. 4, ил. 2, библиогр. 24.

© М. Е. Водян, А. А. Панин, А. В. Плясунов, 2024

Ключевые слова: двухуровневая оптимизация, задача пороговой устойчивости, радиус пороговой устойчивости, размещение предприятий, дискриминационное ценообразование, спуск с чередующимися окрестностями.

Введение

При решении прикладных оптимизационных задач часто необходимо выбрать такое оптимальное или просто допустимое решение, которое приемлемо не только для текущих исходных данных, но и остаётся приемлемым при изменении этих данных в достаточно широком диапазоне. Например, не все исходные данные задачи могут быть определены точно; на качество решения могут влиять ошибки округления на этапе численного решения задачи; если приходится часто решать NP-трудную задачу большой размерности, то естественно использовать ранее полученное оптимальное или приемлемое допустимое решение для той же задачи, но уже с возмущёнными исходными данными.

В зависимости от типа доступной информации такие проблемы исследуются в ряде направлений: стохастическое программирование, оптимизация на основе нечёткого представления данных, робастная оптимизация, постоптимальный анализ чувствительности и устойчивости решений задач линейного и целочисленного программирования [1–7]. В [1] содержатся основные работы до 2000 г., связанные с устойчивостью конечномерных задач, включая идею радиуса устойчивости, идущую от В. К. Леонтьева. В [7] демонстрируется важность исследования проблем, связанных с устойчивостью, для приложений на основе многопараметрического программирования.

С каждым из классических подходов к анализу надёжности решений при различных возмущениях исходных данных связаны определённые недостатки. В основе моделей стохастического программирования лежит информация о вероятностном распределении случайных параметров, которая на практике зачастую недоступна. Разработка моделей на основе нечётких данных — существенно более сложное занятие, чем классическое математическое моделирование. Качество получаемых моделей существенно зависит от качества используемых экспертных оценок. Узкое место робастной оптимизации в том, что её применение ориентируется на учёт худших сценариев. С вычислительной точки зрения это приводит к решению значительно более сложных оптимизационных задач, чем исходная постановка. Некоторые полиномиально разрешимые задачи становятся NP-трудными в робастной постановке. Результаты, полученные в [4], демонстрируют очень высокую сложность многоэтапных задач стохастического программирования, которые оказываются PSPACE-трудными. Таким образом, есть проблемы с получением информации,

которая требуется в том или ином подходе при анализе устойчивости задачи [7].

Относительно недавно в этой области возникло новое направление исследований под названием пороговая устойчивость, свободное от недостатков перечисленных выше подходов, которое использует преобразование оптимизационной задачи на основе понятия радиуса устойчивости [8–12]. В работах [13–16] представлены результаты исследований, связанные с понятием радиуса устойчивости. Это направление исследований в конечном итоге привело к формулировкам, характерным для задач анализа пороговой устойчивости. Однако, впервые и независимо в явном виде эти формулировки появились при исследовании двухкритериальных задач размещения предприятий [8, 9], когда робастный подход к решению задач дискретной оптимизации [2] был реализован на основе пороговой модели, введённой в [9]. В этой работе при решении двухкритериальных задач размещения применялся один из известных методов — метод изменения ограничений (the ε-constraint method). В этом методе выбирается одна из целевых функций, оптимум которой ищется на множестве допустимых решений с учётом дополнительного порогового ограничения, образованного второй целевой функцией, ограниченной параметром $\varepsilon \geqslant 0$. Таким образом, в задаче пороговой устойчивости для заданного набора входных данных задачи размещения на максимум вместо максимизации дохода в новой формулировке будем максимизировать область входных данных (выбранный критерий), близких к начальному набору данных задачи, для которой ищется решение, приводящее к доходу не меньше заданного порога (второй критерий).

Итак, теперь с каждой оптимизационной задачей можно связать задачу пороговой устойчивости, в которой ищется максимальное значение параметра (радиус устойчивости), ограничивающего нормы вариаций, возмущающих исходные данные исследуемой задачи, и допустимое решение, которое при любых вариациях исходных данных допустимо в базовой постановке и удовлетворяет пороговому ограничению.

В настоящей работе продолжается исследование пороговой устойчивости двухуровневых задач, начатое в [17, 18]. Задачи такого типа образуют новый класс двухуровневых задач, для которых не известны ни точные, ни приближённые методы решения. Исследование такого класса задач с алгоритмической точки зрения несомненно является важной теоретической проблемой, поскольку при этом разрабатываются новые точные и приближённые методы для решения оптимизационных задач. Описание современного состояния дел в области двухуровневой оптимизации можно найти в обзорах [19–21].

В этой статье впервые исследована пороговая устойчивость решения двухуровневой задачи с медианным типом размещения предприятий

и дискриминационным ценообразованием. Основным вкладом является подход к разработке алгоритмов решения задачи пороговой устойчивости на основе методов решения исходной задачи и её подзадач и два быстрых приближённых алгоритма, полученных с помощью данного подхода. Оба алгоритма объединяет общая идея генерации хорошего приближения к радиусу устойчивости. С помощью VND-эвристики находится в определённом смысле наилучшее размещение предприятий, которое и используется для построения приближения к радиусу устойчивости [21–23]. Алгоритмы отличаются характеристиками используемых окрестностей. В первом из алгоритмов используется окрестность, в которой текущее размещение предприятий лучше соседнего размещения, если доход, полученный в задаче ценообразования, связанного с текущим размещением, больше, чем в задаче ценообразования соседа. В другом алгоритме используется окрестность, в которой текущее размещение предприятий лучше соседнего размещения, если оценка радиуса пороговой устойчивости, полученного с помощью вектора цен, соответствующего текущему размещению, больше оценки радиуса пороговой устойчивости, полученной с помощью задачи ценообразования соседа.

В разд. 1 вводятся основные определения и формулируется задача пороговой устойчивости для двухуровневой задачи размещения производства и дискриминационного ценообразования в виде двухуровневой модели нелинейного программирования. Здесь же содержится эквивалентное представление задачи пороговой устойчивости в виде одноуровневой линейной задачи смешанного целочисленного программирования. В разд. 2 содержатся результаты о вычислительной сложности задачи пороговой устойчивости. Разд. 3 содержит описание вспомогательных алгоритмов, с помощью которых разрабатываются два приближённых алгоритма на основе VND-эвристики. Разд. 4 содержит результаты вычислительных экспериментов на исходных данных из библиотеки тестовых задач «Дискретные задачи размещения», а также на случайных входных данных. Предлагаемые алгоритмы сравниваются между собой и точным методом из библиотеки Gurobi. В заключении обсуждаются полученные результаты и направление дальнейших исследований.

1. Постановка задачи пороговой устойчивости

Прежде чем сформулировать определение пороговой устойчивости для двухуровневых задач, приведём определение устойчивости, восходящее к работам Леонтьева, Гордеева и ряда других исследователей. Имеется оптимизационная задача P (с критерием максимизации) и некоторый её вход X. Обозначим через $F^*(X)$ $(F(X))$ множество оптимальных решений (множество допустимых решений) задачи P для входа X.

Пусть $\Delta(\rho) = \{\delta \mid \|\delta\| \leqslant \rho\}$ ($\Delta^{=}(\rho) = \{\delta \mid \|\delta\| = \rho\}$) — множество вариаций входа X, где $\rho > 0$. Оптимальное решение $Y^* \in F^*(X)$ называется *устойчивым,* если непусто множество

$$\Gamma^P(X, Y^*) = \{\rho > 0 \mid Y^* \in F^*(X + \delta) \text{ для любого } \delta \in \Delta(\rho)\}.$$

Величина $\sup \Gamma^P(X, Y^*)$ называется *радиусом устойчивости* оптимального решения $Y^* \in F^*(X)$.

Понятие устойчивости, которое исследуется в данной работе, получается путём релаксации условия, что решение $Y^* \in F^*(X)$ остаётся оптимальным при изменении входа X, и замены его условием, что решение остаётся допустимым при изменении входа X и значение целевой функции на нём не меньше заданного порога V. Пусть $f_P(X, Y)$ — целевая функция задачи P с входом X, где Y — произвольное допустимое решение. Решение $Y \in F(X)$ называется *устойчивым относительно порога* V, если непусто множество

$$\Gamma^P(X, Y, V) = \{\rho \geqslant 0 \mid Y \in F(X + \delta), \\ f_P(X + \delta, Y) \geqslant V \text{ для любого } \delta \in \Delta(\rho)\}.$$

Величина $\rho(X, Y, V) = \sup \Gamma^P(X, Y, V)$ называется *радиусом устойчивости* допустимого решения $Y \in F(X)$ *относительно порога* V.

В общем случае задача пороговой устойчивости для входа X и порога V формулируется следующим образом.

