

АЛГОРИТМЫ С ЧЕРЕДУЮЩИМИСЯ ОКРЕСТНОСТЯМИ
ДЛЯ КОНКУРЕНТНОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ
ПРЕДПРИЯТИЙ С ГИБКИМ СПРОСОМ

Т. В. Леванова^{1,2, a}, А. Ю. Гнусарев^{1, b}

¹ Омский филиал Института математики им. С. Л. Соболева,
ул. Певцова, 13, 644099 Омск, Россия

² Омский гос. университет им. Ф. М. Достоевского,
пр. Мира, 55а, 644077 Омск, Россия

E-mail: ^alevanova@ofim.oscsbras.ru, ^balexander.gnussarev@gmail.com

Аннотация. Рассматривается ситуация на конкурентном рынке, когда новая Компания планирует получить прибыль от открытия своих предприятий, предлагающих товары или услуги. При этом ей необходимо учитывать, что имеется несколько проектов открытия для каждого предприятия, а на рынке уже размещены аналогичные предприятия Конкурента. Кроме того, клиенты сами выбирают места удовлетворения спроса в зависимости от того, где и какие предприятия размещены. Цель Компании — определить места и проекты открытия новых предприятий, чтобы привлечь наибольшую долю всего спроса клиентов. Особый характер спроса приводит к нелинейности целевой функции и дополнительным трудностям отыскания оптимального решения. В работе построены варианты алгоритмов поиска с чередующимися окрестностями, выполнен их экспериментальный анализ с использованием верхних оценок, получены апостериорные оценки точности и проведено обсуждение полученных результатов. Табл. 4, ил. 2, библиогр. 44.

Ключевые слова: задача размещения предприятий, конкуренция, гибкий спрос, эвристика, поиск с чередующимися окрестностями.

Разделы 1 и 2 выполнены Т. В. Левановой при поддержке Программы фундаментальных научных исследований государственных академий наук на 2013–2020 гг. № I.5 (проект № 0314–2019–0019). Раздел 3 выполнен А. Ю. Гнусаревым при поддержке Российского фонда фундаментальных научных исследований (проект № 18–07–00599).

© Т. В. Леванова, А. Ю. Гнусарев, 2020

Введение

Среди широкого круга прикладных проблем задачи размещения предприятий занимают отдельную активно развивающуюся область. Под предприятиями понимаются любые организации, устройства и т. п., удовлетворяющие потребности клиентов в каких-либо товарах и услугах. В этих задачах требуется расположить предприятия в некоторых пунктах и удовлетворить спрос клиентов с минимальными затратами или наибольшим доходом. Многие из подобных задач, такие как простейшая задача размещения, задача о p -медиане с ограничениями на мощности производства и др., уже стали классическими, они хорошо изучены и являются предметом значительного числа публикаций, в том числе [1–4]. В них спрос клиентов задан и остаётся неизменным, а решение о том, в каких пунктах он будет удовлетворяться, принимает централизованно одно лицо. Такие ситуации возможны, например, при размещении почтовых отделений, строительстве детских садов и т. п., когда жители закрепляются за территориальными участками. Но в реальных ситуациях чаще всего клиенты сами выбирают, где им удовлетворять потребности, а предприятия обслуживания борются между собой за внимание клиентов с целью увеличения своей прибыли. Подобные случаи описываются с помощью конкурентных моделей размещения. Идея рассмотрения конкурентного поведения принадлежит Хотеллингу [5], который исследовал оптимальное размещение двух конкурирующих предприятий на отрезке. Из дальнейшего наблюдения за реальным поведением покупателей стало очевидно, что существенную роль при выборе пунктов обслуживания (например, магазинов) играет расстояние. Полагают, что чем дальше расположен магазин, тем он менее притягателен для покупателя. Таким образом, выбор магазина происходит по некоторому вероятностному правилу, одно из них получило название правила гравитации (gravity rule). В гравитационной модели предполагается, что вероятность выбора объекта прямо пропорциональна его привлекательности и обратно пропорциональна неубывающей функции расстояния. Если рассматривать процесс размещения с точки зрения торговых компаний, то их доход будет зависеть от выбора клиентами пунктов размещения магазинов. Чем больше суммарная полезность всех открытых предприятий, тем больше спрос. В таком случае можно говорить о гибком спросе, имея в виду как меняющийся спрос каждого клиента, так и изменяющуюся структуру обслуживаемого спроса для компаний. Функция расстояния может быть различной. В исходной модели Рейли [6] принимается, что интерес к предприятию, находящемуся от клиента на расстоянии d , обратно пропорционален квадрату расстояния до него, т. е. $f(d) = \frac{1}{d^2}$.

Хафф [7, 8] предложил использовать $f(d) = \frac{1}{d^\lambda}$, где значение λ зависит от типа торгового предприятия. Уилсон [9] ввёл экспоненциальное правило $f(d) = \exp\{-\lambda d\}$. Дрезнер сравнил правила Хаффа и Уилсона и выявил преимущество экспоненциального подхода на реальных данных. В дальнейшем экспоненциальная функция для описания гибкого спроса применялась в ряде работ, например [10–13]. Берман и Красс изложили в [10] общую идею моделирования размещения предприятий компании в условиях конкуренции. В ней клиенты удовлетворяют свой спрос во всех открытых предприятиях, но в разных пропорциях. Возникающие при этом нелинейные модели осложняют поиск подходов к решению и требуют разработки специальных методов, поэтому число публикаций, посвящённых исследованию задач с гибким спросом (elastic demand), относительно невелико, среди них [14–25], а также обзорные работы [26–31].

Статья посвящена разработке методов приближённого решения конкурентной задачи размещения предприятий с гибким спросом, описанной в [12]. Предлагаются варианты алгоритма локального поиска с чередующимися окрестностями. Построена гибридная схема алгоритма с чередующимися окрестностями и имитации отжига. В ней реализуется возможность в процессе поиска с некоторой вероятностью осуществлять переход в решение, хуже текущего, что позволяет находить более качественные решения. Для улучшения работы, выявления сильных и слабых сторон новых алгоритмов создан набор серий тестовых примеров, обладающих различными характеристиками. На их основе выполнены экспериментальные исследования и проанализированы результаты. Обсуждена возможность использования известного программного обеспечения для нахождения рекордных решений. Тестируемые решатели *Baron* и *CoinBonmin* часто не находили даже допустимых решений за разумное время и требовали значительных вычислительных затрат. Решатель *LocalSolver* в целом справился со всеми тестовыми примерами, но на одной из серий каждому предложенному авторами алгоритму при равном времени счёта удалось найти решения, лучшие по качеству, чем с помощью *LocalSolver*.

Работа организована следующим образом. В разд. 1 приводится содержательная постановка задачи и её математическая модель. В разд. 2 даётся описание оригинальных вариантов алгоритма локального поиска с чередующимися окрестностями для указанной задачи. В разд. 3 излагаются этапы вычислительного эксперимента и проводится анализ его результатов.

1. Задача размещения предприятий с гибким спросом

Авторы [12] продолжили исследования задач размещения с гибким спросом и рассмотрели ситуацию, в которой Компания планирует открыть новую сеть супермаркетов, отличающихся друг от друга размером, ассортиментом и пр. Такие различия называются проектами. Цель Компании — заинтересовать максимальное количество клиентов, т. е. обслужить наибольшую долю их спроса. Эта доля для Компании не фиксирована, она зависит от того, где и по какому проекту открываются новые предприятия и, как следствие, какие предприятия будут выбраны клиентами с учётом привлекательности и расстояния до них. В зарубежной литературе [12] она носит название конкурентной задачи размещения и проектирования предприятий (competitive facility location and design problem). Её можно отнести к классу многовариантных задач размещения [32], при решении которых необходимо не только определить, где и сколько объектов разместить, но и выбрать для них варианты функционирования.

