

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 33 № 1 2026

Новосибирск
Издательство Института математики

МАТЭВРИСТИКА НА ОСНОВЕ ЛОКАЛЬНОГО ПОИСКА
ДЛЯ ОДНОЙ МОДИФИКАЦИИ ОБОБЩЁННОЙ
ЗАДАЧИ О НАЗНАЧЕНИЯХ

Ю. В. Захарова^{1, a}, Л. А. Заозёрская^{1, 2, b}

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

² Омский гос. технический университет,
пр. Мира, 11, 644050 Омск, Россия

E-mail: ^ayzakharova@ofim.oscsbras.ru, ^bzaozer@ofim.oscsbras.ru

Аннотация. Рассматривается новая модификация обобщённой задачи о назначениях с ограничениями на объёмы и типы работ. Заданное множество работ необходимо распределить между агентами (исполнителями). Каждая работа имеет объём и тип. Для каждого агента заданы ограничения на количество типов назначенных работ, общий объём назначенных работ, а также эффективность выполнения работ. Требуется максимизировать суммарную эффективность назначения при заданных ограничениях на загрузку. Исследуется вычислительная сложность задачи. Предлагается алгоритм локального поиска с окрестностями произвольного размера, где учитываются известные подходы к решению обобщённой задачи о назначениях и дополнительные условия рассматриваемой задачи. Окрестности строятся путём частичного фиксирования значений переменных, соответствующих назначению работ. Для решения соответствующих подзадач используются точные алгоритмы из известных пакетов решения задач математического программирования. Экспериментальные результаты показывают, что предложенный алгоритм демонстрирует конкурентоспособные результаты по качеству получаемых решений в сравнении с гибридными алгоритмами на основе методов ветвей и отсечений из известных пакетов Gurobi и SCIP при заданном ограничении времени счёта на серии тестовых примеров с различными структурными свойствами. Табл. 7, библиогр. 29.

Ключевые слова: назначение, цветовой класс, сложность, матэвристика.

Введение

Обобщённая задача о назначениях является известной NP-трудной задачей дискретной оптимизации. В этой задаче каждый агент (станок, рабочий, процессор) может быть назначен на несколько работ (задач, заказов) в отличие от классической задачи о назначениях, где назначение формируется один к одному. Каждая работа выполняется ровно одним агентом. Назначения ограничены ресурсом агентов (время, бюджет, мощность). Требуется минимизировать суммарную стоимость (или максимизировать прибыль) назначений. Задача имеет широкий спектр приложений [1–3]: планирование производственных расписаний, распределение процессов в компьютерных сетях, размещение объектов, управление грузоперевозками, формирование портфелей проектов, распределение заявок по экспертам, распределение учебной загрузки.

Известны различные модификации обобщённой задачи о назначениях [1, 2]: задача о назначениях с несколькими ресурсами, многоуровневая обобщённая задача о назначениях, динамическая обобщённая задача о назначениях, обобщённая задача о назначениях с многократными назначениями и обобщённая задача о назначениях с «узким местом». Исследуются многокритериальные постановки с соответствующими подходами к решению [4]. Рассматривается также максимизационный вариант обобщённой задачи о назначениях, в котором ищется подмножество заданных работ [5]. Важным частным случаем такой постановки является задача о нескольких рюкзаках, в которой затраты ресурса и прибыль от выполнения любой работы не зависят от выбора агента [6].

Известны задача о нескольких рюкзаках и задача об упаковке в контейнеры с цветовыми классами [6, 7]. В них работы имеют «цветовые» характеристики, которые необходимо учитывать при распределении.

В настоящей работе рассматривается модификация обобщённой задачи о назначениях с дополнительными ограничениями на загрузку агентов и цветовыми условиями (классами). Каждая работа дополнительно имеет тип (или цвет). Для каждого агента задано максимально допустимое количество типов назначенных работ. Искомое распределение агентов по работам также должно обеспечивать заданную минимальную загрузку агентов. Такая постановка задачи возникает при решении двукритериальных задач распределения учебной загрузки [3] или управления поставками с дискретными размерами партий [8]. Предыдущие исследования данной задачи, посвящённые анализу моделей целочисленного программирования и структуры множества допустимых решений, представлены на конференциях [3, 8].

В данной статье для обобщённой задачи о назначениях с цветовыми классами предлагается математическая эвристика (матэвристика).

К матэвристикам относятся эвристические методы, основанные на решении серии задач математического программирования (МП) пониженной размерности и их упрощённых версий [9, 10]. В большинстве случаев такие алгоритмы начинают свою работу с допустимого или частично-го решения, найденного для упрощённой версии задачи, и итеративно пытаются улучшить результат за счёт локальных изменений. Приведём некоторые типичные варианты матэвристик, предложенных для решения задачи МП в общем виде [9].

Основная идея алгоритма из [11] заключается в поиске хороших допустимых решений, расположенных вблизи ЛП-релаксации, т. е. оптимального решения соответствующей задачи линейного программирования (ЛП). В [12] алгоритмы локального поиска с большими окрестностями начинают работу с решения упрощённой версии задачи как линейного, так и нелинейного программирования с различными ограничениями, а затем переходят к поиску улучшающих шагов на основе решения задач пониженной размерности. В [13] предложен гибридный эвристический алгоритм, сочетающий процедуры поиска допустимых целочисленных направлений и методы линейной оптимизации.

В [14] разработан алгоритм локального поиска для решения задач целочисленного линейного программирования (ЦЛП) на основе окрестностей двух типов: окрестности, использующей информацию из непрерывной релаксации ЦЛП-модели, и управляемые спуски/подъёмы. В работе [15] предполагается использование универсального пакета для решения задач ЦЛП в качестве чёрного ящика для исследования подпространств решений, которые определяются и контролируются с помощью простой внешней схемы ветвления.

Для NP-трудных модификаций задачи о назначениях известен ряд подходов к приближённому и точному решению на основе ЦЛП и метаэвристик. Среди метаэвристик можно выделить генетические алгоритмы, алгоритмы поиска с запретами, имитацию отжига [1, 16].

Для обобщённой задачи о назначениях известны эвристические алгоритмы на основе различных типов пошаговой релаксации [17], декомпозиционные эвристики на базе лагранжевой релаксации [18]. Предложены эвристика локального поиска с переменной глубиной окрестности [19], алгоритм поиска с запретами [20] и гибридный эволюционный алгоритм с проблемно-ориентированными эвристиками и новыми операторами [21].

