

ISSN 2949-5598

# ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 3 2025

Новосибирск  
Издательство Института математики

МАКСИМИЗАЦИЯ РАДИУСА ПОРОГОВОЙ  
УСТОЙЧИВОСТИ В МОДЕЛИ РАЗМЕЩЕНИЯ  
ПРОИЗВОДСТВА И ФАБРИЧНОГО ЦЕНООБРАЗОВАНИЯ

*М. Е. Водян<sup>а</sup>, А. А. Панин<sup>б</sup>, А. В. Плясунов<sup>с</sup>*

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

E-mail: <sup>а</sup>m.vodyan@ng.su.ru,  
<sup>б</sup>aapanin1988@gmail.com, <sup>с</sup>apljas@math.nsc.ru

**Аннотация.** Исследуется пороговая устойчивость задачи с медианным размещением предприятий и фабричным ценообразованием. Задача пороговой устойчивости имеет следующие отличия от исходной двухуровневой постановки: в задаче верхнего уровня максимизируется отклонение бюджетов потребителей от ожидаемых значений при условии, что доход производителя не меньше заданного порога. Главное отличие исследуемой постановки от задач, чья пороговая устойчивость изучалась ранее, заключается в том, что при фиксированном размещении предприятий задача фабричного ценообразования NP-трудна в сильном смысле.

Для решения задачи пороговой устойчивости предлагается алгоритм на основе спуска с чередующимися окрестностями (VND). Численное исследование алгоритма проводится на известных примерах и случайно сгенерированных данных. Эксперимент показал, что идея итеративного вычитания радиуса пороговой устойчивости из бюджетов потребителей, впервые реализованная в данной работе, сильно снижает время работы алгоритма. На примерах, для которых был найден оптимум, алгоритм ошибся в среднем на 0,63%. На всех примерах алгоритм находит решение в среднем на 2,97% лучше, чем решатель Gurobi. Табл. 4, ил. 2, библиогр. 33.

**Ключевые слова:** двухуровневая задача, пороговая устойчивость, радиус пороговой устойчивости, размещение производства, фабричное ценообразование, спуск с чередующимися окрестностями.

### Введение

В настоящей работе продолжаются исследования пороговой устойчивости двухуровневых задач размещения и ценообразования [1–4]. Неформально устойчивость — это свойство оптимизационной задачи, которое выражает в том или ином смысле меру её нечувствительности к неопределённости в исходных данных. Любой изученный тип устойчивости связан с определённым типом неопределённости в исходных данных. В зависимости от типа доступной информации такие проблемы исследуются в рамках следующих направлений: стохастического программирования, оптимизации на основе нечёткого представления данных, робастной оптимизации, постоптимального анализа чувствительности и устойчивости решений задач линейного и целочисленного программирования [5–10].

С каждым из классических подходов к анализу надёжности решений при различных возмущениях исходных данных связаны определённые проблемы. В основе моделей стохастического программирования лежит информация о вероятностном распределении случайных параметров, которая на практике зачастую недоступна. Ряд исследований демонстрирует очень высокую сложность многоэтапных задач стохастического программирования, которые оказываются PSPACE-трудными [5]. Разработка моделей на основе нечёткого представления данных существенно более сложное занятие, чем классическое математическое моделирование. Качество получаемых моделей существенно зависит от качества используемых экспертных оценок.

Узкое место робастной оптимизации заключается в том, что её применение ориентируется на учёт худших сценариев [7]. 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 [11–15].

Подход к исследованию неопределённости на основе постоптимального анализа чувствительности и устойчивости решений попросту игнорирует влияние неопределённости данных в своих моделях. Сам по себе анализ чувствительности — только лишь инструмент для анализа устойчивости решения, и он не может быть использован для поиска решений, устойчивых к вариации данных. Таким образом, помимо проблем с получением информации, которая требуется в том или ином подходе при анализе устойчивости задачи, возникают и существенные вычислительные трудности.

Относительно недавно в обсуждаемой области возникло новое направление исследований, названное пороговой устойчивостью. При исследовании устойчивости задачи Вебера авторы [16] предложили применить

пороговую модель вместо робастной оптимизации. В качестве неопределённых данных применён вектор спроса, а для того чтобы не допустить слишком большого роста транспортных затрат, введён порог, ограничивающий значение целевой функции сверху.

В задаче пороговой устойчивости для заданного набора входных данных задачи Вебера вместо минимизации транспортных расходов ищется размещение предприятий с максимальным радиусом устойчивости и транспортными затратами, не превосходящими заданного бюджетного порога. В [17] эта идея реализована для задачи о  $p$ -медиане и для простейшей задачи размещения предприятий.

Далее в работе используется близкое по смыслу определение радиуса пороговой устойчивости, в котором с каждым размещением предприятий связывается величина, равная максимальному отклонению спроса от ожидаемых значений при условии, что выполняется пороговое ограничение. Такое определение восходит к впервые введённому в [18] понятию радиуса устойчивости, на основе которого в [1] предложен радиус пороговой устойчивости для двухуровневых задач. Небольшой, но информативный обзор работ, связанных с понятием радиуса устойчивости, и развитых на его основе подходов к исследованию устойчивости оптимизационных задач можно найти в [1, 2, 19–25].

Идея конструкции такова, что с каждой оптимизационной задачей можно связать задачу пороговой устойчивости, в которой ищется максимальное значение параметра (радиус устойчивости), ограничивающего нормы вариаций исходных данных исследуемой задачи, и подходящее допустимое решение базовой постановки, удовлетворяющее пороговому ограничению. В [1] этот подход обобщён и применён к исследованию пороговой устойчивости многоуровневых задач размещения и ценообразования.

Первые результаты в области пороговой устойчивости двухуровневых задач получены в работах [1–4]. Задачи такого типа образуют новый класс двухуровневых задач, для которых не известны ни точные, ни приближённые методы решения. Исследование новых классов двухуровневых задач несомненно является важной теоретической проблемой, поскольку при этом разрабатываются новые точные и приближённые методы для их решения. Современное состояние дел в области двухуровневой оптимизации можно найти в обзорах [26, 27].

В [1] впервые исследована пороговая устойчивость задачи ценообразования с различными ценовыми политиками и задачи конкурентного ценообразования. Обзор по проблемам размещения производства и ценообразования и пороговой устойчивости можно найти в [2]. В [3, 4] изучаются задачи с медианным ограничением на открытие предприятий и двумя стратегиями ценообразования: равномерной и дискриминационной.

В текущей работе впервые исследуется пороговая устойчивость NP-трудной в сильном смысле задачи медианного размещения производства и фабричного ценообразования [30, 31]. Производитель определяет, в каких местах он должен открыть заданное количество предприятий, и решает, какие цены на однородную продукцию нужно установить на каждом из них (фабричное ценообразование), чтобы максимизировать прибыль от обслуживания потребителей. Каждый потребитель имеет ограниченный бюджет, который он готов потратить на покупку и транспортировку продукта в единичном экземпляре. Потребители действуют рационально, т. е. минимизируют свои затраты. Будем полагать, что транспортные затраты известны и неизменны, а информация о бюджетах неточная, но предполагается, что известны ожидаемые значения. В таком случае множество сценариев (неопределённость) задаётся как множество отклонений бюджетов от прогнозируемых значений.

В разд. 1 приводятся постановка и математическая модель исследуемой проблемы. В разд. 2 содержатся результаты о вычислительной сложности задачи пороговой устойчивости. Приближённые алгоритмы для её решения предлагаются в разд. 3, а в разд. 4 описывается численный эксперимент для сравнения разработанных алгоритмов с решателем Gurobi. В заключении обсуждаются полученные результаты и направления дальнейших исследований.

## 1. Постановка задачи пороговой устойчивости

Прежде чем сформулировать постановку задачи пороговой устойчивости, заметим, что робастная оптимизация и пороговая устойчивость исследуют один тип неопределённости. Однако, в робастной оптимизации предполагается, что вектор данных задачи не известен, но принадлежит некоторому множеству, которое описывает неопределённость и называется множеством сценариев. Оптимальное решение в такой задаче строится с учётом всех сценариев и фактически сводится к решению минимаксной задачи, ограничения которой удовлетворяют всем сценариям [7]. В задаче пороговой устойчивости предполагается, что вектор данных известен и необходимо найти такое множество возмущений, для которого найдётся допустимое решение, удовлетворяющее пороговому ограничению. Искомая область возмущений определяется радиусом пороговой устойчивости.

В [1] содержится формальное определение радиуса пороговой устойчивости и постановка задачи пороговой устойчивости для оптимизационных задач. Основная идея этого подхода заключается в следующем. *Пороговое значение* в таких задачах — это величина дохода, которую рациональный лидер считает достаточной. Далее, *радиус устойчивости*

допустимого решения задачи — это такое число, для которого любое одновременное возмущение бюджетов клиентов на величину, не превышающую этого числа, не нарушает допустимости этого решения, а значение целевой функции на этом решении не превышает значения выбранного порога.

Пусть  $V$  — некоторый порог. В задаче пороговой устойчивости необходимо найти допустимое решение, которое удовлетворяет пороговому ограничению и имеет максимальный радиус устойчивости.

Там же приводится модификация этого подхода для двухуровневых задач и реализация для задач ценообразования. Разделим переменные двухуровневой задачи на две группы  $(x, y)$ , где  $x$  — переменные верхнего уровня,  $y$  — переменные нижнего уровня. В определении радиуса устойчивости заменяем требование допустимости решения при возмущении бюджетов условием существования  $y$  такого, что  $(x, y)$  является допустимым решением двухуровневой задачи.

При анализе пороговой устойчивости двухуровневой задачи необходимо найти вектор значений переменных верхнего уровня  $x$  с наибольшим радиусом возможного варьирования входных данных, при котором целевая функция продолжает удовлетворять пороговому ограничению.

В рассматриваемой постановке варьируются не все входные данные, а только бюджеты потребителей, возмущение которых происходит в сторону их уменьшения. При увеличении бюджетов задача пороговой устойчивости становится тривиальной, а радиус пороговой устойчивости стремится к бесконечности. Таким образом, в задаче ниже фиксируем доход производителя и ищем решение, которое предоставляет доход не менее зафиксированного.

Приведём содержательную постановку базовой задачи размещения производства и ценообразования, пороговая устойчивость которой исследуется ниже. Сформулируем её в виде игры Штакельберга «лидер — последователь». В качестве лидера выступает производитель, который размещает  $r$  предприятий и формирует цены на каждом из них. В качестве последователей — потребители, выбирающие предприятия так, чтобы минимизировать суммарные затраты на покупку и транспортировку товаров. При этом потребитель совершает покупку только в том случае, если 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 чего необходимо ввести следующее соглашение. Если у потребителя есть несколько предприятий с одинаковой минимальной суммой платежей, то он выберет предприятие с минимальными транспортными затратами.

Обычно в подобных постановках рассматриваются три следующие стратегии ценообразования [32]:

- равномерное (uniform pricing) — на всех предприятиях устанавливается одна цена;
- фабричное (mill pricing) — на каждом предприятии устанавливается своя цена;
- дискриминационное (discriminatory pricing) — на каждом предприятии для каждого потребителя устанавливается своя цена.

В настоящей статье рассматривается фабричное ценообразование.

Отличие задачи пороговой устойчивости от базовой постановки — в заранее заданном доходе производителя, который определяет пороговое ограничение, и в наличии неопределённости в бюджетах потребителей, для которой необходимо предусмотреть максимально возможное отклонение от ожидаемых (определённых заранее) бюджетов.

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

Обозначения:

- $I = \{1, \dots, n\}$  — множество возможных мест для открытия предприятий;
- $J = \{1, \dots, m\}$  — множество потребителей;
- $r \in \mathbb{Z}$  — число размещаемых предприятий;
- $b_j \in \mathbb{Z}$  — бюджет потребителя  $j$ ;
- $c_{ij} \in \mathbb{Z}$  — транспортные затраты потребителя  $j$  при обслуживании на предприятии  $i$ ;
- $V \in \mathbb{Z}$  — доход производителя.

Переменные:

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

Двухуровневая смешанно целочисленная квадратичная математическая модель задачи пороговой устойчивости имеет следующий вид:

$$\rho \rightarrow \max_{p, y, x, \rho}, \quad (1)$$

$$\sum_{i \in I} \sum_{j \in J} p_i x_{ij} \geq V, \quad (2)$$

$$\sum_{i \in I} y_i = r, \quad (3)$$

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

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

$$\sum_{i \in I} \sum_{j \in J} (b_j - c_{ij} - \rho - p_i) x_{ij} \rightarrow \max_x, \quad (5)$$

$$\sum_{i \in I} x_{ij} \leq 1, \quad j \in J, \quad (6)$$

$$x_{ij} \leq y_i, \quad i \in I, j \in J, \quad (7)$$

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

Максимизируя целевую функцию (1) на верхнем уровне, получим максимально возможное отклонение от ожидаемых (данных) размеров бюджетов. Неравенство (2) устанавливает пороговое ограничение, гарантирующее, что доход производителя не меньше заданного, а равенство (3) требует, чтобы было открыто ровно  $r$  предприятий. Условия (4) определяют область значений переменных верхнего уровня и фиксируют фундаментальное свойство двухуровневых задач: переменные нижнего уровня  $x$  принимают свои значения из множества оптимальных решений задачи нижнего уровня. Таким образом, в работе исследуется оптимистический вариант постановки. Целевая функция нижнего уровня (5) представляет собой сумму неизрасходованных потребителями средств, а ограничения (6)–(8) гарантируют, что каждый потребитель обслуживается не более чем одним предприятием производителя, которое должно быть открыто.