Приведём математическую постановку задачи размещения с гибким спросом, используя обозначения из [12]. Предполагается, что спрос клиентов сконцентрирован в точках множества $N = \{1, 2, \dots, n\}$. Все клиенты имеют одинаковую потребность, поэтому каждый пункт может рассматриваться как отдельный клиент с весом спроса w_i . На N задано метрическое пространство. Расстояние d_{ij} между пунктами $i, j \in N$ может, например, измеряться в евклидовой метрике или быть равным кратчайшему расстоянию между двумя вершинами графа. Подмножество пунктов возможного размещения предприятий обозначено через $P \subseteq N$, его часть $C \subset P$ уже занята Конкурентом. Компания может открывать свои предприятия в $S = P \setminus C$, имея для каждого из них множество проектов R . Таким образом, необходимо найти такие пункты из S и проекты из R для открытия предприятий, при которых доля обслуживания Компании максимальна.

На выбор места размещения также влияют следующие характеристики: B — доступный бюджет; a_{jr} — привлекательность предприятия проекта $r \in R$, размещённого в пункте $j \in S$; β — параметр чувствительности к расстоянию: чем меньше β , тем меньше клиенты реагируют на расстояние до пунктов обслуживания; λ_i — гибкость спроса в пункте $i \in N$; c_{jr} — стоимость открытия предприятия при использовании проекта $r \in R$ в пункте $j \in S$.

Переменные задачи имеют следующий вид:

$$x_{jr} = \begin{cases} 1, & \text{если в пункте } j \text{ открыто предприятие} \\ & \text{Компании проекта } r, \\ 0 & \text{иначе.} \end{cases}$$

Чтобы определить полезность u_{ij} предприятия $j \in S$ для клиента $i \in N$, вводятся коэффициенты $k_{ijr} = a_{jr}(d_{ij} + 1)^{-\beta}$. Тогда u_{ij} вычисляется следующим образом: $u_{ij} = \sum_{r \in R} k_{ijr} x_{jr}$. Общая полезность $U_i(C)$ для пункта спроса $i \in N$ от предприятий, открытых Конкурентом, определяется с помощью формулы

$$U_i(C) = \sum_{j \in C} u_{ij}.$$

Функция спроса имеет экспоненциальный вид: $g(U_i) = 1 - \exp(-\lambda_i U_i)$, где $U_i = \sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr} + U_i(C)$ — общая полезность для клиента $i \in N$ от всех открытых предприятий Компании и Конкурента.

Доля MS_i новых предприятий в общем объеме обслуживания клиента $i \in N$ равна

$$MS_i = \frac{U_i(S)}{U_i(S) + U_i(C)} = \frac{\sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr}}{\sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr} + U_i(C)}.$$

В использованных обозначениях математическая модель имеет следующий вид [12]:

$$\sum_{i \in N} w_i \cdot g(U_i) \cdot MS_i \rightarrow \max, \quad (1)$$

$$\sum_{j \in S} \sum_{r \in R} c_{jr} x_{jr} \leq B, \quad (2)$$

$$\sum_{r \in R} x_{jr} \leq 1, \quad j \in S, \quad (3)$$

$$x_{jr} \in \{0, 1\}, \quad j \in S, \quad r \in R. \quad (4)$$

Функция (1) отражает цель Компании — захватить наибольшую долю всего спроса клиентов. Ограничение (2) говорит о том, что нельзя выходить за рамки установленного бюджета; условие (3) показывает, что предприятие может работать не более чем по одному проекту.

Выпишем целевую функцию более подробно:

$$\begin{aligned} \sum_{i \in N} w_i \left(1 - \exp \left(-\lambda_i \left(\sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr} + U_i(C) \right) \right) \right) \\ \times \frac{\sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr}}{\sum_{j \in S} \sum_{r \in R} k_{ijr} x_{jr} + U_i(C)} \rightarrow \max. \quad (5) \end{aligned}$$

Рассматриваемая задача относится к классу NP-трудных [33]. Нелинейность целевой функции (5) приводит к необходимости разработки оригинальных алгоритмов решения.

В работе [12] обсуждаются ситуации при различных значениях коэффициента гибкости λ . Отмечается, что чем меньше λ , тем более чувствительны клиенты к тому, какие проекты для предприятий выбраны, спрос считается гибким (elastic). Компания следит как за тем, где размещать свои предприятия, т. е. пытается расширить своё влияние на рынке, так и за тем, какие проекты для предприятий выбрать. При этом максимизируется не только доля обслуживаемого рынка, но и спрос клиентов. При больших λ спрос негибкий, суммарный спрос в каждом пункте очень близок к его максимальному значению, выбор проекта существенно не влияет на поведение клиентов, а стратегия Компании нацелена на увеличение её доли на рынке. Авторы в [12] на ряде примеров проанализировали, при каких λ выбор проекта перестаёт влиять на размер захваченной доли рынка, т. е. при каких значениях λ спрос перестаёт быть гибким. Из их исследований можно сделать вывод, что при $0 < \lambda \leq 1$ спрос гибкий. В данной статье в разд. 3 на основе данных из [12] строятся серии тестовых примеров.

2. Локальный поиск с чередующимися окрестностями

Впервые методы локального поиска для задач дискретной оптимизации стали применяться в конце 1950-х — начале 1960-х гг. и в большей степени были связаны с задачей коммивояжёра. Позднее эти подходы применялись для задач размещения, построения расписаний и др. [4, 34]. Вскоре выяснилось, что обычный локальный поиск часто получает решение существенно хуже глобального оптимума, и интерес к данному направлению упал. Однако в последние годы наблюдается возрождение данного подхода. Это связано, в первую очередь, с тем, что появились новые идеи построения таких алгоритмов.

Методы локального поиска получили развитие в метаэвристиках, одной из которых является поиск с чередующимися окрестностями [35, 36]. Конкурентоспособность данного подхода при решении NP-трудных задач объясняется относительной лёгкостью схем, возможностью адаптации к различным моделям и высокой эффективностью с точки зрения близости к глобальному оптимуму. Идея алгоритма применялась в решении задачи о p -медиане, задачи Вебера, задач кластеризации и размещения, коммивояжёра, построения расписаний, построения сетей, поиска деревьев Штейнера, максимальных разрезов, клик максимальной мощности и многих других [1, 37–40].

В данной работе для конкурентной задачи размещения с гибким спросом предложен вариант основного алгоритма, реализованный по базовой

схеме [35] (variable neighborhood search, VNS), а также его релаксированная версия (relaxed neighborhood search algorithm, RVNS) и гибридный вариант алгоритма VNS. Результаты частично докладывались на конференциях и опубликованы в материалах [17].

В начале работы алгоритма определяются множество используемых окрестностей и порядок их просмотра, выбирается допустимое исходное решение и вычисляется значение целевой функции, которое будет служить начальным рекордом. Алгоритм за один шаг специальным образом просматривает список окрестностей и находит улучшающее решение. Поиск заканчивается в случае, если список окрестностей просмотрен и не найдено решения лучше рекорда.

Для построения алгоритмов вместо булевой матрицы $X = (x_{jr})$ удобно использовать целочисленный вектор $z = (z_j)$, компонента j которого равна номеру проекта r предприятия в пункте j , т. е. $z_j = r$ тогда и только тогда, когда $x_{jr} = 1$, $j \in S$, $r \in R$. Будем называть рекордом лучшее найденное значение $F(z)$ целевой функции (5) и соответствующий ему вектор z . Опишем схему алгоритма для задачи на максимум.

ПРЕДВАРИТЕЛЬНЫЙ ШАГ. Выбираем окрестности N_k , $k = 1, \dots, K$, определяем начальное решение z и полагаем $\text{Rec} := F(z)$.

ШАГ 1. Полагаем $k := 1$.

ШАГ 2. Пока $k \neq K$ повторяем следующие шаги.

ШАГ 2.1 (SHAKING). Генерируем точку $z' \in N_k$.

ШАГ 2.2 (LOCAL SEARCH). Применяем какой-либо алгоритм локального поиска к z' ; найденный локальный оптимум обозначим через z'' .

ШАГ 2.3 (MOVE OR NOT). Если $F(z'') > F(z)$, то $\text{Rec} := F(z'')$, $z := z''$, $k := 1$, иначе $k := k + 1$. Идём на шаг 2.1.