Задача 1. *Необходимо найти радиус устойчивости* $\rho(X, \widetilde{Y}, V)$ *и допустимое решение* $\widetilde{Y}$ *исходной задачи, устойчивое относительно порога* V:

$$\rho(X, Y, V) \to \max_{Y \in F(X)}.$$

В двухуровневых задачах в определении как допустимого, так и оптимального решений часть переменных является оптимальным решением задачи нижнего уровня, поэтому в дополнение к задаче 1 сформулируем ещё одну постановку, которая учитывает структуру допустимых и оптимальных решений двухуровневых задач. Разделим переменные двухуровневой задачи на две группы (Y_l, Y_f), где Y_l — переменные верхнего уровня, Y_f — переменные нижнего уровня. Пусть $F_f^*(Y_l)$ — множество оптимальных решений задачи нижнего уровня, а $F(X)|_l = \{Y_l \mid \exists\, Y_f \in F_f^*(Y_l)\colon (Y_l, Y_f) \in F(X)\}$ — проекция множества допустимых решений $F(X)$ двухуровневой задачи P на пространство переменных верхнего уровня. Полудопустимое решение $Y_l \in F(X)|_l$ называется *устойчивым относительно порога* V, если непусто множество

$$\Gamma(X, Y_l, V) = \{\rho \geqslant 0 \mid \text{ для любого } \delta \in \Delta(\rho) \\ \exists\, Y_f(\delta) \in F_f^*(Y_l)\colon (Y_l, Y_f(\delta)) \in F(X + \delta),\ f(X + \delta, Y) \geqslant V\}.$$

Величина $\rho(X, Y_l, V) = \sup \Gamma(X, Y_l, V)$ называется *радиусом устойчивости относительно порога* V или *пороговым радиусом устойчивости полудопустимого решения* $Y_l \in F(X)|_l$. В определении радиуса устойчивости заменяем требование допустимости решения при вариации δ исходных данных X условием существования оптимального решения $Y_f(\delta)$ задачи нижнего уровня с параметрами Y_l такого, что пара $(Y_l, Y_f(\delta))$ является допустимым решением двухуровневой задачи, возмущённой вариацией δ, и выполняется соответствующее пороговое ограничение.

В общем случае задача пороговой устойчивости двухуровневой задачи для входа X и порога V формулируется следующим образом.

Задача 2. *Найти радиус устойчивости относительно входа* X *и вектор переменных верхнего уровня* Y_l, *устойчивый относительно порога* V:

$$\rho(X, Y_l, V) \to \max_{Y_l \in F(X)|_u}.$$

Приведём содержательную постановку базовой задачи с медианным типом размещения предприятий и дискриминационным ценообразованием, пороговая устойчивость которой исследуется далее. Сформулируем её в виде игры Штакельберга «лидер — последователи». В качестве лидера выступает производитель, который размещает r предприятий и формирует цены на каждом из них. В качестве последователей — потребители, каждый из которых выбирает то предприятие, на котором его суммарные затраты на покупку и транспортировку товара минимальны, и совершает покупку только в том случае, когда эти затраты не превышают его бюджета. Требуется выбрать такое размещение предприятий и такие цены, при которых доход производителя максимален. Далее рассматривается оптимистическая постановка двухуровневой задачи. Для этого необходимо ввести следующее соглашение. Если у потребителя есть несколько предприятий с одинаковой минимальной суммой платежей, то он выберет предприятие с минимальными транспортными затратами. В статье рассматривается дискриминационное ценообразование (discriminatory pricing), когда на каждом предприятии для каждого потребителя устанавливается своя цена.

Задача пороговой устойчивости отличается от базовой постановки тем, что заранее задан доход производителя, определяющий пороговое ограничение, и наличием неопределённости в бюджетах потребителей, максимизируя которые сможем получить максимально возможное отклонение от ожидаемых (данных) бюджетов.

Для того чтобы сформулировать математическую модель задачи пороговой устойчивости, введём следующие обозначения и переменные. Обозначения:

- $I = \{1, \ldots, n\}$ — множество возможных мест открытия предприятий;
- $J = \{1, \ldots, m\}$ — множество потребителей;
- $r \in Z^+$ — число размещаемых предприятий;
- $b_j \in \mathbb{Z}^+ \cup \{0\}$ — бюджет потребителя j;
- $c_{ij} \in \mathbb{Z}^+ \cup \{0\}$ — транспортные затраты потребителя j, если он обслуживается на предприятии i;
- $V \in \mathbb{Z}^+$ — доход производителя.

Переменные:

- $\rho \in \mathbb{Q}^+ \cup \{0\}$ — радиус пороговой устойчивости;
- $p_{ij} \in \mathbb{Q}^+ \cup \{0\}$ — цена товара на предприятии i для потребителя j;
- $x_{ij} = \begin{cases} 1, & \text{если потребитель } j \text{ обслуживается на предприятии } i, \\ 0 & \text{иначе;} \end{cases}$
- $y_i = \begin{cases} 1, & \text{если предприятие } i \text{ открыто,} \\ 0 & \text{иначе.} \end{cases}$

Двухуровневая смешанно целочисленная квадратичная математическая модель задачи пороговой устойчивости имеет вид

$$\rho \to \max_{p,y,x,\rho}, \tag{1}$$

$$\sum_{i\in I}\sum_{j\in J} p_{ij}x_{ij} \geqslant V, \tag{2}$$

$$\sum_{i\in I} y_i = r, \tag{3}$$

$$y_i \in \{0,1\}, \quad p_{ij}, \rho \in \mathbb{Q}^+ \cup \{0\}, \quad x \in \mathcal{F}^*(p,y,\rho), \quad i \in I,\ j \in J, \tag{4}$$

$\mathcal{F}^*(p,y,\rho)$ — множество оптимальных решений задачи нижнего уровня:

$$\sum_{i\in I}\sum_{j\in J}(b_j - c_{ij} - \rho - p_{ij})x_{ij} \to \max_x, \tag{5}$$

$$\sum_{i\in I} x_{ij} \leqslant 1, \quad j \in J, \tag{6}$$

$$x_{ij} \leqslant y_i, \quad i \in I,\ j \in J, \tag{7}$$

$$x_{ij} \in \{0,1\}, \quad i \in I,\ j \in J. \tag{8}$$

Максимизируя целевую функцию (1) на верхнем уровне, получим максимально возможное отклонение от ожидаемых (данных) бюджетов. Ограничение (2) — пороговое ограничение, гарантирующее, что доход производителя не меньше заданного. Условие (3) требует, чтобы было открыто ровно r предприятий. Ограничения (4) определяют тип переменных верхнего уровня и фиксируют фундаментальное свойство двухуровневых задач: переменные нижнего уровня x являются его оптимальным

решением. Целевая функция нижнего уровня (5) описывает стратегию каждого потребителя — в максимальной степени экономить свой бюджет, а ограничения (6)–(8) гарантируют, что каждый потребитель обслуживается не более чем одним предприятием производителя, которое должно быть открыто. Также из этих ограничений и определения целевой функции следует, что покупка совершается в том случае, когда это позволяет бюджет потребителя. В базовой постановке отсутствует целевая функция (1), а максимизируется доход производителя $\sum\limits_{i\in I}\sum\limits_{j\in J} p_{ij}x_{ij}$.

Задачу (1)–(8) можно свести к одноуровневой задаче и линеаризовать, введя дополнительные переменные $z_{ij} = p_{ij}x_{ij}$, $r_{ij} = \rho x_{ij}$ и ограничения:

$$\rho \to \max_{x,p,y,\rho,z,r},$$

$$\sum_{i\in I}\sum_{j\in J} z_{ij} \geqslant V,$$

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

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

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

$$\sum_{i\in I}((b_j - c_{ij})x_{ij} - r_{ij} - z_{ij}) \geqslant 0, \quad j \in J, \tag{9}$$

$$\sum_{i\in I}(c_{ij}x_{ij} + z_{ij}) \leqslant c_{kj} + p_{kj}, \quad k \in I,\ j \in J, \tag{10}$$

$$x_{ij}, y_i \in \{0,1\}, \quad \rho, p_{ij} \in \mathbb{Q}^+ \cup \{0\}, \quad i \in I,\ j \in J,$$

$$(1 - x_{ij})W + z_{ij} \geqslant p_{ij}, \tag{11}$$

$$(1 - x_{ij})W + p_{ij} \geqslant z_{ij}, \tag{12}$$

$$z_{ij} \leqslant x_{ij}W, \tag{13}$$

$$z_{ij} \geqslant 0, \tag{14}$$

$$(1 - x_{ij})W + r_{ij} \geqslant \rho, \tag{15}$$

$$(1 - x_{ij})W + \rho \geqslant r_{ij}, \tag{16}$$

$$r_{ij} \leqslant x_{ij}W, \tag{17}$$

$$r_{ij} \geqslant 0, \tag{18}$$

где W — положительная константа и $W \geqslant \max\limits_{i\in I,\, j\in J}\{b_j - c_{ij}\}$.