В [22] предложенная для классической обобщённой задачи о назначениях ЛП-ориентированная эвристика [23] с отсеечениями и локальными улучшениями адаптируется к решению обобщённой задачи о назначениях со специальными упорядоченными множествами.

Для задачи об упаковке в контейнеры с цветовыми ограничениями предложены эвристика на основе метода генерации столбцов, гибридный алгоритм локального поиска с чередующимися окрестностями и точный метод на основе ветвлений [7]. Для обобщённой задачи о назначениях [24] разработана эвристика на основе метода генерации столбцов и алгоритма построения верхних оценок на целевую функцию, как и для задачи о рюкзаке с цветовыми классами [25].

В разд. 1 данной статьи приводится постановка задачи, а в разд. 2 — исследуется её сложность. Матэвристика описывается и анализируется в разд. 3. Результаты эксперимента представлены в разд. 4.

1. Постановка задачи

Рассмотрим следующую модификацию задачи о назначениях, которая имеет приложения в образовании при распределении загрузки между преподавателями и в производстве при распределении задач между сотрудниками подразделений.

Множество работ $J = \{j_1, \dots, j_n\}$ необходимо распределить между агентами (исполнителями) из множества $I = \{i_1, \dots, i_m\}$. Эффективность назначения работы j агенту i равна c_{ij} , $i \in I$, $j \in J$. Объём работы $j \in J$ равен a_j , а суммарный объём загрузки для агента $i \in I$ должен лежать на отрезке $[d_i, b_i]$.

Кроме того, работы делятся на типы, множество которых обозначается через $K = \{k_1, \dots, k_l\}$. Максимальное число типов работ, которые могут быть назначены агенту i , равно T_i , $i \in I$, при этом подмножество $J_k \subset J$ задаёт множество работ типа $k \in K$. Требуется составить такое допустимое назначение работ агентам, при котором максимизируется суммарная эффективность.

Построим для рассматриваемой задачи модель булева программирования (БП). Введём булевы переменные

$$x_{ij} = \begin{cases} 1, & \text{если работа } j \text{ назначена агенту } i, \\ 0 & \text{иначе,} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{если агент } i \text{ выполняет работы типа } k, \\ 0 & \text{иначе,} \end{cases}$$

где $i \in I$, $j \in J$, $k \in K$. Пусть $X = (x_{ij})$ — матрица размера $m \times n$. С помощью указанных переменных модель записывается следующим образом:

$$f(X) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \rightarrow \max, \quad (1)$$

$$d_i \leq \sum_{j \in J} a_j x_{ij} \leq b_i, \quad i \in I, \quad (2)$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J, \quad (3)$$

$$\sum_{j \in J_k} x_{ij} \leq |J_k| y_{ik}, \quad i \in I, k \in K, \quad (4)$$

$$\sum_{k \in K} y_{ik} \leq T_i, \quad i \in I, \quad (5)$$

$$x_{ij}, y_{ik} \in \{0, 1\}, \quad i \in I, j \in J, k \in K. \quad (6)$$

Ограничения (2) задают верхнюю и нижнюю границы на объём загрузки каждого агента. Условие (3) показывает, что каждая работа будет назначена ровно одному агенту. Ограничения (4), (5) устанавливают верхнюю границу на число типов работ для каждого агента.

Задача (1)–(3) с условием булевости переменных x_{ij} из (6) есть вариант известной обобщённой задачи о назначениях [1]. Заметим, что в обобщённой задаче о назначениях b_i часто интерпретируют как объём ресурса у агента i , а a_j как объём ресурса, необходимого для выполнения агентом i работы j , и в общем случае для каждой работы он может иметь неодинаковые значения для разных агентов. Таким образом, задачу (1)–(6) можно рассматривать как обобщённую задачу о назначениях с цветовыми классами [6, 7] и нижними границами на загрузку агентов.

Для обобщённой задачи о назначениях доказано [23], что в ЛП-релаксации число работ, разделённых между двумя и более агентами, не превышает m . Ранее в [26] этот результат был получен для варианта обобщённой задачи о назначениях, в котором расход ресурсов для выполнения работ не зависит от выбора агента, т. е. задачи (1)–(3), (6) при $d_i = 0, i \in I$. Для доказательства используется переход к эквивалентной транспортной задаче. Отсюда легко получается, что число дробных компонент ЛП-релаксации не превышает $2m$, что свидетельствует о применимости методов, основанных на поиске в окрестности оптимального решения линейной релаксации, особенно при $n \gg m$. Результаты эксперимента показали, что для задачи (1)–(6) указанные свойства не нарушаются в рассматриваемых сериях примеров (см. разд. 4).

2. Вычислительная сложность

Теорема 1. Поиск допустимого решения в задаче (1)–(3), (6) является NP-трудной в обычном смысле задачей даже в случае двух агентов.

Доказательство. Чтобы показать справедливость утверждения теоремы, построим полиномиальное сведение задачи РАЗБИЕНИЕ [27] к задаче поиска допустимого решения, указанной в условии.

Задача РАЗБИЕНИЕ. Даны конечное множество $S = \{s_1, s_2, \dots, s_{n_0}\}$ и размер $e(s) \in \mathbb{Z}_+$ каждого элемента $s \in S$. Существует ли такое подмножество $S' \subset S$, что $\sum_{s \in S'} e(s) = \sum_{s \in S \setminus S'} e(s)$?

Имеем конечное множество S , $|S| = n_0$, и веса $e(s)$ задачи РАЗБИЕНИЕ. Построим соответствующий пример задачи (2), (3), (6). Пусть имеются два агента и n_0 работ, которые соответствуют элементам множества S задачи РАЗБИЕНИЕ. Объём a_j работы j равен $e(s_j)$, $j = 1, \dots, n_0$. Ограничения по объёму для агента $i = 1$ равны $d_i = b_i = \frac{\sum_{s \in S} e(s)}{2}$, $i = 1, 2$.

В таком случае множество S имеет разбиение на два подмножества S' и $S \setminus S'$ в задаче РАЗБИЕНИЕ тогда и только тогда, когда каждому агенту можно назначить подмножество работ объёма $\frac{\sum_{s \in S} e(s)}{2}$ в задаче (2), (3), (6). Сводимость строится за полиномиальное время. Теорема 1 доказана.

