

АЛГЕБРО-ЛОГИЧЕСКИЕ МЕТОДЫ В ИНФОРМАТИКЕ  
И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

ALGEBRAIC AND LOGICAL METHODS IN COMPUTER  
SCIENCE AND ARTIFICIAL INTELLIGENCE



Серия «Математика»  
2024. Т. 48. С. 95–110

Онлайн-доступ к журналу:  
<http://mathizv.isu.ru>

---

---

ИЗВЕСТИЯ  
Иркутского  
государственного  
университета

---

---

Научная статья

УДК 519.8+519.6

MSC 90C59, 68W20

DOI <https://doi.org/10.26516/1997-7670.2024.48.95>

Модели и алгоритмы локального поиска  
для маршрутизации транспортных средств  
с возвратами и временными окнами

Л. А. Заозерская<sup>1</sup>, Ю. В. Захарова<sup>1</sup>✉

<sup>1</sup> Институт математики им. С. Л. Соболева СО РАН, Омск, Российская Федерация  
✉ [yzakharova@ofim.oscsbras.ru](mailto:yzakharova@ofim.oscsbras.ru)

**Аннотация.** Рассматривается задача маршрутизации транспортных средств для обслуживания объектов с возможностью распределения работ и учетом временных окон. Обсуждаются подходы к построению моделей целочисленного линейного программирования и их модификаций. Для поиска приближенных решений предлагается алгоритм локального поиска с «большими» окрестностями, где на каждом шаге осуществляется поиск улучшающего решения при помощи стратегий «сломай» и «почини». Строятся серии тестовых примеров различной структуры, в том числе демонстрируется целесообразность рассмотрения постановки с возвратами транспортных средств на объекты. Представлены результаты экспериментального исследования моделей и алгоритмов.

**Ключевые слова:** маршрутизация, локальный поиск, целочисленное программирование, модель

**Благодарности:** Работа выполнена в рамках государственного задания ИМ СО РАН, проект FWNF-2022-0020.

**Ссылка для цитирования:** Заозерская Л. А., Захарова Ю. В. Модели и алгоритмы локального поиска для маршрутизации транспортных средств с возвратами и временными окнами // Известия Иркутского государственного университета. Серия

Математика. 2024. Т. 48. С. 95–110.  
<https://doi.org/10.26516/1997-7670.2024.48.95>

Research article

## Models and Local Search Algorithms for Vehicle Routing with Returns and Time Windows

Lidia A. Zaozerskaya<sup>1</sup>, Yulia V. Zakharova<sup>1</sup>✉

<sup>1</sup> Sobolev Institute of Mathematics SB RAS, Omsk, Russian Federation  
✉ yzakharova@ofim.oscsbras.ru

**Abstract.** A vehicle routing problem for servicing objects with the possibility of distributing work between vehicle taking into account time windows is considered. We discuss approaches to constructing integer linear programming models and their modifications. A large neighborhood search algorithm is proposed to find approximate solutions. Here an improving solution is constructed using the “destroy” and “repair” methods at each step. A series of instances with various structures are proposed, in particular demonstrating the reason of returns to objects for vehicles. The results of an experimental evaluation of models and algorithms are presented.

**Keywords:** vehicle routing, local search, integer programming, model

**Acknowledgements:** The research was funded in accordance with the state task of the IM SB RAS, project FWNF-2022-0020.

**For citation:** Zaozerskaya L. A., Zakharova Yu. V. Models and Local Search Algorithms for Vehicle Routing with Returns and Time Windows. *The Bulletin of Irkutsk State University. Series Mathematics*, 2024, vol. 48, pp. 95–110. (in Russian)  
<https://doi.org/10.26516/1997-7670.2024.48.95>

### 1. Введение

Задачи маршрутизации транспортных средств активно исследуются в различных постановках, где учитываются ограничения на маршруты передвижения, способы доставки, временные окна, варианты и ограничения загрузки, а также другие условия, возникающие из специфики прикладных задач. Более того, в условиях могут учитываться различные типы неопределенностей в исходных данных и динамические составляющие. Рассматриваемые задачи являются NP-трудными, поэтому для их решения строятся модели частично целочисленного линейного программирования (ЧЦЛП), как правило, с целью использования пакетов прикладных программ, а также предлагаются различные конструктивные методы и метаэвристики.

В алгоритмах локального поиска для задач маршрутизации используются различные типы окрестностей: обмены внутри маршрута

– relocate, exchange, 2-opt, 3-opt,  $\lambda$ -opt, or-exchange, GENI; обмены между маршрутами – 2-opt, insert, swap, cross,  $\lambda$ -interchange [7].

В работе [10] предлагается меметический алгоритм для задач маршрутизации, где наряду с основной задачей решаются ее упрощенные версии и накопленные «знания» о перспективных маршрутах передаются в процессе работы в эволюционный поиск для исходной задачи. В локальном поиске используется окрестность типа insert. Для задачи маршрутизации транспортных средств с размещением заказов по упаковочным местам с учетом ограничений по объемам и совместимости построена модель ЧЦЛП и предложен алгоритм имитации отжига, действующий окрестности swap, insertion и inversion [11]. Для близкой задачи, где дополнительно учитывается возможность гибких доставок, разработан гибридный алгоритм локального поиска с чередующимися окрестностями и поиск с запретами [9]. Такие задачи характерны для городской логистики и транспортировки электронных заказов.

Для задачи минимизации числа транспортных средств с учетом временных окон на объектах, различных депо и вариантов перемещения предложен адаптивный алгоритм локального поиска с чередующимися окрестностями [3]. Для задачи из реальной практики, где дополнительно учитываются нижняя граница на загрузку транспортных средств и время на загрузку/отгрузку товара, также построены адаптивный локальный поиск с «большими» окрестностями и модель ЧЦЛП [8].

