S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Том 20, № 2, стр. 144–145 (2023) УДК 519.853
https://doi.org/10.33048/semi.2023.20.??? MSC 90C26

Методы DC оптимизации для задачи взвешенной 3-раскраски графа

Г.А. ЖУКОВ,
А.В. ОРЛОВ

ПРЕДСТАВЛЕНО П.П. ПЕТРОВЫМ

Abstract: The paper addresses the problem of finding a vertex 3-coloring of an edge-weighted graph that minimizes the sum of weights of edges connecting vertices of the same color. The problem is formulated as a continuous DC optimization problem, which allows applying the relevant mathematical framework. To solve the continuous problem, specialized local and global search algorithms have been implemented. In the computational experiment, a comparative analysis was conducted between the developed approach

---

Zhukov, G.A., Orlov, A.V., DC optimization methods for the weighted graph 3-coloring problem .

Работа Орлова А.В. выполнена за счет государственного задания в рамках темы "Теория и методы непрерывной и дискретной оптимизации в прикладных задачах исследования операций с реализацией на высокопроизводительных вычислительных системах"(Шифр научной темы FWEW-2026-0008, № госрегистрации 126021217232-3).

Поступила 1 января 2023 г., опубликована 31 декабря 2023 г.

and reference methods: a greedy algorithm, the Variable Neighborhood Descent (VND) method, and its multistart version. The obtained results confirm the applicability of DC optimization techniques for solving the weighted 3-coloring problem and reveal promising areas for further research.

Keywords: Minimum 3-Partition, 3-coloring, DC optimization, local search, global search, computational experiment.

1 Введение

Задачи раскраски графов являются классическими в комбинаторной оптимизации. Одна из фундаментальных проблем в этой области заключается в определении хроматического числа $\chi(G)$ — минимального количества цветов, необходимого для правильной раскраски вершин графа (раскраски, при которой смежные вершины имеют разные цвета). Данная задача входит в известный список 21 NP-полной задачи Карпа [1].

Теоретический базис оценок хроматического числа опирается на ряд классических результатов. Теорема Брукса [2] устанавливает верхнюю границу $\chi(G) \leq \Delta(G)$ через максимальную степень вершины (для связных графов, отличных от полных графов и нечётных циклов). В [3] показано существование графов с произвольно большим хроматическим числом при отсутствии больших клик. В поисках новых подходов исследователи обратились к методам, далёким от традиционной комбинаторики. Так, в [4] продемонстрирована эффективность использования аппарата алгебраической топологии для получения нижних оценок $\chi(G)$, что заложило основы топологической комбинаторики.

Среди частных случаев задачи о хроматическом числе особое место занимает задача 3-раскрашиваемости (3-Coloring), заключающаяся в проверке существования правильной раскраски вершин графа в 3 цвета. Несмотря на фиксированное число цветов, задача остаётся NP-полной даже для планарных графов с максимальной степенью вершин 4 [5]. Более того, как показано в [6], NP-полнота сохраняется и для планарных 4-регулярных графов.

В данной работе исследуется оптимизационный вариант проблемы 3-раскрашиваемости — задача взвешенной 3-раскраски (в англоязычной литературе известная как Minimum 3-Partition [7]). Она состоит в нахождении раскраски вершин графа в три цвета, при которой суммарный вес рёбер, соединяющих вершины одного цвета, минимален.

Известно, что дискретные задачи допускают непрерывную формулировку [8, разд. 3.3][9]. В настоящем исследовании предлагается осуществить переход от исходной дискретной постановки задачи взвешенной 3-раскраски к непрерывному аналогу с последующим применением аппарата DC оптимизации, точнее Теории глобального поиска (ТГП), разработанной А. С. Стрекаловским [9, 10]. Центральную роль в ней играют конструктивные условия глобальной оптимальности (УГО). Они позволяют сводить исходную невыпуклую проблему к семейству выпуклых задач, для решения которых применимы современные программные комплексы (например, *CPLEX*, *Gurobi* и т.д.). Конструктивность УГО лежит в основе стратегии глобального поиска, которая объединяет процедуры локального поиска с методами выхода из локальных экстремумов. В последние годы удалось распространить теорию на общие задачи DC оптимизации с невыпуклыми ограничениями типа равенства и неравенства [11, 12]. Ключевыми особенностями новых результатов являются применение теории точного штрафа и динамическое изменение параметра штрафа в процессе счёта.