Условия (9) и (10) гарантируют, что любой потребитель обслуживается только в том случае, если ему позволяет бюджет и его суммарные затраты минимальны. Группа ограничений (11)–(14) гарантирует, что если

j-й потребитель обслуживается на i-м предприятии, то цена на продукт для него равна p_{ij}, а если он не обслуживается, то индикатором этого будет значение z_{ij}, равное нулю. Таким образом, если $x_{ij} = 1$, то $z_{ij} = p_{ij}$, а если $x_{ij} = 0$, то $z_{ij} = 0$. Аналогично для группы ограничений (15)–(18).

2. Вычислительная сложность задачи пороговой устойчивости (1)–(8)

Напомним обозначения, используемые в теории сложности для описания полиномиальной иерархии классов сложности [20]. Первые два основных класса задач распознавания (P и NP) определяются с помощью детерминированных и недетерминированных машин Тьюринга [20]. Класс P содержит задачи распознавания, решаемые за полиномиальное время на детерминированных машинах Тьюринга, а класс NP — это класс задач распознавания, решаемых за полиномиальное время на недетерминированных машинах Тьюринга. Третий основной класс co-NP состоит из задач распознавания, дополнения которых принадлежат NP. Эти классы образуют первый уровень полиномиальной иерархии. В [18] было показано, что базовая задача с медианным типом размещения предприятий и дискриминационным ценообразованием NP-трудна в сильном смысле.

Обозначим через D_ρ и D стандартную задачу распознавания задачи (1)–(8) и стандартную задачу распознавания базовой задачи соответственно.

Теорема 1. *Задача D_ρ* NP*-полна в сильном смысле.*

Доказательство. Покажем, что D_ρ принадлежит классу NP. Предположим, что для некоторого целого числа $\widehat{\rho}$ существует такое допустимое решение (ρ, y, p, x), что $\rho \geqslant \widehat{\rho}$. Можно считать, что $\rho = \widehat{\rho}$. Действительно, если $\rho > \widehat{\rho}$, то при уменьшении ρ до $\widehat{\rho}$ при фиксированных (y, p) будет изменяться только оптимальное решение x задачи нижнего уровня, так как могут появиться клиенты, бюджеты которых увеличатся, и они будут обслужены на открытых предприятиях. Доход лидера при этом только возрастёт, т. е. пороговое ограничение не будет нарушено. Учитывая, что при заданных y и $\widehat{\rho}$ базовая задача полиномиально разрешима, также можно считать, что вектор цен p оптимален. Таким образом, существование для некоторого целого числа $\widehat{\rho}$ такого допустимого решения (ρ, y, p, x), что $\rho \geqslant \widehat{\rho}$, эквивалентно существованию такого размещения предприятий y, при котором в базовой задаче с бюджетами $b_j - \widehat{\rho}$, $j \in J$, и множеством открытых предприятий $\{i \mid y_i = 1\}$ доход лидера больше заданного порога V. Теперь заметим, что для заданного

целого числа $\widehat{\rho}$ требуемый вектор y может быть найден за недетерминированное полиномиальное время, если в задаче D_ρ ответ «да». Отсюда следует, что задача D_ρ принадлежит классу NP.

Покажем, что задача D_ρ полна в NP. Этот результат следует из полиномиальной сводимости задачи D к задаче D_ρ. Действительно, в задаче D для заданного порога V надо найти допустимое решение (y, p, x), которое приносит лидеру доход, не меньший порога. В качестве исходных данных задачи D_ρ возьмём $\widehat{\rho} = 0$ и исходные данные задачи D. Из результатов, полученных в [3], следует, что задача D NP-полна в сильном смысле. Теорема 1 доказана.

Теорема 2. *Для задачи* (1)–(8) *не существует детерминированных полиномиальных приближённых алгоритмов с абсолютной или относительной оценкой уклонения от оптимального решения при условии, что* $\mathrm{P} \neq \mathrm{NP}$.

Доказательство. Предположим, что существует детерминированный приближённый полиномиальный алгоритм для задачи (1)–(8). Покажем, что тогда задача D полиномиально разрешима. Рассмотрим произвольный вход данной задачи с порогом V. Применим приближённый алгоритм к входу задачи (1)–(8), который получается из входа задачи D. Если в задаче D для порога V ответ «да», то алгоритм выдаст некоторое приближённое допустимое решение (ρ, y, p, x) задачи (1)–(8). Если $\rho > 0$, то, рассуждая как в доказательстве теоремы 1, получим её допустимое решение $(0, y, p, \widehat{x})$, которое является подтверждением, что в задаче D для текущего входа ответ «да». Таким образом, получен полиномиальный алгоритм для задачи D, что противоречит условию $\mathrm{P} \neq \mathrm{NP}$. Теорема 2 доказана.

3. Алгоритмы

Для нахождения оптимального размещения и радиуса устойчивости предлагаются два алгоритма, основанных на методе спуска с чередующимися окрестностями (метаэвристика VND), который выполняет некоторое количество итераций с разными окрестностями до тех пор, пока не будет получен локальный оптимум относительно всех используемых окрестностей. Алгоритмы различаются выбором целевой функции и критерием оптимизации при сравнении двух размещений предприятий. Далее приводится описание нескольких вспомогательных алгоритмов.

3.1. Вспомогательные алгоритмы. В этом пункте оптимальную цену для j-го потребителя будем обозначать через p_j. Она всегда достигается на предприятии с минимальными транспортными затратами, поэтому можем убрать индекс i из обозначения p_{ij}.

Для заданного размещения y оптимальную цену для каждого потребителя можно посчитать при помощи алгоритма PC (price calculation).

Алгоритм 1. PC(y)

Вход: размещение y.
Выход: вектор цен $p = (p_1, \ldots, p_m)$, где p_j — цена для потребителя j.

1: $I(y) = \{i \mid y_i = 1\}$; $j \leftarrow 1$;
2: **if** $j > m$ **then** STOP;
3: **if** $b_j > \min\limits_{i \in I(y)} \{c_{ij}\}$ **then** $p_j \leftarrow b_j - \min\limits_{i \in I(y)} \{c_{ij}\}$;
4: **else** $p_j \leftarrow 0$; $j \leftarrow j + 1$; **goto** 1.

Таким образом, для каждого потребителя находится то предприятие, на котором затраты на транспортировку минимальны, а если бюджет потребителя превосходит эти затраты, то цена устанавливается как разность между бюджетом потребителя и минимальными транспортными затратами, тем самым принося максимальный доход производителю. Если бюджет j-го потребителя меньше минимальных транспортных затрат, то ни на одном из открытых предприятий клиент не может быть обслужен, принося положительную прибыль; индикатором этого является значение переменной p_j, равное нулю. Данный алгоритм решает задачу ценообразования для заданного размещения за полиномиальное время. Временна́я сложность алгоритма $O(mr)$.

Следующий алгоритм PR (price recalculation) понадобится для пересчёта вектора цен при вычислении радиуса пороговой устойчивости.

Алгоритм 2. PR(p, Δ)

Вход: вектор цен $p = (p_1, \ldots, p_m)$, некоторое значение Δ.
Выход: вектор цен $\overline{p} = (\overline{p}_1, \ldots, \overline{p}_m)$.

1: $j \leftarrow 1$;
2: **if** $j > m$ **then** STOP;
3: **if** $p_j > \Delta$ **then** $\overline{p}_j \leftarrow p_j - \Delta$;
4: **else** $\overline{p}_j \leftarrow 0$; $j \leftarrow j + 1$; **goto** 2.

Данный алгоритм пересчитывает цену для каждого потребителя. Если можно уменьшить цену на заданную величину и прибыль от клиента останется положительной, то уменьшаем цену, иначе перестаём обслуживать клиента. Временна́я сложность алгоритма $O(m)$.

Радиус пороговой устойчивости для заданного размещения y можно найти при помощи алгоритма RC (radius calculation).

Алгоритм 3. RC(y)

Вход: размещение y.
Выход: радиус устойчивости ρ.

1: $\rho \leftarrow 0$; $p \leftarrow \mathrm{PC}(y)$;
2: **if** $\sum\limits_{j \in J} p_j < V$ **then** STOP;
3: $d(p) \leftarrow \sum\limits_{j \in J} p_j - V$;
4: **if** $d(p) = 0$ **then** STOP;
5: $c(p) = |\{j \in J \mid p_j \neq 0\}|$; $\rho \leftarrow \rho + d(p)/c(p)$; $p \leftarrow \mathrm{PR}(p, d(p)/c(p))$;
6: **goto** 3.