В настоящей работе рассматривается задача маршрутизации транспортных средств с обслуживанием объектов, на которых допускается разделение работ при условии, что они должны быть выполнены в указанные временные окна. Технические свойства транспортных средств влияют на длительность обслуживания и перемещения, в том числе они могут задавать запреты на возможность обслуживания и перемещения. Данная задача возникает, например, при маршрутизации буровых установок для бурения скважин на объектах, а также описывает другие обслуживающие системы с несколькими клиентами на объектах [2; 5]. Поиск допустимого решения этой задачи NP-труден [2; 5].

Два подхода к построению моделей ЧЦЛП для указанной задачи представлены в [2; 5]. В [2] предложена метаэвристика поиска с чередующимися окрестностями, где действует аппарат решения задач ЧЦЛП пониженной размерности. В [5; 6] теоретически исследована стохастическая постановка задачи, а также разработаны жадный метод и генетический алгоритм для ее решения. Текущее исследование продолжает экспериментальный анализ моделей ЧЦЛП для рассматриваемой задачи маршрутизации [1; 5]. Построено семейство примеров с евклидовыми расстояниями между объектами, где оптимальное решение гарантированно предполагает многократное посещение объекта буровой установкой для обслуживания скважин. Построен и экспериментально исследован рандомизированный алгоритм локального поис-

ка с «большими» окрестностями, основанный на решении серии подзадач пониженной размерности. Для построения начального решения предлагается два конструктивных подхода.

## 2. Постановка задачи

Исследуется задача маршрутизации буровых установок для обслуживания объектов с заданными временными окнами в следующей постановке. Объекты из множества  $I = \{i_1, \dots, i_{|I|}\}$  должны быть обслужены транспортными средствами (буровыми установками) из множества  $U = \{u_1, \dots, u_{|U|}\}$ . Каждый объект  $i \in I$  характеризуется наличием  $n_i$  скважин для бурения и временным окном  $(a_i, b_i]$ , в котором все работы по бурению должны быть выполнены. Обозначим через  $U_i$  множество транспортных средств, которые могут обслуживать объект  $i \in I$ .

Бурение скважины осуществляется одной буровой установкой без прерываний. Различные скважины объекта могут обслуживаться различными буровыми установками. Обозначим через  $d_{ui}$  длительность бурения одной скважины на объекте  $i \in I$  буровой установкой  $u \in U$ . Для каждого транспортного средства  $u \in U$  указано подмножество объектов  $I_u$ , которые оно может обслуживать. Изначально всякое транспортное средство  $u$  находится в некотором депо, обозначаемом через  $id_u$ . Пусть величина  $s_{uij}$  представляет длительность перемещения между объектами  $i$  и  $j$  для транспортного средства  $u$ .

Требуется найти маршруты передвижения между объектами для транспортных средств и распределить скважины для обслуживания между транспортными средствами, посещающими соответствующие объекты, таким образом, чтобы минимизировать суммарную длительность перемещений и гарантировать соблюдение ограничений по временным окнам для объектов. Напомним, что для этой задачи поиск допустимого решения NP-труден [5].

Рассмотрим вариант задачи маршрутизации, когда буровая установка может посещать объект не более одного раза (это задача  $\mathcal{P}$ ). В [1; 5] предлагаются две модели, одна из которых основана на назначении объектов в позиции перестановок, а вторая использует свойство смежности при посещении объектов. Далее будем использовать модель второго типа в варианте со штрафами, так как она показала более перспективные результаты в предварительном вычислительном эксперименте [1; 5].

Введем фиктивный объект  $f_s$ , соответствующий начальной и конечной точке маршрута любого транспортного средства, и обозначим  $I_f = I \cup \{f_s\}$ . Положим длительности перемещения  $s_{uif_s} = 0$ ,  $s_{uf_si} = d_{id_{ui}}$  для  $i \in I$ . Будем считать, что буровые установки могут посещать  $f_s$ , т. е.  $U_{f_s} = U$ .

Определим булевы переменные  $x_{uij}$  так, что  $x_{uij} = 1$ , если транспортное средство  $u$  посещает объект  $i$  и перемещается к объекту  $j$ , и  $x_{uij} = 0$  иначе. Пусть вещественные переменные  $t_{ui}^s$  задают момент начала работ для буровой установки  $u$  на объекте  $i$  и целочисленные переменные  $y_{ui}$  определяют число скважин объекта  $i$ , обслуживаемых буровой установкой  $u$ . Добавим неотрицательные переменные  $\tau_{ui}$ ,  $i \in I$ ,  $u \in U$ , значения которых не меньше величины превышения буровой установкой  $u$  правой границы временного окна для объекта  $i$ , и  $\tau_i^0$ , значения которых не меньше величины наибольшего среди запаздываний буровых установок для объекта  $i$ ,  $i \in I$ .

Тогда для задачи  $\mathcal{P}$  модель ЧЦЛП со штрафами может быть записана следующим образом: минимизировать

$$f = \sum_{u \in U} \sum_{i \in I_f} \sum_{j \in I_f} s_{uij} x_{uij} + M_f \sum_{i \in I} \tau_i^0, \quad (2.1)$$

$$\sum_{j \in I_f} x_{uij} = \sum_{j \in I_f} x_{uji}, \quad u \in U, \quad i \in I_u \setminus \{f_s\}, \quad (2.2)$$

$$\sum_{i \in I_u} x_{uij} \leq 1, \quad u \in U_j, \quad j \in I, \quad (2.3)$$

$$\sum_{u \in U_i} y_{ui} = n_i, \quad i \in I, \quad (2.4)$$

$$\sum_{j \in I_u} x_{uji} \leq y_{ui} \leq n_i \sum_{j \in I_u} x_{uji}, \quad i \in I, \quad u \in U_i, \quad (2.5)$$