Практическая эффективность ТГП подтверждена на широком спектре задач. В работах [13, 14, 15] теория успешно применена к различным двухуровневым задачам. В исследовании [10] рассматриваются линейная задача о дополнительности, поиск равновесий Нэша в биматричных играх и квадратично-линейные двухуровневые задачи. Применению подхода к полиматричным играм посвящена, например, работа [16].

Особый интерес в контексте настоящего исследования представляют результаты, полученные с помощью ТГП для дискретной задачи о максимальной (взвешенной) клике [17, 18]. В указанных работах эта изначально комбинаторная NP-трудная задача на графе сформулирована и исследована как задача DC оптимизации. Применение специализированного локального поиска, учитывающего структуру задачи, в рамках общей схемы глобального поиска, позволило достичь здесь высокой вычислительной эффективности. Успешный опыт решения задачи о клике служит непосредственным обоснованием целесообразности применения ТГП к задаче взвешенной 3-раскраски.

Статья организована следующим образом. В разделе 2 задача взвешенной 3-раскраски сводится к задаче DC минимизации на выпуклом допустимом множестве. Раздел 3 посвящён специальному методу локального поиска, основанному на последовательном решении линеаризованных по базовой невыпуклости задач, и его численному тестированию. В разделе 4 в терминах исследуемой задачи сформулированы конструктивные условия глобальной оптимальности и общая схема глобального поиска; затем конкретизированы её ключевые элементы и построен соответствующий алгоритм глобального поиска. Результаты вычислительных экспериментов на трёх группах графов приведены в разделе 5. Заключение содержит основные выводы и направления дальнейших исследований.

2 Постановка задачи и её редукция

Пусть $G = (V, E)$ — простой неориентированный граф, где V — множество вершин, $|V| = n$, а E — множество рёбер. Для каждого ребра $e \in E$ задан положительный вес $w(e)$. Требуется раскрасить вершины графа в 3 цвета так, чтобы суммарный вес рёбер, соединяющих вершины одного цвета, был минимален. Данная задача является NP-трудной, поскольку к ней сводится NP-полная задача 3-раскрашиваемости графа [6].

Сформулируем эту задачу как задачу дискретной оптимизации. Введем бинарные переменные x_i^q, принимающие значение 1, если вершина i цвета q, и 0 в противном случае. Для каждого цвета q сформируем вектор переменных $x^q := (x_1^q, x_2^q, \ldots, x_n^q)$, $q \in \{1, 2, 3\}$. Обозначим $x := (x^1, x^2, x^3) \in \mathbb{R}^{3n}$. Определим симметричную матрицу весов $W := (w_{ij})_{n\times n}$, где элемент w_{ij} равен весу ребра (i, j), если оно существует, и 0 в противном случае. Тогда задачу можно сформулировать следующим образом [19]:

$$\left.\begin{aligned} F(x) := \sum_{q=1}^{3} \langle x^q,\, Wx^q\rangle \downarrow \min_x, \\ \sum_{q=1}^{3} x_i^q = 1 \quad \forall i \in V, \\ x_i^q \in \{0, 1\} \quad \forall i \in V,\ q \in \{1, 2, 3\}. \end{aligned}\right\}$$

В практических приложениях матрица W, как правило, является знаконеопределённой, поэтому в дальнейшем будем считать, что целевая функция $F(x)$ не является выпуклой.

Осуществим переход от дискретной постановки к её непрерывному аналогу [8, разд. 3.3]. Заменим условие дискретности $x_i^q \in \{0, 1\}$ на требования принадлежности переменных единичному отрезку $x_i^q \in [0, 1]$ (данное ограничение формально избыточно, однако необходимо для дальнейших преобразований) и одновременное выполнение равенства $(x_i^q)^2 - x_i^q = 0$. Просуммировав эти равенства по i

для каждого q, получим:

$$\sum_{i=1}^{n}\big((x_i^q)^2 - x_i^q\big) = 0, \quad q \in \{1,2,3\}.$$

После выделения полного квадрата это условие принимает вид:

$$\left\|x^q - \frac{1}{2}e\right\|^2 - \frac{n}{4} = 0, \quad q \in \{1,2,3\},$$