Если алгоритм остановился на шаге 2, т. е. максимальный доход производителя на размещении y меньше порогового ограничения (ожидаемого дохода), то на данном размещении невозможно найти допустимого значения радиуса пороговой устойчивости. На шаге 3 расчитываем сверхприбыль $d(p)$. Если она положительная, то можем увеличить радиус пороговой устойчивости. На шаге 5 вычисляем число обслуживаемых клиентов и увеличиваем радиус устойчивости на величину, полученную делением сверхприбыли на число всех обслуживаемых клиентов. Таким образом, если на данном размещении есть допустимое решение, то алгоритм останавливается только тогда, когда сверхприбыль $d(p)$ будет нулевая. Временна́я сложность алгоритма RC в худшем случае $O(mr + m^2)$.

Утверждение 1. *Если для размещения предприятий y оптимальный набор цен и оптимальное назначение потребителей удовлетворяют пороговому ограничению, то радиус устойчивости* RC(y) *максимальный для данного размещения y.*

Доказательство. Предположим, напротив, что существует допустимое решение (ρ, y, p) и $\rho > \overline{\rho} = \mathrm{RC}(y)$. Каждая компонента $\overline{p}_j$ вектора цен $\overline{p}$, полученного алгоритмом RC, является максимально возможной ценой обслуживания j-го потребителя при радиусе пороговой устойчивости $\overline{\rho}$. Из шага 1 алгоритма RC следует, что $\sum\limits_{j \in J} \overline{p}_j = V$. Так как $\rho > \overline{\rho}$, имеем $\overline{p}_j > p_j$. В результате получаем $V = \sum\limits_{j \in J} \overline{p}_j < \sum\limits_{j \in J} p_j$. Значит, решение (ρ, y, p, x) не допустимо; противоречие. Утверждение 1 доказано.

3.2. Критерии выбора. Используем два критерия для попарного сравнения мест расположения объектов. Первый критерий использует целевую функцию исходной задачи. Второй критерий при оценке двух вариантов размещения предприятий основывается на сравнении соответствующих радиусов пороговой устойчивости.

Пусть имеется два допустимых размещения y и $\widetilde{y}$ и соответствующие им векторы цен p и $\widetilde{p}$, полученные при помощи алгоритма PC.

Первый критерий. Если $\sum\limits_{j\in J} p_j > \sum\limits_{j\in J} \widetilde{p}_j$, то y и соответствующий вектор цен p лучше, чем $\widetilde{y}$ и $\widetilde{p}$.

Второй критерий. Пусть $\rho = \mathrm{RC}(y)$, $\widetilde{\rho} = \mathrm{RC}(\widetilde{y})$. Если $\rho > \widetilde{\rho}$, то y и соответствующий вектор цен p лучше, чем $\widetilde{y}$ и $\widetilde{p}$.

Следует отметить, что не всегда там, где доход больше, больше и радиус пороговой устойчивости. В доказательство этого рассмотрим

Пример 1. Пусть общее число предприятий, число клиентов и число открываемых предприятий равны n, $m > 2$ и r соответственно. Задано пороговое ограничение $V = k \in Z^+$, и есть два допустимых размещения y_f и y_s таких, что открытые объекты из y_f не входят в y_s. Пусть на первом размещении алгоритм PC возвращает вектор цен $p^f = \mathrm{PC}(y_f) = (k, k, \ldots, k)$, $\left|\left\{j \mid p_j^f \neq 0\right\}\right| = m$, а на втором — $p^s = \mathrm{PC}(y_s) = (k(m-1), 0, \ldots, 0)$, $\left|\left\{j \mid p_j^s \neq 0\right\}\right| = 1$. Тогда доход производителя на размещении y_f равен km, а радиус пороговой устойчивости равен $k\frac{m-1}{m} < k$. На размещении y_s доход производителя равен $k(m-1) < km$, а радиус пороговой устойчивости равен $k(m-2) \geqslant k$.

Утверждение 2. *Пусть (y^*, p^*, x^*) — оптимальное решение исходной задачи, $\rho^* = \mathrm{RC}(y^*)$, $\widetilde{p}^* = \mathrm{PR}(p^*, \rho^*)$. Решение $(\rho^*, y^*, \widetilde{p}^*)$ будет оптимальным в задаче пороговой устойчивости тогда и только тогда, когда $\sum\limits_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) \geqslant 0$ для любого допустимого решения (y, p, x) исходной задачи и $\widetilde{p} = \mathrm{PR}(p, \rho^*)$.*

Доказательство. Предположим, напротив, что $\sum\limits_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) < 0$ для некоторого решения (y, p, x) исходной задачи. Тогда

$$\sum_{j=1}^{m} (\widetilde{p}_j^* - \widetilde{p}_j) = \sum_{j=1}^{m} \widetilde{p}_j^* - \sum_{j=1}^{m} \widetilde{p}_j = V - \sum_{j=1}^{m} \widetilde{p}_j < 0,$$

откуда $\sum\limits_{j=1}^{m} \widetilde{p}_j - V > 0$. Следовательно, на шаге 2 алгоритма RC можем увеличить радиус пороговой устойчивости. Утверждение 2 доказано.

Максимальный радиус пороговой устойчивости будет меньше при неоптимальном наборе цен. Тогда имеет место

Следствие 1. *Пусть (y^*, p^*, x^*) — оптимальное решение исходной задачи, $\rho^* = \mathrm{RC}(y^*)$, $\widetilde{p}^* = \mathrm{PR}(p^*, \rho^*)$. Решение $(\rho^*, y^*, \widetilde{p}^*)$ будет оптимальным в задаче пороговой устойчивости тогда и только тогда, когда*

$\sum\limits_{j=1}^{m}(\widetilde{p}_j^* - \widetilde{p}_j) \geqslant 0$ *для любого допустимого размещения предприятий* y, $\widetilde{p} = \mathrm{PR}(p, \rho^*)$ *и набора цен* p, *оптимального в исходной задаче для размещения* y.

3.3. Основной алгоритм. В качестве основного алгоритма использована метаэвристика VND (variable neighborhood descent). Приведём описание окрестностей, по которым будет производиться спуск. Обозначим через $d(x, y)$ расстояние Хэмминга между векторами x и y, а через $wt(x)$ — вес Хэмминга вектора x. Определим окрестность k-Swap(y) как множество, содержащее все допустимые размещения такие, что относительно размещения y было закрыто k предприятий и открыто k новых: k-Swap$(y) = \{\widetilde{y} \mid wt(\widetilde{y}) = r,\ d(y, \widetilde{y}) = 2k\}$. Вес Хэмминга размещения $\widetilde{y} \in k$-Swap(y), равный r, гарантирует, что число открываемых предприятий будет равно r.

Определим процедуру встряски k-Shake(y): последовательно просматриваем окрестности 2-Swap(y), 3-Swap(y), ..., k-Swap(y), пока не будет найдено размещение лучше y. Если такое размещение найдено, то на выходе имеем новое размещение, лучшее y, иначе сохраняем размещение y.

Алгоритм 4. VND

Вход: $I_{\max}$ — максимальное число итераций алгоритма, k — параметр процедуры встряски k-Shake.

Выход: Наилучшее размещение y.

1: $I \leftarrow 0$; $y = \mathrm{rand}\{0,1\}^n$ — случайный булев вектор;
2: Применить локальный поиск относительно 1-Swap(y), y^* — локальный оптимум;
3: $I \leftarrow I + 1$; $y \leftarrow k$-Shake(y^*);
4: **if** $y = y^*$ или $I > I_{\max}$ **then** STOP;
5: **else goto** 1.

В результате работы алгоритма VND получим некоторое размещение y^*. При помощи алгоритма RC найдём радиус пороговой устойчивости для размещения y^*. В зависимости от критерия выбора получаются различные размещения предприятий на выходе. Следовательно, возникает два алгоритма, поведение которых исследуется в ходе численного эксперимента.

4. Численный эксперимент

Тестирование алгоритмов проведено на компьютере с процессором Intel(R) Core(TM) i7-8750H и 16 ГБ оперативной памяти. Алгоритм VND сравнивался с решателем Gurobi. Для сравнения использованы входные

данные из библиотеки «Discrete Location Problems» (табл. 1, 2) и входные данные, порождённые случайным образом с равномерным распределением (табл. 3, 5).

Если I — вход для исходной задачи, то $I \cup \{V\}$ — вход для задачи пороговой устойчивости с пороговым ограничением V (минимальным ожидаемым доходом). Для каждого примера решаем исходную задачу и определяем максимальный доход производителя. В качестве порогового ограничения V выбираем некоторую часть найденного максимального дохода, каждую из которых назовём уровнем со своим порядковым номером. Таким образом для одного набора входных данных исходной задачи имеем несколько наборов входных данных для задачи пороговой устойчивости, которые отличаются друг от друга пороговым ограничением.