$$t_{ui}^s + y_{ui} d_{ui} + s_{uij} \leq t_{uj}^s + M_r(1 - x_{uij}), \quad i, j \in I, \quad i \neq j, \quad u \in U_i \cap U_j, \quad (2.6)$$

$$s_{u,f_s,j} \leq t_{uj}^s + M_r(1 - x_{u,f_s,j}), \quad j \in I, \quad u \in U_j, \quad (2.7)$$

$$t_{ui}^s \geq \sum_{j \in I_u} a_i x_{uji}, \quad i \in I, \quad u \in U_i, \quad (2.8)$$

$$t_{ui}^s + y_{ui} d_{ui} \leq \sum_{j \in I_u} b_i x_{uji} + \tau_{ui}, \quad i \in I, \quad u \in U_i, \quad (2.9)$$

$$\tau_i^0 \geq \tau_{ui}, \quad u \in U, \quad i \in I, \quad (2.10)$$

$$\sum_{i \in I_u} x_{u,f_s,i} = 1, \quad \sum_{i \in I_u} x_{uif_s} = 1, \quad u \in U. \quad (2.11)$$

Здесь и далее  $M_r$  и  $M_f$  обозначают достаточно большие константы. Ограничения (2.2)–(2.3) гарантируют, что в маршрутах буровые установки посещают объекты не более одного раза. Ограничения (2.4)–(2.5) позволяют учесть выполнение работ по бурению всех скважин на объектах и использование ровно одной буровой установки для бурения

каждой скважины. Ограничения (2.6)–(2.7) задают время начала работ на объектах для буровых установок. Ограничения (2.8)–(2.9) гарантируют условную допустимость маршрутов в контексте временных окон (правая граница временного окна может быть нарушена). Равенства (2.11) показывают, что транспортные средства начинают и заканчивают маршрут в фиктивном объекте  $f$ .

В отличие от  $\mathcal{P}$  задача (2.1)–(2.11) всегда имеет допустимое решение. Если при этом  $\tau_{ui} = 0$  и  $\tau_i^0 = 0$ ,  $i \in I$ ,  $u \in U$ , то такое решение является допустимым и для  $\mathcal{P}$  (задачи маршрутизации с временными окнами).

Модель для случая возможности возвратов буровой установки на объект строится по правилу, когда для каждого объекта  $i \in I$  создается  $m_i$  его копий, что позволяет смоделировать посещение объекта буровой установкой не более  $m_i$  раз за счет посещения копий не более одного раза. Переменные и ограничения обобщаются естественным образом на копии объектов (см., например, [1]).

## 2.1. ПРИМЕРЫ

Рассмотрим пример с 6 объектами, 3 буровыми установками и следующими входными данными. Количество скважин  $n = (5, 8, 5, 8, 5, 8)$ , временные окна определяют вектора  $a = (20, 10, 30, 20, 40, 30)$  и  $b = (30, 39, 40, 49, 50, 59)$ ,  $d_{ui} = 2$ ,  $u \in U$ ,  $i \in I$ . Для всех  $i, j$  из  $I$   $s_{uij}$  не зависят от  $u$  и указаны на рис. 1, если возможен непосредственный переезд между объектами  $i$  и  $j$ . Остальные  $s_{uij} = \infty$ . Вершина  $id$  соответствует начальному расположению буровых установок.

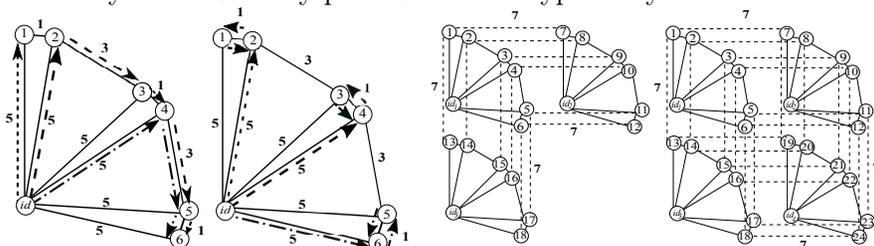


Рис. 1. Слева – решение задачи с  $|I|=6$  и  $|U| = 3$  без возвратов и с возвратами, справа – схема соединения объектов для блоков 3 и 4.

Для  $i = 1, 3, 5$  выполнено условие  $(a_i, b_i] \subset (a_{i+1}, b_{i+1}]$ . Легко видеть, что если возврат буровых установок на ранее посещенные объекты запрещен, то переход с объекта  $i + 1$  на объект  $i$  в этих парах нецелесообразен. Оптимальное решение со значением  $f = 27$  представлено на рис. 1 (слева). Маршрут каждой буровой установки имеет уникальный тип штриховки линии. Число обслуживаемых буровыми установками скважин:  $y_{11} = 5$ ,  $y_{22} = 8$ ,  $y_{23} = 5$ ,  $y_{24} = 1$ ,  $y_{25} = 2$ ,  $y_{26} = 4$ ,  $y_{34} = 7$ ,  $y_{35} = 3$ ,  $y_{36} = 4$ . Если разрешены возвраты на ранее посещенные объекты, то буровая установка может пробурить 4 скважины на объекте

$i+1$ , затем перейти на объект  $i$ , выполнить там всю работу, вернуться на объект  $i+1$  для обработки оставшихся там скважин. Маршрут, соответствующий оптимальному решению с  $f = 21$ , показан на рис. 1 (слева). Отметим, что малые изменения в расположении объектов, в частности сдвиг объектов 1 и 2 на расстояние 2, приводят к нецелесообразности многократного посещения этих объектов буровыми установками.