В качестве критерия останова может использоваться, например, максимальное время счёта или заданное число итераций без смены лучшего найденного решения.

Идея алгоритма основана на том соображении, что локальный оптимум относительно одной окрестности может не быть таковым относительно другой. Смена окрестности позволяет продлить процесс поиска локального оптимума и улучшить текущее решение, поэтому выбор окрестности играет важную роль в локальном поиске.

Для описания окрестностей введём V — множество номеров открытых предприятий и Q — множество номеров закрытых предприятий. Определим вектор перспективности α размерности $|S|$ по следующим правилам: 1) полагаем $\alpha = \vec{0}$; 2) изменяем компоненты вектора $\alpha_v := \alpha_v - 1$,

$\alpha_q := \alpha_q + 1$, если при уменьшении номера проекта $z_v, v \in V$, и увеличении номера проекта $z_q, q \in Q$, вектора z получаем вектор z' такой, что $F(z') > F(z)$.

Для построения рассматриваемых алгоритмов были предложены новые виды окрестностей N_1 и N_2 , учитывающие специфику задачи [17].

Окрестностью N_1 вектора z называется множество, состоящее из векторов z' , полученных следующим образом: 1) закрываем открытое предприятие с номером v и проектом z_v ; 2) открываем с проектом z_v одно из закрытых предприятий с номером q , имеющее наибольшую перспективность $\alpha_q, z'_q = z_v$.

Окрестностью N_2 вектора z называется множество, состоящее из векторов z' , построенных по правилам: 1) выбираем номер предприятия v для закрытия; 2) выбираем номер предприятия q для открытия; 3) выбираем целое число $\delta \in [1, \min(z_v, R - z_q)]$; 4) изменяем номера проектов $z_v := z_v - \delta, z_q := z_q + \delta$.

В качестве окрестности N_3 использовалась известная окрестность Лина — Кернигана (см., например, [3]). Для её формирования необходимо задать число точек в этой окрестности k и окрестность N , по которой она будет строиться. Для рассматриваемой задачи окрестностью Лина — Кернигана вектора z называется множество векторов $N_{LK} = \{z^1, \dots, z^k\}$, полученное следующим образом. Полагаем $z^1 = z$, повторяем k раз такие шаги: 1) рассматриваем окрестность N_2 вектора z^k , исключаем из неё вектор z^k , среди оставшихся векторов указанной окрестности находим локальный максимум z' ; 2) полагаем $z^k = z'$.

Возможны также различные модификации основной схемы алгоритма. Например, в базовой схеме на шаге 2.3 осуществляется переход в следующую точку только тогда, когда происходит увеличение значения целевой функции. В данной работе предлагается заменить это условие по аналогии с алгоритмом имитации отжига (simulated annealing), в котором допускается произвольное ухудшение по целевой функции с некоторой вероятностью, зависящей от этого ухудшения. В нашей задаче вероятность перехода в решение с меньшим значением целевой функции обратно пропорциональна величине уменьшения [41] и зависит от значения на текущей итерации k параметра «температура» t_k . Данная версия алгоритма получила обозначение VNS_{SA}, в ней шаг 2.3 реализован следующим образом.

ШАГ 2.3 (MOVE OR NOT). Переход из текущего решения z в новое z'' осуществляется с вероятностью

$$P_{zz''} = \begin{cases} 1, & \text{если } F(z'') \geq F(z), \\ \exp \frac{F(z'') - F(z)}{t_k}, & \text{если } F(z'') < F(z). \end{cases}$$

Кроме того, в данной работе развивается сокращённая версия базовой схемы (RVNS [35]), в которой отсутствует шаг локального поиска 2.2.

3. Вычислительный эксперимент

Все алгоритмы реализованы на языке C++ и тестировались на компьютере Intel(R) Xeon(R) CPU X5675 @ 3,07 GHz с оперативной памятью 96,0 Гб. Численные эксперименты проводились с целью апробации и сравнительного анализа предложенных алгоритмов решения рассматриваемой конкурентной задачи размещения. Построено 5 серий примеров для изучения алгоритмов при различных условиях. Для выявления качества получаемых решений использовались верхние оценки целевой функции, построенные авторами в работах [17, 42], а также известное программное обеспечение: решатели Baron и CoinBonmin [43] и LocalSolver [44].

В данном разделе сначала будут описаны правила, по которым строились серии тестовых примеров, а затем представлены этапы экспериментальных исследований. Вычисления можно разделить на две части. В первой рассматривались задачи с коэффициентом гибкости $\lambda = 1$ (medium elasticity). Здесь с использованием серий 1 и 2 тестовых примеров выполнено сравнение алгоритмов и решателей CoinBonmin, LocalSolver, а также результатов работы предложенных алгоритмов и значений верхних оценок целевой функции из [17]. На серии 3 показано сравнение решателя LocalSolver и разработанных алгоритмов. Во второй части численных экспериментов на сериях 4 и 5 проведено сравнение алгоритмов с верхними оценками [42] при малых значениях коэффициента гибкости спроса $\lambda \ll 1$ (high elasticity).

Необходимо отметить, что в связи с наличием в алгоритмах локального поиска нескольких параметров возникает задача настройки алгоритма, т. е. выбора таких значений параметров, при которых достигалось бы достаточно хорошее поведение алгоритма на большинстве задач. В алгоритмах локального поиска с чередующимися окрестностями VNS и RVNS ими являются количество перезапусков без улучшения рекорда, а также порядок окрестностей. Для настройки параметров проведены специальные экспериментальные исследования, наименьшие средние отклонения от верхних оценок достигнуты при наборе окрестностей N_1, N_2, N_3 и 100 перезапусках; количество точек в окрестности Лина — Кернигана равно 7. В алгоритме VNS_{SA} помимо уже описанных настроек применялся следующий закон изменения параметра «температура»: $t_l = r \cdot t_{l-1}$, где r — коэффициент «остывания», l — итерация алгоритма. Начальными значениями параметров приняты $t_0 = 150$, $r = 0,99$, а минимальное значение параметра «температура» равно $t_{\min} = 5$ (так называемая

«температура остывания»). Если в процессе просмотра окрестностей значение параметра t_l оказывалось равным t_{\min} , то его величина поднималась до начального значения. В алгоритме имитации отжига такой приём называется «разогреванием». Пока идёт поиск решения, «разогревание» могло происходить несколько раз. Когда просмотрены все окрестности, алгоритм заканчивал свою работу.

Опишем серии тестовых задач, используемых в экспериментальных исследованиях. Для проведения вычислительных экспериментов созданы пять серий тестовых заданий. Во всех сериях

- в каждом пункте спроса может быть открыто чьё-либо предприятие, т. е. $P = N$;

- для каждого пункта возможного размещения с вероятностью $1/2$ определялось, чьё именно предприятие будет в нём размещаться — Конкурента или Компании;

- возможны три проекта работы предприятий ($R = 3$): основной с затратами на открытие $c_{j1} = 1$; средний с затратами на открытие $c_{j2} = 2$; улучшенный с затратами на открытие $c_{j3} = 3$ для всех $j \in S$.

Серии 1 и 2 созданы на основе реальных данных в соответствии с их описанием в [12] и отличаются способом задания расстояний между объектами. В серии 1 рассматривались задачи, в которых расстояния можно интерпретировать как длину кратчайшего пути между пунктами на некоторой сети ($d_{ij} \in [0, 30]$, $i \neq j$, $d_{ij} = d_{ji}$, $i \in N$, $j \in S$). В серии 2 содержатся примеры с расстояниями между пунктами, удовлетворяющими неравенству треугольника (координата $x \in [0, 100]$, координата $y \in [0, 150]$). Остальные параметры для них одинаковы, а именно

- бюджет изменяется от 3 до 9 с шагом в 2 единицы. В рамках доступного бюджета Компания имеет возможность рассматривать различные варианты развития предприятий. Например, располагая бюджетом в 9 единиц, можно открыть 5 предприятий: 2 основных, 2 средних и 1 улучшенное;

- количество пунктов размещения равно 60, 80, 100, 150, 200, 300;

- задачи имеют высокую чувствительность к расстоянию ($\beta = 2$);

- привлекательности предприятий a_{jr} выбираются следующим образом: $a_{j1} = 3$, $a_{j2} = a_{j1} + t_1$, $a_{j3} = a_{j2} + t_2$, $j \in S$, где $t_1 \in [1, 6]$, $t_2 \in [1, 9]$; веса спроса $w_i \in [1, 10]$, $i \in N$.