где $e = (1,1,\ldots,1) \in \mathbb{R}^n$. Заметим, что квадратичная функция $(x_i^q - 1/2)^2$ на отрезке $[0,1]$ достигает максимума на концах отрезка, следовательно, справедлива оценка:

$$\left(x_i^q - \frac{1}{2}\right)^2 \leq \frac{1}{4}. \tag{1}$$

Из оценки (1) следует, что равенство $\sum\limits_{i=1}^{n}(x_i^q - 1/2)^2 = n/4$ возможно лишь тогда, когда каждое слагаемое в точности равно $1/4$, то есть когда все x_i^q принимают значения 0 или 1. Таким образом, при дополнительном условии $x_i^q \in [0,1]$ ограничение $\|x^q - \frac{1}{2}e\|^2 = n/4$ эквивалентно условию дискретности переменных x_i^q.

Обозначим через S выпуклое множество, определяемое линейными ограничениями:

$$S := \left\{ x \in \mathbb{R}^{3n} \;\middle|\; \begin{array}{l} \sum\limits_{q=1}^{3} x_i^q = 1 \quad \forall i \in V; \\ 0 \leq x_i^q \leq 1 \quad \forall i \in V, \quad q \in \{1,2,3\} \end{array} \right\}$$

Тогда непрерывная постановка задачи принимает следующий вид:

$$\left.\begin{array}{l} F(x) \downarrow \min\limits_{x}, \\ \left\|x^q - \dfrac{1}{2}e\right\|^2 = \dfrac{n}{4}, \quad q \in \{1,2,3\}, \\ x \in S. \end{array}\right\} \tag{$\mathcal{P}$}$$

Задача ($\mathcal{P}$) характеризуется невыпуклостью как в целевой функции, так и в системе ограничений (поскольку квадратичное равенство задаёт невыпуклое множество), что затрудняет её исследование и решение. При этом целевая функция $F(x)$ является DC функцией. Действительно, симметричность матрицы W позволяет построить разложение $W = A - B$, где A и B — положительно определённые матрицы [9, с. 269–270]. Это даёт возможность представить $F(x)$ в виде разности двух выпуклых квадратичных форм:

$$F(x) = \sum_{q=1}^{3} \big(\langle x^q,\ Ax^q \rangle - \langle x^q,\ Bx^q \rangle \big).$$

Далее оштрафуем невыпуклое ограничение для того, чтобы перейти от задачи $(\mathcal{P})$ к задаче с невыпуклостью только в целевой функции. Получим следующую задачу:

$$\Phi_\mu(x) := \sum_{q=1}^{3} \big(\langle x^q,\ Ax^q \rangle - \langle x^q,\ Bx^q \rangle \big) + \mu P(x) \downarrow \min_x, \quad x \in S, \quad (\mathcal{P}_\mu)$$

где $\mu > 0$ — параметр штрафа, а функция $P(x)$ представляет собой классический штраф для ограничений равенств [20, гл. 15] и имеет следующий вид:

$$P(x) := \sum_{q=1}^{3} \left| \left\| x^q - \frac{1}{2} e \right\|^2 - \frac{n}{4} \right|.$$

Исследуем свойства функции $P(x)$. В силу (1) справедлива оценка:

$$\left\| x^q - \frac{1}{2} e \right\|^2 = \sum_{i=1}^{n} \left(x_i^q - \frac{1}{2} \right)^2 \le \frac{n}{4}, \quad q \in \{1, 2, 3\}, \quad x \in S.$$

Данный факт позволяет однозначно раскрыть модули в штрафной функции с обратными знаками:

$$P(x) = \sum_{q=1}^{3} \left| \left\| x^q - \frac{1}{2} e \right\|^2 - \frac{n}{4} \right| = \sum_{q=1}^{3} \Big(\frac{n}{4} - \left\| x^q - \frac{1}{2} e \right\|^2 \Big) \quad \forall x \in S.$$