На основе этого примера мы построили семейство задач с  $6k$  объектами и  $3k$  установками с оптимальным значением целевой функции  $21k$  (если возвраты разрешены) и  $27k$  (если возвраты запрещены),  $k \in N$ . Здесь  $k$  блоков из 6 объектов объединяются по принципу «окаймляющих миноров» в направлении сверху вниз и слева направо. Объекты каждого блока обслуживаются своими тройками буровых установок. Схемы соединения блоков для  $k$ , равного 3 и 4, представлены на рис. 1 (справа). Штрихованные горизонтальные и вертикальные линии, соединяющие блоки, означают, что разрешены перемещения между одноименными объектами соседних блоков с одинаковым расстоянием, равным 7.

### 3. Эвристика локального поиска

В этом разделе опишем эвристический алгоритм поиска в «большой» окрестности с этапами «сломай» и «почини» для приближенного решения задачи  $\mathcal{P}$  маршрутизации буровых установок без возвратов. Алгоритм основан на использовании рандомизированного локального поиска, где решается серия задач со штрафами вида (2.1)–(2.11) пониженной размерности с помощью пакета решения задач ЧЦЛП. На этапе «сломай» выбираются объекты для удаления из маршрутов, а на этапе «почини» удаленные объекты перераспределяются между маршрутами.

Обозначим через  $W = (X, Y, T, \tau, \tau^0)$  решение задачи (2.1)–(2.11), где  $X, Y, T, \tau, \tau^0$  – множества значений всех переменных  $x_{uij}, y_{ui}, t_{ui}^s, \tau_{ui}, \tau_i^0$  соответственно. По компоненте  $X$  легко восстановить маршрут  $route(u)$  для каждой буровой установки  $u$  как последовательность номеров объектов в порядке их прохождения, т.е.  $route(u) = (f_s, i_1^u, i_2^u, \dots, i_{l(u)}^u, f_s)$ , где  $l(u)$  – длина маршрута и объекты  $i_j^u \in I$ ,  $j = 1, \dots, l(u)$ . Для двух выделенных объектов на маршруте подпоследовательность объектов маршрута между этими объектами называется подмаршрутом, например  $subroute(u, f_s, i_{l(u)}) = (i_1, i_2, \dots, i_{l(u)-1})$ .

Определим «большую» окрестность решения, которая описывается только перемещениями буровых установок [4]. Для решения  $W$  зададим «большую»  $(r, k)$ -окрестность как множество решений, которые отличаются от  $W$  не более чем на  $r$  подмаршрутах, принадлежащих разным маршрутам, и общее число объектов в этих подмаршрутах не

превосходит  $k$ . Данная окрестность включает в себя окрестности типа  $N_2(\sigma)$  (при  $r = 2, k = 2$ ),  $N_3(\sigma)$  (при  $r = 2, k = 2$ ),  $N_4(\sigma)$  из [2].

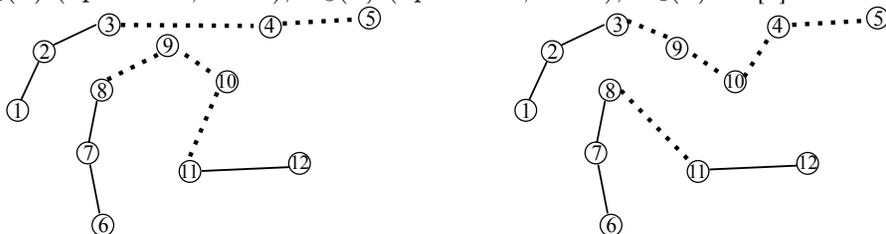


Рис. 2. Переход от одного решения к другому в  $(r, k)$ -окрестности (пунктиром выделены подмаршруты до и после изменения)

Предлагаемый алгоритм начинает работу с допустимого решения задачи (2.1)–(2.11). На каждой итерации случайным образом выбирается подмножество буровых установок. Для каждой из них также случайным образом определяется подмаршрут. Далее выполняется попытка улучшить текущее решение за счет обмена объектами между выбранными подмаршрутами и/или перераспределения обслуживаемых скважин для всех объектов между всеми установками. Если получено решение с лучшим значением целевой функции, то осуществляется переход в это решение. На рис. 2 приведена иллюстрация модификации решения.

### 3.1. ПОИСК НАЧАЛЬНОГО РЕШЕНИЯ

Предлагается два алгоритма получения начального решения, где буровые установки назначаются объектам, упорядоченным по неубыванию границ временных окон. Первый из них IS1 описывается правилом «случайный выбор из разрешенных буровых установок», а второй IS2 – «назначение ближайшей буровой установки из числа разрешенных».

Далее будем описывать перемещение буровых установок как через значения переменных  $x_{uij}$ ,  $u \in U$ ,  $i, j \in I$ , так и через множество их маршрутов  $route(u)$ .

**Алгоритм 1.** Построение решения по правилу «случайный выбор из разрешенных буровых установок» (алгоритм IS1):

**Шаг 1.** Упорядочить объекты по неубыванию значений  $a_i$ ,  $i \in I$ , при одинаковых  $a_i$  по неубыванию правых границ  $b_i$ . Далее под  $I$  понимаем множество упорядоченных объектов.

**Шаг 2.** Выполнить для всех объектов  $i \in I$ :

положить  $\bar{U}_i = U_i$  и  $\bar{n}_i = n_i$ ;

если  $|\bar{U}_i| > 1$ , то

пока  $\bar{n}_i > 0$ :

выбираем случайным образом  $u$  из  $\bar{U}_i$ ;

если  $\bar{n}_i > 1$ , то назначаем случайным образом  $y_{ui}$  (число обрабатываемых установкой  $u$  скважин),  $1 \leq y_{ui} \leq \bar{n}_i$ ;

иначе  $y_{ui} = 1$ ;  
 полагаем  $\bar{U}_i = \bar{U}_i \setminus \{u\}$  и  $\bar{n}_i = \bar{n}_i - y_{ui}$ ;  
 иначе назначаем буровую установку  $u$  из  $\bar{U}_i$  и полагаем  
 $\bar{U}_i = \emptyset$ ,  $y_{ui} = \bar{n}_i$ .