Серия 3 отличается от первых двух следующими значениями параметров:

- коэффициент чувствительности к расстоянию β равен 1, т. е. задачи имеют меньшую чувствительность по сравнению с примерами серий 1 и 2;

- количество пунктов размещения равно $N = 300$;

- бюджет равен 20;

— веса пунктов спроса для пунктов от 1 до 100 задаются с равномерным распределением в интервале $[1, 3]$, для пунктов от 101 до 200 — в интервале $[4, 6]$, для пунктов от 201 до 300 — в интервале $[7, 10]$;

— привлекательность предприятий генерируется с равномерным распределением следующим образом: для пунктов от 1 до 100 их $a_{j1} = 1$, $a_{j2} = a_{j1} + t_1$, t_1 в интервале $[1, 2]$, $a_{j3} = a_{j2} + t_2$, t_2 в интервале $[1, 3]$; для пунктов от 101 до 200 их $a_{j1} = 2$, $a_{j2} = a_{j1} + t_1$, t_1 в интервале $[2, 3]$, $a_{j3} = a_{j2} + t_2$, t_2 в интервале $[3, 5]$; для пунктов от 201 до 300 их $a_{j1} = 3$, $a_{j2} = a_{j1} + t_1$, t_1 в интервале $[4, 6]$, $a_{j3} = a_{j2} + t_2$, t_2 в $[5, 9]$, $j \in S$;

— пункты спроса размещаются на евклидовой плоскости, где точки определяются по следующему правилу: для первых 100 пунктов координаты x и y выбираются в интервале $[1, 1000]$, для пунктов с номерами от 101 до 200 — в интервале $[2000, 3000]$, для пунктов от 201 до 300 — в $[4000, 5000]$.

Серии 4 и 5 сформированы по тем же правилам, что и серии 1 и 2. Значение коэффициента гибкости λ мало и изменяется в серии 4 от 0,000606 до 0,002850, а в серии 5 — от 0,000052 до 0,000714.

Всего для серий 1 и 2 было сгенерировано по 96 тестовых примеров, для серии 3 создано 100 тестовых примеров, для серий 4 и 5 также было по 96 примеров. Исходя из их свойств, при необходимости будем использовать названия: серия 1 — UniformModerate (UM), серия 2 — EuclideanModerate (EM), серия 3 — LocalModerate (LM), серия 4 — UniformHigh (UH), серия 5 — EuclideanHigh (EH).

Таблица 1

Среднее время работы на серии 1, с

UM				
N	VNS	RVNS	VNS _{SA}	CoinBonmin
60	8,9	0,7	9,1	106,1
80	15,7	1,2	16,4	160,5
100	22,8	1,5	24,1	294,8
150	51,6	2,8	49,9	999,9
200	113,9	5,1	81,4	1325,2
300	168,8	10,0	209,7	4603,4

В первой части исследований с помощью решателей CoinBonmin [43] и LocalSolver (LS) [44] проведены численные эксперименты на сериях 1 и 2. Для серии 1 CoinBonmin нашёл допустимые решения для всех примеров. Лучшие из них по значению целевой функции будем называть рекордом. Среднее время работы пакета приведено в табл. 1. Время работы

алгоритмов на обеих сериях было одинаковым, поэтому для них приводятся данные только одной серии. Для примеров с неравенством треугольника были получены рекорды для всех задач размерности $n = 60$ со следующими минимальным (t_{\min}), средним (t_{avg}) и максимальным (t_{\max}) временами работы: $t_{\min} = 675$ с, $t_{\text{avg}} = 28864$ с, $t_{\max} = 189704$ с ($t_{\min} = 11$ мин, $t_{\text{avg}} = 8$ ч, $t_{\max} = 52$ ч 41 мин). Максимальное время, требуемое решателю для нахождения рекорда, составило более 52 часов, поэтому для остальных размерностей было принято решение ограничить его время счёта 1 часом. Из оставшихся 80 примеров решатель нашёл рекорды только для 18 из них, а именно для 10 задач размерности 80, по 4 — для размерностей 100 и 150, а для размерностей 200 и 300 не было найдено ни одного допустимого решения. Оказалось невозможным задействовать коммерческое программное обеспечение для анализа качества предложенных алгоритмов, поэтому применялась верхняя оценка целевой функции из работы [17]. Из-за большого времени счёта уже на малых размерностях на серии 3 использовался только LocalSolver, который хорошо зарекомендовал себя в предварительных экспериментах. Так, LocalSolver за 20 секунд показывал результаты, сравнимые с CoinBonmin при работе последнего более 52 часов.

Поскольку особый интерес представляют результаты, полученные на задачах большой размерности, опишем их более подробно. Для серии 1 при размерности 300 среднее отклонение от верхних оценок для алгоритмов составило: VNS — 0,327 %, RVNS — 3,604 %, VNS_{SA} — 0,43 %; для решателя CoinBonmin отклонение составило 2,017 %; для LocalSolver — 0,009 % (для остальных размерностей см. рис. 1). Важно отметить, что в 15 примерах найденное значение целевой функции отклоняется от верхней оценки менее чем на 0,001 % для алгоритма VNS (RVNS — 7 примеров, VNS_{SA} — 15 примеров, CoinBonmin — 11 примеров, LocalSolver — 16 примеров). Сравнение векторов Z , полученных при решении оценочной и исходной задач, показало, что для VNS они совпадают в 51 примере (RVNS — 40 примеров, VNS_{SA} — 55 примеров, CoinBonmin — 90 примеров, LocalSolver — 90 примеров). Время работы алгоритма VNS не превысило 327,5 с на размерности 300 (RVNS — 25,9 с, для VNS_{SA} — 297,7 с, CoinBonmin — 13719,3 с, время работы LocalSolver было равно 20 с). На данной серии тестовых примеров результаты разработанных алгоритмов и коммерческих решателей CoinBonmin и LocalSolver были сравнимы. Таким образом, на этой серии лучший из новых алгоритмов и коммерческий LocalSolver способны находить решения с отклонением менее 1 % при небольшом времени счёта.

Для тестовых примеров с неравенством треугольника при размерности 300 среднее отклонение от верхних оценок для алгоритмов составило: VNS — 14,730 %, RVNS — 14,992 %, VNS_{SA} — 14,749 %. Из рис. 2 также