## 2. Вычислительная сложность задачи пороговой устойчивости

Используемые далее понятия и обозначения классов сложности, связанные с полиномиальной и аппроксимационной иерархиями, можно найти в [2, 4, 33]. Будем предполагать, что переменные  $\rho$  и  $p$  целочисленные. Обозначим через  $D_\rho$  и  $D$  стандартные задачи распознавания для задачи (1)–(8) и базовой задачи соответственно. Приведём доказательство следующей теоремы в варианте, который демонстрирует тесную связь между этими задачами распознавания.

**Теорема 1.** *Задача  $D_\rho$  NP-полна в сильном смысле.*

**Доказательство.** Будем следовать идее, использованной при обосновании аналогичного результата в [4], при этом необходимо учесть особенности новой постановки. Покажем, что задача  $D_\rho$  полиномиально сводится к некоторому модифицированному варианту задачи  $D$ . Для этого рассмотрим произвольный пример с ответом «да» задачи  $D_\rho$  с некоторой целой константой  $\hat{\rho}$ .



Из определения задачи  $D_\rho$  следует, что существует такое допустимое решение  $(\rho, y, p, x)$ , что  $\rho \geq \hat{\rho}$ . Можно считать, что  $\rho = \hat{\rho}$ . Действительно, если  $\rho > \hat{\rho}$ , то при уменьшении  $\rho$  до  $\hat{\rho}$  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 возрастет, т. е. пороговое ограничение не будет нарушено.

Тем самым существование для некоторого целого  $\hat{\rho}$  такого допустимого решения  $(\rho, y, p, x)$ , что  $\rho \geq \hat{\rho}$ , эквивалентно существованию такого размещения предприятий  $y$  и такого набора цен, при которых в базовой задаче с бюджетами  $b_j - \hat{\rho}$ ,  $j \in J$ , множеством открытых предприятий  $\{i \mid y_i = 1\}$  и множеством цен  $p$  доход лидера больше заданного порога  $V$ .

Таким образом, для заданного целого  $\hat{\rho}$  требуемые в базовой задаче размещение  $y$  и набор фабричных цен  $p$  могут быть найдены за недетерминированное полиномиальное время, если в задаче  $D_\rho$  ответ «да». Отсюда следует, что задача  $D_\rho$  принадлежит классу NP.

Полнота задачи  $D_\rho$  в классе NP следует из полиномиальной сводимости задачи  $D$  к задаче  $D_\rho$ . Действительно, в  $D$  для заданного порога  $V$  надо найти допустимое решение  $(y, p, x)$ , которое приносит лидеру доход не меньше порога. В качестве исходных данных задачи  $D_\rho$  возьмём  $\hat{\rho} = 0$  и исходные данные задачи  $D$ . Из результатов, полученных в [2], 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 отклонения от оптимального решения, если  $P \neq NP$ .

**ДОКАЗАТЕЛЬСТВО.** Предположим, что существует детерминированный полиномиальный приближённый 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  $D$  так же, как в доказательстве теоремы 1. Если в задаче  $D$  для порога  $V$  ответ «да», то алгоритм выдаст некоторое приближённое допустимое решение  $(\rho, y, p, x)$  задачи (1)–(8). Если  $\rho > 0$ , то, рассуждая как в доказательстве теоремы 1, получим её допустимое решение  $(0, y, p, \hat{x})$ , которое подтверждает, что в задаче  $D$  для текущего входа ответ «да». Таким образом получаем полиномиальный алгоритм для задачи  $D$ , что противоречит условию  $P \neq NP$ . Теорема 2 доказана.

Сформулируем ещё одно утверждение, которое следует из доказанных теорем и позволяет уточнить возможности полиномиальных алгоритмов относительно исследуемой задачи.

**Следствие 1.** Для задачи (1)–(8) можно разработать точный полиномиальный алгоритм только в классе детерминированных алгоритмов с оракулами из класса NP, если  $P \neq NP$ .

**ДОКАЗАТЕЛЬСТВО.** Из теорем 1 и 2 следует, что задача (1)–(8) принадлежит классу NPO, причём лежит выше класса Exр-APX. В силу предположения классы NPO и PO не совпадают, следовательно, для задачи (1)–(8) не существует полиномиального детерминированного алгоритма. Из включения  $NPO \subseteq \Delta_2^P O$ , где  $\Delta_2^P O$  — класс оптимизационных задач, разрешимых детерминированными полиномиальными алгоритмами с оракулами из класса NP, следует требуемый результат. Следствие 1 доказано.

Полученные результаты о неаппроксимируемости утверждают, что невозможно разработать детерминированные полиномиальные точные и приближённые алгоритмы с оценками относительного уклонения. Это означает, что исследуемая задача пороговой устойчивости либо NPO-полна относительно подходящей сводимости, сохраняющей аппроксимируемость, либо лежит в промежуточном классе задач выше класса Exр-APX.

### 3. Алгоритмы

При разработке алгоритмов для решения задачи (1)–(8) использована VND-эвристика, идея которой содержится в [28]. Позднее эта эвристика применена в [29–31] для разработки эффективных алгоритмов решения задачи размещения и фабричного ценообразования. В настоящей работе для решения задачи пороговой устойчивости разработана модификация этой эвристики — двухэтапная VND-эвристика. Для определения и выбора лучшего размещения необходимы критерии их сравнения.

Введём некоторые вспомогательные величины и опишем алгоритм определения радиуса пороговой устойчивости на основе поиска цены. Если для поиска цены применить точный алгоритм, то алгоритм определения радиуса пороговой устойчивости будет точным. Для поиска цены при фиксированном размещении используем VND-эвристику, основываясь на идеях [30, 31].

Пусть  $d(y, p) = V - \sum_{i \in I} \sum_{j \in J} x_{ij} p_i$  — сверхприбыль производителя относительно порога  $V$  при фиксированном размещении;  $c(y, p) = \sum_{i \in I} \sum_{j \in J} x_{ij}$  — число обслуживаемых клиентов при фиксированном размещении;  $B = (b_1, \dots, b_m)$  — вектор бюджетов потребителей;  $PC(y, B)$  — алгоритм поиска цены при фиксированном размещении [30, 31]. Для поиска радиуса пороговой устойчивости использовался следующий алгоритм  $RC(y, B)$ .

**Алгоритм 1.** Алгоритм  $RC(y, B)$ **Вход:**  $y, B$ .**Выход:**  $\rho$  — радиус пороговой устойчивости.

- 1:  $\rho \leftarrow 0; p \leftarrow PC(y, B);$
- 2: **if**  $d(y, p) < 0$  **then stop**;
- 3: **else**  $\Delta\rho \leftarrow d(y, p)/c(y, p); \rho \leftarrow \rho + \Delta\rho;$
- 4: **if**  $\Delta\rho < \min_{i \in I} p_i$  **then stop**;
- 5: **else**  $B \leftarrow B - (\Delta\rho, \dots, \Delta\rho); p \leftarrow PC(y, B);$
- 6: **goto** 2;

Предлагаемая ниже реализация алгоритма для поиска цены основана на VND-эвристике, поэтому в функции  $PC$  и  $RC$  добавим аргумент  $\text{flip}$ , ограничивающий число просматриваемых окрестностей. Этот аргумент используется для остановки алгоритма, а именно для процедуры улучшения, описанной в [4].

Если алгоритм остановился после шага 1, то считаем размещение  $y$  недопустимым. На шаге 2 выполняется проверка возможности увеличить радиус пороговой устойчивости, и если сверхприбыль положительная, то увеличиваем радиус. На шаге 4 происходит проверка необходимости поиска цены, и если есть необходимость, то на шаге 5 ищем цены относительно новых бюджетов. Тем самым на каждой итерации алгоритма имеем бюджеты потребителей, содержащие радиус пороговой устойчивости, размещение и цены для него такие, что доход производителя больше порога  $V$ , т. е. получаем допустимое решение задачи пороговой устойчивости.

Пусть  $y_1$  и  $y_2$  — различные размещения, для сравнения которых выпишем два критерия. Первый критерий выбирает размещение, при котором доход производителя больше, а второй — то, для которого больше радиус пороговой устойчивости.

**Критерий 1.** Полагаем  $p_1 = PC(y_1, B, \text{flip})$ ,  $p_2 = PC(y_2, B, \text{flip})$ , и если  $\sum_{i \in I} \sum_{j \in J} p_{1i} x_{ij} > \sum_{i \in I} \sum_{j \in J} p_{2i} x_{ij}$ , то считаем размещение  $y_1$  лучше  $y_2$ .

**Критерий 2.** Полагаем  $\rho_1 = RC(y_1, B, \text{flip})$ ,  $\rho_2 = RC(y_2, B, \text{flip})$ , и если  $\rho_1 > \rho_2$ , то считаем размещение  $y_1$  лучше чем  $y_2$ .

С целью построения основного алгоритма для размещения  $y$  определим окрестность  $\bar{k}\text{-Swar}(y)$  и процедуру улучшения  $k\text{-Improve}(y)$ , как это сделано в [3]. Здесь  $k, \bar{k} \in \mathbb{N}$  — некоторые параметры. Для сравнения размещений в окрестности используем критерии 1 и 2. Основным алгоритм

**Алгоритм 2.** Алгоритм VND<sub>1</sub>**Вход:**  $I_{\max}$ ,  $B$ ,  $k$ , flip.**Выход:** размещение  $y$  и радиус пороговой устойчивости  $\rho$ .

- 1:  $I \leftarrow 0$ ;  $y \leftarrow \text{rand}\{y' \in \{0, 1\}^n \mid \sum_i y'_i = r\}$  — случайный булев вектор;  
 $\rho \leftarrow \text{RC}(y, B, \text{flip})$ ;  $B \leftarrow B - (\rho, \dots, \rho)$ ;
- 2: применить локальный поиск для 1-Swap( $y$ ) и найти локальный оптимум  $y^*$ ;
- 3:  $\rho^* \leftarrow \text{RC}(y^*, B, \text{flip})$ ;  $B \leftarrow B - (\rho^*, \dots, \rho^*)$ ;  $\rho \leftarrow \rho + \rho^*$ ;
- 4:  $I \leftarrow I + 1$ ;  $(\hat{y}, \hat{\rho}) \leftarrow k\text{-Improve}(y^*)$ ;
- 5: **if**  $\hat{y} = y^*$  **or**  $I > I_{\max}$  **then stop**;
- 6: **else**  $\hat{\rho} \leftarrow \text{RC}(\hat{y}, B, \text{flip})$ ;  $B \leftarrow B - (\hat{\rho}, \dots, \hat{\rho})$ ;  $\rho \leftarrow \rho + \hat{\rho}$ ;  $y \leftarrow \hat{y}$ ;
- 7: **goto** 2;

представим в виде вложенной VND-эвристики с двумя этапами. На первом этапе выбираем размещение, фиксируем его и строим относительно него окрестность. На втором этапе просматриваем элементы окрестности и для каждого размещения считаем доход производителя или радиус пороговой устойчивости — в зависимости от применяемого критерия.

В алгоритме VND<sub>1</sub> для выбора наилучшего размещения используем критерий 1. Для каждого размещения  $y$  из окрестности 1-Swap( $y$ ) при помощи алгоритма RC с параметром flip = 1 находим доход производителя и выбираем то размещение, на котором доход производителя наибольший. Далее вычисляем радиус пороговой устойчивости при помощи

**Алгоритм 3.** Алгоритм VND<sub>2</sub>**Вход:**  $I_{\max}$ ,  $B$ ,  $k$ , flip.**Выход:** размещение  $y$  и радиус пороговой устойчивости  $\rho$ .

- 1:  $I \leftarrow 0$ ;  $y \leftarrow \text{rand}\{y' \in \{0, 1\}^n \mid \sum_i y'_i = r\}$  — случайный булев вектор;  
 $\rho \leftarrow \text{RC}(y, B, \text{flip})$ ;  $B \leftarrow B - (\rho, \dots, \rho)$ ;
- 2: применить локальный поиск для 1-Swap( $y$ ) и найти локальный оптимум  $y^*$  с радиусом пороговой устойчивости  $\rho^* = \text{RC}(y^*, B, 1)$ ;
- 3:  $B \leftarrow B - (\rho^*, \dots, \rho^*)$ ;  $\rho \leftarrow \rho + \rho^*$ ;
- 4: **if** flip > 1 **then**
- 5:    $\rho^{**} \leftarrow \text{RC}(y^*, B, \text{flip})$ ;  $B \leftarrow B - (\rho^{**}, \dots, \rho^{**})$ ;  $\rho \leftarrow \rho + \rho^{**}$ ;
- 6:  $I \leftarrow I + 1$ ;  $(\hat{y}, \hat{\rho}) \leftarrow k\text{-Improve}(y^*)$ ;
- 7: **if**  $\hat{y} = y^*$  **or**  $I > I_{\max}$  **then stop**;
- 8: **else**  $B \leftarrow B - (\hat{\rho}, \dots, \hat{\rho})$ ;  $\rho \leftarrow \rho + \hat{\rho}$ ;  $y \leftarrow \hat{y}$ ;
- 9: **goto** 2;

алгоритма RC. Отметим, что алгоритме VND<sub>1</sub> просмотр окрестности занимает меньшее время, чем в VND<sub>2</sub>, так как здесь цены для размещения вычисляются один раз.