**Шаг 3.** Вычислить целевую функцию  $f$  на полученном решении  $W = (X, Y, T)$ .

Алгоритм IS2 отличается от IS1 правилом выбора буровой установки. Через  $PreviousObject(u)$  обозначим объект, в котором находится буровая установка  $u$  на текущей итерации. Приведем отличия в описании алгоритма IS2. На Шаге 1 для всех  $u$  из  $U$  полагаем  $PreviousObject(u) = f_s$ . На Шаге 2 пока  $\bar{n}_i > 0$  выбираем  $u$  из  $\bar{U}_i$  по правилу

$$s(PreviousObject(u), i, u) = \min_{\bar{u} \in \bar{U}_i} s(PreviousObject(\bar{u}), i, \bar{u}).$$

Если таких  $u$  несколько, то выбираем любое среди них случайным образом, иначе  $u$  определяется однозначно. При изменении  $\bar{U}_i$  и  $\bar{n}_i$  дополнительно полагаем  $PreviousObject(u) = i$ .

Для решения, построенного алгоритмом IS1 или IS2, выполнены все ограничения задачи  $\mathcal{P}$  за исключением, возможно, ограничений, связанных с соблюдением правых границ временных окон.

Вычислить целевую функцию  $f$  можно непосредственно или решить для этого задачу (2.1)–(2.11) при фиксированных значениях  $x_{uij}$  и  $y_{ui}$ , соответствующих назначениям алгоритма. Другим вариантом является решение задачи (2.1)–(2.11), когда фиксируются только значения  $x_{uij}$ .

### 3.2. АЛГОРИТМ РАНДОМИЗИРОВАННОГО ЛОКАЛЬНОГО ПОИСКА

Приведем формальное описание предлагаемого алгоритма рандомизированного локального поиска (РЛП) с «большими» окрестностями.

**Алгоритм 2.** Локальный поиск с «большими» окрестностями (два основных этапа: «сломай» и «почини»)

**Шаг 1.** Построение начального решения.

Применяя алгоритм IS1 и/или IS2 заданное число раз, получить множество допустимых решений для задачи (2.1)–(2.11). Решение с наименьшим значением целевой функции  $f_{best}$  становится начальным решением  $W_{best} = (X, Y, T)$ .

**Шаг 2.** Улучшение маршрутов. Выполнять заданное время.

2.1. Этап «сломай».

2.1.1. Выбор маршрутов. Сформировать случайным образом множество  $\bar{U} \subseteq U$  такое, что  $|\bar{U}| = r$ . Положить  $O = \emptyset$ . Для каждого  $u \in \bar{U}$  случайным образом выбрать на маршруте  $route(u)$  два объекта  $start(u)$  и  $finish(u)$ , где  $start(u)$  предшествует  $finish(u)$ . Суммарная длина подмаршрутов

для  $u$  из  $\bar{U}$  не превосходит  $k$ .

2.1.2. *Выбор объектов.* В  $O$  включить все объекты подмаршрутов без объектов  $start(u)$ ,  $finish(u)$ ,  $u \in U$ .

2.2. *Этап «почини».*

2.2.1. *Подготовка модели.* Зафиксировать значения переменных  $x_{uij}$  для  $i, j \notin O$ . Для всех  $u \in \bar{U}$  положить  $\bar{O}(u) = O$  и удалить из  $\bar{O}(u)$  те объекты, которые встречаются для  $u$  на подмаршруте от  $f_s$  до  $start(u)$  и на подмаршруте от  $finish(u)$  до  $f_s$ .

2.2.2. *Решение подзадачи.* Заданное время timeMIP решать задачу ЧЦЛП (2.1)–(2.11), в которой для  $u \in \bar{U}$  булевые переменные  $x_{uij}$  имеют только индексы, соответствующие следующим возможным перемещениям установок:

$$\{(i, j) \mid i \in \{start(u)\} \cup \bar{O}(u); j \in \bar{O}(u) \cup \{finish(u)\}\}.$$

Если пакет за отведенное время нашел допустимое решение  $W$  и  $f(W) < f_{best}$ , то  $f_{best} = f(W)$  и  $W_{best} = W$ .

**Шаг 3.** Если для полученного решения  $W_{best}$  задачи (2.1)–(2.11) все  $\tau_i^0 = 0$ , то  $W_{best}$  является приближенным решением задачи  $\mathcal{P}$ .

Алгоритм легко обобщается на случай возможности возвратов буровых установок на объект заменой объектов на их копии и применением соответствующей модификации модели ЧЦЛП.

#### 4. Результаты численных экспериментов

Вычислительный эксперимент проводился на сервере с двумя процессорами AMD EPYC 7502 (каждый имеет 32 ядра, включен режим гиперпоточности). Выполнялось исследование модели ЧЦЛП при ее решении коммерческим пакетом Gurobi, а также анализ предложенного алгоритма РЛП на задачах трех серий: 1) задачи блочной структуры; 2) задачи серии  $S1$  с  $n = 50$  из [2]; 3) задачи серии  $S1'$  с  $n = 50$  из [2].

Задачи серии  $S1'$  отличаются от задач серии  $S1$  более узкими временными окнами и большим числом буровых установок. В экспериментах рассматривалось два варианта: допускается не более одного посещения буровой установкой объекта и не более двух посещений. Значения для параметров пакета threads=4, presolve=2, gomorypasses=0, method=0, minrelnodes=10627, improvestarttime=8640 взяты из статьи [2].