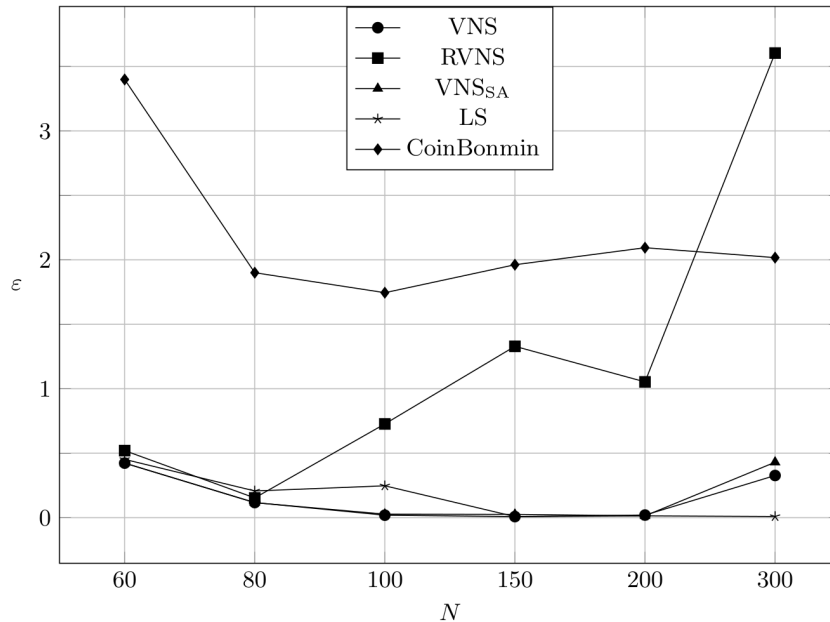


Рис. 1. Отклонения алгоритмов локального поиска на серии 1, ε %

можно заметить, что результаты алгоритмов не уступают коммерческому решателю LocalSolver в качестве получаемых решений (по значению целевой функции). Кроме того, было выполнено сравнение векторов Z , полученных при нахождении верхних оценок и решении задачи алгоритмами: для VNS они совпадают в 21 примере (RVNS — 21 пример, VNS_{SA} — 21 пример, CoinBonmin — 3 примера). Время работы алгоритма VNS не превысило 676,9 с на размерности 300 (RVNS — 35,7 с, VNS_{SA} — 746,5 с). На данной серии результаты работы алгоритмов сравнивались только с верхними оценками и решателем LocalSolver, так как CoinBonmin не находил даже допустимого решения за отведённое время в 1 час. Можно сделать вывод, что эта серия оказалась более сложной и для коммерческого решателя, и для алгоритмов локального поиска. По сравнению с первой серией отклонения новых алгоритмов от верхних оценок резко возросли и оказались примерно одинаковыми. Этот факт дал повод предположить, что для подобных задач оценка из [17] достаточно груба и не позволяет сделать объективных выводов о качестве решений. Поэтому в работе [42] была построена новая оценка с малыми значениями коэффициента гибкости λ , экспериментальные исследования на этих задачах описаны ниже.

Для анализа алгоритмов на серии 3 использовался только решатель LocalSolver, так как он оказался самым быстрым на предыдущих сериях.

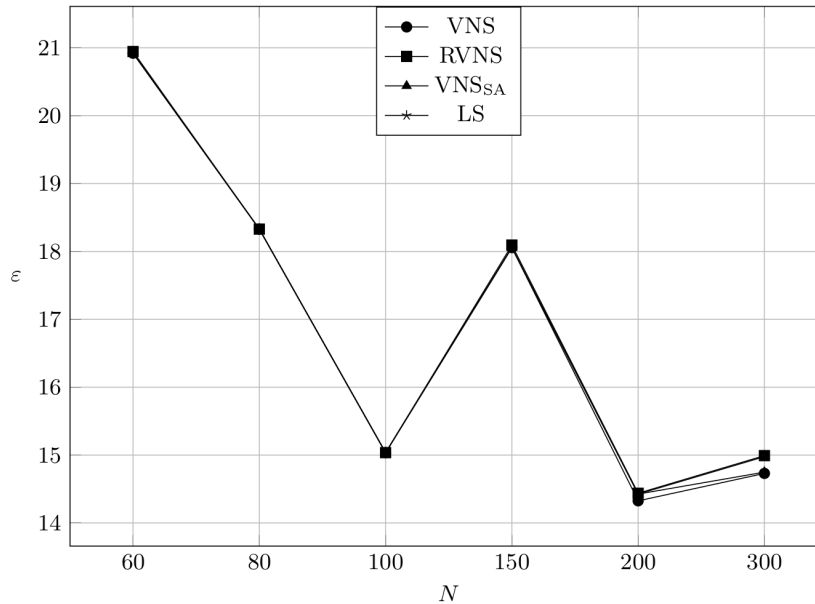


Рис. 2. Отклонения алгоритмов локального поиска на серии 2, ε %

Сравнение LocalSolver и предложенных алгоритмов проводились в одинаковых условиях по времени работы — на решение одного примера давалось 20 с. Результаты приведены в табл. 2. Отклонение вычислялось по следующей формуле: $\varepsilon = \frac{VS-LS}{LS} \cdot 100$ %, где LS и VS — лучшие значения целевой функции задачи, найденные решателем LocalSolver и каким-либо алгоритмом локального поиска с чередующимися окрестностями. Положительные отклонения означают, что предложенный алгоритм превзошёл рекорд LocalSolver. Среднее отклонение рекорда алгоритма (ε_{avg}) VNS_{SA} от LocalSolver составило 1,543 %. Видно, что по средним отклонениям алгоритм VNS получает результаты лучше решателя LocalSolver, в дополнение к этому, на одном из примеров ему удалось превзойти достижения решателя на 6,378 % (ε_{max} — максимальные отклонения). Алгоритм RVNS нашёл новые рекорды для 2 примеров, VNS — для 78 и VNS_{SA} — для 81 примера из 100. При этом частота получения нового рекорда алгоритмом VNS_{SA} составляет 0,81, для RVNS — 0,020, для VNS — 0,780. Также были построены доверительные интервалы для вероятности улучшения рекорда LocalSolver при уровне доверия 95 %: RVNS — [0,000; 0,047], VNS — [0,699; 0,861], VNS_{SA} — [0,733; 0,877]. Из данных результатов следует, что найдена серия заданий, для которой с достаточно большой надёжностью можно утверждать превосходство новых алгоритмов перед коммерческим продуктом LocalSolver по значению рекордного решения при малом времени счёта.

Таблица 2

Отклонения алгоритмов от LocalSolver на серии 3, ε %

LM			
ε	RVNS	VNS	VNS _{SA}
ε_{avg}	-5,713	1,316	1,543
ε_{min}	-13,049	-3,215	-3,040
ε_{max}	1,313	6,378	5,787

Во второй части исследований использованы верхние оценки целевой функции задачи CFLDP, предложенные в работе [42] при малых значениях гибкости спроса. При этих значениях клиенты имеют высокую чувствительность к виду открываемых предприятий. Для таких задач был проведён отдельный вычислительный эксперимент на сериях 4 и 5.

Таблица 3

Среднее отклонение алгоритмов от верхних оценок при малых значениях гибкости спроса, ε %

N	UH			EH		
	VNS	RVNS	VNS _{SA}	VNS	RVNS	VNS _{SA}
60	6,394	1,719	1,425	3,421	2,514	2,506
80	8,535	1,363	1,451	4,274	2,694	2,375
100	8,078	2,289	1,788	4,690	2,047	2,085
150	8,005	1,846	1,823	5,182	2,101	2,921
200	10,179	2,871	2,935	6,061	1,498	2,424
300	8,988	2,411	3,570	7,148	3,685	4,742

Для серии 4 при размерности 300 средние отклонения от верхних оценок составили: VNS — 2,411 %, RVNS — 8,988 %, VNS_{SA} — 3,573 %. Для остальных размерностей полученные результаты приведены в табл. 3. Доверительные интервалы для вероятности нахождения рекорда с отклонением от верхних оценок менее чем на 3 % при уровне доверия 95 % оказались равны для VNS — [0,829; 0,843], для RVNS — [0,318; 0,337], для VNS_{SA} — [0,787; 0,803].

Для серии 5 при размерности 300 средние отклонения от верхних оценок составили: VNS — 3,685 %, RVNS — 7,148 %, VNS_{SA} — 4,742 %. Для остальных размерностей результаты приведены в табл. 3. Доверительные интервалы для вероятности нахождения рекорда с отклонением от верхних оценок менее чем на 3 % при уровне доверия 95 % получились следующие: VNS — [0,847; 0,861], RVNS — [0,438; 0,458], VNS_{SA} — [0,768; 0,785].