В алгоритме VND<sub>2</sub> на шаге 2 применяется локальный поиск в окрестности 1-Swap( $y$ ). Для каждого размещения из окрестности вычисляем радиус пороговой устойчивости при помощи алгоритма RC с параметром flip = 1 и при помощи критерия 2 выбираем локальный оптимум. Если использовать параметр flip, переданный на вход, то поиск локального оптимума сильно замедлится. Для выбранного на шаге 2 локального оптимума пытаемся улучшить результат на шаге 4, используя алгоритм RC и параметр flip, полученный на входе алгоритма.

Таблица 1

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

Пример		Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	%	gap	Время	gap	Время	gap	Время	gap	Время
FLPr01	10	0	1,1 <sub>+2</sub>	6,5 <sub>-3</sub>	1,5 <sub>+0</sub>	0	7,3 <sub>-1</sub>	5,6 <sub>-3</sub>	1,1 <sub>+2</sub>
	30	0	4,2 <sub>+3</sub>	0	1,8 <sub>+1</sub>	0	8,0 <sub>+0</sub>	3,6 <sub>-2</sub>	1,3 <sub>+2</sub>
	50	9,5 <sub>-3</sub>	4,3 <sub>+4</sub>	4,7 <sub>-3</sub>	2,7 <sub>+1</sub>	1,4 <sub>-2</sub>	8,3 <sub>+0</sub>	1,4 <sub>-2</sub>	1,5 <sub>+2</sub>
	70	1,3 <sub>-2</sub>	4,3 <sub>+4</sub>	3,5 <sub>-3</sub>	2,0 <sub>+1</sub>	4,6 <sub>-2</sub>	2,2 <sub>+1</sub>	3,2 <sub>-4</sub>	1,5 <sub>+2</sub>
	90	7,1 <sub>-2</sub>	4,3 <sub>+4</sub>	-3,2 <sub>-2</sub>	5,6 <sub>+1</sub>	-6,7 <sub>-2</sub>	2,8 <sub>+1</sub>	-6,7 <sub>-2</sub>	2,1 <sub>+2</sub>
FLPr02	10	0	1,1 <sub>+2</sub>	1,3 <sub>-9</sub>	4,8 <sub>+0</sub>	1,3 <sub>-9</sub>	1,4 <sub>+0</sub>	1,2 <sub>-2</sub>	2,6 <sub>+2</sub>
	30	0	4,8 <sub>+3</sub>	1,6 <sub>-2</sub>	1,3 <sub>+1</sub>	0	3,9 <sub>+0</sub>	7,1 <sub>-2</sub>	1,6 <sub>+2</sub>
	50	5,0 <sub>-3</sub>	4,3 <sub>+4</sub>	4,7 <sub>-3</sub>	2,3 <sub>+1</sub>	-1,8 <sub>-4</sub>	8,9 <sub>+0</sub>	3,4 <sub>-2</sub>	1,6 <sub>+2</sub>
	70	1,1 <sub>-2</sub>	4,3 <sub>+4</sub>	9,1 <sub>-2</sub>	1,0 <sub>+2</sub>	7,0 <sub>-2</sub>	1,8 <sub>+1</sub>	2,1 <sub>-4</sub>	1,4 <sub>+2</sub>
	90	6,4 <sub>-2</sub>	4,3 <sub>+4</sub>	2,9 <sub>-1</sub>	1,4 <sub>+2</sub>	2,8 <sub>-2</sub>	2,7 <sub>+1</sub>	2,8 <sub>-2</sub>	1,8 <sub>+2</sub>
FLPr03	10	0	8,3 <sub>+1</sub>	0	1,1 <sub>+0</sub>	0	6,4 <sub>-1</sub>	7,5 <sub>-3</sub>	1,3 <sub>+2</sub>
	30	0	2,9 <sub>+3</sub>	6,9 <sub>-4</sub>	9,7 <sub>+0</sub>	2,3 <sub>-3</sub>	4,3 <sub>+0</sub>	5,3 <sub>-2</sub>	8,0 <sub>+1</sub>
	50	4,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-5,4 <sub>-5</sub>	3,2 <sub>+1</sub>	1,7 <sub>-2</sub>	6,8 <sub>+0</sub>	9,8 <sub>-3</sub>	1,5 <sub>+2</sub>
	70	1,8 <sub>-2</sub>	4,3 <sub>+4</sub>	1,9 <sub>-4</sub>	5,2 <sub>+1</sub>	1,9 <sub>-4</sub>	1,2 <sub>+1</sub>	1,9 <sub>-4</sub>	9,4 <sub>+1</sub>
	90	6,8 <sub>-2</sub>	4,3 <sub>+4</sub>	2,8 <sub>-1</sub>	6,8 <sub>+1</sub>	7,7 <sub>-4</sub>	1,7 <sub>+1</sub>	7,7 <sub>-4</sub>	1,0 <sub>+2</sub>
FLPr04	10	0	5,6 <sub>+1</sub>	3,6 <sub>-3</sub>	2,3 <sub>+0</sub>	3,6 <sub>-3</sub>	3,8 <sub>-1</sub>	6,0 <sub>-3</sub>	5,4 <sub>+1</sub>
	30	0	2,3 <sub>+2</sub>	2,9 <sub>-4</sub>	4,3 <sub>+0</sub>	6,9 <sub>-3</sub>	1,7 <sub>+0</sub>	7,9 <sub>-3</sub>	1,1 <sub>+2</sub>
	50	0	1,6 <sub>+3</sub>	0	2,3 <sub>+1</sub>	0	4,0 <sub>+0</sub>	3,0 <sub>-2</sub>	7,3 <sub>+1</sub>
	70	0	8,0 <sub>+3</sub>	0	3,9 <sub>+1</sub>	7,4 <sub>-2</sub>	7,0 <sub>+0</sub>	2,2 <sub>-2</sub>	8,8 <sub>+1</sub>
	90	3,7 <sub>-2</sub>	4,3 <sub>+4</sub>	2,7 <sub>-1</sub>	6,8 <sub>+1</sub>	5,9 <sub>-3</sub>	1,6 <sub>+1</sub>	9,1 <sub>-2</sub>	1,4 <sub>+2</sub>
FLPr05	10	0	1,1 <sub>+2</sub>	7,6 <sub>-3</sub>	9,7 <sub>-1</sub>	4,0 <sub>-3</sub>	9,3 <sub>-1</sub>	1,2 <sub>-2</sub>	2,3 <sub>+2</sub>
	30	0	2,4 <sub>+3</sub>	8,5 <sub>-3</sub>	2,1 <sub>+1</sub>	1,0 <sub>-2</sub>	3,3 <sub>+0</sub>	4,2 <sub>-2</sub>	2,1 <sub>+2</sub>
	50	0	1,3 <sub>+4</sub>	7,4 <sub>-3</sub>	9,3 <sub>+1</sub>	1,5 <sub>-2</sub>	9,2 <sub>+0</sub>	0	4,4 <sub>+2</sub>
	70	7,9 <sub>-3</sub>	4,3 <sub>+4</sub>	7,1 <sub>-5</sub>	1,4 <sub>+2</sub>	1,5 <sub>-2</sub>	3,0 <sub>+1</sub>	1,2 <sub>-1</sub>	2,5 <sub>+2</sub>
	90	5,4 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,2 <sub>-1</sub>	1,2 <sub>+2</sub>	-2,0 <sub>-1</sub>	4,7 <sub>+1</sub>	-9,1 <sub>-2</sub>	3,1 <sub>+2</sub>

Таблица 1 (окончание)

Пример		Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	%	gap	Время	gap	Время	gap	Время	gap	Время
FLPr06	10	0	6,3 <sub>+1</sub>	0	1,7 <sub>+0</sub>	0	7,3 <sub>-1</sub>	2,3 <sub>-2</sub>	9,6 <sub>+1</sub>
	30	0	3,1 <sub>+3</sub>	9,0 <sub>-3</sub>	1,7 <sub>+1</sub>	0	3,1 <sub>+0</sub>	4,8 <sub>-2</sub>	8,1 <sub>+1</sub>
	50	0	1,0 <sub>+4</sub>	9,1 <sub>-3</sub>	1,6 <sub>+1</sub>	1,1 <sub>-2</sub>	6,3 <sub>+0</sub>	6,2 <sub>-2</sub>	1,1 <sub>+2</sub>
	70	9,7 <sub>-3</sub>	4,3 <sub>+4</sub>	1,2 <sub>-1</sub>	6,6 <sub>+1</sub>	1,3 <sub>-4</sub>	2,7 <sub>+1</sub>	9,7 <sub>-2</sub>	8,2 <sub>+1</sub>
	90	5,2 <sub>-2</sub>	4,3 <sub>+4</sub>	2,6 <sub>-1</sub>	7,9 <sub>+1</sub>	6,1 <sub>-2</sub>	4,5 <sub>+1</sub>	-1,4 <sub>-2</sub>	1,6 <sub>+2</sub>
FLPr07	10	0	8,9 <sub>+1</sub>	1,0 <sub>-4</sub>	1,6 <sub>+0</sub>	6,5 <sub>-9</sub>	7,2 <sub>-1</sub>	4,3 <sub>-3</sub>	2,5 <sub>+2</sub>
	30	0	2,7 <sub>+3</sub>	8,7 <sub>-3</sub>	1,5 <sub>+1</sub>	6,6 <sub>-3</sub>	8,5 <sub>+0</sub>	5,1 <sub>-2</sub>	2,6 <sub>+2</sub>
	50	0	1,5 <sub>+4</sub>	1,3 <sub>-2</sub>	1,7 <sub>+1</sub>	0	8,4 <sub>+0</sub>	4,9 <sub>-2</sub>	1,2 <sub>+2</sub>
	70	7,4 <sub>-3</sub>	4,3 <sub>+4</sub>	-9,0 <sub>-3</sub>	5,3 <sub>+1</sub>	-1,7 <sub>-2</sub>	2,5 <sub>+1</sub>	-1,3 <sub>-2</sub>	2,7 <sub>+2</sub>
	90	3,6 <sub>-2</sub>	4,3 <sub>+4</sub>	1,4 <sub>-1</sub>	6,8 <sub>+1</sub>	9,3 <sub>-3</sub>	2,6 <sub>+1</sub>	9,3 <sub>-3</sub>	1,4 <sub>+2</sub>
FLPr08	10	0	5,6 <sub>+1</sub>	0	1,6 <sub>+0</sub>	0	6,1 <sub>-1</sub>	4,9 <sub>-3</sub>	2,7 <sub>+2</sub>
	30	0	3,9 <sub>+3</sub>	0	1,2 <sub>+1</sub>	0	7,4 <sub>+0</sub>	2,9 <sub>-1</sub>	1,3 <sub>+2</sub>
	50	2,1 <sub>-3</sub>	4,3 <sub>+4</sub>	1,6 <sub>-4</sub>	5,4 <sub>+1</sub>	9,2 <sub>-3</sub>	9,1 <sub>+0</sub>	3,5 <sub>-2</sub>	2,1 <sub>+2</sub>
	70	8,6 <sub>-3</sub>	4,3 <sub>+4</sub>	3,3 <sub>-3</sub>	7,1 <sub>+1</sub>	3,3 <sub>-3</sub>	2,8 <sub>+1</sub>	3,2 <sub>-2</sub>	2,0 <sub>+2</sub>
	90	7,2 <sub>-2</sub>	4,3 <sub>+4</sub>	1,5 <sub>-1</sub>	1,6 <sub>+2</sub>	-8,6 <sub>-2</sub>	3,1 <sub>+1</sub>	-7,6 <sub>-2</sub>	1,3 <sub>+2</sub>
FLPr09	10	0	7,3 <sub>+1</sub>	1.4e-14	1,0 <sub>+0</sub>	1.4e-14	6,0 <sub>-1</sub>	3,1 <sub>-1</sub>	1,9 <sub>+0</sub>
	30	0	1,9 <sub>+3</sub>	8,8 <sub>-3</sub>	1,5 <sub>+1</sub>	2,6 <sub>-3</sub>	2,6 <sub>+0</sub>	4,6 <sub>-2</sub>	1,2 <sub>+2</sub>
	50	0	6,3 <sub>+3</sub>	0	3,1 <sub>+1</sub>	6,9 <sub>-3</sub>	6,0 <sub>+0</sub>	1,0 <sub>-3</sub>	1,1 <sub>+2</sub>
	70	2,7 <sub>-3</sub>	4,3 <sub>+4</sub>	2,4 <sub>-2</sub>	3,9 <sub>+1</sub>	2,1 <sub>-4</sub>	1,0 <sub>+1</sub>	2,1 <sub>-4</sub>	2,1 <sub>+2</sub>
	90	3,2 <sub>-2</sub>	4,3 <sub>+4</sub>	1,9 <sub>-1</sub>	6,5 <sub>+1</sub>	5,1 <sub>-4</sub>	1,7 <sub>+1</sub>	2,1 <sub>-2</sub>	9,6 <sub>+1</sub>
FLPr10	10	0	5,5 <sub>+1</sub>	8,9 <sub>-9</sub>	3,2 <sub>+0</sub>	1,0 <sub>-3</sub>	1,3 <sub>+0</sub>	5,6 <sub>-3</sub>	2,3 <sub>+2</sub>
	30	0	3,2 <sub>+3</sub>	2,2 <sub>-3</sub>	1,4 <sub>+1</sub>	4,8 <sub>-3</sub>	3,9 <sub>+0</sub>	4,7 <sub>-3</sub>	1,5 <sub>+2</sub>
	50	8,0 <sub>-4</sub>	4,3 <sub>+4</sub>	5,2 <sub>-3</sub>	6,6 <sub>+1</sub>	1,8 <sub>-4</sub>	8,7 <sub>+0</sub>	1,7 <sub>-3</sub>	9,5 <sub>+1</sub>
	70	5,7 <sub>-3</sub>	4,3 <sub>+4</sub>	2,8 <sub>-4</sub>	5,2 <sub>+1</sub>	2,8 <sub>-4</sub>	1,6 <sub>+1</sub>	2,8 <sub>-4</sub>	9,8 <sub>+1</sub>
	90	6,0 <sub>-2</sub>	4,3 <sub>+4</sub>	2,1 <sub>-1</sub>	7,8 <sub>+1</sub>	-9,1 <sub>-4</sub>	2,5 <sub>+1</sub>	-9,1 <sub>-4</sub>	1,1 <sub>+2</sub>

В обоих алгоритмах найденный радиус пороговой устойчивости сразу вычитается из бюджетов потребителей. При вызове алгоритма RC внутри VND вектор бюджетов меняется локально внутри RC. Вычитание радиуса из бюджетов не влияет на результат алгоритма VND (пройденный им путь), но сильно снижает время работы. В численном эксперименте сравним алгоритм VND<sub>2</sub> с его версией, реализованной без вычитания радиуса пороговой устойчивости из бюджетов потребителей.