В табл. 1 представлены результаты решения задач блочной структуры пакетом Gurobi для моделей без возвратов и с возвратами (ограничение по времени счета — 10 ч). Используются обозначения:  $k$  — число блоков в задаче;  $f^*$ ,  $f_{best}$  — оптимальное и лучшее найденное пакетом значение целевой функции;  $LB$  — нижняя оценка целевой функции, полученная Gurobi;  $t_{solve}$ ,  $t_{best}$  — время решения задачи и время получения  $f_{best}$  в секундах. В задачах блочной структуры  $|I| = 6k$ ,  $|U| = 3k$ .

Отметим, что во всех задачах (кроме примера с 9 блоками в модели с возвратами) пакетом найдено оптимальное решение (т. е.  $f^* = f_{best}$ ), но доказана оптимальность полученного решения только для случаев не более двух блоков при модели с возвратами и до трех блоков при модели без возвратов. Задачи с возвратами оказались более сложными для пакета Gurobi в том числе из-за роста размерности. В среднем относительное отклонение  $\frac{f_{best}-LB}{f_{best}}$  в задачах с равным числом блоков более чем в два раза больше для модели с возвратами по сравнению с моделью без возвратов. Время нахождения лучшего решения также существенно возрастает при возможности возвратов.

Таблица 1

Результаты пакета Gurobi для задач блочной структуры

$k$	Модель без возвратов				Модель с возвратами				
	$f^*$	$[LB]$	$t_{solve}$	$t_{best}$	$f^*$	$f_{best}$	$[LB]$	$t_{solve}$	$t_{best}$
1	27	27	<1	<1	21	21	21	<1	<1
2	54	54	16	1	42	42	42	776	81
3	81	81	916	24	63	63	53	36000	216
4	108	100	36000	30	84	84	49	36000	165
5	135	111	36000	101	105	105	56	36000	3557
6	162	130	36000	61	126	126	58	36000	11366
7	189	142	36000	940	147	147	67	36000	2324
8	216	151	36000	468	168	168	68	36000	9573
9	243	172	36000	636	189	210	76	36000	10527
10	270	183	36000	536	210	210	80	36000	10639

В табл. 2, 3 приведены результаты расчетов для серий  $S1$  и  $S1'$  в задачах без возвратов, полученные Gurobi с ограничением по времени счета в 3 ч и алгоритмом РЛП с ограничением по времени счета в 1 ч (на каждом примере запускался 30 раз). Для алгоритма РЛП используются обозначения:  $f_a$  — среднее значение целевой функции;  $f_b$  — лучшее найденное значение целевой функции;  $fr$  — частота получения  $f_b$ ;  $f_w$  — худшее полученное значение целевой функции;  $\Delta f_a^b, \Delta f_w^b$  — отклонение лучшего найденного решения от среднего и от худшего в процентах. Результаты Gurobi для задач с возвратами представлены в табл. 4.

Только задача  $S1.9$  (отмечена звездочкой) без возвратов решена пакетом Gurobi за 890 с. Как и в случае задач блочной структуры, примеры с возвратами являются более сложными для Gurobi, однако есть примеры, когда по модели с возвратами получено решение с лучшим значением целевой функции, чем по модели без возвратов (задачи  $S1.3$  и  $S1.5$ ). В задачах  $S1.9$  и  $S1.10$  допустимое решение не найдено.

Алгоритм РЛП запускался с параметрами (вариант 1): начальное решение строится вызовом агоритма  $IS1$  10 раз; максимальное время

Таблица 2

Результаты для серии  $S1$  с 50 объектами

№	Gurobi, 10800 с			Алгоритм РЛП					
	$f_{best}$	$LB$	$t_{best}$	$f_a$	$f_b$	$fr$	$f_w$	$\Delta f_a^b$	$\Delta f_w^b$
S1.1	63	60,9	7824	63,03	63	29	64	0,05	1,59
S1.2	74	71,3	907	74,90	74	12	77	1,20	4,05
S1.3	80	71,8	446	79,53	79	15	81	0,67	2,53
S1.4	62	53,1	5894	61,83	61	13	63	1,34	3,28
S1.5	83	75,1	1490	82,30	82	21	83	0,36	1,22
S1.6	64	59,9	259	64,03	64	29	65	0,05	1,56
S1.7	67	60,6	173	66,33	66	23	68	0,50	3,03
S1.8	65	56,8	224	65,87	65	9	67	1,32	3,08
S1.9*	67	67,0	654	67,47	67	17	69	0,70	2,99
S1.10	73	63,6	5395	72,50	72	15	73	0,69	1,39
ср. зн.	69,80			69,78	69,30	18,3	71	0,69	2,47

timeMIP на решение подзадачи равно 30 с (шаг 2.2.2); используется окрестность (3, 40); иницируется перезапуск, если за 100 итераций рекорд не улучшается.

Таблица 3

Результаты для серии  $S1'$  с 50 объектами

№	Gurobi, 10800 sec.			Алгоритм РЛП					
	$f_{best}$	$LB$	$t_{best}$	$f_a$	$f_b$	$fr$	$f_w$	$\Delta f_a^b$	$\Delta f_w^b$
$S1'.1$	123	102,1	548	124,60	122	1	128	2,09	4,92
$S1'.2$	133	119,4	1054	135,43	134	4	138	1,06	2,99
$S1'.3$	147	127,9	3734	150,27	147	2	154	2,18	4,76
$S1'.4$	117	92,9	1834	117,83	115	2	125	2,40	8,70
$S1'.5$	153	130,3	149	155,73	154	5	159	1,11	3,25
$S1'.6$	137	119,5	438	142,20	139	1	145	2,25	4,32
$S1'.7$	146	131,4	303	148,20	146	2	150	1,48	2,74
$S1'.8$	152	128,3	2106	154,17	152	1	156	1,41	2,63
$S1'.9$	148	119,3	8700	150,97	149	7	156	1,30	4,70
$S1'.10$	121	102,9	3401	122,73	121	3	126	1,41	4,13
ср. зн.	137,7			140,21	137,9	2,8	143,7	1,67	4,31