Теорема 2. Поиск допустимого решения в задаче (1)–(3), (6) является NP-трудной в сильном смысле задачей в случае произвольного числа агентов.

ДОКАЗАТЕЛЬСТВО. Достаточно построить полиномиальное сведение задачи 3-РАЗБИЕНИЕ [27] к задаче поиска допустимого решения, указанной в условии теоремы.

Задача 3-РАЗБИЕНИЕ. Даны множество $S = \{s_1, s_2, \dots, s_{3n_0}\}$, содержащее $3n_0$ элементов, граница $E \in \mathbb{Z}_+$ и размер $e(s) \in \mathbb{Z}_+$ каждого элемента $s \in S$ такие, что $\frac{E}{4} < e(s) < \frac{E}{2}$ и $\sum_{s \in S} e(s) = n_0 E$. Можно ли множество S разбить на n_0 попарно не пересекающихся трёхэлементных подмножеств $S_1, S_2, \dots, S_{n_0}$ так, что $\sum_{s \in S_i} e(s) = E$ для любого $i = 1, \dots, n_0$?

Построим соответствующий пример задачи (2), (3), (6) следующим образом. Элементы множества S задачи 3-РАЗБИЕНИЕ соответствуют работам (2), (3), (6). Число агентов равно n_0 , для каждого из которых ограничения по объёму равны $d_i = b_i = E$, $i = 1, \dots, n_0$. Веса $e(s_j)$ задачи 3-РАЗБИЕНИЕ соответствуют объёмам работ a_j , $j = 1, 2, \dots, n_0$.

В таком случае множество S может быть разбито на n_0 подмножеств $S_1, S_2, \dots, S_{n_0}$ в задаче 3-РАЗБИЕНИЕ тогда и только тогда, когда каждому агенту можно назначить подмножество работ объёма E в задаче (2), (3), (6). Сводимость строится за полиномиальное время. Теорема 2 доказана.

Теорема 3. Задача (1), (3)–(5), (6) полиномиально разрешима.

ДОКАЗАТЕЛЬСТВО. В указанном варианте задачи учитываются только ограничения на число типов без условий на объём. Зададим новое

множество агентов, где для каждого исходного агента i создадим T_i его копий с одинаковой эффективностью c_{ij} , $j \in J$. Если в итоге число агентов будет меньше числа работ, то добавим фиктивных агентов с нулевой эффективностью назначения. Аналогично поступим, если число агентов будет больше числа работ. В результате получим классическую задачу о назначениях, которая полиномиально разрешима [28]. Теорема 3 доказана.

Открытым остаётся вопрос об NP-трудности в сильном смысле рассматриваемой задачи в случае ограниченного константой числа агентов. Возможно, доказательство удастся построить на основе ограничений как по объёму загрузки, так и по числу типов работ.

3. Матэвристика на основе локального поиска

Для решения рассматриваемой задачи предлагается математическая эвристика, в которой на каждой итерации строится задача БП вида (1)–(6) меньшей размерности за счёт фиксирования части переменных. Полученные задачи решаются универсальными пакетами решения задач МП в течение заданного времени. Матэвристика состоит из двух этапов. На первом этапе начальное допустимое решение ищется случайным поиском вблизи допустимого (или оптимального) решения релаксационной задачи для исходной задачи БП. При получении допустимого решения на втором этапе улучшающее решение ищется вблизи допустимого решения с наибольшим значением целевой функции. К найденным допустимым решениям применяется процедура локального поиска.

Релаксацию задачи (1)–(6) определим, заменив условие булевости переменных x_{ij} условием

$$0 \leq x_{ij} \leq 1, \quad i \in I, j \in J. \quad (7)$$

Если условие булевости y_{ik} заменить условием

$$0 \leq y_{ik} \leq 1, \quad i \in I, k \in K,$$

то получим релаксационную задачу ЛП.

Для описания алгоритма введём следующие обозначения:

- h — номер итерации алгоритма;
- H — число итераций случайного поиска вблизи допустимого решения релаксации (1)–(6);
- X_{best} — допустимое решение задачи (1)–(6) с наибольшим значением целевой функции f ;
- f_{best} — значение целевой функции решения X_{best} ;
- X — вектор, определяющий текущую стартовую точку для поиска.

Алгоритм МН_{LS}**Этап 1. Построение начального решения.**

ШАГ 1. Решение релаксационной задачи.

ШАГ 1.1. Решить задачу (1)–(5), (7), используя пакет МП в течение времени t_0 .

ШАГ 1.2. Если допустимое решение $X_{\text{МП}}$ найдено, то $\tilde{X} := X_{\text{МП}}$, иначе $\tilde{X} := X_{\text{ЛР}}$, где $X_{\text{ЛР}}$ — оптимальное или допустимое решение ЛП-релаксации (1)–(6), полученное пакетом МП за время t_0 .

ШАГ 1.3. Положить $\mathcal{P} = \{(i, j) \mid \tilde{x}_{ij} = 1\}$, $X_{\text{best}} := \tilde{X}$, $f_{\text{best}} = 0$, $h = 1$.

ШАГ 2. Случайный поиск. Пока $h \leq H$ или $f_{\text{best}} = 0$ выполнять

ШАГ 2.1. С вероятностью p_{rand} выполнять

- выбрать работу j_{rand} случайным образом с равномерным распределением и назначить её случайно выбранному агенту i_{rand} ;
- обновить множество $\mathcal{P}$.

ШАГ 2.2. В задаче (1)–(6) положить $x_{ij} = 1$ с вероятностью p для пар индексов $(i, j) \in \mathcal{P}$ и решать полученную задачу в течение указанного времени t_1 , используя пакет МП.

ШАГ 2.3. Если получено допустимое решение X и $f(X) > f_{\text{best}}$, то обновить X_{best} и f_{best} .

ШАГ 2.4. Положить $h := h + 1$ и перейти на ШАГ 2.1.

ШАГ 3. Положить $\tilde{X} := X_{\text{best}}$ и обновить $\mathcal{P}$.

Этап 2. Локальное улучшение.

Выполнять, пока не сработает критерий остановки:

ШАГ 4. Построение и решение редуцированной задачи.

ШАГ 4.1. С вероятностью p_{rand} выполнять

- выбрать работу j_{rand} случайным образом с равномерным распределением и назначить её случайно выбранному агенту i_{rand} ;
- обновить множество $\mathcal{P}$.