Заметим, что при фиксированном μ задача $(\mathcal{P}_\mu)$ представляет собой задачу DC минимизации на выпуклом допустимом множестве S. Действительно, подставив выражение для $P(x)$ в целевую функцию оштрафованной задачи, получим следующее представление:

$$\Phi_\mu(x) = G(x) - H(x), \tag{2}$$

где $G(x) = \sum_{q=1}^{3} G_q(x)$, $H(x) = \sum_{q=1}^{3} H_q(x)$ и выпуклые компоненты определены как:

$$G_q(x) = \langle x^q, Ax^q \rangle + \frac{\mu n}{4}, H_q(x) = \langle x^q, Bx^q \rangle + \mu \left\| x^q - \frac{1}{2} e \right\|^2, q \in \{1, 2, 3\}.$$

Из классической теории штрафа известно, что если для некоторого значения параметра $\mu > 0$ глобальное решение $z(\mu)$ задачи $(\mathcal{P}_\mu)$ является допустимым решением задачи $(\mathcal{P})$ (т.е. выполняется условие $P(z(\mu)) = 0$, то $z(\mu)$ будет глобальным решением задачи $(\mathcal{P})$ [20, гл. 17]. Более того, если для некоторого значения $\hat{\mu}$ выполняется условие $P(z(\hat{\mu})) = 0$, то при всех $\mu \geq \hat{\mu}$ глобальное решение оштрафованной задачи совпадает с глобальным решением исходной задачи $(\mathcal{P})$ [20, гл. 17].

Ключевым моментом теории точного штрафа [11, 12, 20] является существование такого порогового значения $\mu_* > 0$, что для всех $\mu \geq \mu_*$ выполняется равенство $P(z(\mu)) = 0$. Сформулируем и докажем теорему о точности штрафа для рассматриваемой задачи.

Теорема 1. *Существует пороговое значение $\mu_* > 0$, такое что для любого $\mu \geq \mu_*$ оптимальное решение $z(\mu)$ оштрафованной задачи $(\mathcal{P}_\mu)$ удовлетворяет условию $P(z(\mu)) = 0$.*

Доказательство. Рассмотрим допустимое множество S. В силу блочной структуры ограничений S представимо в виде декартова произведения n стандартных симплексов в пространстве $\mathbb{R}^3$. Известно, что вершинами такого симплекса являются единичные орты координатных осей [21, разд. 2.2]. Следовательно, все вершины политопа S являются бинарными векторами. Заметим, что для любого бинарного вектора штрафная функция $P(x)$ обращается в ноль, поскольку $P(x)$ — штраф за нарушение целочисленности переменных.

Исследуем свойства функции $\Phi_\mu(x)$. Гессиан функции $\Phi_\mu(x)$ является блочно-диагональной матрицей, где каждый диагональный блок имеет вид $2(W - \mu I)$, где I — единичная матрица. Выберем $\mu_* > \max\{\lambda_{\max}(W), 0\}$, где $\lambda_{\max}(W)$ — максимальное собственное значение матрицы W. Тогда при $\mu \geq \mu_*$ матрица $2(W - \mu I)$ оказывается отрицательно определённой (поскольку все её собственные значения отрицательны [21, разд. A.5]). Следовательно, целевая функция оштрафованной задачи $\Phi_\mu(x)$ становится строго вогнутой на $\mathbb{R}^{3n}$. Поскольку S — выпуклое множество, то $\Phi_\mu(x)$ будет строго вогнутой и на S.

Далее заметим, что множество S ограничено (поскольку $S \subset [0, 1]^{3n}$) и замкнуто (как декартово произведение замкнутых множеств). Значит, по теореме Гейне–Бореля [22, гл. 2, теор. 2.41], S является компактом. Согласно фундаментальному свойству задач вогнутой минимизации [8, гл. 2, теор. I.1], глобальный минимум строго вогнутой функции на компактном выпуклом множестве достигается в одной из его крайних точек. Таким образом, оптимальное решение $z(\mu)$ задачи $(\mathcal{P}_\mu)$ при $\mu \geq \mu_*$ является вершиной S. Следовательно, ограничение $P(z(\mu)) = 0$ выполнено, что завершает доказательство. □

Опираясь на упомянутые выше результаты теории штрафа и теорему 1, заключаем, что решение исходной задачи ($\mathcal{P}$) может быть сведено к решению всего одной задачи ($\mathcal{P}_\mu$) при любом фиксированном значении параметра $\mu \geq \mu_*$.