Максимальное время работы алгоритмов для серий 4 и 5 до достижения критерия останова для VNS не превысили 327,5 с, для VNS_{SA} — 297,7 с, а RVNS уже за 25,9 с заканчивал поиск. Поэтому были проведены дополнительные исследования на указанных тестовых примерах при одинаковом времени счёта в 20 с. Если алгоритм завершал работу раньше установленного времени, то он запускался снова. Итоги этих вычислений для всех трёх алгоритмов, а также решателя LocalSolver, приведены в табл. 4. На малых размерностях, таких как 60, 80, 100, результаты всех алгоритмов оказались сравнимыми между собой: на серии 4 отклонения не превысили 1,569 %, а на серии 5 отклонения были меньше 3,210 %. На размерностях 150, 200, 300 отклонения алгоритма RVNS стали расти, в то время как для VNS, VNS_{SA}, LocalSolver они остались на прежнем невысоком уровне. К примеру, на размерности 300 средние отклонения от верхних оценок на серии 4 таковы: VNS — 3,327 %, RVNS — 10,032 %, VNS_{SA} — 0,850 %, LocalSolver — 0,187 %. На серии 5 получены следующие результаты: VNS — 4,703 %, RVNS — 13,318 %, VNS_{SA} — 0,620 %, LocalSolver — 0,556 %. Таким образом, алгоритму RVNS при больших размерностях для нахождения рекорда с отклонением менее 3,000 % потребуется значительно больше времени, чем алгоритмам VNS, VNS_{SA} и решателю LocalSolver. Алгоритм VNS_{SA} уже за 20 с показывает своё преимущество, особенно на тестовых примерах большой размерности.

Таблица 4

Среднее отклонение от верхних оценок при малых значениях гибкости спроса и фиксированном времени 20 с, ε %

<i>N</i>	UH				EH			
	VNS	RVNS	VNS _{SA}	LS	VNS	RVNS	VNS _{SA}	LS
60	1,082	1,100	1,332	1,082	1,974	1,991	3,210	1,974
80	0,834	1,375	1,034	0,831	1,610	1,954	2,894	1,603
100	0,707	1,569	1,378	0,654	1,433	2,210	2,098	1,339
150	0,908	3,705	0,648	0,391	2,029	5,190	1,776	1,099
200	1,586	6,027	0,905	0,291	2,298	7,012	1,190	0,695
300	3,327	10,032	0,850	0,187	4,703	13,318	0,620	0,556

В целом, вторая часть экспериментальных исследований, как и первая, показала способность новых алгоритмов находить достаточно качественные решения за небольшое время счёта. Оценки из [42] при малых λ оказались точнее и позволили сделать более объективные выводы о проводимых разработках.

Заключение

В работе рассмотрена NP-трудная задача конкурентного размещения и проектирования с гибким спросом. Разработаны алгоритмы её приближённого решения с использованием поиска с чередующимися окрестностями (VNS), а также предложена гибридная схема, сочетающая идеи алгоритма имитации отжига (SA) и VNS. Для выявления преимуществ и недостатков разрабатываемых методов, а также для настройки их параметров созданы серии тестовых примеров различного уровня сложности. Для определения качества построенных алгоритмов проведён численный эксперимент по их сравнению с коммерческим программным обеспечением (Baron, CoinBonmin, LocalSolver) и верхними оценками. Можно отметить, что известные решатели Baron и CoinBonmin оказались ограниченно пригодными для данных исследований, поскольку требовали большого времени счёта, а при разумном его ограничении часто не находили даже допустимого решения. Выявлены задачи и создана серия тестовых примеров, сложных для LocalSolver, на которых он уступает предложенным в работе алгоритмам. Построены апостериорные оценки точности новых алгоритмов, которые демонстрируют достаточно высокое качество разработок. Так, лучшая из предложенных гибридная схема на задачах большой размерности показывает отклонение от верхних оценок в интервале от 0,001 % до 4,742 %. При этом статистический анализ свидетельствует о том, что достаточно высока вероятность нахождения решения с отклонением меньше 3 %. В дальнейших исследованиях интересно реализовать параллельные версии данных алгоритмов и провести анализ улучшения качества получаемых решений. Также для рассматриваемой задачи представляет интерес построение точных методов решения.

Авторы благодарят проф. Н. Младеновича и проф. Ю. А. Кочетова за полезные советы.

ЛИТЕРАТУРА

1. **Леванова Т. В., Федоренко А. С.** Локальный поиск с чередующимися окрестностями для двустадийной задачи размещения // Дискрет. анализ и исслед. операций. 2008. Т. 15, № 3. С. 43–57.
2. **Discrete location theory.** New York: Wiley, 1990. 555 p.
3. **Kochetov Yu. A., Alekseeva E. V., Levanova T. V., Loresh M. A.** Large neighborhood local search for the p -median problem // Yugoslav J. Oper. Res. 2005. Vol. 15, No. 1. P. 53–63.
4. **Sridharan R.** The capacitated plant location problem // Eur. J. Oper. Res. 1995. Vol. 87, No. 1. P. 203–213.
5. **Hotelling H.** Stability in competition // Econ. J. 1929. Vol. 39. P. 41–57.

6. **Reilly W. J.** The law of retail gravitation. New York: Knickerbocker Press, 1931. 75 p.
7. **Huff D. L.** Defining and estimating a trade area // J. Mark. 1964. Vol. 28. P. 34–38.
8. **Huff D. L.** A programmed solution for approximating an optimum retail location // Land Econ. 1966. Vol 42. P. 293–303.
9. **Wilson A. G.** Retailers' profits and consumers' welfare in a spatial interaction shopping mode // Theory and practice in regional science. London: Pion, 1976. P. 42–59.
10. **Berman O., Krass D.** Locating multiple competitive facilities: Spatial interaction models with variable expenditures // Ann. Oper. Res. 2002. Vol. 111. P. 197–225.
11. **Aboolian R., Berman O., Krass D.** Competitive facility location model with concave demand // Eur. J. Oper. Res. 2007. Vol. 181, No. 2. P. 598–619.
12. **Aboolian R., Berman O., Krass D.** Competitive facility location and design problem // Eur. J. Oper. Res. 2007. Vol. 182, No. 1. P. 40–62.
13. **Drezner T., Drezner Z.** Lost demand in a competitive environment // J. Oper. Res. Soc. 2008. Vol. 59. P. 362–371.
14. **Drezner Z., Scott C. H.** Optimizing the location of a production firm // Netw. Spat. Econ. 2010. Vol. 10. P. 411–425.
15. **Drezner T., Drezner Z., Zerom D.** Competitive facility location with random attractiveness // Oper. Res. Lett. 2018. Vol. 46. P. 312–317.
16. **Küçükkaydin H., Necati Aras N., Altinel İ. K.** A leader–follower game in competitive facility location // Comput. Oper. Res. 2012. Vol. 39, No. 2. P. 437–448.
17. **Levanova T. V., Gnusarev A. Yu.** Variable neighborhood search approach for the location and design problem // Discrete Optimization and Operations Research. Proc. 9th Int. Conf. DOOR 2016 (Vladivostok, Russia, Sep. 19–23, 2016). Heidelberg: Springer, 2016. P. 159–170. (Lect. Notes Comput. Sci.; Vol. 9869).
18. **Levanova T. V., Gnusarev A. Yu.** Ant colony optimization for competitive facility location problem with elastic demand // Optimization Problems and Their Applications. Proc. School-Seminar OPTA-SCL 2018 (Omsk, Russia, July 8–14, 2018). Aachen: RWTH Aachen Univ., 2018. P. 239–248. (CEUR Workshop Proc.; Vol. 2098). Available at <http://ceur-ws.org/Vol-2098> (accessed July 2, 2020).
19. **Levanova T. V., Gnusarev A. Yu.** Development of threshold algorithms for a location problem with elastic demand // Large-Scale Scientific Computing. Rev. Sel. Pap. 11th Int. Conf. LSSC 2017 (Sozopol, Bulgaria, June 5–9, 2017). Cham: Springer, 2017. P. 382–389. (Lect. Notes Comput. Sci.; Vol. 10665).
20. **Levanova T. V., Gnusarev A. Yu.** Simulated annealing for competitive p -median facility location problem // J. Phys. Conf. Ser. 2018. Vol. 1050. P. 012044.1–5. Available at <https://iopscience.iop.org/article/10.1088/1742-6596/1050/1/012044/pdf> (accessed July 2, 2020).