ШАГ 4.2. В задаче (1)–(6) с дополнительным условием $f(X) \geq f_{\text{best}} + 1$ положить $x_{ij} = 1$ с вероятностью p для $(i, j) \in \mathcal{P}$ и решать полученную задачу пакетом МП в течение заданного времени t_1 .

ШАГ 4.3. Если допустимое решение получено, то

- обновить X_{best} и f_{best} ;
- к X_{best} применить локальный поиск с вероятностью p_{LS} ;
- положить $\tilde{X} := X_{\text{best}}$ и обновить множество $\mathcal{P}$.

ШАГ 4.4. Перейти на ШАГ 4.1.

Опишем процедуру локального поиска для решения X_{best} .

Пока значение целевой функции увеличивается, выполнять для всех пар агентов:

- для каждой пары агентов, применяя пакет МП, найти оптимальное перераспределение назначенных им работ;
- если переназначение улучшает текущее лучшее решение X_{best} , то обновить это решение.

Теорема 4. Алгоритм МН_{LS} сходится к оптимуму почти наверное при числе итераций, стремящемся к бесконечности.

Доказательство. Для сходимости почти наверное главным образом служит оператор шага 2.1 и шага 3.1 случайного поиска, который с ненулевой вероятностью назначает случайно выбранную работу случайно выбранному агенту на итерациях. Действительно, за счёт указанного оператора можно за n итераций с ненулевой вероятностью назначить работы агентам в соответствии с оптимальным решением. Назначения каждой итерации согласно множеству $\mathcal{P}$ сохраняются с вероятностью не менее $p^{|\mathcal{P}|}$ на этапах 2 и 3. Отметим, что локальный поиск не применяется на шаге 3.2 с вероятностью $1 - p_{\text{LS}}$. Далее сходимость почти наверное при числе итераций, стремящемся к бесконечности, следует из результатов, аналогичных теореме 2.1 в [29]. Теорема 4 доказана.

4. Вычислительный эксперимент

Предложенная матэвристика была протестирована на сериях примеров задачи (1)–(6) со случайно сгенерированными исходными данными и сериях примеров обобщённой задачи о назначениях из библиотеки OR-library.

Опишем правила генерации тестовых примеров. Напомним, что ограничение по объёму загрузки агентов имеет вид

$$d_i \leq \sum_{j \in J} a_j x_{ij} \leq b_i, \quad i \in I.$$

Обозначим через Δ отклонение суммарного объёма работ от суммы нижних границ загрузки агентов, т. е.

$$\Delta = \frac{\sum_{j \in J} a_j - \sum_{i \in I} d_i}{\sum_{i \in I} d_i} 100.$$

Построено три серии по десять примеров со случайными данными и следующими значениями диапазонов для отклонения Δ :

- серия L , где Δ от 2 до 5,2%,
- серия M , где Δ от 1 до 2%,

- серия H , где Δ до 1%.

Для всех задач число агентов равно $|I| = 22$, число типов работ — $|K| = 45$. Число работ каждого типа $|J_k|$ генерировалось с равномерным распределением из множества $\{1, \dots, 9\}$, в результате число работ $|J| \in \{208, \dots, 253\}$. Размерности задач БП после предобработки Gurobi по паре (число переменных $\times$ число ограничений) варьируются от 5434×1132 , 5469×1091 до 6314×1235 и существенно не отличаются по сериям (табл. 1).

Таблица 1

Размерность задач БП

Серия	Число переменных $\times$ число ограничений		
	средняя	минимальная	максимальная
L	5794×1156	5434×1132	6314×1235
M	5958×1191	5676×1143	6182×1271
H	5812×1148	5469×1091	5962×1198
	5812×1148	5469×1091	6028×1138

Значения остальных параметров генерируются также с равномерным распределением из следующих диапазонов: $b_i \in [300, 1400]$, $a_j \in [40, 120]$, $c_{ij} \in [1, 10]$. Значения d_i и T_i вычисляются по формулам

$$d_i = [0, 95b_i], \quad T_i = \left\lceil \frac{6b_i}{b_{\max}} + 0,5 \right\rceil, \quad b_{\max} = \max_{i \in I} b_i.$$

В серии включены только те задачи, для которых выполняется необходимое условие разрешимости

$$\sum_{i \in I} d_i \leq \sum_{j \in J} a_j \leq \sum_{i \in I} b_i.$$

Эксперимент проводился на компьютере с процессором AMD EPYC 7502 CPU. Использовались пакеты МП Gurobi, SCIP и среда разработки GAMS 23.2. Алгоритм МН_{LS} запускался на каждой задаче 30 раз с критерием остановки в виде ограничения на время счёта 1000 секунд, значениями параметров $p = 0,7$, $H = 10$, $p_{\text{rand}} = 0,000001$, $p_{LS} = 0,999999$.

Проводился предварительный вычислительный эксперимент по сравнению результатов для разного числа потоков (4, 8, 10, 12 потоков) при точном решении сгенерированных примеров пакетом Gurobi, а также при решении матэвристикой МН_{LS}. Результаты показали, что с ростом числа потоков результаты МН_{LS} несколько улучшались, но статистически значимых отличий в результатах эксперимента (использовался тест Вилкоксона) не установлено. При точном решении примеров из всех серий пакет Gurobi, т. е. метод на основе ветвлений и отсечений различного

типа, завершал работу через 7–19 часов ввиду нехватки памяти. В итоге число потоков было установлено равным 12, как при точном решении задач, так и с помощью матэвристики. Отметим, что узким местом тестовых примеров являются нижние границы и условия на число типов работ, без которых задачи решаются за приемлемое время без остановки из-за ограничений по памяти.

В таблицах с результатами экспериментов используются следующие обозначения:

- N_{sr} — число запусков, на которых получено допустимое решение матэвристикой,
- $f_a, f_{\min}, f_{\max}$ — среднее, минимальное, максимальное значения целевой функции f , найденные матэвристикой;
- f_G — лучшее значение целевой функции f , найденное Gurobi,
- f_{1000} — значение целевой функции f , найденное Gurobi за 1000 с,
- UB — верхняя граница на целевую функцию, найденная пакетом МП,
- $\delta_B^A = \frac{f_B - f_A}{f_B} \cdot 100$ — отклонение величины f_A от f_B (%).