Далее в работе представлен процесс построения методов локального и глобального поисков в задаче ($\mathcal{P}_\mu$) при фиксированном μ (при этом считаем, что $\mu > \mu_*$). Также принимаем во внимание, что ($\mathcal{P}_\mu$) с фиксированным μ является задачей DC минимизации на выпуклом множестве. Вопрос практического отыскания $\mu > \mu_*$ будет отдельно решён ниже во время вычислительных экспериментов.

3 Специальный метод локального поиска и его тестирование

Важнейшей составляющей метода глобального поиска является локальный поиск. В данном случае будем использовать специальный метод локального поиска (СМЛП) [9, 13] для задачи DC минимизации ($\mathcal{P}_\mu$), который заключается в последовательном решении линеаризованных по базовой невыпуклости задач. Формулировка линеаризованной в точке y^s задачи ($\mathcal{P}_\mu$) имеет вид:

$$\left.\begin{aligned} \Psi_s(x) = G(x) - \langle \nabla H(y^s),\ x\rangle = \\ = \sum_{q=1}^{3}\big(G_q(x) - \langle \nabla H_q(y^s),\ x^q\rangle\big) \downarrow \min_x,\ x \in S. \end{aligned}\right\} \qquad (\mathcal{PL}(y^s))$$

Приведём явное выражение для функции $\Psi_s(x)$. Для этого вычислим градиент функций $H_q(x)$, $q \in \{1,2,3\}$:

$$\nabla H_q(x) = \nabla\left(\langle x^q, Bx^q\rangle + \mu\left\|x^q - \frac{1}{2}e\right\|^2\right) = \\ = 2Bx^q + 2\mu\left(x^q - \frac{1}{2}e\right), \quad q \in \{1,2,3\}.$$

Тогда функция $\Psi_s(x)$ записывается следующим образом (считаем, что $y^s := (y^{s1}, y^{s2}, y^{s3}) \in \mathbb{R}^{3n}$):

$$\Psi_s(x) = \sum_{q=1}^{3}\left[\langle x^q,\ Ax^q\rangle - 2\left\langle By^{sq} + \mu\left(y^{sq} - \frac{1}{2}e\right),\ x^q\right\rangle\right] + \frac{3\mu n}{4}.$$

Задача $(\mathcal{PL}(y^s))$ является выпуклой, что позволяет использовать для её решения классические методы выпуклой оптимизации и соответствующие программные комплексы. Общая схема СМЛП имеет следующий вид (y_0 — некоторая стартовая точка):

Шаг 0: Положить $s := 0$, $y^s := y_0$.
Шаг 1: Найти приближенное решение y^{s+1} задачи $(\mathcal{PL}(y^s))$ с точностью ρ_s.
Шаг 2: Положить $s := s + 1$ и перейти к шагу 1.

Согласно результатам из [9, 13] эта простая схема сходится к так называемой *критической точке* $\hat{x}$ в задаче $(\mathcal{P}_\mu)$. Точка $\hat{x}$ оказывается решением линеаризованной задачи $(\mathcal{PL}(\hat{x}))$ (линеаризованной в ней же самой).

При практической реализации СМЛП мы также используем понятие *приближённой критической точки* и в вычислительную схему вводится дополнительный шаг 1a (проверка критерия остановки) [9, 13]:

$$\Psi_s(y^s) - \Psi_s(y^{s+1}) \leq \tau/2. \tag{3}$$

При выполнении критерия остановки (3) можно показать, что точка y^s, полученная с помощью СМЛП, оказывается $(\tau/2+\rho_s)$-решением задачи $(\mathcal{PL}(y^s))$ (см. [9, 13]). Иными словами, y^s является $(\tau/2 + \rho_s)$-приближённой критической точкой в задаче $(\mathcal{P}_\mu)$ (если требуется отыскать, скажем, τ-критическую точку, то достаточно выбрать $\rho_s \leq \tau/2$).