Например, в табл. 1, 2 максимальный доход производителя разбит на 14 равных частей, и в качестве порогового ограничения V взяты 1, 4, 7, 10 и 13 частей, т. е. определено пять уровней. В табл. 1, 2 первый столбец означает номер примера, столбцы level, % — номер уровня и рассматриваемую часть максимального дохода производителя. Столбцы Gurobi, VND_1 и VND_2 содержат характеристики решений Gurobi и алгоритма VND с первым и вторым критерием лучшего размещения соответственно: time — время поиска решения; opt — оптимальное решение Gurobi; best, GAP — лучшее решение алгоритма VND и его относительное отклонение от оптимума, рассчитанное по формуле $\text{GAP} = \frac{\text{opt}-\text{best}}{\text{opt}}$. В столбце $\text{RC}(\overline{y})$ указан радиус пороговой устойчивости, полученный на оптимальном размещении $\overline{y}$ исходной задачи. В табл. 5 столбец dim содержит размеры входа n, m и r — число предприятий, число клиентов и число открываемых предприятий соответственно.

Таблица 1

Результаты численного эксперимента $n = 40$, $m = 100$, $r = 5$

№	V		Gurobi		VND_1			VND_2			$\text{RC}(\overline{y})$
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	7,12	8,13	64,05	0,02	64,05	0	0,02	64,05	0	63,33
	2	28,56	16,91	38,44	0,03	38,44	0	0,03	38,44	0	38,29
	3	49,97	31,85	24,58	0,03	24,55	0,0012	0,03	24,58	0	24,55
	4	71,41	20,23	13,08	0,03	13,08	0	0,03	13,08	0	13,08
	5	92,85	7,89	3,16	0,04	3,16	0	0,03	3,16	0	3,16
2	1	7,14	9,50	65,17	0,02	64,85	0,0049	0,02	65,17	0	64,0
	2	28,56	18,33	40,7	0,02	40,7	0	0,02	40,7	0	40,33
	3	50,00	36,23	26,1	0,04	26,1	0	0,03	26,1	0	25,71
	4	71,42	18,24	13,65	0,03	13,52	0,0095	0,03	13,65	0	13,52
	5	92,83	9,16	3,21	0,03	3,21	0	0,03	3,21	0	3,21
3	1	7,13	6,09	63,21	0,02	63,0	0,0033	0,02	63,21	0	62,69
	2	28,57	17,77	36,79	0,03	36,71	0,0021	0,03	36,79	0	36,31
	3	50,00	22,92	23,54	0,03	23,44	0,0042	0,03	23,54	0	23,44
	4	71,40	18,68	12,58	0,04	12,58	0	0,04	12,58	0	12,58
	5	92,84	9,18	3,0	0,04	3,0	0	0,03	3,0	0	3,0

4	1	7,12	5,70	61,43	0,03	60,65	0,0126	0,02	61,43	0	58,44
	2	28,55	12,69	36,64	0,03	36,57	0,0019	0,02	36,64	0	34,42
	3	49,98	15,43	22,04	0,03	22,04	0	0,02	22,04	0	20,98
	4	71,41	10,01	11,49	0,03	11,48	0,0008	0,03	11,48	0,0008	11,3
	5	92,84	6,68	2,68	0,03	2,63	0,0186	0,03	2,68	0	2,68
5	1	7,13	10,05	68,67	0,03	68,56	0,0016	0,03	68,67	0	64,41
	2	28,56	19,82	43,58	0,04	43,58	0	0,03	43,58	0	41,56
	3	49,99	41,12	27,25	0,03	26,89	0,0132	0,03	27,25	0	26,81
	4	71,42	30,46	14,51	0,03	14,51	0	0,03	14,51	0	14,45
	5	92,85	13,31	3,44	0,04	3,42	0,0058	0,05	3,44	0	3,42
6	1	7,12	6,33	64,47	0,02	63,47	0,0155	0,02	64,47	0	61,94
	2	28,56	17,34	40,11	0,04	40,11	0	0,03	40,11	0	38,76
	3	50,00	25,45	24,63	0,04	24,48	0,006090	0,03	24,63	0	24,48
	4	71,41	14,88	13,29	0,05	13,29	0	0,03	13,29	0	13,29
	5	92,84	7,99	3,2	0,06	3,2	0	0,04	3,2	0	3,2
7	1	7,12	8,98	65,58	0,02	65,58	0	0,02	65,58	0	63,16
	2	28,56	19,27	42,47	0,04	42,42	0,0011	0,03	42,47	0	41,69
	3	50,00	31,80	27,17	0,03	27,17	0	0,02	27,17	0	26,6
	4	71,41	19,72	13,95	0,03	13,95	0	0,03	13,95	0	13,85
	5	92,85	9,13	3,23	0,04	3,23	0	0,04	3,23	0	3,23
8	1	7,13	9,71	65,35	0,03	65,33	0,0003	0,03	65,35	0	62,45
	2	28,57	16,46	40,34	0,03	40,34	0	0,02	40,34	0	39,77
	3	50,00	41,41	25,11	0,05	25,11	0	0,03	25,11	0	25,0
	4	71,40	25,25	13,28	0,03	13,28	0	0,03	13,28	0	13,2
	5	92,84	11,26	3,26	0,03	3,19	0,0214	0,03	3,26	0	3,19
9	1	7,12	7,72	65,53	0,03	65,47	0,0009	0,02	65,53	0	62,58
	2	28,55	12,79	41,23	0,03	41,23	0	0,03	41,23	0	40,59
	3	49,98	40,21	25,45	0,03	25,45	0	0,03	25,45	0	25,35
	4	71,41	17,47	12,73	0,04	12,73	0	0,03	12,73	0	12,73
	5	92,85	7,63	2,96	0,06	2,96	0	0,06	2,96	0	2,96
10	1	7,14	8,76	66,56	0,03	65,76	0,0120	0,02	66,56	0	63,19
	2	28,55	13,82	41,83	0,03	41,83	0	0,02	41,83	0	39,93
	3	49,99	29,60	25,42	0,03	25,31	0,0043	0,03	25,42	0	25,0
	4	71,42	15,18	13,29	0,03	13,2	0,0067	0,03	13,29	0	13,2
	5	92,84	9,10	3,15	0,04	3,15	0	0,03	3,15	0	3,15

Результат, отражённый в табл. 1, получен на небольшой размерности $n = 40$. VND_1 находит оптимальное решение в 28 из 50 случаев, VND_2 выдаёт оптимум в 49 из 50 примеров. Время работы обоих алгоритмов примерно одинаково и в среднем более чем в 500 раз меньше времени работы решателя.

Увеличим число открываемых предприятий на 60 и посмотрим на следующий результат.

Таблица 2

Результаты численного эксперимента $n = 100$, $m = 100$, $r = 5$

№	V		Gurobi		VND_1			VND_2			$\mathrm{RC}(\overline{y})$
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	7,14	35,07	66,89	0,14	66,48	0,0061	0,14	66,89	0	63,19
	2	28,55	81,63	42,18	0,16	42,18	0	0,15	42,18	0	39,93
	3	49,99	143,16	25,51	0,27	25,51	0	0,16	25,51	0	25,0
	4	71,42	115,21	13,29	0,19	13,26	0,0022	0,31	13,29	0	13,2
	5	92,84	68,12	3,18	0,19	3,15	0,0063	0,17	3,18	0	3,15