Предложенный алгоритм допускает использование мултистарта: запуск на разных стартовых решениях и выбор лучшего из найденных решений. Результаты такого подхода показаны далее.

#### 4. Численный эксперимент

Для численного эксперимента использованы входные данные, взятые из библиотеки «Дискретные задачи размещения», и данные, сгенерированные случайным образом. Сначала готовим входные данные для исходной задачи, решаем её и в результате определяем максимально возможный доход производителя  $V$ . Далее фиксируем вход исходной задачи, берём некоторую часть от  $V$  в качестве порогового ограничения и получаем вход для задачи пороговой устойчивости. Варьируя пороговое ограничение таким образом, для одного входа исходной задачи можно сгенерировать несколько входов задачи пороговой устойчивости. Более подробно эта процедура описана в [3].

Таблица 2

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

Пример		Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	%	gap	Время	gap	Время	gap	Время	gap	Время
FLPr01	10	0	5,4 <sub>+2</sub>	3,3 <sub>-9</sub>	2,3 <sub>+1</sub>	3,3 <sub>-9</sub>	4,2 <sub>+0</sub>	1,1 <sub>-2</sub>	1,3 <sub>+3</sub>
	30	0	1,3 <sub>+4</sub>	8,0 <sub>-4</sub>	6,6 <sub>+1</sub>	0	2,4 <sub>+1</sub>	8,0 <sub>-4</sub>	1,2 <sub>+3</sub>
	50	1,1 <sub>-2</sub>	4,3 <sub>+4</sub>	4,1 <sub>-3</sub>	2,1 <sub>+2</sub>	4,1 <sub>-3</sub>	5,8 <sub>+1</sub>	-4,2 <sub>-5</sub>	1,1 <sub>+3</sub>
	70	1,6 <sub>-2</sub>	4,3 <sub>+4</sub>	-4,5 <sub>-3</sub>	3,9 <sub>+2</sub>	2,2 <sub>-4</sub>	1,4 <sub>+2</sub>	-6,6 <sub>-3</sub>	8,5 <sub>+2</sub>
	90	1,2 <sub>-1</sub>	4,3 <sub>+4</sub>	1,8 <sub>-1</sub>	6,2 <sub>+2</sub>	-3,7 <sub>-2</sub>	3,0 <sub>+2</sub>	-1,8 <sub>-2</sub>	1,1 <sub>+3</sub>
FLPr02	10	0	6,2 <sub>+2</sub>	0	2,5 <sub>+1</sub>	0	1,1 <sub>+1</sub>	9,2 <sub>-3</sub>	2,1 <sub>+3</sub>
	30	7,1 <sub>-3</sub>	4,3 <sub>+4</sub>	5,3 <sub>-3</sub>	1,5 <sub>+2</sub>	5,5 <sub>-3</sub>	5,7 <sub>+1</sub>	4,8 <sub>-2</sub>	1,3 <sub>+3</sub>
	50	1,7 <sub>-2</sub>	4,3 <sub>+4</sub>	-2,9 <sub>-2</sub>	4,1 <sub>+2</sub>	-3,4 <sub>-2</sub>	1,3 <sub>+2</sub>	-3,4 <sub>-2</sub>	3,8 <sub>+3</sub>
	70	2,4 <sub>-2</sub>	4,3 <sub>+4</sub>	-4,5 <sub>-2</sub>	4,0 <sub>+2</sub>	-4,8 <sub>-2</sub>	1,2 <sub>+2</sub>	-4,5 <sub>-2</sub>	2,2 <sub>+3</sub>
	90	1,0 <sub>-1</sub>	4,3 <sub>+4</sub>	2,5 <sub>-1</sub>	7,1 <sub>+2</sub>	1,7 <sub>-1</sub>	3,6 <sub>+2</sub>	-4,1 <sub>-2</sub>	1,1 <sub>+3</sub>
FLPr03	10	0	4,0 <sub>+2</sub>	1,7 <sub>-3</sub>	1,4 <sub>+1</sub>	0	4,7 <sub>+0</sub>	1,0 <sub>-2</sub>	9,2 <sub>+2</sub>
	30	0	9,2 <sub>+3</sub>	0	1,0 <sub>+2</sub>	0	2,0 <sub>+1</sub>	2,0 <sub>-2</sub>	2,0 <sub>+3</sub>
	50	1,3 <sub>-2</sub>	4,3 <sub>+4</sub>	3,9 <sub>-3</sub>	3,0 <sub>+2</sub>	6,9 <sub>-4</sub>	5,7 <sub>+1</sub>	6,1 <sub>-5</sub>	1,7 <sub>+3</sub>
	70	2,3 <sub>-2</sub>	4,3 <sub>+4</sub>	-5,2 <sub>-2</sub>	4,2 <sub>+2</sub>	-4,6 <sub>-2</sub>	9,1 <sub>+1</sub>	9,1 <sub>-3</sub>	1,2 <sub>+3</sub>
	90	8,6 <sub>-2</sub>	4,3 <sub>+4</sub>	2,9 <sub>-1</sub>	5,0 <sub>+2</sub>	1,5 <sub>-1</sub>	1,5 <sub>+2</sub>	1,0 <sub>-1</sub>	1,9 <sub>+3</sub>
FLPr04	10	0	4,2 <sub>+2</sub>	2,0 <sub>-4</sub>	1,4 <sub>+1</sub>	2,2 <sub>-3</sub>	6,5 <sub>+0</sub>	1,2 <sub>-2</sub>	1,0 <sub>+3</sub>
	30	0	1,2 <sub>+4</sub>	0	1,8 <sub>+2</sub>	0	3,6 <sub>+1</sub>	4,7 <sub>-2</sub>	2,3 <sub>+3</sub>
	50	1,6 <sub>-2</sub>	4,3 <sub>+4</sub>	-9,2 <sub>-5</sub>	4,1 <sub>+2</sub>	-9,2 <sub>-5</sub>	6,8 <sub>+1</sub>	1,0 <sub>-2</sub>	1,0 <sub>+3</sub>
	70	3,6 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,5 <sub>-2</sub>	3,3 <sub>+2</sub>	6,8 <sub>-3</sub>	1,5 <sub>+2</sub>	-1,5 <sub>-2</sub>	1,6 <sub>+3</sub>
	90	1,1 <sub>-1</sub>	4,3 <sub>+4</sub>	1,4 <sub>-1</sub>	9,6 <sub>+2</sub>	-1,3 <sub>-1</sub>	3,1 <sub>+2</sub>	-1,3 <sub>-1</sub>	2,1 <sub>+3</sub>
FLPr05	10	0	5,2 <sub>+2</sub>	1,5 <sub>-8</sub>	9,0 <sub>+0</sub>	1,5 <sub>-8</sub>	8,2 <sub>+0</sub>	3,7 <sub>-3</sub>	1,4 <sub>+3</sub>
	30	0	1,1 <sub>+4</sub>	0	9,2 <sub>+1</sub>	0	3,2 <sub>+1</sub>	1,6 <sub>-2</sub>	9,6 <sub>+2</sub>
	50	1,1 <sub>-2</sub>	4,3 <sub>+4</sub>	5,3 <sub>-3</sub>	1,8 <sub>+2</sub>	8,5 <sub>-3</sub>	5,1 <sub>+1</sub>	1,2 <sub>-2</sub>	1,4 <sub>+3</sub>
	70	1,5 <sub>-2</sub>	4,3 <sub>+4</sub>	6,7 <sub>-3</sub>	2,5 <sub>+2</sub>	1,5 <sub>-2</sub>	1,6 <sub>+2</sub>	6,7 <sub>-5</sub>	9,6 <sub>+2</sub>
	90	7,8 <sub>-2</sub>	4,3 <sub>+4</sub>	1,9 <sub>-1</sub>	3,9 <sub>+2</sub>	0	2,2 <sub>+2</sub>	0	1,2 <sub>+3</sub>

Таблица 2 (окончание)

Пример		Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	%	gap	Время	gap	Время	gap	Время	gap	Время
FLPr06	10	0	6,1 <sub>+2</sub>	1,8 <sub>-3</sub>	1,3 <sub>+1</sub>	7,0 <sub>-10</sub>	5,2 <sub>+0</sub>	4,6 <sub>-3</sub>	1,7 <sub>+3</sub>
	30	0	1,0 <sub>+4</sub>	0	1,3 <sub>+2</sub>	0	2,4 <sub>+1</sub>	6,4 <sub>-3</sub>	1,5 <sub>+3</sub>
	50	4,6 <sub>-3</sub>	4,3 <sub>+4</sub>	-1,5 <sub>-2</sub>	3,4 <sub>+2</sub>	-1,5 <sub>-2</sub>	8,0 <sub>+1</sub>	-1,3 <sub>-2</sub>	1,0 <sub>+3</sub>
	70	1,4 <sub>-2</sub>	4,3 <sub>+4</sub>	2,2 <sub>-2</sub>	4,4 <sub>+2</sub>	-4,4 <sub>-2</sub>	1,2 <sub>+2</sub>	-3,4 <sub>-2</sub>	1,6 <sub>+3</sub>
	90	1,0 <sub>+7</sub>	4,3 <sub>+4</sub>	-1,0 <sub>+0</sub>	9,1 <sub>+2</sub>	-1,0 <sub>+0</sub>	1,9 <sub>+2</sub>	-1,0 <sub>+0</sub>	1,0 <sub>+3</sub>
FLPr07	10	0	4,8 <sub>+2</sub>	7,6 <sub>-4</sub>	2,2 <sub>+1</sub>	7,6 <sub>-4</sub>	4,0 <sub>+0</sub>	3,6 <sub>-3</sub>	1,7 <sub>+3</sub>
	30	0	9,1 <sub>+3</sub>	0	8,4 <sub>+1</sub>	0	2,4 <sub>+1</sub>	1,5 <sub>-2</sub>	8,4 <sub>+2</sub>
	50	5,0 <sub>-3</sub>	4,3 <sub>+4</sub>	8,3 <sub>-3</sub>	2,6 <sub>+2</sub>	8,0 <sub>-3</sub>	5,8 <sub>+1</sub>	-1,6 <sub>-4</sub>	1,3 <sub>+3</sub>
	70	2,9 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,9 <sub>-3</sub>	4,0 <sub>+2</sub>	1,2 <sub>-2</sub>	1,4 <sub>+2</sub>	2,7 <sub>-2</sub>	8,3 <sub>+2</sub>
	90	4,5 <sub>-2</sub>	4,3 <sub>+4</sub>	2,5 <sub>-1</sub>	5,0 <sub>+2</sub>	2,1 <sub>-2</sub>	1,8 <sub>+2</sub>	-9,1 <sub>-2</sub>	1,3 <sub>+3</sub>
FLPr08	10	0	4,6 <sub>+2</sub>	3.5e-14	1,8 <sub>+1</sub>	3.7e-14	4,0 <sub>+0</sub>	1,6 <sub>-2</sub>	8,2 <sub>+2</sub>
	30	0	1,2 <sub>+4</sub>	0	5,6 <sub>+1</sub>	0	4,3 <sub>+1</sub>	4,4 <sub>-3</sub>	8,4 <sub>+2</sub>
	50	1,4 <sub>-2</sub>	4,3 <sub>+4</sub>	9,5 <sub>-5</sub>	1,8 <sub>+2</sub>	2,8 <sub>-2</sub>	5,2 <sub>+1</sub>	1,3 <sub>-2</sub>	1,7 <sub>+3</sub>
	70	2,9 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,2 <sub>-2</sub>	2,5 <sub>+2</sub>	4,4 <sub>-3</sub>	9,7 <sub>+1</sub>	-2,1 <sub>-2</sub>	8,7 <sub>+2</sub>
	90	1,3 <sub>-1</sub>	4,3 <sub>+4</sub>	8,1 <sub>-2</sub>	8,2 <sub>+2</sub>	-1,6 <sub>-1</sub>	2,5 <sub>+2</sub>	-1,6 <sub>-1</sub>	9,5 <sub>+2</sub>
FLPr09	10	0	7,4 <sub>+2</sub>	1,5 <sub>-3</sub>	2,5 <sub>+1</sub>	0	4,3 <sub>+0</sub>	2,1 <sub>-3</sub>	9,0 <sub>+2</sub>
	30	0	1,1 <sub>+4</sub>	0	9,7 <sub>+1</sub>	0	2,4 <sub>+1</sub>	5,1 <sub>-2</sub>	7,2 <sub>+2</sub>
	50	1,9 <sub>-2</sub>	4,3 <sub>+4</sub>	-3,3 <sub>-2</sub>	3,7 <sub>+2</sub>	-4,0 <sub>-2</sub>	1,0 <sub>+2</sub>	1,1 <sub>-1</sub>	5,4 <sub>+2</sub>
	70	3,9 <sub>-2</sub>	4,3 <sub>+4</sub>	-5,4 <sub>-2</sub>	3,8 <sub>+2</sub>	-5,0 <sub>-2</sub>	9,8 <sub>+1</sub>	-6,1 <sub>-2</sub>	6,1 <sub>+2</sub>
	90	1,4 <sub>-1</sub>	4,3 <sub>+4</sub>	2,1 <sub>-1</sub>	5,0 <sub>+2</sub>	-5,1 <sub>-2</sub>	2,7 <sub>+2</sub>	-5,1 <sub>-2</sub>	5,7 <sub>+2</sub>
FLPr10	10	0	6,8 <sub>+2</sub>	3,0 <sub>-4</sub>	1,6 <sub>+1</sub>	2,8 <sub>-9</sub>	8,3 <sub>+0</sub>	3,0 <sub>-4</sub>	1,4 <sub>+3</sub>
	30	2,1 <sub>-3</sub>	4,3 <sub>+4</sub>	-3,3 <sub>-3</sub>	1,3 <sub>+2</sub>	-8,1 <sub>-5</sub>	5,0 <sub>+1</sub>	1,4 <sub>-2</sub>	1,3 <sub>+3</sub>
	50	1,3 <sub>-2</sub>	4,3 <sub>+4</sub>	3,1 <sub>-3</sub>	2,0 <sub>+2</sub>	1,2 <sub>-4</sub>	1,6 <sub>+2</sub>	1,2 <sub>-4</sub>	1,4 <sub>+3</sub>
	70	1,7 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,0 <sub>-2</sub>	6,6 <sub>+2</sub>	4,9 <sub>-3</sub>	1,2 <sub>+2</sub>	-2,6 <sub>-2</sub>	1,1 <sub>+3</sub>
	90	9,4 <sub>-2</sub>	4,3 <sub>+4</sub>	4,3 <sub>-1</sub>	4,7 <sub>+2</sub>	-3,0 <sub>-3</sub>	2,0 <sub>+2</sub>	5,4 <sub>-2</sub>	2,0 <sub>+3</sub>