21. **Levanova T. V., Gнусarev A. Yu.** Development of ant colony optimization algorithm for competitive p -median facility location problem with elastic demand // Mathematical Optimization Theory and Operations Research. Rev. Sel. Pap. 18th Int. Conf. MOTOR 2019 (Ekaterinburg, Russia, July 8–12, 2019). Cham: Springer, 2019. P. 68–78. (Commun. Comput. Inf. Sci.; Vol. 1090).
22. **McGarvey R. G., Cavalier T. M.** Determining the location and capacity of competitive facilities // Int. J. Math. Oper. Res. 2010. Vol. 2, No. 6. P. 694–723.
23. **Perl J., Ho P.** Public facilities location under elastic demand // Transp. Sci. 1990. Vol. 24, No. 2. P. 117–136.
24. **Pahlavani A., Saidi-Mehrabad M.** A competitive facility location model with elastic demand and patronising behaviour sensitive to location, price and waiting time // Int. J. Logist. Syst. Manage. 2011. Vol. 10, No. 3. P. 293–312.
25. **Wang X.** Location and design decisions of facilities in a distribution system with elastic customer demand // J. Shanghai Jiaotong Univ. (Sci.). 2009. Vol. 14, No. 5. P. 606–612.
26. **Ashtiani M. G.** Competitive location: A state-of-art review // Int. J. Ind. Eng. Comput. 2016. Vol. 7. P. 1–18.
27. **Berman O., Drezner T., Drezner Z., Krass D.** Modeling competitive facility location problems: New approaches and results // Decision Technologies and Applications. Catonsville: Inst. Oper. Res. Manage. Sci., 2009. P. 156–181. (INFORMS Tutor. Oper. Res.).
28. **Drezner T.** Gravity models in competitive facility location // Contributions to Location Analysis. Cham: Springer, 2019. P. 253–275. (Int. Ser. Oper. Res. Manage. Sci.; Vol. 281).
29. **Eiselt H. A., Marianov V., Drezner T.** Competitive location models // Location Science. Cham: Springer, 2015. P. 365–398.
30. **Karakitsiou A.** Modeling discrete competitive facility location. Cham: Springer, 2015. 54 p.
31. **Kress D., Pesch E.** Sequential competitive location on networks // Eur. J. Oper. Res. 2012. Vol. 217, No. 3. P. 483–499.
32. **Бахтин А. Е., Колоколов А. А., Коробкова З. В.** Дискретные задачи производственно-транспортного типа. Новосибирск: Наука, 1978. 160 с.
33. **Aboolian R., Berman O., Krass D.** Capturing market share: Facility location and design problem // Мат. Междунар. конф. «Дискретная оптимизация и исследование операций» (Новосибирск, Россия, 24–28 июня 2013 г.). Новосибирск: Изд-во Ин-та математики, 2013. С. 7–11.
34. **Nicholson T. A. J.** A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems // J. Inst. Math. Appl. 1965. Vol. 13. P. 362–375.
35. **Кочетов Ю. А., Младенович Н., Хансен П.** Локальный поиск с чередующимися окрестностями // Дискрет. анализ и исслед. операций. Сер. 2. 2003. Т. 10, № 1. С. 11–43.
36. **Hansen P., Mladenović N.** Variable neighborhood search: Principles and applications // Eur. J. Oper. Res. 2001. Vol. 130, No. 3. P. 449–467.

37. Anokić A., Stanimirović Z., Davidović T., Stakić Đ. Variable neighborhood search based approaches to a vehicle scheduling problem in agriculture // Int. Trans. Oper. Res. 2020. Vol. 27, No. 1. P. 26–56.
38. Handbook of metaheuristics. New York: Springer, 2010. 648 p. (Int. Ser. Oper. Res. Manage. Sci.; Vol. 146).
39. Кононова П. А., Кочетов Ю. А. Локальный поиск с чередующимися окрестностями для задачи Джонсона с пассивным буфером // Дискрет. анализ и исслед. операций. 2012. Т. 19, № 5. С. 63–82.
40. Pei J., Mladenović N., Urošević D., Brimberg J., Liu X. Solving the traveling repairman problem with profits: A novel variable neighborhood search approach // Inf. Sci. 2020. Vol. 507. P. 108–123.
41. Dréo J., Pétrowski A., Siarry P., Taillard E. Metaheuristics for hard optimization. Heidelberg: Springer, 2006. 369 p.
42. Леванова Т. В., Белан С. Е. Применение верхних оценок для анализа алгоритмов приближённого решения конкурентной задачи размещения с гибким спросом // Вестн. Омск. гос. ун-та. 2017. № 4. С. 4–10.
43. The general algebraic modeling system. Fairfax: GAMS Development, 2020. Available at <http://www.gams.com> (accessed July 2, 2020).
44. LocalSolver. Paris: LocalSolver, 2020. Available at <http://www.localsolver.com> (accessed July 2, 2020).

Татьяна Валентиновна Леванова
Александр Юрьевич Гнусарев

Статья поступила
17 апреля 2017 г.
После доработки —
7 июня 2020 г.
Принята к публикации
19 июня 2020 г.

VARIABLE NEIGHBORHOOD SEARCH ALGORITHMS
FOR A COMPETITIVE LOCATION PROBLEM
WITH ELASTIC DEMAND

T. V. Levanova^{1,2, a} and A. Yu. Gnusarev^{1, b}

¹Omsk Branch of Sobolev Institute of Mathematics,
13 Pevtsov Street, 644099 Omsk, Russia

²Dostoevsky Omsk State University,
55a Mir Avenue, 644077 Omsk, Russia

E-mail: ^alevanova@ofim.oscsbras.ru, ^balexander.gnussarev@gmail.com

Abstract. Under consideration is the situation in a competitive market when a new Company plans to make profit from opening its own facilities that offer goods or services. The Company have to take it into account that there are several projects for opening each facility, and similar facilities of the Competitor are already placed on the market. Moreover, customers themselves choose the places to meet their demand in dependence on where and which facilities are located. The Company's goal is to choose locations and projects for opening new facilities in order to attract the largest share of all customer demand. The special type of demand leads to nonlinearity of the objective function and to additional difficulties in finding an optimal solution. In this article we construct some variants of variable neighborhood search algorithms, perform their experimental analysis by using the upper estimates, obtain a posteriori accuracy estimates, and discuss the results. Tab. 4, illustr. 2, bibliogr. 44.

Keywords: location problem, competition, elastic demand, heuristic, variable neighborhood search.

Sections 1 and 2 are performed by T. V. Levanova with the support of the Program for Fundamental Scientific Research of the State Academies of Sciences for 2013–2020 No. I.5 (Project 0314–2019–0019). Section 3 is performed by A. Yu. Gnusarev with the support of the Russian Foundation for Basic Research (Project 18–07–00599).

English version: Journal of Applied and Industrial Mathematics **14** (4), 693–705 (2020), DOI 10.1134/S1990478920040080.