2	1	7,14	37,77	64,24	0,13	63,19	0,0163	0,12	64,24	0	60,22
	2	28,55	129,49	39,11	0,16	39,02	0,0023	0,15	39,11	0	38,74
	3	49,99	275,46	24,55	0,19	24,31	0,0097	0,23	24,55	0	24,28
	4	71,42	109,75	13,12	0,19	13,0	0,0091	0,28	13,12	0	13,0
	5	92,84	226,93	3,11	0,20	3,11	0	0,21	3,06	0,0160	3,11
3	1	7,14	34,08	69,18	0,14	68,85	0,0047	0,12	69,18	0	64,21
	2	28,57	77,52	41,75	0,15	40,82	0,0222	0,14	41,75	0	40,1
	3	50,00	212,27	26,09	0,17	26,09	0	0,17	26,09	0	25,38
	4	71,43	80,97	14,13	0,18	13,48	0,0460	0,20	14,13	0	13,66
	5	92,86	66,34	3,19	0,19	3,13	0,0188	0,20	3,14	0,0156	3,13
4	1	7,13	36,20	66,79	0,14	65,76	0,0154	0,14	66,79	0	63,84
	2	28,57	88,10	41,55	0,17	40,4	0,0276	0,29	41,55	0	39,81
	3	49,99	220,34	25,63	0,17	25,25	0,0148	0,34	25,63	0	25,51
	4	71,43	111,33	13,45	0,34	13,45	0	0,18	13,45	0	13,37
	5	92,84	208,84	3,17	0,34	3,17	0	0,21	3,17	0	3,17
5	1	7,13	37,07	66,88	0,13	66,39	0,0073	0,21	66,88	0	62,11
	2	28,56	60,72	41,43	0,15	41,05	0,0091	0,15	41,43	0	39,37
	3	50,00	144,30	24,68	0,21	24,68	0	0,18	24,68	0	23,91
	4	71,41	155,15	12,93	0,21	12,93	0	0,20	12,91	0,0015	12,68
	5	92,84	73,71	3,09	0,21	3,08	0,0032	0,33	3,09	0	3,08
6	1	7,13	34,45	67,26	0,13	66,68	0,0086	0,12	67,26	0	66,05
	2	28,57	66,79	41,74	0,17	41,66	0,0019	0,20	41,74	0	41,63
	3	50,00	124,89	26,33	0,19	26,33	0	0,17	26,33	0	25,85
	4	71,40	97,23	14,2	0,19	14,2	0	0,22	14,2	0	13,87
	5	92,84	201,63	3,39	0,21	3,28	0,0324	0,30	3,39	0	3,28
7	1	7,14	36,30	59,57	0,14	59,57	0	0,12	59,57	0	53,25
	2	28,55	74,21	36,0	0,19	36,0	0	0,15	36,0	0	33,84
	3	50,00	143,13	22,31	0,39	22,25	0,0026	0,16	22,25	0,0026	21,62
	4	71,41	150,51	11,74	0,19	11,31	0,0366	0,18	11,74	0	11,54
	5	92,83	72,89	2,78	0,19	2,26	0,1870	0,19	2,78	0	2,77
8	1	7,12	36,14	65,12	0,14	64,26	0,0132	0,18	65,12	0	62,84
	2	28,55	91,18	38,73	0,15	38,62	0,0028	0,16	38,73	0	37,63
	3	49,98	217,38	23,19	0,20	23,13	0,0025	0,19	23,19	0	22,88
	4	71,41	170,15	11,92	0,21	11,84	0,0067	0,20	11,84	0,0067	11,85
	5	92,85	231,70	2,86	0,22	2,86	0	0,18	2,86	0	2,84
9	1	7,14	36,27	65,73	0,14	65,06	0,0101	0,20	65,73	0	62,84
	2	28,56	80,06	37,7	0,17	37,7	0	0,17	37,7	0	37,65
	3	49,98	167,22	23,17	0,27	23,17	0	0,24	23,17	0	23,07
	4	71,41	155,83	12,0	0,22	12,0	0	0,23	12,0	0	12,0
	5	92,83	51,49	2,76	0,21	2,76	0	0,21	2,76	0	2,76
10	1	7,12	39,72	65,33	0,14	65,1	0,0035	0,14	65,33	0	62,35
	2	28,55	111,89	40,79	0,16	40,71	0,0019	0,17	40,79	0	40,71
	3	50,00	255,10	25,81	0,18	25,81	0	0,21	25,81	0	25,81
	4	71,42	202,17	13,53	0,20	13,53	0	0,21	13,53	0	13,53
	5	92,85	752,71	3,22	0,21	3,22	0	0,19	3,22	0	3,22

Этот результат очень похож на предыдущий. Алгоритм VND_2 находит оптимум в 45 из 50 случаев, что на 25 превосходит число оптимумов, найденных алгоритмом VND_1. Время работы решателя Gurobi в среднем, более чем в 500 раз больше времени алгоритмов VND.

Диаграмма на рис. 1 отражает усреднённые значения результатов из табл. 1, 2 по каждому уровню разбиения. По горизонтали отмечены номера уровней, по вертикали — усреднённые значения относительного отклонения. Сплошная линия относится к алгоритму VND_1, штриховая — VND_2. Штрихпунктирная линия с двумя точками показывает целевую функцию на оптимальном размещении исходной задачи.

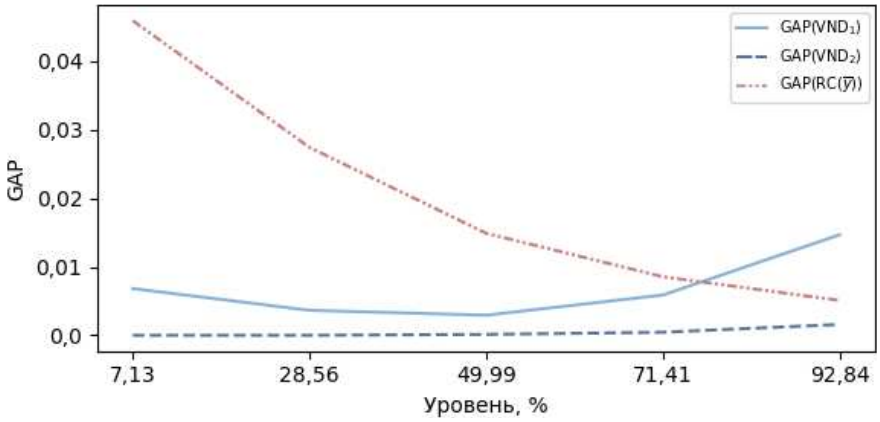


Рис. 1. Усреднённые значения по каждому уровню (табл. 1, 2)

Решения, найденные алгоритмами VND_1 и VND_2, близки к оптимальному. Значение целевой функции, полученное на оптимальном размещении исходной задачи, приближается к оптимальному при приближении порогового ограничения к максимальному доходу производителя.

Следующий результат получен на случайных входных данных с равномерным распределением.

Таблица 3

Результаты численного эксперимента на случайных входных данных

№	V		Gurobi		VND_1			VND_2			RC($\overline{y}$)
	level	%	time	opt	time	best	GAP	time	best	GAP	
1	1	32,54	1,20	45,2	0,01	45,2	0	0,01	45,2	0	45,2
	2	65,90	1,08	22,85	0,01	22,85	0	0,01	22,85	0	22,85
	3	99,25	0,30	0,5	0,01	0,5	0	0,01	0,5	0	0,5
2	1	32,66	1,18	52,05	0,00	52,05	0	0,00	52,05	0	52,05
	2	65,98	0,62	26,3	0,00	26,3	0	0,00	26,3	0	26,3
	3	99,35	0,26	0,5	0,00	0,5	0	0,00	0,5	0	0,5
3	1	33,01	25,00	53,98	0,09	53,95	0,0005	0,07	53,95	0,0005	53,98
	2	66,34	23,33	27,13	0,08	27,1	0,0011	0,07	27,1	0,0011	27,12
	3	99,69	4,00	0,25	0,10	0,23	0,0800	0,07	0,0	1	0,25
4	1	32,93	18,14	44,65	0,16	44,65	0	0,15	44,65	0	44,65
	2	66,28	15,46	22,45	0,16	22,45	0	0,15	22,45	0	22,45
	3	99,62	2,51	0,25	0,18	0,25	0	0,16	0,25	0	0,25
5	1	33,13	140,46	56,98	0,26	56,98	0	0,39	56,95	0,0005	56,98
	2	66,46	184,77	28,58	0,27	28,58	0	0,39	28,55	0,0010	28,58
	3	99,80	58,76	0,17	0,26	0,17	0	0,20	0,0	1	0,17
6	1	33,11	123,96	53,43	0,58	53,42	0,0001	0,51	53,4	0,0005	53,43
	2	66,45	117,26	26,8	0,56	26,78	0,0007	0,54	26,77	0,0011	26,8
	3	99,79	18,99	0,17	0,61	0,15	0,1176	0,54	0,0	1	0,17
7	1	33,18	17801,16	48,13	1,45	48,13	0	0,76	48,08	0,0010	48,13
	2	66,51	23961,22	24,12	1,56	24,12	0	0,82	24,07	0,0020	24,12
8	1	33,20	168378,12	53,38	1,93	53,38	0	1,8	53,38	0	53,20

В табл. 4 указаны размеры входа для примеров из табл. 3.

Таблица 4

Размеры входа для табл. 3

№	n, m, r	№	n, m, r	№	n, m, r	№	n, m, r
1	$20, 20, 10$	3	$40, 40, 10$	5	$60, 60, 10$	7	$90, 90, 10$
2	$20, 20, 15$	4	$40, 40, 15$	6	$60, 60, 15$	8	$100, 100, 10$

На примерах в табл. 3 алгоритм VND_1 находит оптимум чаще, чем алгоритм VND_2. Значение целевой функции, полученное на оптимальном решении исходной задачи, в 19 случаях из 21 оказывается оптимальным. В примере 7 на первом и втором уровнях решатель находит оптимум за 5–6 часов, в то время как алгоритму VND требуется менее двух секунд. В примере 8 решатель потратил на поиск оптимального решения около двух суток, а алгоритм VND — около двух секунд.

На рис. 2 представлены усреднённые значения результатов из табл. 3 по каждому уровню разбиения.