Для серии  $S1$  частота получения рекордного значения алгоритмом РЛП является высокой, причем для 5 из 10 примеров удалось получить решения с меньшим значением целевой функции, чем найдено непосредственно Gurobi. Среднее значение целевой функции также меньше, чем у пакета, а отклонение между худшим и лучшим значениями около

Таблица 4

Результаты пакета Gurobi для модели с возвратами

№ задачи в серии	Серия S1				Серия S1'			
	$f$	$f_{best}$	$\lceil LB \rceil$	$t_{best}$	$f$	$f_{best}$	$\lceil LB \rceil$	$t_{best}$
1	42,83	63	50	7004	76,68	123	88	4782
2	49,58	74	64	393	86,56	133	122	1015
3	50,58	78	63	2827	94,39	147	120	1881
4	37,41	63	44	2000	66,12	118	88	2116
5	52,56	82	63	8628	97,52	153	124	1036
6	45,33	66	54	634	92,53	140	117	1141
7	45,13	68	54	9962	86,56	147	116	5382
8	43,38	69	50	7756	84,67	151	127	2369
9	45,85	–	53	–	91,79	148	120	9593
10	38,26	–	46	–	79,69	121	100	9182

Таблица 5

Сравнение параметров для алгоритма РЛП

Номер варианта	$f_a$	$f_b$	$fr$	$f_w$	$\Delta f_a^b$	$\Delta f_w^b$
1	69,78	69,3	18,3	71,0	0,69	2,47
2	70,98	69,6	7,4	73,1	1,93	5,03
3	69,83	69,3	17,7	71,0	0,76	2,44
4	69,80	69,3	18,3	71,0	0,71	2,46

4,3% в среднем. Для серии S1' результаты алгоритма РЛП несколько хуже, чем для S1. Это обусловлено тем, что решения данных примеров содержат много коротких маршрутов и перестройка их малой части не приводит к существенным улучшениям.

Алгоритм РЛП запускался с другими параметрами: вариант 2 — окрестность (2, 40) при timeMIP=30; вариант 3 — вероятность 0,1, использовать окрестность (2, 4–6; 20) при timeMIP=30, иначе — окрестность (3, 20) при timeMIP=15; вариант 4 — окрестность (3, 40) при timeMIP=30. Во всех вариантах выполняется перезапуск. Алгоритмы IS1 и IS2 запускались по 10 раз. Из результатов, представленных через средние значения в табл. 5, видим, что вариант 1 демонстрирует лучшие показатели с точки зрения различных характеристик.

Генетический алгоритм из [6] и гибридный алгоритм из [2] показывают на задачах серий S1 и S1' результаты, близкие к результатам Gurobi и, соответственно, близкие к результатам РЛП. Отметим, что на 2–3 задачах алгоритм РЛП или один из [6], [2] демонстрирует небольшое, но преимущество. Подчеркнем, что основной целью проведенного эксперимента было показать перспективность и целесообразность подхода

к решению рассматриваемой задачи с помощью предложенной модели ЧЦЛП посредством решения серии подзадач пониженной размерности.

## 5. Заключение

Проведено исследование задачи маршрутизации с возвратами и временными окнами. Предложен алгоритм локального поиска с «большими» окрестностями, основанный на решении серии задач пониженной размерности методами ветвей и отсечений. Результаты эксперимента показали перспективность используемого подхода. Дальнейшее исследование может быть направлено на использование методов с чередующимися окрестностями и жадных стратегий для решения подзадач, а также экспериментальный анализ на примерах большей размерности.

## Список источников

1. Еремеев А. В., Заозерская Л. А., Захарова Ю. В. Исследование моделей целочисленного программирования для задачи маршрутизации буровых установок с возвратами и временными окнами // PREPRINTS.RU. 2022. <https://doi.org/10.24108/preprints-3112582>
2. Кулаченко И. Н., Кононова П. А. Гибридный алгоритм решения задачи маршрутизации буровых установок // Дискретный анализ и исследование операций. 2021. Т. 28, вып. 2. С. 35–59. <https://doi.org/10.33048/daio.2021.28.703>
3. Bezerra S. N., Souza M. J. F., de Souza S. R. A variable neighborhood search-based algorithm with adaptive local search for the Vehicle Routing Problem with Time Windows and multi-depots aiming for vehicle fleet reduction // Comp. & Oper. Res. 2023. Vol. 149. 106016. <https://doi.org/10.1016/j.cor.2022.106016>
4. Blum C., Eremeev A. V., Zakharova Y. V. Hybridizations of evolutionary algorithms with Large Neighborhood Search // Comp. Sci. Rev. 2022. Vol. 46. 100512. <https://doi.org/10.1016/j.cosrev.2022.100512>
5. Rig routing with possible returns and stochastic drilling times / P. Borisovsky, A. Eremeev, Y. Kovalenko, L. Zaozerskaya // Proc. of Mathematical Optimization Theory and Operations Research. MOTOR 2021. LNCS. 2021. Vol. 12755. P. 51–66. [https://doi.org/10.1007/978-3-030-77876-7\\_4](https://doi.org/10.1007/978-3-030-77876-7_4)
6. Borisovsky P. A parallel greedy approach enhanced by genetic algorithm for the stochastic rig routing problem // Opt. Let. OnLine-First. 2023. P. 235–255. <https://doi.org/10.1007/s11590-023-01986-x>
7. Groer C., Golden B., Wasil E. A library of local search heuristics for the vehicle routing problem // Math. Prog. Comp. 2010. Vol. 2. P. 79–101. <https://doi.org/10.1007/s12532-010-0013-5>
8. Modeling and solving a multi-trip multi-distribution center vehicle routing problem with lower-bound capacity constraints / V. S. Nguyen, Q. D. Pham, T. H. Nguyen, Q. T. Bui // Comp. and Ind. Eng. 2022. Vol. 172, Part A. 108597. <https://doi.org/10.1016/j.cie.2022.108597>
9. Sadati M. E. H., Akbari V., Çatay B. Electric vehicle routing problem with flexible deliveries // Int. Jour. of Prod. Res. 2022. Vol. 60, N 13. P. 4268–4294. <https://doi.org/10.1080/00207543.2022.2032451>