На лёгкой серии L (табл. 2) относительное отклонение результата МН_{LS} от результата Gurobi f_G составило в среднем 1,18%, а от верхней границы UB — не более 2,4%. При равном времени счёта МН_{LS} показывает несколько лучшие результаты, чем Gurobi, в среднем превосходство составляет 0,1%. Число успешных запусков для всех задач равно 30.

На средней по сложности серии M (табл. 3) среднее относительное отклонение результата МН_{LS} от результата Gurobi схоже с серией L ,

Таблица 2

Сравнение результатов Gurobi и МН_{LS} на серии L

№	Gurobi			МН _{LS}			Отклонение, %		
	f_{1000}	f_G	UB	f_a	$f_{\min}$	$f_{\max}$	δ_{1000}^a	δ_G^a	δ_{UB}^a
1	1913	1961	1969	1938,7	1922	1947	−1,35	1,14	1,54
2	1914	1928	1937	1911,4	1894	1927	0,13	0,86	1,32
3	1913	1954	1961	1914,6	1897	1930	−0,08	2,02	2,37
4	1939	1947	1959	1915,4	1901	1933	1,22	1,62	2,22
5	1872	1890	1902	1856,3	1842	1871	0,84	1,78	2,40
6	2070	2113	2127	2092,1	2076	2103	−1,07	0,99	1,64
7	2025	2054	2077	2040,4	2025	2052	−0,76	0,66	1,76
8	2085	2106	2117	2094,4	2083	2101	−0,45	0,55	1,07
9	2076	2077	2093	2052,9	2043	2067	1,11	1,16	1,92
10	2185	2221	2249	2198,7	2184	2210	−0,63	1,00	2,24
ср.	1999,2	2025,1	2039,1	2001,5	1986,7	2014,1	−0,1	1,18	1,85

Таблица 3

Сравнение результатов Gurobi и MH_{LS} на серии M

№	Gurobi			MH_{LS}			Отклонение, %		
	f_{1000}	f_G	UB	f_a	$f_{\min}$	$f_{\max}$	δ_{1000}^a	δ_G^a	δ_{UB}^a
1	—	2018	2057	2006,5	1982	2024	—	0,57	2,45
2	—	2006	2043	1999,9	1990	2009	—	0,30	2,11
3	2246	2256	2279	2216,9	2154	2239	1,29	1,73	2,72
4	1924	1953	1980	1923,3	1903	1936	0,04	1,52	2,87
5	2163	2175	2187	2148,7	2134	2162	0,66	1,21	1,75
6	2099	2126	2131	2086,3	2043	2103	0,61	1,87	2,10
7	2036	2057	2074	2041,0	2031	2053	−0,25	0,77	1,59
8	1972	1991	2007	1977,4	1962	1989	−0,27	0,68	1,47
9	2088	2100	2113	2071,6	2062	2080	0,79	1,35	1,96
10	2181	2196	2207	2161,2	2142	2175	0,91	1,58	2,08
ср.	2088,6	2087,8	2107,8	2063,3	2040,3	2077	0,47	1,16	2,11

а отклонение от верхней границы — не более 2,9%. При равном времени счёта MH_{LS} уступает Gurobi в среднем на 0,47%. Число успешных запусков для задачи 3 равно 29, для остальных задач равно 30.

На сложной серии H (табл. 4) алгоритмом MH_{LS} найдено допустимое решение для каждой задачи в отличие от Gurobi (при ограничении времени счёта в 1000 с Gurobi найдено допустимое решение только для двух задач). Для трудного примера (номер 2) алгоритмом MH_{LS} допустимое решение найдено в 12 запусках из 30, для примеров 1 — в 14 запусках, в примерах 3 и 4 — в 19 запусках, для остальных примеров все запуски были успешными, а в среднем по всей серии — в 24,4 запусках

Таблица 4

Сравнение результатов Gurobi и MH_{LS} на серии H

№	Gurobi			MH_{LS}			Отклонение, %		
	f_{1000}	f_G	UB	f_a	$f_{\min}$	$f_{\max}$	δ_{1000}^a	δ_G^a	δ_{UB}^a
1	—	—	2089	1987,0	1972	2009	—	—	4,88
2	—	—	2110	1994,4	1967	2013	—	—	5,48
3	—	—	2030	1939,2	1891	1963	—	—	4,47
4	—	—	2021	1930,6	1892	1947	—	—	4,47
5	—	2084	2119	2050,2	2019	2074	—	1,62	3,25
6	—	2006	2060	2012,7	1993	2025	—	−0,33	2,30
7	1960	2026	2069	1984,4	1947	2011	−1,25	2,05	4,09
8	2038	2054	2083	2027,6	2006	2040	0,51	1,29	2,66
9	—	1869	1916	1849,2	1812	1867	—	1,06	3,49
10	—	2137	2166	2104,1	2082	2126	—	1,54	2,86
ср.	1999	2029,3	2066,3	1987,9	1958,1	2007,5	−0,37	1,20	3,79

из 30. Среднее относительное отклонение результата MH_{LS} от результата $Guobi$ составило в среднем 1,2%, а отклонение от верхней границы — не более 5,5%.

Таблица 5

Сравнение алгоритмов MH_{RS} и MH_{LS}

Серия	Среднее значение f		Отклонение, %
	MH_{RS}	MH_{LS}	
L	2003,21	2001,50	0,086
M	2066,63	2063,29	0,161
H	1978,78	1987,94	-0,463

Алгоритм MH_{LS} также сравнивался с версией алгоритма, обозначаемой далее через MH_{RS} , где вторая стадия не задействуется (см. табл. 5, где представлены средние значения рекордов по сериям и средние относительные отклонения рекордов MH_{RS} от рекордов MH_{LS}). На сериях L и M алгоритмы показали похожие результаты (результаты теста Вилкоксона показали наличие статистически значимых отличий при уровне значимости $\alpha = 0,05$ на четырёх примерах серии L и пяти примерах серии M), а на серии H несколько лучшие результаты показал MH_{LS} (статистически значимые отличия выявлены на двух примерах), что свидетельствует о целесообразности применения рандомизированного поиска и перспективности использования локального поиска при малых отличиях между суммарным объёмом работ и суммы нижних границ на загрузку агентов.