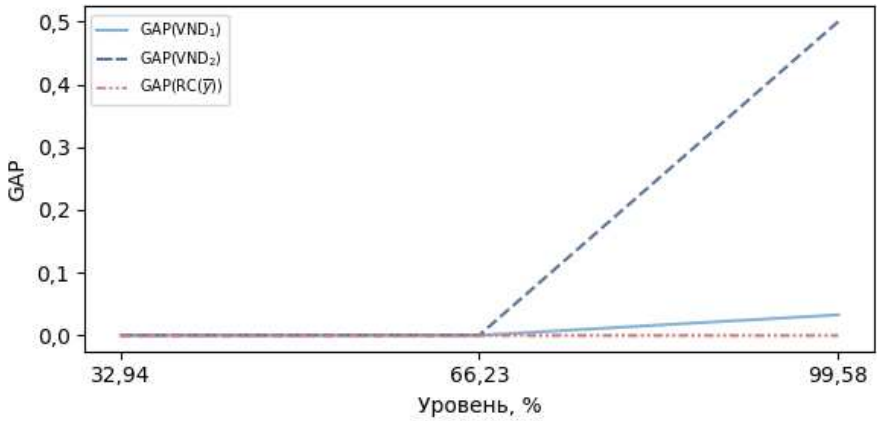


Рис. 2. Усреднённые значения по каждому уровню (табл. 3)

Таким образом, на небольшой с точки зрения времени вычисления размерности при наличии оптимального размещения $\overline{y}$ исходной задачи, имеет смысл применить алгоритм $\mathrm{RC}(\overline{y})$ для получения нижней оценки радиуса устойчивости.

Следующий эксперимент проводился на случайных данных большой с точки зрения времени вычисления размерности. Каждый пример разбит на два уровня. В столбце time для Gurobi указан прочерк, если решателю не удалось найти оптимального решения за три часа работы.

Лучшее найденное решение отмечено в столбце best. Синим цветом отмечено лучшее из решений решателя и алгоритмов, если оптимальное неизвестно.

Таблица 5

Результаты численного эксперимента на случайных входных данных

№	dim	V		Gurobi		VND_1		VND_2	
	n, m, r	level	%	time	best	time	best	time	best
1	100, 100, 10	1	33,32	—	53,26	4,59	53,28	4,33	53,28
	100, 100, 10	2	66,66	8616	26,63	3,42	26,63	3,05	26,63
2	100, 100, 15	1	33,33	3531	46,89	8,57	46,8	8,07	46,79
	100, 100, 15	2	66,66	2995	23,45	6,36	23,36	5,56	23,35
3	200, 200, 10	1	33,33	—	43,91	20,49	44,22	20,86	44,27
	200, 200, 10	2	66,67	—	22,69	25,98	23,3	23,71	23,35
4	200, 200, 15	1	33,33	—	50,55	154,51	51,65	56,69	51,54
	200, 200, 15	2	66,67	—	26,24	105,94	26,98	40,09	26,87
5	300, 300, 10	1	33,33	—	57,12	53,34	57,68	62,43	57,62
	300, 300, 10	2	66,66	—	27,69	76,90	30,24	83,31	30,19
6	300, 300, 15	1	33,33	—	31,70	274,46	52,14	195,93	52,16
	300, 300, 15	2	66,67	—	26,51	151,43	27,07	186,05	27,09
7	400, 400, 10	1	33,33	—	47,30	240,62	55,7	111,34	55,65
	400, 400, 10	2	66,67	—	21,20	443,30	29,61	210,62	29,56
8	400, 400, 15	1	33,33	—	43,33	611,13	49,58	553,10	49,58
	400, 400, 15	2	66,67	—	0,13	1009,20	25,9	910,73	25,9

В этих примерах решения, найденные алгоритмами VND_1 и VND_2, близки друг к другу. В большинстве случаев оба алгоритма находят решение лучше и за гораздо меньшее относительно решателя Gurobi время.

Заключение

Изучение пороговой устойчивости комбинаторных оптимизационных задач является новым направлением исследований в современной теории экстремальных задач. В ходе таких исследований возникают новые классы задач со своими структурными особенностями. В связи с этим возникает необходимость разработки точных и приближённых алгоритмов их решения. Основная доля работ в этой области связана с исследованием пороговой устойчивости одноуровневых оптимизационных задач [8–12]. В работах [8, 9] начато исследование пороговой устойчивости бикритериальных задач размещения объектов. Исследование пороговой

устойчивости двух- и трёхуровневых задач размещения объектов и ценообразования начато в [17, 18].

В настоящей работе предлагается подход к разработке алгоритмов решения задач пороговой устойчивости, основанный на методах решения исходной задачи и её подзадач. В статье показано, что используя решение исходной задачи, можно эффективно найти решение задачи пороговой устойчивости для двухуровневой задачи размещения производства и дискриминационного ценообразования при помощи предложенного полиномиального алгоритма поиска оптимальной цены и гибридной эвристики, которая основана на эвристике VND и локальном поиске. Приводится экспериментальное сравнение двух версий алгоритма и решателя Gurobi. Аналогичные результаты содержатся в работе [24], в которой показана эффективность предлагаемого подхода при разработке приближённых алгоритмов решения для двухуровневой задачи с медианным типом размещения предприятий и равномерным ценообразованием как с точки зрения времени работы алгоритмов, так и качества получаемых решений. Однако наибольший интерес представляет исследование пороговой устойчивости двухуровневых задач размещения предприятий с фабричным ценообразованием, поскольку в этом случае задача ценообразования NP-трудна для фиксированного размещения объектов. По этой причине важно оценить, насколько эффективными окажутся приближённые алгоритмы, разработанные на основе предложенного подхода для данной постановки.

Теоремы 1 и 2 приводят к следующей гипотезе: возможно, задачи пороговой устойчивости, исследованные в данной работе, полны в классе NPO относительно подходящей сводимости, сохраняющей аппроксимируемость.

Финансирование работы

Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 23–21–00424).

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Greenberg H. J.** An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization // Advances in computational and stochastic optimization, logic programming, and heuristic search. New York: Springer, 1998. P. 97–147. DOI: 10.1007/978-1-4757-2807-1_4.

2. **Ben-Tal A., Nemirovski A.** Robust optimization: Methodology and applications // Math. Program. 2002. V. 92. P. 453–480.
3. **Snyder L. V.** Facility location under uncertainty: A review // IIE Trans. 2006. V. 38. P. 537–554. DOI: 10.1080/07408170500216480.
4. **Dyer M., Stougie L.** Computational complexity of stochastic programming problems // Math. Program. Ser. A. 2006. V. 106. P. 423–432. DOI: 10.1007/s10107-005-0597-0.
5. **Кибзун А. И., Кан Ю. С.** Задачи стохастического программирования с вероятностными критериями. М.: Физматлит, 2009. 371 с.
6. **Correia I., da Gama F. S.** Facility location under uncertainty // Location science. Cham: Springer, 2015. P. 177–203.
7. **Charitopoulos V. M., Papageorgiou L. G., Dua V.** Multiparametric mixed integer linear programming under global uncertainty // Comput. Chem. Eng. 2018. V. 116. P. 279–295.
8. **Carrizosa E., Nickel S.** Robust facility location // Math. Methods Oper. Res. 2003. V. 58. P. 331–349.
9. **Carrizosa E., Ushakov A., Vasilyev I.** Threshold robustness in discrete facility location problems: A bi-objective approach // Optim. Lett. 2015. V. 9. P. 1297–1314.
10. **Rossi A., Gurevsky E., Battaïa O., Dolgui A.** Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations // Discrete Appl. Math. 2016. V. 208. P. 123–136.
11. **Pirogov A., Gurevsky E., Rossi A., Dolgui A.** Robust balancing of transfer lines with blocks of uncertain parallel tasks under fixed cycle time and space restrictions // Eur. J. Oper. Res. 2021. V. 290. P. 946–955.
12. **Sotskov Yu. N.** Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives // Algorithms. 2023. V. 16, No. 2. Paper ID 100. 43 p.
13. **Леонтьев В. К.** Устойчивость задачи коммивояжера // Вычисл. математика и мат. физика. 1975. Т. 15, № 5. С. 1298–1309.
14. **Леонтьев В. К., Гордеев Э. Н.** Качественное исследование траекторных задач // Кибернетика. 1986. № 5. С. 82–89.
15. **Sotskov Yu. N., Leontiev V. K., Gordeev Eh. N.** Some concepts of stability analysis in combinatorial optimization // Discrete Appl. Math. 1995. V. 58, No. 2. P. 169–190.
16. **Кузьмин К. Г.** Единый подход к нахождению радиусов устойчивости в многокритериальной задаче о максимальном разрезе графа // Дискрет. анализ и исслед. операций. 2015. Т. 22, № 5. С. 30–51.
17. **Panin A. A., Plyasunov A. V.** Stability analysis for pricing // Mathematical optimization theory and operations research. Rev. Sel. Pap. 19th Int. Conf. (Novosibirsk, Russia, July 6–10, 2020). Cham: Springer, 2020. P. 57–69. (Commun. Comput. Inf. Sci.; V. 1275). DOI: 10.1007/978-3-030-58657-7_7.
18. **Panin A. A., Plyasunov A. V.** The multilevel facility location and pricing problems: the computational complexity and the stability analysis // Optim. Lett. 2023. V. 17. P. 1295–1315.