10. Solving vehicle routing problem by memetic search with evolutionary multitasking / Q. Shang, Y. Huang, Y. Wang [et al.] // *Memetic Comp.* 2022. Vol. 14. P. 31–44. <https://doi.org/10.1007/s12293-021-00352-7>
11. A simulated annealing algorithm for the vehicle routing problem with parcel lockers / V. F. Yu, H. Susanto, P. Jodiawan, T.-W. Ho, S.-W. Lin, Y.-T. Huang // *IEEE Access.* 2022. Vol. 10. P. 20764–20782. <https://doi.org/10.1109/ACCESS.2022.3152062>

## References

1. Eremeev A.V., Zaozerskaya L.A., Zakharova Yu.V. Investigation of integer programming models for rig routing problem with returns and time windows. *PREPRINTS.RU*, 2022, <https://doi.org/10.24108/preprints-3112582>. (in Russian)
2. Kulachenko I.N., Kononova P.A. A hybrid algorithm for the drilling rig routing problem. *Journal of Applied and Industrial Mathematics*, 2021, vol. 15, iss. 2, pp. 261–276. <https://doi.org/10.33048/daio.2021.28.703>
3. Bezerra S.N., Souza M.J.F., de Souza S.R. A variable neighborhood search-based algorithm with adaptive local search for the Vehicle Routing Problem with Time Windows and multi-depots aiming for vehicle fleet reduction. *Comp. & Oper. Res.*, 2023, vol. 149. 106016. <https://doi.org/10.1016/j.cor.2022.106016>
4. Blum C., Eremeev A.V., Zakharova Y.V. Hybridizations of evolutionary algorithms with Large Neighborhood Search. *Comp. Sci. Rev.*, 2022, vol. 46. 100512. <https://doi.org/10.1016/j.cosrev.2022.100512>
5. Borisovsky P., Eremeev A., Kovalenko Y., Zaozerskaya L. Rig routing with possible returns and stochastic drilling times. *Proc. of Mathematical Optimization Theory and Operations Research. MOTOR 2021. LNCS*, 2021, vol. 12755, pp. 51–66. [https://doi.org/10.1007/978-3-030-77876-7\\_4](https://doi.org/10.1007/978-3-030-77876-7_4)
6. Borisovsky P. A parallel greedy approach enhanced by genetic algorithm for the stochastic rig routing problem. *Opt. Let. OnLine-First*, 2023, 15 p. <https://doi.org/10.1007/s11590-023-01986-x>
7. Groer C., Golden B., Wasil E. A library of local search heuristics for the vehicle routing problem. *Math. Prog. Comp.*, 2010, vol. 2, pp. 79–101. <https://doi.org/10.1007/s12532-010-0013-5>
8. Nguyen V.S., Pham Q.D., Nguyen T.H., Bui Q.T. Modeling and solving a multi-trip multi-distribution center vehicle routing problem with lower-bound capacity constraints. *Comp. and Ind. Eng.*, 2022, vol. 172, part A. 108597. <https://doi.org/10.1016/j.cie.2022.108597>
9. Sadati M.E.H., Akbari V., Çatay B. Electric vehicle routing problem with flexible deliveries. *Int. Jour. of Prod. Res.*, 2022, vol. 60, no. 13, pp. 4268–4294. <https://doi.org/10.1080/00207543.2022.2032451>
10. Shang Q., Huang Y., Wang Y. et al. Solving vehicle routing problem by memetic search with evolutionary multitasking. *Memetic Comp.*, 2022, vol. 14, pp. 31–44. <https://doi.org/10.1007/s12293-021-00352-7>
11. Yu V.F., Susanto H., Jodiawan P., Ho T.-W., Lin S.-W., Huang Y.-T. A simulated annealing algorithm for the vehicle routing problem with parcel lockers. *IEEE Access*, 2022, vol. 10, pp. 20764–20782. <https://doi.org/10.1109/ACCESS.2022.3152062>

**Об авторах**

**Заозерская Лидия Анатольевна**,  
канд. физ.-мат. наук, доц., Институт  
математики им. С. Л. Соболева СО  
РАН, Омск, 644090, Российская  
Федерация, zaozer@ofim.oscsbras.ru,  
<https://orcid.org/0000-0002-0287-3759>

**Захарова Юлия Викторовна**,  
канд. физ.-мат. наук, Институт  
математики им. С. Л. Соболева СО  
РАН, Омск, 644090, Российская  
Федерация,  
yzakharova@ofim.oscsbras.ru,  
<https://orcid.org/0000-0003-4791-7011>

**About the authors**

**Lidia A. Zaozerskaya**, Cand. Sci.  
(Phys.Math.), Assoc. Prof., Sobolev  
Institute of Mathematics SB RAS,  
Omsk, 644090, Russian Federation,  
zaozer@ofim.oscsbras.ru,  
<https://orcid.org/0000-0002-0287-3759>

**Yulia V. Zakharova**, Cand. Sci.  
(Phys.Math.), Sobolev Institute of  
Mathematics SB RAS, Omsk, 644090,  
Russian Federation,  
yzakharova@ofim.oscsbras.ru,  
<https://orcid.org/0000-0003-4791-7011>

*Поступила в редакцию / Received 26.10.2023*

*Поступила после рецензирования / Revised 21.12.2023*

*Принята к публикации / Accepted 05.02.2024*