REFERENCES

1. **T. V. Levanova** and **A. S. Fedorenko**, Variable neighborhood search for two-stage facility location problem, *Diskretn. Anal. Issled. Oper.* **15** (3), 43–57 (2008) [Russian].
2. *Discrete Location Theory* (Wiley, New York, 1990).
3. **Yu. A. Kochetov**, **E. V. Alekseeva**, **T. V. Levanova**, and **M. A. Loresh**, Large neighborhood local search for the p -median problem, *Yugoslav J. Oper. Res.* **15** (1), 53–63 (2005).
4. **R. Sridharan**, The capacitated plant location problem, *Eur. J. Oper. Res.* **87** (1), 203–213 (1995).
5. **H. Hotelling**, Stability in competition, *Econ. J.* **39**, 41–57 (1929).
6. **W. J. Reilly**, *The Law of Retail Gravitation* (Knickerbocker Press, New York, 1931).
7. **D. L. Huff**, Defining and estimating a trade area, *J. Mark.* **28**, 34–38 (1964).
8. **D. L. Huff**, A programmed solution for approximating an optimum retail location, *Land Econ.* **42**, 293–303 (1966).
9. **A. G. Wilson**, Retailers' profits and consumers' welfare in a spatial interaction shopping mode, in *Theory and Practice in Regional Science* (Pion, London, 1976), pp. 42–59.
10. **O. Berman** and **D. Krass**, Locating multiple competitive facilities: Spatial interaction models with variable expenditures, *Ann. Oper. Res.* **111**, 197–225 (2002).
11. **R. Aboolian**, **O. Berman**, and **D. Krass**, Competitive facility location model with concave demand, *Eur. J. Oper. Res.* **181** (2), 598–619 (2007).
12. **R. Aboolian**, **O. Berman**, and **D. Krass**, Competitive facility location and design problem, *Eur. J. Oper. Res.* **182** (1), 40–62 (2007).
13. **T. Drezner** and **Z. Drezner**, Lost demand in a competitive environment, *J. Oper. Res. Soc.* **59**, 362–371 (2008).
14. **Z. Drezner** and **C. H. Scott**, Optimizing the location of a production firm, *Netw. Spat. Econ.* **10**, 411–425 (2010).
15. **T. Drezner**, **Z. Drezner**, and **D. Zerom**, Competitive facility location with random attractiveness, *Oper. Res. Lett.* **46**, 312–317 (2018).
16. **H. Küçükaydin**, **N. Necati Aras**, and **İ. K. Altinel**, A leader–follower game in competitive facility location, *Comput. Oper. Res.* **39** (2), 437–448 (2012).
17. **T. V. Levanova** and **A. Yu. Gnusarev**, Variable neighborhood search approach for the location and design problem, in *Discrete Optimization and Operations Research* (Proc. 9th Int. Conf. DOOR 2016, Vladivostok, Russia, Sep. 19–23, 2016) (Springer, Heidelberg, 2016), pp. 159–170 (Lect. Notes Comput. Sci., Vol. 9869).
18. **T. V. Levanova** and **A. Yu. Gnusarev**, Ant colony optimization for competitive facility location problem with elastic demand, in *Optimization Problems and Their Applications* (Proc. School-Seminar OPTA-SCL 2018, Omsk, Russia, July 8–14, 2018) (RWTH Aachen Univ., Aachen, 2018), pp. 239–248 (CEUR Workshop Proc., Vol. 2098). Available at <http://ceur-ws.org/Vol-2098> (accessed July 2, 2020).

19. **T. V. Levanova** and **A. Yu. Gnusarev**, Development of threshold algorithms for a location problem with elastic demand, in *Large-Scale Scientific Computing* (Rev. Sel. Pap. 11th Int. Conf. LSSC 2017, Sozopol, Bulgaria, June 5–9, 2017) (Springer, Cham, 2017), pp. 382–389 (Lect. Notes Comput. Sci., Vol. 10665).
20. **T. V. Levanova** and **A. Yu. Gnusarev**, Simulated annealing for competitive p -median facility location problem, *J. Phys. Conf. Ser.* **1050**, 012044.1–5 (2018). Available at <https://iopscience.iop.org/article/10.1088/1742-6596/1050/1/012044/pdf> (accessed July 2, 2020).
21. **T. V. Levanova** and **A. Yu. Gnusarev**, Development of ant colony optimization algorithm for competitive p -median facility location problem with elastic demand, in *Mathematical Optimization Theory and Operations Research* (Rev. Sel. Pap. 18th Int. Conf. MOTOR 2019, Ekaterinburg, Russia, July 8–12, 2019) (Springer, Cham, 2019), pp. 68–78 (Commun. Comput. Inf. Sci., Vol. 1090).
22. **R. G. McGarvey** and **T. M. Cavalier**, Determining the location and capacity of competitive facilities, *Int. J. Math. Oper. Res.* **2** (6), 694–723 (2010).
23. **J. Perl** and **P. Ho**, Public facilities location under elastic demand, *Transp. Sci.* **24** (2), 117–136 (1990).
24. **A. Pahlavani** and **M. Saidi-Mehrabad**, A competitive facility location model with elastic demand and patronising behaviour sensitive to location, price and waiting time, *Int. J. Logist. Syst. Manage.* **10** (3), 293–312 (2011).
25. **X. Wang**, Location and design decisions of facilities in a distribution system with elastic customer demand, *J. Shanghai Jiaotong Univ. (Sci.)* **14** (5), 606–612 (2009).
26. **M. G. Ashtiani**, Competitive location: A state-of-art review, *Int. J. Ind. Eng. Comput.* **7**, 1–18 (2016).
27. **O. Berman**, **T. Drezner**, **Z. Drezner**, and **D. Krass**, Modeling competitive facility location problems: New approaches and results, in *Decision Technologies and Applications* (Inst. Oper. Res. Manage. Sci., Catonsville, 2009), pp. 156–181 (INFORMS Tutor. Oper. Res.).
28. **T. Drezner**, Gravity models in competitive facility location, in *Contributions to Location Analysis* (Springer, Cham, 2019), pp. 253–275 (Int. Ser. Oper. Res. Manage. Sci., Vol. 281).
29. **H. A. Eiselt**, **V. Marianov**, and **T. Drezner**, Competitive location models, in *Location Science* (Springer, Cham, 2015), pp. 365–398.
30. **A. Karakitsiou**, *Modeling Discrete Competitive Facility Location* (Springer, Cham, 2015).
31. **D. Kress** and **E. Pesch**, Sequential competitive location on networks, *Eur. J. Oper. Res.* **217** (3), 483–499 (2012).
32. **A. E. Bakhtin**, **A. A. Kolokolov**, and **Z. V. Korobkova**, *Discrete Problems of Production and Transport Type* (Nauka, Novosibirsk, 1978) [Russian].
33. **R. Aboolian**, **O. Berman**, and **D. Krass**, Capturing market share: Facility location and design problem, in *Proc. Int. Conf. “Discrete Optimization and Operations Research”*, Novosibirsk, Russia, June 24–28, 2013 (Izd. Inst. Mat., Novosibirsk, 2013), pp. 7–11.

34. **T. A. J. Nicholson**, A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems, *J. Inst. Math. Appl.* **13**, 362–375 (1965).
35. **Yu. A. Kochetov**, **N. Mladenović** and **P. Hansen**, Local search with alternating neighborhoods, *Diskretn. Anal. Issled. Oper., Ser. 2*, **10** (1), 11–43 (2003) [Russian].
36. **P. Hansen** and **N. Mladenović**, Variable neighborhood search: Principles and applications, *Eur. J. Oper. Res.* **130** (3), 449–467 (2001).
37. **A. Anokić**, **Z. Stanimirović**, **T. Davidović** and **Đ. Stakić**, Variable neighborhood search based approaches to a vehicle scheduling problem in agriculture, *Int. Trans. Oper. Res.* **27** (1), 26–56 (2020).
38. *Handbook of Metaheuristics* (Springer, New York, 2010) (Int. Ser. Oper. Res. Manage. Sci., Vol. 146).
39. **P. A. Kononova** and **Yu. A. Kochetov**, The variable neighborhood search for the two machine flow shop problem with a passive prefetch, *Diskretn. Anal. Issled. Oper.* **19** (5), 63–82 (2012) [Russian] [*J. Appl. Ind. Math.* **7** (1), 54–67 (2013)].
40. **J. Pei**, **N. Mladenović**, **D. Urošević**, **J. Brimberg**, and **X. Liu**, Solving the traveling repairman problem with profits: A novel variable neighborhood search approach, *Inf. Sci.* **507**, 108–123 (2020).
41. **J. Dréo**, **A. Pétrowski**, **P. Siarry**, and **E. Taillard**, *Metaheuristics for Hard Optimization* (Springer, Heidelberg, 2006).
42. **T. V. Levanova** and **S. E. Belan**, Application of upper bounds for analysis of approximate algorithms for a competitive location problem with elastic demand, *Vestn. Omsk. Gos. Univ.*, No. 4, 4–10 (2017) [Russian].
43. *The General Algebraic Modeling System* (GAMS Development, Fairfax, 2020). Available at <http://www.gams.com> (accessed July 2, 2020).
44. *LocalSolver* (LocalSolver, Paris, 2020). Available at <http://www.localsolver.com> (accessed July 2, 2020).

Tatyana V. Levanova
Aleksandr Yu. Gnusarev

Received April 17, 2017
Revised June 7, 2020
Accepted June 19, 2020