19. Dempe S., Zemkoho A. Bilevel optimization. Advances and next challenges. Cham: Springer, 2020. 672 p. (Springer Optim. Its Appl.; V. 161). DOI: 10.1007/978-3-030-52119-6.

20. Kochetov Yu. A., Plyasunov A. V., Panin A. A. Bilevel discrete optimisation: Computational complexity and applications // The Palgrave handbook of operations research. Cham: Palgrave Macmillan, 2022. P. 3–42. DOI: 10.1007/978-3-030-96935-6_1.

21. Talbi E.-G. Metaheuristics: From design to implementation. Berlin: Wiley, 2009. 624 p.

22. Mladenovic N., Hansen P. Variable neighbourhood search // Comput. Oper. Res. 1997. V. 24. P. 1097–1100.

23. 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.

24. Vodyan M. E., Panin A. A., Plyasunov A. V. Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem // 2023 19th Int. Asian School-Seminar Optimization Problems of Complex Systems (Novosibirsk, Russia, Aug. 14–22, 2023). Piscataway: IEEE, 2023. P. 130–135. DOI: 10.1109/OPCS59592.2023.10275325.

Водян Максим Евгеньевич
Панин Артём Александрович
Плясунов Александр Владимирович

Статья поступила
10 ноября 2023 г.
После доработки —
23 января 2024 г.
Принята к публикации
22 марта 2024 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2024. Vol. 31, No. 3. P. 79–104

UDC 519.8+518.25 DOI: 10.33048/daio.2024.31.788

A STUDY OF THE THRESHOLD STABILITY OF THE BILEVEL PROBLEM OF FACILITY LOCATION AND DISCRIMINATORY PRICING

M. E. Vodyan[1, a], *A. A. Panin*[2, b], and *A. V. Plyasunov*[2, c]

[1] Novosibirsk State University,
2 Pirogov Street, 630090 Novosibirsk, Russia
[2] Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia
E-mail: [a] m.vodyan@g.nsu.ru,
[b] aapanin1988@gmail.com, [c] apljas@math.nsc.ru

Abstract. The problem of threshold stability for a bilevel problem with a median type of facility location and discriminatory pricing is considered. When solving such a problem, it is necessary to find the threshold stability radius and a semifeasible solution of the original bilevel problem such that the leader's revenue is not less than a predetermined value (threshold) for any deviation of budgets that does not exceed the threshold stability radius and which preserves its semifeasibility. Thus, the threshold stability radius determines the limit of disturbances of consumer budgets with which these conditions are satisfied.

Two approximate algorithms for solving the threshold stability problem based on the heuristic of descent with alternating neighborhoods are developed. These algorithms are based on finding a good approximate location of facilities as well as on calculating the optimal set of prices for the found location of facilities. The algorithms differ in the way they compare various locations of facilities; this ultimately leads to different estimates of threshold stability radius. A numerical experiment has shown the efficiency of the chosen approach both in terms of the running time of the algorithms and the quality of the solutions obtained. Tab. 4, illustr. 2, bibliogr. 24.

Keywords: bilevel optimization, threshold stability problem, threshold stability radius, facility location, discriminatory pricing, variable neighborhood descent.

English transl.: *Journal of Applied and Industrial Mathematics* **18** (3), 558–574 (2024), DOI: 10.1134/S1990478924030165.

References

1. **H. J. Greenberg,** An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization, in *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search* (Springer, New York, 1998), pp. 97–147, DOI: 10.1007/978-1-4757-2807-1_4.
2. **A. Ben-Tal** and **A. Nemirovski,** Robust optimization: Methodology and applications, *Math. Program.* **92**, 453–480 (2002).
3. **L. V. Snyder,** Facility location under uncertainty: A review, *IIE Trans.* **38**, 537–554 (2006), DOI: 10.1080/07408170500216480.
4. **M. Dyer** and **L. Stougie,** Computational complexity of stochastic programming problems, *Math. Program., Ser. A,* **106**, 423–432 (2006), DOI: 10.1007/s10107-005-0597-0.
5. **A. I. Kibzun** and **Yu. S. Kan,** *Stochastic Programming Problems with Probabilistic Criteria* (Fizmatlit, Moscow, 2009) [Russian].
6. **I. Correia** and **F. S. da Gama,** Facility location under uncertainty, in *Location Science* (Springer, Cham, 2015), pp. 177–203.
7. **V. M. Charitopoulos, L. G. Papageorgiou,** and **V. Dua,** Multiparametric mixed integer linear programming under global uncertainty, *Comput. Chem. Eng.* **116**, 279–295 (2018).
8. **E. Carrizosa** and **S. Nickel,** Robust facility location, *Math. Methods Oper. Res.* **58**, 331–349 (2003).
9. **E. Carrizosa, A. Ushakov,** and **I. Vasilyev,** Threshold robustness in discrete facility location problems: A bi-objective approach, *Optim. Lett.* **9**, 1297–1314 (2015).
10. **A. Rossi, E. Gurevsky, O. Battaïa,** and **A. Dolgui,** Maximizing the robustness for simple assembly lines with fixed cycle time and limited number of workstations, *Discrete Appl. Math.* **208**, 123–136 (2016).
11. **A. Pirogov, E. Gurevsky, A. Rossi,** and **A. Dolgui,** Robust balancing of transfer lines with blocks of uncertain parallel tasks under fixed cycle time and space restrictions, *Eur. J. Oper. Res.* **290**, 946–955 (2021).
12. **Yu. N. Sotskov,** Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives, *Algorithms* **16** (2), ID 100 (2023).
13. **V. K. Leontiev,** Stability of the travelling salesman problem, *Vychisl. Mat. Mat. Fiz.* **15** (5), 1298–1309 (1975) [Russian] [*USSR Comput. Math. Math. Phys.* **15** (5), 199–213 (1975)].
14. **V. K. Leontiev** and **Eh. N. Gordeev,** Qualitative investigation of path problems, *Kibern.,* No. 5, 82–89 (1986) [Russian] [*Cybern.* **22**, 636–646 (1986)].
15. **Yu. N. Sotskov, V. K. Leontiev,** and **Eh. N. Gordeev,** Some concepts of stability analysis in combinatorial optimization, *Discrete Appl. Math.* **58** (2), 169–190 (1995).
16. **K. G. Kuz'min,** A general approach to the calculation of stability radii for the max-cut problem with multiple criteria, *Diskretn. Anal. Issled. Oper.* **22** (5), 30–51 (2015) [Russian] [*J. Appl. Ind. Math.* **9** (4), 527–539 (2015)].

17. A. A. Panin and **A. V. Plyasunov,** Stability analysis for pricing, in *Mathematical Optimization Theory and Operations Research* (Rev. Sel. Pap. 19th Int. Conf., Novosibirsk, Russia, July 6–10, 2020) (Springer, Cham, 2020), pp. 57–69 (Commun. Comput. Inf. Sci., Vol. 1275), DOI: 10.1007/978-3-030-58657-7_7.

18. A. A. Panin and **A. V. Plyasunov,** The multilevel facility location and pricing problems: the computational complexity and the stability analysis, *Optim. Lett.* **17**, 1295–1315 (2023).

19. S. Dempe and **A. Zemkoho,** *Bilevel Optimization. Advances and Next Challenges* (Springer, Cham, 2020) (Springer Optim. Its Appl., Vol. 161), DOI: 10.1007/978-3-030-52119-6.

20. Yu. A. Kochetov, A. V. Plyasunov, and **A. A. Panin,** Bilevel discrete optimisation: Computational complexity and applications, in *The Palgrave Handbook of Operations Research* (Palgrave Macmillan, Cham, 2022), pp. 3–42, DOI: 10.1007/978-3-030-96935-6_1.

21. E.-G. Talbi, *Metaheuristics: From Design to Implementation* (Wiley, Berlin, 2009).

22. N. Mladenovic and **P. Hansen,** Variable neighbourhood search, *Comput. Oper. Res.* **24**, 1097–1100 (1997).

23. 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) [Russian] [*J. Appl. Ind. Math.* **9** (3), 392–401 (2015)].

24. M. E. Vodyan, A. A. Panin, and **A. V. Plyasunov,** Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem, in *2023 19th Int. Asian School-Seminar Optimization Problems of Complex Systems, Novosibirsk, Russia, Aug. 14–22, 2023* (IEEE, Piscataway, 2023), pp. 130–135, DOI: 10.1109/OPCS59592.2023.10275325.

Maksim E. Vodyan
Artyom A. Panin
Aleksandr V. Plyasunov

Received November 10, 2023
Revised January 23, 2024
Accepted March 22, 2024