В завершение раздела представим результаты численного тестирования СМЛП, выполненного с целью определения порогового значения параметра штрафа μ. Данное тестирование было проведено для всех тестовых примеров (подробности о тестовых данных см. в разделе 5). Для иллюстрации в таблице 1 приведено по одному примеру из каждой группы тестовых графов при значениях параметров $\tau/2 = 10^{-3}$ и $\rho_s = 10^{-6}$. В таблице использованы следующие обозначения:

- n — число вершин графа;
- m — число рёбер;
- μ — значение параметра штрафа;
- $\widehat{\Phi}_\mu$ — значение целевой функции оштрафованной задачи в приближённой критической точке;
- $\widehat{P}$ — значение штрафной функции в приближённой критической точке;
- N_{lin} — количество решённых линеаризованных задач (итераций СМЛП).

Стартовая точка y_0 задавалась как случайный вектор, равномерно распределённый на $[0,1]^{3n}$. Из таблицы 1 можно видеть, что во всех тестовых примерах при $\mu = 9$ значение штрафной функции в критической точке $\widehat{P}$ не превышает 0.01, а при $\mu = 10$ обращается в ноль. Это же свойство сохраняется для всех тестовых примеров. Поэтому для выполнения равенства $\widehat{P} = 0$ в дальнейших вычислительных экспериментах будем использовать $\mu = 10$.

Таблица 1. Результаты тестирования СМЛП

n	m	μ	$\widehat{\Phi}_\mu$	$\widehat{P}$	N_{lin}
53	293	5	1121.62	4.24	18
		7	1123.05	5.54	18
		9	1069.18	**0**	42
		10	1069.18	**0**	38
100	1000	5	14893.8	2.4	157
		7	14898.6	1.54	163
		9	14949.2	0.01	171
		10	14949.1	**0**	177
100	400	5	2744.35	2	103
		7	2742.42	0.08	106
		9	2742.34	**0**	107
		10	2742.34	**0**	104

4 Метод глобального поиска

Спецификой невыпуклых задач является наличие множества локальных экстремумов, вследствие чего применение исключительно методов локального поиска не гарантирует нахождения глобального решения [8, гл. 1], [20, гл. 1]. Для достижения глобального оптимума необходимо применять подходы, позволяющие выходить из локальных решений и стационарных точек. В данном случае будем использовать подход, базирующийся на так называемых конструктивных условиях глобальной оптимальности, доказанных А. С. Стрекаловским (далее — УГО) [9, 13]. В терминах исследуемой задачи с учётом DC разложения (2) необходимые УГО записываются следующим образом.

Теорема 2 ([9, 13])**.** *Если $z \in S$ является глобальным решением задачи $(\mathcal{P}_\mu)$ и $\zeta := \Phi_\mu(z)$, то*

$$\forall (y, \gamma) \,:\, y \in S, \quad \gamma - H(y) = \zeta, \tag{4}$$

$$G(y) \leq \gamma \leq \sup_{x \in S} G(x), \tag{5}$$

$$G(x) - \gamma \geq \langle \nabla H(y),\ x - y \rangle \quad \forall x \in S. \tag{6}$$

Неравенство (6) называют основным неравенством УГО. Нетрудно видеть, что УГО сводят решение невыпуклой задачи $(\mathcal{P}_\mu)$ к решению семейства линеаризованных выпуклых задач, зависящих от параметра y:

$$G(x) - \langle \nabla H(y),\ x \rangle \downarrow \min_x, \quad x \in S, \tag{$\mathcal{PL}(y)$}$$

а также последующей проверке нарушения неравенства (6). Пусть $z^k \in S$ — текущая допустимая точка. Если для некоторой пары (y, γ) из (4) на уровне $\zeta = \zeta_k = \Phi_\mu(z^k)$ и для допустимой точки $u \in S$ неравенство (6) нарушается:

$$G(u) < \gamma + \langle \nabla H(y), u - y \rangle,$$

то из выпуклости $H(\cdot)$ следует:

$$\Phi_\mu(u) = G(u) - H(u) < \gamma + \langle \nabla H(y), u - y \rangle - H(u) \leq$$
$$\leq \gamma + H(u) - H(y) - H(u) = \gamma - H(y) = \Phi_\mu(z^k)$$

или $\Phi_\mu(u) < \Phi_\mu(z^k)$. Значит, $u \in S$ «лучше», чем z^k. При этом z^k может быть критической точкой в задаче и даже её локальным решением. Описанное конструктивное свойство лежит в основе схемы (стратегии) глобального поиска (СГП) [9, 13]. Запишем её в терминах задачи $(\mathcal{P}_\mu)$. Пусть заданы:

- стартовая точка $x^0 \in S$;
- числовые последовательности $\{\tau_k\}$ и $\{\delta_k\}$ такие, что τ_k, $\delta_k > 0$, $\tau_k \downarrow 0$, $\delta_k \downarrow 0$ при $k \to \infty$;
- множество направлений $Dir = \{v^l := (v^{l1}, v^{l2}, v^{l3}) \in \mathbb{R}^{3n} \mid v^l \neq 0, l = 1, \ldots, N\}$;
- оценки глобального минимума и максимума функции $G(x)$ на S:

$$\gamma_- \triangleq \inf_{x \in S} G(x), \quad \gamma_+ \triangleq \sup_{x \in S} G(x);$$

- параметр $M > 0$.

Шаг 0: Положить $k := 0$, $\gamma := \gamma_-$, $\Delta\gamma := (\gamma_+ - \gamma_-)/M$, $\bar{x}^k := x^0$, $l := 1$.

Шаг 1: Начиная из точки $\bar{x}^k$, с помощью метода локального поиска из раздела 3 построить τ_k-приближённо критическую точку $x^k \in S$ в задаче $(\mathcal{P}_\mu)$. Положить $\zeta_k := \Phi_\mu(x^k)$.

Шаг 2: По направлению $v^l \in Dir$ построить точку u^l аппроксимации $\mathcal{A}_k$ поверхности уровня функции $H(\cdot)$:

$$\mathcal{A}_k := \{u^l \mid H(u^l) = \gamma - \zeta_k, \quad l = 1, \dots, N\}.$$

Шаг 3: Если $G(u^l) > \gamma$, $l < N$ и $\gamma < \gamma_+$, то положить $l := l+1$ и перейти на шаг 2.

Шаг 4: Если $G(u^l) > \gamma$, $l = N$ и $\gamma < \gamma_+$, то положить $l := 1$, $\gamma := \gamma + \Delta\gamma$ и перейти на шаг 2.

Шаг 5: Если $G(u^l) > \gamma$, $l = N$ и $\gamma = \gamma_+$, то остановиться; x^k — найденное решение задачи $(\mathcal{P}_\mu)$.

Шаг 6: Найти δ_k-решение $\bar{u}^l$ линеаризованной задачи в точке u^l $(\mathcal{PL}(u^l))$.

Шаг 7: Начиная из точки $\bar{u}^l$, построить с помощью СМЛП τ_k-приближённо критическую точку $\hat{x}^l \in S$ в задаче $(\mathcal{P}_\mu)$.

Шаг 8: Если $\Phi_\mu(\hat{x}^l) < \Phi_\mu(x^k)$, то положить $\bar{x}^{k+1} := \hat{x}^l$, $k := k+1$, $\gamma := \gamma_-$, $l := 1$. Перейти на шаг 2.

Шаг 9: Если $\Phi_\mu(\hat{x}^l) \geq \Phi_\mu(x^k)$ и $l < N$, то положить $l := l+1$ и вернуться на шаг 2.

Шаг 10: Если $\Phi_\mu(\hat{x}^l) \geq \Phi_\mu(x^k)$, $l = N$ и $\gamma < \gamma_+$, то $\gamma := \gamma + \Delta\gamma$, $l := 1$ и вернуться на шаг 2.

Шаг 11: Если $\Phi_\mu(\hat{x}^l) \geq \Phi_\mu(x^k)$, $l = N$ и $\gamma = \gamma_+$, то завершить работу алгоритма. Точка x^k — найденное решение задачи $(\mathcal{P}_\mu)$.

Поясним основные шаги СГП:

- на шаге 2 строится конечная аппроксимация поверхности уровня функции, задающей базовую невыпуклость в исследуемой задаче;
- на шагах 3–5 с помощью неравенства из (5) в УГО проверяется перспективна ли текущая точка аппроксимации;
- на шагах 6–7 находится новая приближённая критическая точка в задаче $(\mathcal{P}_\mu)$;
- на шагах 8–11 анализируется, даёт ли найденная приближённая критическая точка улучшение текущего значения целевой функции.