Сначала рассмотрим примеры задач из библиотеки. Для поиска точного решения применяем решатель Gurobi версии 10. Тестирование производится на сервере с двумя процессорами AMD EPYC 7502 32-Core и 512 ГБ оперативной памяти. При этом решатель Gurobi может использовать свободные ядра, и каждую задачу решаем параллельно на 7 ядрах. Время работы решателя для каждой задачи ограничено 12 ч. Основной алгоритм выполняем одним потоком, т. е. на одном ядре.

Эмпирически подобраны наилучшие параметры  $k = 2$  и  $\text{flip} = 2$ , которые применены в дальнейших экспериментах. Результаты, отражённые в табл. 1 и 2, получены на входных данных из библиотеки «Дискретные задачи размещения», а в табл. 3 и 4 — на входных данных, сгенерированных случайным образом. Полученные значения представлены в экспоненциальной записи, при этом порядок числа приводится в нижнем



индексе мантииссы для экономии места. В колонке % указана доля максимального дохода производителя, взятая в качестве порога  $V$ . Красным отмечено оптимальное значение gap, а синим — значение gap в случае, если решение предложенного алгоритма лучше в сравнении с решением Gurobi.

Размерность каждого входа в табл. 1 фиксирована числами  $n = 100$ ,  $m = 40$ ,  $r = 5$ .  $VND'_2$  — это версия алгоритма  $VND_2$  без вычитания радиуса пороговой устойчивости из бюджетов потребителей. Время работы  $VND'_2$  больше, чем  $VND_2$ , на всех примерах в среднем в 13 раз. Ускорение происходит в силу того, что при вычитании радиуса пороговой устойчивости из бюджетов потребителей алгоритм чаще останавливается на шаге 4 алгоритма RC, тем самым экономя время на поиске цен. В 24 из 50 случаев решатель не смог найти оптимального решения. Алгоритм  $VND_2$  с критерием 2 находит в среднем более качественное решение.

Размерность каждого входа в табл. 2 фиксирована числами  $n = 100$ ,  $m = 100$ ,  $r = 5$ . Как в предыдущем случае, время работы алгоритма

Таблица 3

## Результаты эксперимента на средней и большой размерностях

Пример						Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	$n$	$m$	$r$	$V$	%	gap	Время	gap	Время	gap	Время	gap	Время
1	100	100	10	621	10	3,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-7,5 <sub>-3</sub>	6,4 <sub>+4</sub>	-8,7 <sub>-5</sub>	4,8 <sub>+3</sub>	2,0 <sub>-1</sub>	2,1 <sub>+4</sub>
				1863	30	9,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-3,3 <sub>-2</sub>	1,4 <sub>+5</sub>	-2,9 <sub>-2</sub>	1,4 <sub>+4</sub>	-6,4 <sub>-3</sub>	1,1 <sub>+4</sub>
				3105	50	1,8 <sub>-2</sub>	4,3 <sub>+4</sub>	-2,5 <sub>-2</sub>	1,5 <sub>+5</sub>	-1,3 <sub>-2</sub>	1,5 <sub>+4</sub>	-1,4 <sub>-2</sub>	2,6 <sub>+4</sub>
				4347	70	2,2 <sub>-2</sub>	4,3 <sub>+4</sub>	-5,9 <sub>-2</sub>	1,6 <sub>+5</sub>	-6,0 <sub>-2</sub>	1,9 <sub>+4</sub>	-5,1 <sub>-2</sub>	1,3 <sub>+4</sub>
				5589	90	1,0 <sub>+7</sub>	4,3 <sub>+4</sub>	-1,0 <sub>+0</sub>	1,1 <sub>+5</sub>	-1,0 <sub>+0</sub>	1,8 <sub>+4</sub>	-1,0 <sub>+0</sub>	6,2 <sub>+4</sub>
2	20	20	10	101	10	0	9,7 <sub>+0</sub>	1,5 <sub>-2</sub>	4,6 <sub>-1</sub>	1,3 <sub>-8</sub>	3,1 <sub>-1</sub>	5,8 <sub>-3</sub>	1,2 <sub>+1</sub>
				303	30	0	2,3 <sub>+1</sub>	2,4 <sub>-1</sub>	3,1 <sub>+0</sub>	2,1 <sub>-1</sub>	7,7 <sub>-1</sub>	2,4 <sub>-1</sub>	5,7 <sub>+0</sub>
				505	50	0	2,5 <sub>+2</sub>	7,9 <sub>-2</sub>	1,4 <sub>+1</sub>	7,5 <sub>-2</sub>	1,6 <sub>+0</sub>	7,0 <sub>-2</sub>	1,6 <sub>+1</sub>
				707	70	0	4,6 <sub>+2</sub>	1,2 <sub>-1</sub>	2,8 <sub>+0</sub>	5,0 <sub>-2</sub>	2,2 <sub>+0</sub>	3,7 <sub>-2</sub>	3,7 <sub>+0</sub>
				909	90	0	5,1 <sub>+2</sub>	8,8 <sub>-2</sub>	1,1 <sub>+1</sub>	1,5 <sub>-1</sub>	2,5 <sub>+0</sub>	5,0 <sub>-2</sub>	4,1 <sub>+1</sub>
3	20	20	15	128	10	0	2,6 <sub>+1</sub>	2,3 <sub>-2</sub>	1,1 <sub>+1</sub>	1,5 <sub>-1</sub>	4,1 <sub>-1</sub>	1,7 <sub>-1</sub>	4,1 <sub>+1</sub>
				384	30	0	9,9 <sub>+1</sub>	2,0 <sub>-2</sub>	4,9 <sub>+0</sub>	1,5 <sub>-2</sub>	3,4 <sub>+0</sub>	1,3 <sub>-1</sub>	7,0 <sub>+0</sub>
				641	50	0	8,4 <sub>+1</sub>	1,1 <sub>-1</sub>	2,2 <sub>+1</sub>	1,1 <sub>-1</sub>	3,5 <sub>+0</sub>	9,9 <sub>-2</sub>	8,3 <sub>+0</sub>
				898	70	0	1,5 <sub>+2</sub>	1,5 <sub>-1</sub>	4,1 <sub>+0</sub>	8,5 <sub>-2</sub>	5,6 <sub>+0</sub>	8,5 <sub>-2</sub>	3,7 <sub>+1</sub>
				1154	90	0	1,3 <sub>+2</sub>	3,6 <sub>-1</sub>	2,8 <sub>+1</sub>	2,1 <sub>-1</sub>	3,2 <sub>+0</sub>	1,9 <sub>-1</sub>	3,4 <sub>+1</sub>
4	40	40	10	288	10	2,0 <sub>-4</sub>	4,3 <sub>+4</sub>	5,9 <sub>-3</sub>	1,6 <sub>+2</sub>	4,8 <sub>-3</sub>	2,6 <sub>+1</sub>	7,7 <sub>-2</sub>	1,1 <sub>+3</sub>
				866	30	1,3 <sub>-3</sub>	4,3 <sub>+4</sub>	1,6 <sub>-3</sub>	5,8 <sub>+2</sub>	1,2 <sub>-2</sub>	2,2 <sub>+2</sub>	1,6 <sub>-2</sub>	1,8 <sub>+3</sub>
				1443	50	2,6 <sub>-3</sub>	4,3 <sub>+4</sub>	-2,2 <sub>-3</sub>	1,0 <sub>+3</sub>	3,5 <sub>-3</sub>	1,2 <sub>+2</sub>	1,1 <sub>-2</sub>	8,6 <sub>+2</sub>
				2020	70	4,0 <sub>-3</sub>	4,3 <sub>+4</sub>	3,7 <sub>-3</sub>	1,1 <sub>+3</sub>	-6,7 <sub>-3</sub>	1,3 <sub>+2</sub>	1,4 <sub>-3</sub>	1,2 <sub>+3</sub>
				2598	90	5,9 <sub>-3</sub>	4,3 <sub>+4</sub>	1,5 <sub>-2</sub>	1,1 <sub>+3</sub>	7,0 <sub>-4</sub>	2,5 <sub>+2</sub>	8,8 <sub>-2</sub>	5,8 <sub>+2</sub>

Таблица 3 (окончание)