Таблица 6

Оценка разрыва двойственности в среднем по сериям

Серия	f_{LP}	record	$gap_{rec}, \%$	$gap_a, \%$	$gap_{min}, \%$	$gap_{max}, \%$
L	2216,41	2025,10	8,63	9,71	10,38	9,14
M	2280,82	2088,70	8,42	9,53	10,52	8,93
H	2228,74	2014,40	9,62	10,80	12,15	9,92

В табл. 6 представлена оценка разрыва двойственности по сериям. Здесь f_{LP} — оптимальное решение линейной релаксации в среднем по серии, record — наибольшее значение f , среди найденных $Guobi$ или алгоритмами MH_{LS} и MH_{RS} , gap_{rec} , gap_a , gap_{min} , gap_{max} — отклонения лучшего известного, среднего, минимального, максимального решений соответственно, найденных алгоритмом MH_{LS} , от оптимального решения линейной релаксации.

Величина gar_x , $x \in \{a, \min, \max\}$, показывает отклонение финальных решений, найденных алгоритмами, от решений линейной релаксации. Полученные значения для gar_x свидетельствуют о сложности рассматриваемых тестовых примеров ввиду удалённости по целевой функции найденных решений от оптимума линейной релаксации. Это показывает потенциал для алгоритмов на рассматриваемых сериях тестовых примеров различной структуры.

Был проведён вычислительный эксперимент, где для решения подзадач частично целочисленного линейного программирования в MH_{LS} используется свободно распространяемый пакет SCIP вместо коммерческого пакета Gurobi . Результаты представлены в табл. 7. Здесь f_{best} — значение f , найденное пакетом МП. Как можно видеть, при использовании SCIP результаты алгоритма статистически значительно ухудшаются, но сохраняются на хорошем уровне (в среднем отклонение от f_{best} составляет не более 2,3% при использовании SCIP и не более 1,2% при использовании Gurobi). Результаты при использовании SCIP в матэвристике согласуются с тем, что бесплатные ресурсы, как правило, демонстрируют конкурентоспособные результаты при решении тестовых задач малой размерности, которые как раз и возникают в предлагаемом алгоритме.

Отметим, что итерации алгоритма MH_{LS} основаны на фиксации части значений x -переменных. Была также исследована версия алгоритма, где осуществлялась фиксация y -переменных, но она показала статистически значимо худшие результаты.

Таблица 7

Сравнение версий MH_{LS} при использовании пакетов SCIP и Gurobi

Пакет МП	N_{sr}	Отклонение, %				
		δ_{1000}^a	δ_{best}^a	$\delta_{\text{best}}^{\min}$	$\delta_{\text{best}}^{\max}$	δ_{UB}^a
Серия L						
Gurobi	30	−0,10	1,18	1,91	0,55	1,85
SCIP	39	0,68	1,95	3,10	1,19	2,62
Серия M						
Gurobi	29,90	0,47	1,16	2,24	0,51	2,11
SCIP	29,78	0,79	1,69	3,15	0,80	2,64
Серия H						
Gurobi	24,40	−0,37	1,20	2,60	0,26	3,79
SCIP	19,40	2,23	2,28	3,85	1,07	4,21

Разработанный метод универсален для модификаций задачи о назначениях, где ключевую роль играет назначение работ агентам с различными дополнительными ограничениями. Для его применения достаточно использовать соответствующую модель задачи ЦЛП. Алгоритм MN_{LS} был применён для обобщённой задачи о назначениях. Для тестирования использовались примеры библиотеки OR-library из серий C , D , E размерности 20×400 , 30×900 , 40×400 , 40×1600 , 60×900 , 80×1600 . Разрыв двойственности для этих примеров в среднем не превосходит 0,1%. Среднее отклонение f_a от f_G равно 0,082%, а от f_{LP} — 0,178%.

Заключение

В статье представлен анализ вычислительной сложности варианта обобщённой задачи о назначениях с цветовыми классами. Предложена и экспериментально исследована математическая эвристика, учитывающая особенности примеров типа задач о назначениях с дополнительными ограничениями. Используемые правила настройки эвристики привели к конкурентоспособным результатам на задачах различной структуры. Важной опцией алгоритма является возможность использования некоммерческих пакетов решения задач математического программирования ввиду решения подзадач настраиваемой размерности на итерациях.

Для дальнейшей работы представляет интерес адаптивная настройка параметров и исследование популяционных методов поиска. Перспективным направлением также выступает использование параллельных вычислений и масштабируемость решений.

Авторы выражают благодарность Омскому филиалу Института математики им. С. Л. Соболева СО РАН за возможность проведения экспериментов на суперкомпьютерном кластере Tesla.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF-2026-0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Kundakcioglu O. E., Alizamir S.** Generalized assignment problem // Encyclopedia of optimization. Boston, MA: Springer, 2008. P. 1153–1162.
2. **Ross G. T., Soland R. M.** Modeling facility location problems as generalized assignment problems // Manag. Sci. 1977. V. 24. P. 345–357.