Пример						Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	n	m	r	V	%	gap	Время	gap	Время	gap	Время	gap	Время
5	40	40	15	225	10	0	2,3 <sub>+2</sub>	1,5 <sub>-2</sub>	2,3 <sub>+2</sub>	1,5 <sub>-2</sub>	2,1 <sub>+1</sub>	9,1 <sub>-3</sub>	2,9 <sub>+3</sub>
				677	30	1,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-7,4 <sub>-4</sub>	1,0 <sub>+3</sub>	7,4 <sub>-4</sub>	1,9 <sub>+2</sub>	4,0 <sub>-2</sub>	2,7 <sub>+3</sub>
				1129	50	3,0 <sub>-3</sub>	4,3 <sub>+4</sub>	8,3 <sub>-4</sub>	2,5 <sub>+3</sub>	1,2 <sub>-2</sub>	2,1 <sub>+2</sub>	2,8 <sub>-3</sub>	5,4 <sub>+2</sub>
				1581	70	4,7 <sub>-3</sub>	4,3 <sub>+4</sub>	1,7 <sub>-2</sub>	1,4 <sub>+3</sub>	-1,7 <sub>-2</sub>	5,4 <sub>+2</sub>	1,4 <sub>-2</sub>	1,2 <sub>+3</sub>
				2033	90	5,9 <sub>-3</sub>	4,3 <sub>+4</sub>	1,6 <sub>-1</sub>	1,7 <sub>+3</sub>	1,5 <sub>-1</sub>	4,8 <sub>+2</sub>	-3,6 <sub>-3</sub>	8,5 <sub>+2</sub>
6	60	60	10	411	10	1,0 <sub>-3</sub>	4,3 <sub>+4</sub>	5,4 <sub>-3</sub>	4,8 <sub>+3</sub>	3,6 <sub>-3</sub>	2,1 <sub>+2</sub>	1,5 <sub>-1</sub>	1,2 <sub>+4</sub>
				1234	30	4,6 <sub>-3</sub>	4,3 <sub>+4</sub>	-1,9 <sub>-2</sub>	6,6 <sub>+3</sub>	-1,6 <sub>-2</sub>	7,6 <sub>+2</sub>	-9,6 <sub>-3</sub>	5,5 <sub>+3</sub>
				2058	50	8,9 <sub>-3</sub>	4,3 <sub>+4</sub>	-7,4 <sub>-3</sub>	5,4 <sub>+3</sub>	-1,5 <sub>-2</sub>	1,4 <sub>+3</sub>	5,2 <sub>-3</sub>	4,0 <sub>+3</sub>
				2881	70	1,0 <sub>-2</sub>	4,3 <sub>+4</sub>	1,8 <sub>-2</sub>	1,2 <sub>+4</sub>	3,0 <sub>-2</sub>	4,0 <sub>+3</sub>	4,6 <sub>-2</sub>	4,9 <sub>+3</sub>
				3704	90	5,0 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,8 <sub>-1</sub>	1,8 <sub>+4</sub>	-1,3 <sub>-1</sub>	2,8 <sub>+3</sub>	-1,4 <sub>-1</sub>	3,6 <sub>+3</sub>
7	60	60	15	381	10	1,5 <sub>-3</sub>	4,3 <sub>+4</sub>	8,9 <sub>-3</sub>	2,0 <sub>+3</sub>	3,2 <sub>-3</sub>	2,0 <sub>+2</sub>	6,5 <sub>-3</sub>	2,0 <sub>+4</sub>
				1143	30	3,5 <sub>-3</sub>	4,3 <sub>+4</sub>	-1,2 <sub>-2</sub>	3,6 <sub>+4</sub>	-1,0 <sub>-2</sub>	4,1 <sub>+3</sub>	-5,5 <sub>-3</sub>	1,3 <sub>+4</sub>
				1906	50	7,4 <sub>-3</sub>	4,3 <sub>+4</sub>	8,4 <sub>-2</sub>	2,0 <sub>+4</sub>	7,3 <sub>-2</sub>	3,7 <sub>+3</sub>	8,0 <sub>-2</sub>	1,4 <sub>+4</sub>
				2669	70	1,8 <sub>-2</sub>	4,3 <sub>+4</sub>	-5,3 <sub>-2</sub>	3,6 <sub>+4</sub>	-4,5 <sub>-2</sub>	5,1 <sub>+3</sub>	-7,6 <sub>-3</sub>	8,1 <sub>+3</sub>
				3431	90	5,0 <sub>-2</sub>	4,3 <sub>+4</sub>	-8,9 <sub>-2</sub>	9,8 <sub>+3</sub>	6,3 <sub>-3</sub>	4,0 <sub>+3</sub>	-3,1 <sub>-2</sub>	1,2 <sub>+4</sub>
8	90	90	10	583	10	1,5 <sub>-3</sub>	4,3 <sub>+4</sub>	1,4 <sub>-3</sub>	4,9 <sub>+3</sub>	1,8 <sub>-3</sub>	5,7 <sub>+2</sub>	7,8 <sub>-2</sub>	1,6 <sub>+4</sub>
				1749	30	6,6 <sub>-3</sub>	4,3 <sub>+4</sub>	-7,1 <sub>-3</sub>	7,5 <sub>+3</sub>	-1,0 <sub>-2</sub>	3,4 <sub>+3</sub>	-1,7 <sub>-2</sub>	3,2 <sub>+4</sub>
				2915	50	4,6 <sub>-3</sub>	4,3 <sub>+4</sub>	-3,1 <sub>-2</sub>	1,7 <sub>+4</sub>	-4,7 <sub>-2</sub>	1,1 <sub>+4</sub>	-4,0 <sub>-2</sub>	1,3 <sub>+4</sub>
				4081	70	8,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-5,0 <sub>-2</sub>	3,4 <sub>+4</sub>	-5,1 <sub>-2</sub>	1,5 <sub>+4</sub>	-1,8 <sub>-2</sub>	3,2 <sub>+4</sub>
				5247	90	5,1 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,4 <sub>-1</sub>	3,7 <sub>+4</sub>	-1,6 <sub>-1</sub>	1,2 <sub>+4</sub>	-5,8 <sub>-2</sub>	1,9 <sub>+4</sub>

VND'<sub>2</sub> в сравнении с VND<sub>2</sub> больше на всех примерах в среднем в 13 раз. В 32 случаях из 50 решателю не удалось найти оптимального решения, а в одном случае — хотя бы допустимого. Алгоритм VND<sub>2</sub> с критерием 2 в среднем находит решение лучше. Это происходит, скорее всего, в силу того, что критерий 2 нацелен на поиск максимального радиуса, а не дохода производителя. Время работы решателя значительно превышает время работы алгоритмов VND.

Согласно табл. 3 на примерах средней или большой размерности алгоритм VND работает продолжительное время. Здесь отчётливо видно различие во времени работы алгоритмов VND<sub>2</sub> и VND'<sub>2</sub>. Количество найденных оптимумов — 11 из 40. Так же есть пример, в котором решатель Gurobi не смог найти допустимого решения за отведённое время работы. Критерий 2 на этих примерах наиболее предпочтителен.

Результаты в табл. 4 получены на входных данных малой размерности. Решатель Gurobi нашёл 29 оптимальных решений на 50 примерах. Алгоритм VND<sub>2</sub> находит решение в среднем немного хуже, чем решатель, но время работы в тысячи раз меньше.

На рис. 1 отображены средние показатели алгоритмов с разными параметрами. По горизонтали отмечен алгоритм  $VND(k, \text{flip})$  с параметрами  $k$  и  $\text{flip}$ . Запись  $VND(k, \text{flip})(x)$  означает, что использована идея мультистарта и этот алгоритм запускался  $x$  раз. Приведены значение  $\text{gap}$  и время работы алгоритмов, масштабированные по максимальному значению, а максимальное указано в легенде. Из рис. 1 видно, что идея мультистарта позволяет найти решение лучше, при этом время работы увеличивается пропорционально количеству запусков.

Если ограничиться примерами, для которых найден оптимум, то для алгоритма  $VND$  наилучшими параметрами будут  $k = 2$  и  $\text{flip} = 2$ . При увеличении одного из параметров время счёта сильно возрастает, а значение целевой функции ухудшается незначительно.

Таблица 4

## Результаты численного эксперимента на малой размерности

Пример						Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	$n$	$m$	$r$	$V$	%	gap	Время	gap	Время	gap	Время	gap	Время
1	30	30	5	177	10	4,0 <sub>-4</sub>	4,3 <sub>+4</sub>	1,9 <sub>-3</sub>	6,4 <sub>+0</sub>	3,8 <sub>-5</sub>	1,5 <sub>+0</sub>	4,9 <sub>-2</sub>	8,5 <sub>+1</sub>
				531	30	2,1 <sub>-3</sub>	4,3 <sub>+4</sub>	1,5 <sub>-1</sub>	9,2 <sub>+0</sub>	1,4 <sub>-1</sub>	5,0 <sub>+0</sub>	1,7 <sub>-2</sub>	1,4 <sub>+2</sub>
				885	50	5,5 <sub>-3</sub>	4,3 <sub>+4</sub>	8,1 <sub>-2</sub>	1,5 <sub>+1</sub>	7,8 <sub>-2</sub>	5,1 <sub>+0</sub>	8,3 <sub>-2</sub>	3,9 <sub>+1</sub>
				1239	70	1,1 <sub>-2</sub>	4,3 <sub>+4</sub>	3,0 <sub>-2</sub>	9,3 <sub>+0</sub>	1,1 <sub>-2</sub>	6,7 <sub>+0</sub>	-1,9 <sub>-4</sub>	3,9 <sub>+1</sub>
				1593	90	3,0 <sub>-2</sub>	4,3 <sub>+4</sub>	2,8 <sub>-2</sub>	1,2 <sub>+1</sub>	5,6 <sub>-2</sub>	6,2 <sub>+0</sub>	5,1 <sub>-3</sub>	3,6 <sub>+1</sub>
2	35	35	5	167	10	0	1,6 <sub>+2</sub>	2,2 <sub>-3</sub>	1,8 <sub>+0</sub>	2,2 <sub>-3</sub>	5,8 <sub>-1</sub>	6,8 <sub>-3</sub>	7,6 <sub>+1</sub>
				503	30	2,5 <sub>-3</sub>	4,3 <sub>+4</sub>	2,8 <sub>-2</sub>	2,4 <sub>+1</sub>	2,0 <sub>-2</sub>	3,6 <sub>+0</sub>	9,3 <sub>-2</sub>	2,3 <sub>+2</sub>
				839	50	7,8 <sub>-3</sub>	4,3 <sub>+4</sub>	5,5 <sub>-3</sub>	4,0 <sub>+1</sub>	-3,4 <sub>-3</sub>	9,4 <sub>+0</sub>	8,6 <sub>-3</sub>	1,6 <sub>+2</sub>
				1175	70	1,2 <sub>-2</sub>	4,3 <sub>+4</sub>	7,4 <sub>-3</sub>	1,9 <sub>+1</sub>	-2,4 <sub>-2</sub>	2,0 <sub>+1</sub>	-2,3 <sub>-2</sub>	7,3 <sub>+1</sub>
				1511	90	2,2 <sub>-2</sub>	4,3 <sub>+4</sub>	-8,6 <sub>-3</sub>	1,6 <sub>+1</sub>	-8,6 <sub>-3</sub>	1,5 <sub>+1</sub>	-8,6 <sub>-3</sub>	6,6 <sub>+1</sub>
3	35	35	5	170	10	0	1,3 <sub>+2</sub>	2,3 <sub>-3</sub>	1,8 <sub>+0</sub>	1,3 <sub>-8</sub>	7,5 <sub>-1</sub>	2,3 <sub>-2</sub>	9,6 <sub>+1</sub>
				512	30	2,3 <sub>-3</sub>	4,3 <sub>+4</sub>	9,8 <sub>-3</sub>	2,0 <sub>+1</sub>	1,0 <sub>-2</sub>	4,9 <sub>+0</sub>	9,3 <sub>-2</sub>	1,4 <sub>+2</sub>
				854	50	9,3 <sub>-3</sub>	4,3 <sub>+4</sub>	-4,2 <sub>-2</sub>	5,6 <sub>+1</sub>	-3,4 <sub>-2</sub>	8,4 <sub>+0</sub>	-3,3 <sub>-2</sub>	1,2 <sub>+2</sub>
				1196	70	1,3 <sub>-2</sub>	4,3 <sub>+4</sub>	3,1 <sub>-2</sub>	4,3 <sub>+1</sub>	-1,8 <sub>-3</sub>	1,0 <sub>+1</sub>	2,5 <sub>-2</sub>	6,2 <sub>+1</sub>
				1538	90	2,1 <sub>-2</sub>	4,3 <sub>+4</sub>	5,4 <sub>-2</sub>	4,5 <sub>+1</sub>	6,4 <sub>-2</sub>	1,8 <sub>+1</sub>	-3,4 <sub>-2</sub>	6,4 <sub>+1</sub>
4	35	35	5	169	10	0	1,0 <sub>+2</sub>	1,1 <sub>-3</sub>	3,2 <sub>+0</sub>	2,3 <sub>-3</sub>	9,5 <sub>-1</sub>	1,3 <sub>-2</sub>	2,5 <sub>+2</sub>
				507	30	3,2 <sub>-3</sub>	4,3 <sub>+4</sub>	3,6 <sub>-2</sub>	3,6 <sub>+1</sub>	3,7 <sub>-2</sub>	6,0 <sub>+0</sub>	1,7 <sub>-1</sub>	1,3 <sub>+2</sub>
				845	50	8,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-3,4 <sub>-2</sub>	3,1 <sub>+1</sub>	-4,0 <sub>-2</sub>	9,1 <sub>+0</sub>	4,3 <sub>-3</sub>	7,0 <sub>+1</sub>
				1183	70	1,2 <sub>-2</sub>	4,3 <sub>+4</sub>	6,2 <sub>-4</sub>	3,6 <sub>+1</sub>	6,2 <sub>-4</sub>	2,4 <sub>+1</sub>	2,4 <sub>-2</sub>	6,1 <sub>+1</sub>
				1521	90	2,7 <sub>-2</sub>	4,3 <sub>+4</sub>	7,6 <sub>-2</sub>	3,0 <sub>+1</sub>	-7,6 <sub>-2</sub>	1,8 <sub>+1</sub>	9,0 <sub>-2</sub>	5,6 <sub>+1</sub>
5	35	35	5	174	10	0	1,2 <sub>+2</sub>	3,4 <sub>-3</sub>	4,5 <sub>+0</sub>	2,3 <sub>-3</sub>	7,2 <sub>-1</sub>	1,4 <sub>-2</sub>	1,3 <sub>+2</sub>
				524	30	2,7 <sub>-3</sub>	4,3 <sub>+4</sub>	1,7 <sub>-2</sub>	2,2 <sub>+1</sub>	5,9 <sub>-3</sub>	3,2 <sub>+0</sub>	5,8 <sub>-2</sub>	2,3 <sub>+2</sub>
				874	50	7,2 <sub>-3</sub>	4,3 <sub>+4</sub>	-1,6 <sub>-3</sub>	2,0 <sub>+1</sub>	-4,2 <sub>-3</sub>	1,5 <sub>+1</sub>	1,9 <sub>-2</sub>	6,4 <sub>+1</sub>
				1224	70	1,2 <sub>-2</sub>	4,3 <sub>+4</sub>	-1,9 <sub>-3</sub>	2,8 <sub>+1</sub>	-1,9 <sub>-3</sub>	1,4 <sub>+1</sub>	2,3 <sub>-2</sub>	5,6 <sub>+1</sub>
				1574	90	3,4 <sub>-2</sub>	4,3 <sub>+4</sub>	-6,8 <sub>-2</sub>	2,1 <sub>+1</sub>	-6,8 <sub>-2</sub>	1,3 <sub>+1</sub>	-6,8 <sub>-2</sub>	5,9 <sub>+1</sub>

Таблица 4 (окончание)

Пример						Gurobi		VND <sub>1</sub>		VND <sub>2</sub>		VND' <sub>2</sub>	
№	n	m	r	V	%	гар	Время	гар	Время	гар	Время	гар	Время
6	20	20	5	108	10	0	9,6 <sub>+0</sub>	7,2 <sub>-3</sub>	1,7 <sub>-1</sub>	1,2 <sub>-3</sub>	5,5 <sub>-2</sub>	5,0 <sub>-2</sub>	7,6 <sub>+0</sub>
				324	30	0	1,2 <sub>+2</sub>	2,0 <sub>-2</sub>	1,4 <sub>+0</sub>	3,0 <sub>-2</sub>	2,7 <sub>-1</sub>	4,1 <sub>-2</sub>	9,4 <sub>+0</sub>
				541	50	0	1,1 <sub>+2</sub>	3,3 <sub>-2</sub>	7,2 <sub>-1</sub>	1,6 <sub>-2</sub>	5,8 <sub>-1</sub>	3,0 <sub>-2</sub>	5,1 <sub>+0</sub>
				757	70	0	1,2 <sub>+2</sub>	3,9 <sub>-3</sub>	1,9 <sub>+0</sub>	4,1 <sub>-3</sub>	1,2 <sub>+0</sub>	5,5 <sub>-2</sub>	3,4 <sub>+0</sub>
				973	90	0	1,4 <sub>+2</sub>	1,6 <sub>-2</sub>	2,6 <sub>+0</sub>	1,2 <sub>-2</sub>	7,0 <sub>-1</sub>	1,5 <sub>-3</sub>	4,6 <sub>+0</sub>
7	25	25	5	151	10	0	1,5 <sub>+2</sub>	7,6 <sub>-3</sub>	1,5 <sub>+0</sub>	3,9 <sub>-3</sub>	2,5 <sub>-1</sub>	7,0 <sub>-2</sub>	3,1 <sub>+1</sub>
				454	30	0	4,3 <sub>+3</sub>	5,5 <sub>-3</sub>	6,9 <sub>+0</sub>	1,1 <sub>-2</sub>	1,2 <sub>+0</sub>	1,1 <sub>-2</sub>	2,6 <sub>+1</sub>
				757	50	0	4,8 <sub>+3</sub>	1,5 <sub>-2</sub>	8,1 <sub>+0</sub>	5,6 <sub>-3</sub>	2,2 <sub>+0</sub>	5,6 <sub>-3</sub>	1,6 <sub>+1</sub>
				1060	70	0	4,5 <sub>+3</sub>	2,2 <sub>-2</sub>	1,1 <sub>+1</sub>	2,2 <sub>-2</sub>	3,6 <sub>+0</sub>	2,2 <sub>-2</sub>	2,7 <sub>+1</sub>
				1363	90	0	1,7 <sub>+3</sub>	3,4 <sub>-2</sub>	1,1 <sub>+1</sub>	2,9 <sub>-1</sub>	2,4 <sub>+0</sub>	2,3 <sub>-2</sub>	1,4 <sub>+1</sub>
8	20	20	5	126	10	0	1,3 <sub>+1</sub>	2,3 <sub>-3</sub>	9,5 <sub>-2</sub>	2,3 <sub>-3</sub>	1,2 <sub>-1</sub>	1,8 <sub>-2</sub>	1,2 <sub>+1</sub>
				378	30	0	1,4 <sub>+2</sub>	0	8,6 <sub>-1</sub>	2,2 <sub>-2</sub>	3,1 <sub>-1</sub>	3,7 <sub>-2</sub>	4,5 <sub>+0</sub>
				631	50	0	1,8 <sub>+2</sub>	1,2 <sub>-2</sub>	1,5 <sub>+0</sub>	1,1 <sub>-2</sub>	5,7 <sub>-1</sub>	1,1 <sub>-2</sub>	4,5 <sub>+0</sub>
				884	70	0	1,4 <sub>+2</sub>	2,8 <sub>-2</sub>	2,9 <sub>+0</sub>	1,8 <sub>-2</sub>	6,4 <sub>-1</sub>	1,6 <sub>-2</sub>	7,2 <sub>+0</sub>
				1136	90	0	1,9 <sub>+2</sub>	6,4 <sub>-2</sub>	4,6 <sub>+0</sub>	3,2 <sub>-2</sub>	7,7 <sub>-1</sub>	3,2 <sub>-2</sub>	4,8 <sub>+0</sub>
9	15	15	5	98	10	0	2,0 <sub>+1</sub>	7,4 <sub>-4</sub>	7,3 <sub>-2</sub>	5,8 <sub>-3</sub>	4,3 <sub>-2</sub>	1,4 <sub>-2</sub>	2,7 <sub>+0</sub>
				294	30	0	3,6 <sub>+1</sub>	1,4 <sub>-3</sub>	2,2 <sub>-1</sub>	2,8 <sub>-9</sub>	2,4 <sub>-1</sub>	2,8 <sub>-9</sub>	1,4 <sub>+0</sub>
				491	50	0	3,1 <sub>+1</sub>	2,0 <sub>-3</sub>	2,9 <sub>-1</sub>	2,4 <sub>-9</sub>	2,9 <sub>-1</sub>	2,4 <sub>-2</sub>	8,0 <sub>-1</sub>
				688	70	0	4,1 <sub>+1</sub>	3,3 <sub>-3</sub>	2,6 <sub>-1</sub>	8,8 <sub>-9</sub>	3,0 <sub>-1</sub>	8,8 <sub>-9</sub>	1,2 <sub>+0</sub>
				884	90	0	4,1 <sub>+1</sub>	5,5 <sub>-9</sub>	8,7 <sub>-1</sub>	5,5 <sub>-9</sub>	2,9 <sub>-1</sub>	5,5 <sub>-9</sub>	1,4 <sub>+0</sub>
10	15	15	10	87	10	0	8,8 <sub>+0</sub>	7,0 <sub>-2</sub>	3,2 <sub>-1</sub>	8,1 <sub>-2</sub>	1,3 <sub>-1</sub>	1,2 <sub>-1</sub>	1,2 <sub>+1</sub>
				263	30	0	7,7 <sub>+1</sub>	6,7 <sub>-2</sub>	1,2 <sub>+0</sub>	1,0 <sub>-2</sub>	5,0 <sub>-1</sub>	2,3 <sub>-2</sub>	7,1 <sub>+0</sub>
				438	50	0	5,8 <sub>+1</sub>	6,0 <sub>-2</sub>	8,8 <sub>-1</sub>	4,9 <sub>-2</sub>	6,7 <sub>-1</sub>	3,8 <sub>-2</sub>	1,2 <sub>+0</sub>
				613	70	0	3,8 <sub>+1</sub>	3,7 <sub>-2</sub>	5,9 <sub>-1</sub>	1,5 <sub>-2</sub>	2,9 <sub>-1</sub>	1,8 <sub>-2</sub>	7,8 <sub>-1</sub>
				789	90	0	6,6 <sub>+1</sub>	2,2 <sub>-2</sub>	1,1 <sub>+0</sub>	4,8 <sub>-9</sub>	3,2 <sub>-1</sub>	4,8 <sub>-9</sub>	3,1 <sub>+0</sub>

Из рис. 2 видно, что наименьший гар, отмеченный по вертикали, имеет реализация алгоритма с параметрами  $k = 2$ ,  $\text{flip} = 2$  при запуске 10 раз. Алгоритм с параметрами  $k = 2$ ,  $\text{flip} = 3$  и запуском 10 раз справился в среднем хуже: так происходит из-за того, что стартовое решение выбирается случайным образом. Эксперименты показывают, что для идеи мультистарта наилучшее число запусков алгоритма равно 10.

### Заключение

Исследование пороговой устойчивости двух- и трёхуровневых задач размещения и ценообразования начато в работах [1, 2]. На их основе рассмотрена пороговая устойчивость двухуровневых задач размещения и ценообразования с медианным типом размещения предприятий [3, 4].

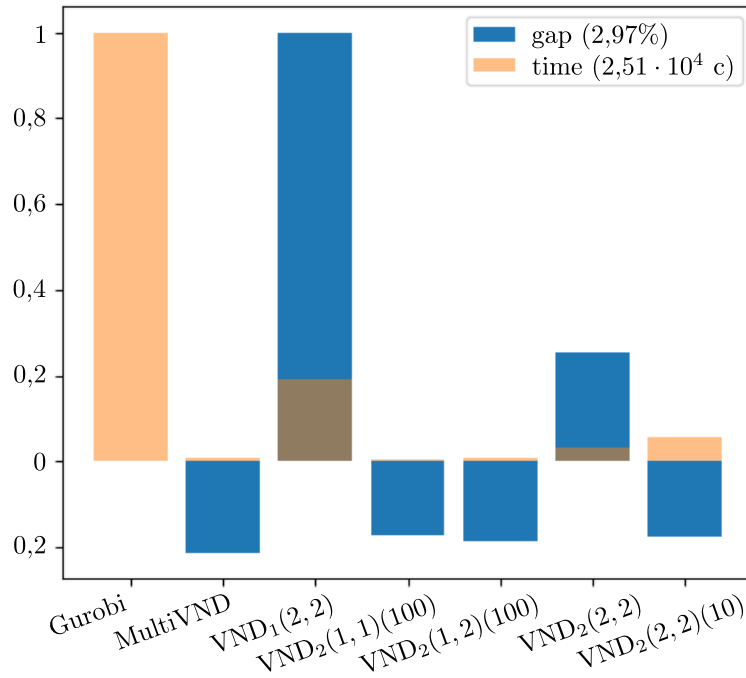


Рис. 1. Средние значения gap и времени по всем примерам

В указанных работах развит оригинальный подход к разработке алгоритмов решения соответствующих задач, основанный на использовании методов решения исходной задачи и её подзадач. Численный эксперимент показал весьма высокую эффективность разработанных алгоритмов как по качеству приближённых решений, так и по трудоёмкости алгоритмов [3, 4]. Однако базовые задачи, для которых исследована пороговая устойчивость, объединяет общее свойство — при фиксированном размещении предприятий соответствующие задачи ценообразования полиномиально разрешимы. Стало быть, для объективной оценки подхода, развитого в указанных работах, желательно исследовать пороговую устойчивость, например, задачи размещения и фабричного ценообразования, поскольку при фиксированном размещении предприятий задача фабричного ценообразования NP-трудна в сильном смысле.

В настоящей работе показано, что развитый в [3, 4] подход к разработке эффективных приближённых алгоритмов для определения пороговой устойчивости двухуровневой задачи размещения производства и ценообразования оказывается продуктивным и в случае базовой задачи с фабричным ценообразованием. Для решения задачи пороговой устойчивости предлагается алгоритм, основанный на спуске с чередующимися окрестностями (VND).

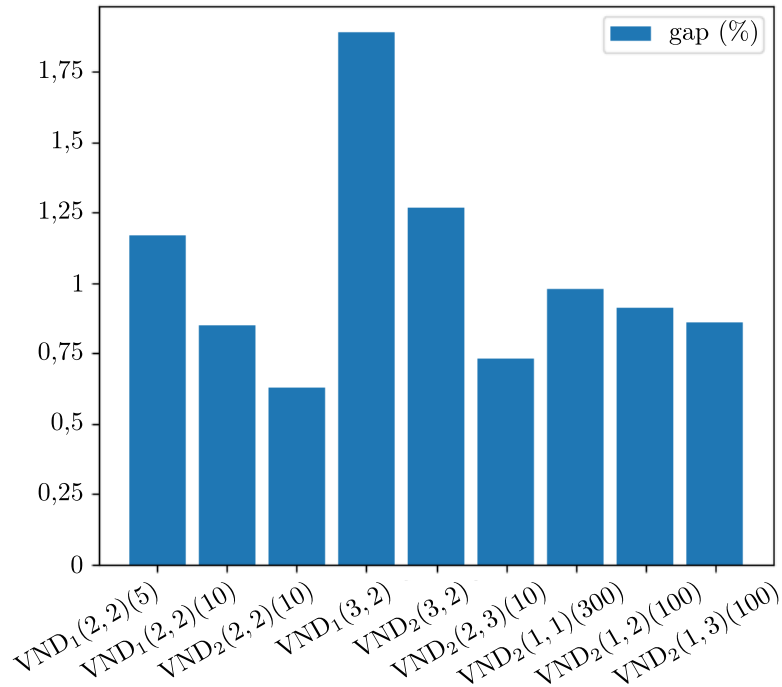


Рис. 2. Среднее значение gap для примеров с известным оптимумом

Численное исследование алгоритма проводится на известных примерах и случайно сгенерированных данных. Эксперименты показывают, что итеративное вычитание радиуса пороговой устойчивости из бюджетов потребителей значительно снижает время работы алгоритма. На примерах с найденным оптимумом алгоритм ошибается в среднем на 0,63% относительно оптимального значения целевой функции. На всех примерах алгоритм находит решение со значением целевой функции в среднем на 2,97% лучше в сравнении с решением Gurobi.

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

Вторая гипотеза вытекает из следствия 1, анализа доказательств теорем 1, 2 и представленных алгоритмов решения. Точный детерминированный полиномиальный алгоритм с оракулом из класса NP, о котором говорится в следствии 1, легко получить, взяв в качестве оракула стандартную задачу распознавания. В силу этого высокая эффективность разработанных алгоритмов связана, возможно, с тем, что использованный в работе подход позволяет хорошо аппроксимировать оракул, доставляющий информацию для детерминированного полиномиального

точного алгоритма из класса  $\Delta_2^P$ . С учётом того, что оракул представляет собой NP-полную задачу, в алгоритме используется итеративная процедура поиска  $(\rho, y, p)$  в обход фазы недетерминированного угадывания этих величин, которые и образуют сертификат.