3. **Zaozerskaya L.** Analysis of integer programming model of academic load distribution // Mathematical Optimization Theory and Operations Research. Rev. Sel. Pap. 18th Int. Conf. (Yekaterinburg, Russia, July 8–12, 2019). Cham: Springer, 2019. P. 266–279. (Commun. Comput. Inf. Sci.; V. 1090). DOI: 10.1007/978-3-030-33394-2_21.
4. **Zhang C. W., Ong H. L.** An efficient solution to biobjective generalized assignment problem // Adv. Eng. Softw. 2007. V. 38. P. 50–58.
5. **Cohen R., Katzir L., Raz D.** An efficient approximation for the generalized assignment problem // Inform. Process. Lett. 2006. V. 100, No. 4. P. 162–166.
6. **Dawande M., Kalagnanam J.** The multiple knapsack problem with color constraints. Yorktown Heights, NY: IBM TJ Watson Res. Center, 1998.
7. **Kondakov A., Kochetov Y.** A core heuristic and the branch-and-price method for a bin packing problem with a color constraint // Optimization Problems and Their Applications. Rev. Sel. Pap. 7th Int. Conf. (Omsk, Russia, July 8–14, 2018). Cham: Springer, 2018. P. 309–320. (Commun. Comput. Inf. Sci.; V. 871). DOI: 10.1007/978-3-319-93800-4_25.
8. **Zaozerskaya L. A., Plankova V. A.** Researching and solving a bicriteria supply management problem with the given volumes of batches // J. Phys.: Conf. Ser. 2019. V. 1210. Article ID 012164. 7 p.
9. **Ngoo C. M., Goh S. L., Sze S. N., Sabar N. R., Hijazi M. H. A., Kendall G.** A survey of mat-heuristics for combinatorial optimisation problems: Variants, trends and opportunities // Appl. Soft Comput. 2024. V. 164. Article ID 111947. 17 p.
10. **Заозёрская Л. А., Захарова Ю. В.** Модели и алгоритмы локального поиска для маршрутизации транспортных средств с возвратами и временными окнами // Изв. Иркутск. гос. ун-та. Сер. Математика. 2024. Т. 48. С. 95–110.
11. **Ibaraki T., Ohashi T., Mine H.** A heuristic algorithm for mixed-integer programming problems // Approaches to integer programming. Heidelberg: Springer, 2009. P. 115–136.
12. **Berthold T., Heinz S., Pfetsch M. E., Vigerske S.** Large neighborhood search beyond MIP. ZIB-Report 11-21. Berlin: Konrad-Zuse-Zentrum Inform., 2011. 12 p.
13. **Genova K.** A heuristic algorithm for solving mixed integer problems // Cybern. Inf. Technol. 2011. V. 11, No. 2. P. 3–12.
14. **Danna E., Rothberg E., Pape C. L.** Exploring relaxation induced neighborhoods to improve MIP solutions // Math. Program. 2005. V. 102, No. 1. P. 71–90.
15. **Fischetti M., Lodi A.** Local branching // Math. Program. 2003. V. 98, No. 1. P. 23–47.
16. **Balachandar S. R., Kannan K.** A new heuristic approach for the large-scale generalized assignment problem // Int. J. Math. Comput. Phys. Elect. Comput. Eng. 2009. V. 3. P. 969–974.
17. **Narciso M. G., Lorena L. A. N.** Lagrangian/surrogate relaxation for generalized assignment problems // Eur. J. Oper. Res. 1999. V. 114. P. 165–177.

18. **Haddadi S.** Lagrangian decomposition based heuristic for the generalized assignment problem // *Inf. Syst. Oper. Res.* 1999. V. 37, No. 4. P. 392–402.
19. **Amini M. M., Racer M.** A rigorous computational comparison of alternative solution methods for the generalized assignment problem // *Manag. Sci.* 1994. V. 40, No. 7. P. 868–890.
20. **D7.0az J. A., Fernandez E.** A tabu search heuristic for the generalized assignment problem // *Eur. J. Oper. Res.* 2001. V. 132, No. 1. P. 22–38.
21. **Feltl H., Raidl G. R.** An improved hybrid genetic algorithm for the generalized assignment problem // *Proc. 2004 ACM Symp. Appl. Comput.* (Nicosia, Cyprus, Mar. 14–17, 2004). New York: ACM Press, 2004. P. 990–995.
22. **French A. P., Wilson J. M.** An LP-based heuristic procedure for the generalized assignment problem with special ordered sets // *Comput. Oper. Res.* 2007. V. 34, No. 8. P. 2359–2369.
23. **Trick M. A.** A linear relaxation heuristic for the generalized assignment problem // *Nav. Res. Logist.* 1992. V. 39, No. 2. P. 137–151.
24. **Cattrysse D. G., Salomon M., Van Wassenhove L. N.** A set partitioning heuristic for the generalized assignment problem // *Eur. J. Oper. Res.* 1994. V. 72, No. 1. P. 167–174.
25. **Forrest J. J. H., Kalagnanam J., Ladanyi L.** A column-generation approach to the multiple knapsack problem with color constraints // *INFORMS J. Comput.* 2006. V. 18, No. 1. P. 129–134.
26. **Srinivasan V., Thompson G. L.** An algorithm for assigning uses to sources in a special class of transportation problem // *Oper. Res.* 1973. V. 21, No. 1. P. 284–295.
27. **Garey M. R., Johnson D. S.** *Computers and intractability: A guide to the theory of NP-completeness.* San Francisco, CA: Freeman, 1979. 338 p.
28. **Munkres J.** Algorithms for the assignment and transportation problems // *J. Soc. Ind. Appl. Math.* 1957. V. 5, No. 1. P. 32–38.
29. **Rudolph G.** Finite Markov chain results in evolutionary computation: A tour d’horizon // *Fund. Inform.* 1998. V. 35, No. 1–4. P. 67–89.

Захарова Юлия Викторовна
Заозёрская Лидия Анатольевна

Статья поступила
15 октября 2025 г.
После доработки —
12 ноября 2025 г.
Принята к публикации
22 декабря 2025 г.

A LOCAL-SEARCH-BASED MATHEURISTIC
FOR A MODIFICATION OF THE GENERALIZED
ASSIGNMENT PROBLEMYu. V. Zakharova^{1, a} and L. A. Zaozerskaya^{1, 2, b}¹ Omsk Branch of Sobolev Institute of Mathematics,
13 Pevtsov Street, 123456 Omsk, Russia² Omsk State Technical University,
11 Mir Avenue, 644050 Omsk, RussiaE-mail: ^ayzakharova@ofim.oscsbras.ru, ^bzaozer@ofim.oscsbras.ru

Abstract. The paper considers a new modification of the generalized assignment problem with constraints on volumes and types of jobs. A set of jobs must be assigned to a set of agents (performers). Each job has volume and type. For each agent, constraints are defined on the number of assigned job types, the total volume of assigned jobs, and the efficiency of job execution. The objective is to maximize the total assignment efficiency subject to the given loading constraints. The computational complexity is analyzed. We propose a local search algorithm with neighborhoods of variable size which takes into account known approaches for solving the generalized assignment problem and the additional constraints of the considered problem. The neighborhoods are constructed by partially fixing the values of the variables corresponding to the assignment of jobs. Exact algorithms from the well-known mathematical programming problem-solving packages are used to solve the corresponding subproblems. Experimental results show that the proposed algorithm achieves results competitive to the hybrid algorithms based on branch-and-cut methods from Gurobi and SCIP solvers in terms of solution quality under given time limit on a series of test instances with various structural properties. Tab. 7, bibliogr. 29.

Keywords: assignment, color class, complexity, matheuristic.

References

1. **O. E. Kundakcioglu** and **S. Alizamir**, Generalized assignment problem, in *Encyclopedia of Optimization* (Springer, Boston, MA, 2008), pp. 1153–1162.
2. **G. T. Ross** and **R. M. Soland**, Modeling facility location problems as generalized assignment problems, *Manag. Sci.* **24**, 345–357 (1977).
3. **L. Zaozerskaya**, Analysis of integer programming model of academic load distribution, in *Mathematical Optimization Theory and Operations Research*, Rev. Sel. Pap. 18th Int. Conf. (Yekaterinburg, Russia, July 8–12, 2019) (Springer, Cham, 2019), pp. 266–279 (Commun. Comput. Inf. Sci., Vol. 1090), DOI: 10.1007/978-3-030-33394-2_21.
4. **C. W. Zhang** and **H. L. Ong**, An efficient solution to biobjective generalized assignment problem, *Adv. Eng. Softw.* **38**, 50–58 (2007).
5. **R. Cohen**, **L. Katzir**, and **D. Raz**, An efficient approximation for the generalized assignment problem, *Inform. Process. Lett.* **100** (4), 162–166 (2006).
6. **M. Dawande** and **J. Kalagnanam**, *The Multiple Knapsack Problem with Color Constraints* (IBM TJ Watson Res. Center, Yorktown Heights, NY, 1998).
7. **A. Kondakov** and **Y. Kochetov**, A core heuristic and the branch-and-price method for a bin packing problem with a color constraint, in *Optimization Problems and Their Applications*, Rev. Sel. Pap. 7th Int. Conf. (Omsk, Russia, July 8–14, 2018) (Springer, Cham, 2018), pp. 309–320 (Commun. Comput. Inf. Sci., Vol. 871), DOI: 10.1007/978-3-319-93800-4_25.
8. **L. A. Zaozerskaya** and **V. A. Plankova**, Researching and solving a bicriteria supply management problem with the given volumes of batches, *J. Phys., Conf. Ser.* **1210**, ID 012164 (2019).
9. **C. M. Ngoo**, **S. L. Goh**, **S. N. Sze**, **N. R. Sabar**, **M. H. A. Hijazi**, and **G. Kendall**, A survey of mat-heuristics for combinatorial optimisation problems: Variants, trends and opportunities, *Appl. Soft Comput.* **164**, ID 111947 (2024).
10. **L. A. Zaozerskaya** and **Yu. V. Zakharova**, Models and local search algorithms for vehicle routing with returns and time windows, *Izv. Irkutsk. Gos. Univ., Ser. Mat.* **48**, 95–110 (2024).
11. **T. Ibaraki**, **T. Ohashi**, and **H. Mine**, A heuristic algorithm for mixed-integer programming problems, in *Approaches to Integer Programming* (Springer, Heidelberg, 2009), pp. 115–136.
12. **T. Berthold**, **S. Heinz**, **M. E. Pfetsch**, and **S. Vigerske**, Large neighborhood search beyond MIP. *ZIB-Report 11-21* (Konrad-Zuse-Zentrum Inform., Berlin, 2011).
13. **K. Genova**, A heuristic algorithm for solving mixed integer problems, *Cybern. Inf. Technol.* **11** (2), 3–12 (2011).
14. **E. Danna**, **E. Rothberg**, and **C. L. Pape**, Exploring relaxation induced neighborhoods to improve MIP solutions, *Math. Program.* **102** (1), 71–90 (2005).
15. **M. Fischetti** and **A. Lodi**, Local branching, *Math. Program.* **98** (1), 23–47 (2003).

16. **S. R. Balachandar** and **K. Kannan**, A new heuristic approach for the large-scale generalized assignment problem, *Int. J. Math. Comput. Phys. Elect. Comput. Eng.* **3**, 969–974 (2009).
17. **M. G. Narciso** and **L. A. N. Lorena**, Lagrangian/surrogate relaxation for generalized assignment problems, *Eur. J. Oper. Res.* **114**, 165–177 (1999).
18. **S. Haddadi**, Lagrangian decomposition based heuristic for the generalized assignment problem, *Inf. Syst. Oper. Res.* **37** (4), 392–402 (1999).
19. **M. M. Amini** and **M. Racer**, A rigorous computational comparison of alternative solution methods for the generalized assignment problem, *Manag. Sci.* **40** (7), 868–890 (1994).
20. **J. A. D7.0az** and **E. Fernandez**, A tabu search heuristic for the generalized assignment problem, *Eur. J. Oper. Res.* **132** (1), 22–38 (2001).
21. **H. Feltl** and **G. R. Raidl**, An improved hybrid genetic algorithm for the generalized assignment problem, in *Proc. 2004 ACM Symp. Appl. Comput.* (Nicosia, Cyprus, Mar. 14–17, 2004) (ACM Press, New York, 2004), pp. 990–995.
22. **A. P. French** and **J. M. Wilson**, An LP-based heuristic procedure for the generalized assignment problem with special ordered sets, *Comput. Oper. Res.* **34** (8), 2359–2369 (2007).
23. **M. A. Trick**, A linear relaxation heuristic for the generalized assignment problem, *Nav. Res. Logist.* **39** (2), 137–151 (1992).
24. **D. G. Cattrysse**, **M. Salomon**, and **L. N. Van Wassenhove**, A set partitioning heuristic for the generalized assignment problem, *Eur. J. Oper. Res.* **72** (1), 167–174 (1994).
25. **J. J. H. Forrest**, **J. Kalagnanam**, and **L. Ladanyi**, A column-generation approach to the multiple knapsack problem with color constraints, *INFORMS J. Comput.* **18** (1), 129–134 (2006).
26. **V. Srinivasan**, **G. L. Thompson**, An algorithm for assigning uses to sources in a special class of transportation problem, *Oper. Res.* **21** (1), 284–295 (1973).
27. **M. R. Garey** and **D. S. Johnson**, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).
28. **J. Munkres**, Algorithms for the assignment and transportation problems, *J. Soc. Ind. Appl. Math.* **5** (1), 32–38 (1957).
29. **G. Rudolph**, Finite Markov chain results in evolutionary computation: A tour d’horizon, *Fund. Inform.* **35** (1–4), 67–89 (1998).

Yulia V. Zakharova
Lidia A. Zaozerskaya

Received October 15, 2025
Revised November 12, 2025
Accepted December 22, 2025