### Финансирование работы

Исследование выполнено при финансовой поддержке Российского научного фонда (проект № 23–21–00424). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

### Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

### Литература

1. **Panin A. A., Plyasunov A. V.** Stability analysis for pricing // Mathematical optimization theory and operations research. Rev. Sel. Pap. 19th Int. Conf. MOTOR 2020 (Novosibirsk, Russia, July 6–10, 2020). Cham: Springer, 2020. P. 57–69. (Commun. Comput. Inf. Sci.; V. 1275).
2. **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, No. 6. P. 1295–1315. DOI: 10.1007/s11590-022-01924-3.
3. **Vodyan M. E., Panin A. A., Plyasunov A. V.** Metaheuristics for finding the stability radius in the bilevel facility location and uniform pricing problem // Proc. 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.
4. **Водян М. Е., Панин А. А., Плясунов А. В.** Исследование пороговой устойчивости двухуровневой задачи размещения производства и дискриминационного ценообразования // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 3. С. 79–104.
5. **Dyer M., Stougie L.** Computational complexity of stochastic programming problems // Math. Program. Ser. A. 2006. V. 106, No. 3. P. 423–432. DOI: 10.1007/s10107-005-0597-0.
6. **Кибзун А. И., Кан Ю. С.** Задачи стохастического программирования с вероятностными критериями. М.: Физматлит, 2009. 372 с.
7. **Ben-Tal A., Nemirovski A.** Robust optimization — Methodology and applications // Math. Program. Ser. B. 2002. V. 92, No. 3. P. 453–480. DOI: 10.1007/s101070100286.
8. **Snyder L. V.** Facility location under uncertainty: A review // IIE Trans. 2006. V. 38. P. 537–554. DOI: 10.1080/07408170500216480.
9. **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.

10. **Correia I., da Gama F. S.** Facility location under uncertainty // Location science. Cham: Springer, 2015. P. 177–203.
11. **Ben-Tal A., Nemirovski A.** Robust solutions of uncertain linear programs // Oper. Res. Lett. 1999. V. 25, No. 1. P. 1–13.
12. **Yu G., Yang J.** On the shortest path problem // Comput. Oper. Res. 1988. V. 25, No. 6. P. 457–68.
13. **Kouvelis P., Yu G.** Robust discrete optimization and its applications. Dordrecht: Kluwer Acad. Publ., 1997. 358 p.
14. **Averbakh I., Lebedev V.** Interval data minmax regret network optimization problems // Discrete Appl. Math. 2004. V. 138. P. 289–301.
15. **Aissi H., Bazgan C., Vanderpooten D.** Complexity of the min-max and min-max regret assignment problems // Oper. Res. Lett. 2005. V. 33, No. 6. P. 634–640.
16. **Carrizosa E., Nickel S.** Robust facility location // Math. Methods Oper. Res. 2003. V. 58, No. 2. P. 331–349.
17. **Carrizosa E., Ushakov A., Vasilyev I.** Threshold robustness in discrete facility location problems: A bi-objective approach // Optim. Lett. 2015. V. 9, No. 7. P. 1297–1314.
18. **Леонтьев В. К.** Устойчивость задачи коммивояжера // Журн. вычисл. математики и мат. физики. 1975. Т. 15, № 5. С. 1298–1309.
19. **Rossi A., Gurevsky E., Battaia 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.
20. **Gurevsky E., Rasamimanana A., Pirogov A., Dolgui A., Rossi A.** Stability factor for robust balancing of simple assembly lines under uncertainty // Discrete Appl. Math. 2022. V. 318. P. 113–132.
21. **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.
22. **Sotskov Yu. N.** Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives // Algorithms. 2023. V. 16, No. 2. Article ID 100. 43 p.
23. **Леонтьев В. К., Гордеев Э. Н.** Качественное исследование траекторных задач // Кибернетика. 1986. № 5. С. 82–89.
24. **Sotskov Yu. N., Leontiev V. K., Gordeev E. N.** Some concepts of stability analysis in combinatorial optimization // Discrete Appl. Math. 1995. V. 58, No. 2. P. 169–190.
25. **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.
26. **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.
27. **Talbi E.-G.** Metaheuristics: From design to implementation. Berlin: Wiley, 2009. 624 p.



28. Mladenovic N., Hansen P. Variable neighbourhood search // Comput. Oper. Res. 1997. V. 24. P. 1097–1100.
29. Кочетов Ю. А., Младенович Н., Хансен П. Локальный поиск с чередующимися окрестностями // Дискрет. анализ и исслед. операций. 2003. Т. 10, № 1. С. 11–43.
30. Diakova Z. S., Kochetov Yu. A. A double VNS heuristic for the facility location and pricing problem // Electron. Notes Discrete Math. 2012. V. 39. P. 29–34. DOI: 10.1016/j.endm.2012.10.005.
31. 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.
32. Hanjoul P., Hansen P., Peeters D., Thisse J.-F. Uncapacitated plant location under alternative spatial price policies // Manage. Sci. 1990. V. 36, No. 1. P. 41–57. DOI: 10.1287/mnsc.36.1.41.
33. Панин А. А., Пащенко М. Г., Плясунов А. В. Двухуровневые модели конкурентного размещения производства и ценообразования // Автоматика и телемеханика. 2014. № 4. С. 153–169.

Водян Максим Евгеньевич

Панин Артём Александрович

Плясунов Александр Владимирович

Статья поступила

29 ноября 2024 г.

После доработки —

2 февраля 2025 г.

Принята к публикации

22 июня 2025 г.

MAXIMIZING THE THRESHOLD STABILITY IN THE MODEL  
OF FACILITY LOCATION AND THE MILL PRICING

*M. E. Vodyan<sup>a</sup>, A. A. Panin<sup>b</sup>, and A. V. Plyasunov<sup>c</sup>*

Sobolev Institute of Mathematics,  
4 Acad. Koptuyug Avenue, 630090 Novosibirsk, Russia

E-mail: <sup>a</sup>m.vodyan@g.nsu.ru,  
<sup>b</sup>aapanin1988@gmail.com, <sup>c</sup>apljas@math.nsc.ru

**Abstract.** We study the threshold stability of the problem with the median location of facilities and mill pricing. The problem of threshold stability has the following differences from the original two-level formulation: in the top-level problem, the deviation of consumer budgets from expected values is maximized provided that the producer's income is not less than a given threshold. The problem statement considered in this paper differs from those previously studied in that the pricing problem is NP-hard in the strong sense when the location of facilities is fixed.

A variable neighborhoods descent based algorithm (VND) to solve the threshold stability problem is proposed. Numerical investigation of the algorithm is carried out on known examples and randomly generated data. The experiment shows that iteratively subtracting the threshold stability radius from the consumer budgets, which is first implemented in this paper, strongly reduces the running time of the algorithm. On the examples with the optimum known, the algorithm was wrong on average by 0.63%. In all the examples, the algorithm finds a solution on average 2.97% better than the Gurobi solver. Tab. 4, illustr. 2, bibliogr. 33.

**Keywords:** bilevel problem, threshold stability, radius of threshold stability, facility location, mill pricing, variable neighborhood descent.

References

1. **A. A. Panin** and **A. V. Plyasunov**, Stability analysis for pricing, in *Mathematical Optimization Theory and Operations Research*, Rev. Sel. Pap. 19th Int. Conf. MOTOR 2020 (Novosibirsk, Russia, July 6–10, 2020) (Springer, Cham, 2020), pp. 57–69 (Commun. Comput. Inf. Sci., Vol. 1275).

2. **A. A. Panin** and **A. V. Plyasunov**, The multilevel facility location and pricing problems: The computational complexity and the stability analysis, *Optim. Lett.* **17** (6), 1295–1315 (2023), DOI: 10.1007/s11590-022-01924-3.
3. **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 *Proc. 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.
4. **M. E. Vodyan**, **A. A. Panin**, and **A. V. Plyasunov**, A study of the threshold stability of the bilevel problem of facility location and discriminatory pricing, *Diskretn. Anal. Issled. Oper.* **31** (3), 79–104 (2024) [Russian] [*J. Appl. Ind. Math.* **18** (3) 558–574 (2024)], DOI: 10.1134/S1990478924030165].
5. **M. Dyer** and **L. Stougie**, Computational complexity of stochastic programming problems, *Math. Program, Ser. A*, **106** (3), 423–432 (2006), DOI: 10.1007/s10107-005-0597-0.
6. **A. I. Kibzun** and **Yu. S. Kan**, *Stochastic Programming Problems with Probability Criteria* (Fizmatlit, Moscow, 2009) [Russian].
7. **A. Ben-Tal** and **A. Nemirovski**, Robust optimization — Methodology and applications, *Math. Program, Ser. B*, **92** (3), 453–480 (2002), DOI: 10.1007/s101070100286.
8. **L. V. Snyder**, Facility location under uncertainty: A review, *IIE Trans.* **38**, 537–554 (2006), DOI: 10.1080/07408170500216480.
9. **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.
10. **I. Correia**, **F. S. da Gama**, Facility location under uncertainty, in *Location Science* (Springer, Cham, 2015), pp. 177–203.
11. **A. Ben-Tal** and **A. Nemirovski**, Robust solutions of uncertain linear programs, *Oper. Res. Lett.* **25** (1), 1–13 (1999).
12. **G. Yu** and **J. Yang**, On the shortest path problem, *Comput. Oper. Res.* **25** (6), 457–68 (1988).
13. **P. Kouvelis** and **G. Yu**, *Robust Discrete Optimization and Its Applications* (Kluwer Acad. Publ., Dordrecht, 1997).
14. **I. Averbakh** and **V. Lebedev**, Interval data minmax regret network optimization problems, *Discrete Appl. Math.* **138**, 289–301 (2004).
15. **H. Aissi**, **C. Bazgan**, and **D. Vanderpooten**, Complexity of the min-max and min-max regret assignment problems, *Oper. Res. Lett.* **33** (6), 634–640 (2005).
16. **E. Carrizosa** and **S. Nickel**, Robust facility location, *Math. Methods Oper. Res.* **58** (2), 331–349 (2003).
17. **E. Carrizosa**, **A. Ushakov**, and **I. Vasilyev**, Threshold robustness in discrete facility location problems: A bi-objective approach, *Optim. Lett.* **9** (7), 1297–1314 (2015).

18. **V. K. Leontiev**, Stability of the travelling salesman problem, *Zh. Vychisl. Mat. Mat. Fiz.* **15** (5), 1298–1309 (1975) [Russian] [*USSR Comput. Math. Math. Phys.* **15** (5), 199–213 (1975), 10.1016/0041-5553(75)90116-0].
19. **A. Rossi, E. Gurevsky, O. Battaia, 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).
20. **E. Gurevsky, A. Rasamimanana, A. Pirogov, A. Dolgui, and A. Rossi**, Stability factor for robust balancing of simple assembly lines under uncertainty, *Discrete Appl. Math.* **318**, 113–132 (2022).
21. **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).
22. **Yu. N. Sotskov**, Assembly and production line designing, balancing and scheduling with inaccurate data: A survey and perspectives, *Algorithms* **16** (2), ID 100 (2023).
23. **V. K. Leontiev and È. N. Gordeev**, Qualitative investigation of path problems, *Kibernetika*, No. 5, 82–89 (1986) [Russian] [*Cybernetics* **22** (5) 636–646 (1986), DOI: 10.1007/BF01068361].
24. **Yu. N. Sotskov, V. K. Leontiev, and È. N. Gordeev**, Some concepts of stability analysis in combinatorial optimization, *Discrete Appl. Math.* **58** (2), 169–190 (1995).
25. **K. G. Kuzmin**, 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), DOI: 10.1134/S1990478915040092].
26. **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.
27. **E.-G. Talbi**, *Metaheuristics: From Design to Implementation* (Wiley, Berlin, 2009).
28. **N. Mladenović and P. Hansen**, Variable neighbourhood search, *Comput. Oper. Res.* **24**, 1097–1100 (1997).
29. **Yu. A. Kochetov, N. Mladenović, and P. Hansen**, Local variable neighbourhood search, *Diskretn. Anal. Issled. Oper., Ser. 2*, **10** (1), 11–43 (2003) [Russian].
30. **Z. S. Diakova and Yu. A. Kochetov**, A double VNS heuristic for the facility location and pricing problem, *Electron. Notes Discrete Math.* **39**, 29–34 (2012), DOI: 10.1016/j.endm.2012.10.005.
31. **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), DOI: 10.1134/S1990478915030102].
32. **P. Hanjoul, P. Hansen, D. Peeters, and J.-F. Thisse**, Uncapacitated plant location under alternative spatial price policies, *Manag. Sci.* **36** (1), 41–57 (1990), DOI: 10.1287/mnsc.36.1.41.

- 33. A. A. Panin, M. G. Pashchenko, and A. V. Plyasunov**, Bilevel competitive facility location and pricing problems, *Avtom. Telemekh.*, No. 4, 153–169 (2014) [Russian] [*Autom. Remote Control* **75** (4), 715–727 (2014), DOI: 10.1134/S0005117914040110].

*Maksim E. Vodyan*

*Artyom A. Panin*

*Aleksandr V. Plyasunov*

Received November 29, 2024

Revised February 2, 2025

Accepted June 22, 2025