Представленная СГП носит достаточно общий характер. Для реализации алгоритма на её основе необходимо исследовать следующие моменты: выбор параметра M (определяет число подотрезков в разбиении интервала $[\gamma_-, \gamma_+]$ для одномерного поиска по γ); выбор множества направлений Dir и построение точек аппроксимации поверхности уровня на основе этого множества (см. шаг 2);

поиск чисел γ_- и γ_+. Настройка этих элементов имеет эвристический характер, базируется на данных задачи и существенно влияет на качество получаемых решений.

Особое внимание следует уделить выбору множества направлений Dir, поскольку аппроксимация должна быть достаточно репрезентативной для решения вопроса о том, является ли x^k глобальным решением задачи [13–18]. Рассмотрим два варианта построения данного множества:

(1) Множество $Dir_1 = \{(e^{i_1}, e^{i_2}, e^{i_3}) \in \mathbb{R}^{3n} \mid i_1, i_2, i_3 \in \{1, \ldots, n\}\}$, где $e^i \in \mathbb{R}^n$, $i \in \{1, \ldots, n\}$ — векторы стандартного базиса пространства $\mathbb{R}^n$.

Выбор данного множества обусловлен тем, что целевая функция $\Phi_\mu(x)$ представляет собой квадратичную форму в пространстве $\mathbb{R}^{3n}$. С учётом этого факта, а также блочной структуры вектора $x = (x^1, x^2, x^3)$, для построения аппроксимации естественно использовать стандартные орты в каждой из его компонент. Количество точек в этой аппроксимации зависит от числа вершин графа n и составляет n^3.

(2) Множество $Dir_2 = \{(b^1, b^2, b^3) \in \mathbb{R}^{3n} \mid b^1, b^2, b^3 \in \{0, 1\}^n\}$.

Выбор данного множества обусловлен изначально дискретным характером исходной задачи и блочной структурой вектора $x = (x^1, x^2, x^3)$. Поэтому для построения аппроксимации естественно использовать в каждой из компонент x произвольный булев вектор. Количество точек в этой аппроксимации составляет 2^{3n}.

Далее на шаге 2 по точкам из множества направлений необходимо построить аппроксимацию поверхности уровня функции $H(\cdot)$. Для этого используем следующий приём. Для заданного множества направлений:

$$Dir = \{v^l = (v^{l1}, v^{l2}, v^{l3}) \in \mathbb{R}^{3n} \mid v^l \neq 0, \quad l = 1, \ldots, N\},$$

множество $\mathcal{A}_k$ строится из векторов, коллинеарных векторам из Dir и лежащих на поверхности уровня:

$$u^{lq} = \lambda v^{lq}, \quad q = 1, 2, 3; \quad H(u^l) = \gamma - \zeta_k := \alpha_k.$$

Для рассматриваемой задачи это условие принимает вид:

$$\sum_{q=1}^{3} \left(\langle u^{lq},\ Bu^{lq} \rangle + \mu \left\| u^{lq} - \frac{1}{2} e \right\|^2 \right) = \alpha_k.$$

Подставив $u^{lq} = \lambda v^{lq}$ в уравнение, получаем:

$$\sum_{q=1}^{3}\left(\lambda^2\langle v^{lq},\ Bv^{lq}\rangle + \mu\left\|\lambda v^{lq} - \frac{1}{2}e\right\|^2\right) = \alpha_k. \tag{7}$$

Раскроем квадрат нормы, учитывая, что $\langle e,\ e\rangle = n$:

$$\left\|\lambda v^{lq} - \frac{1}{2}e\right\|^2 = \lambda^2\langle v^{lq},\ v^{lq}\rangle - \lambda\langle v^{lq},\ e\rangle + \frac{n}{4}, \quad q \in \{1,2,3\}. \tag{8}$$

Подставляя (8) в (7) и группируя слагаемые при степенях λ, приходим к квадратному уравнению относительно λ:

$$\Big(\sum_{q=1}^{3}\big[\langle v^{lq}, Bv^{lq}\rangle + \mu\langle v^{lq}, v^{lq}\rangle\big]\Big)\lambda^2 - \Big(\mu\sum_{q=1}^{3}\langle v^{lq}, e\rangle\Big)\lambda + \Big(\tfrac{3\mu n}{4} - \alpha_k\Big) = 0. \tag{9}$$

Известно, что для существования вещественных корней уравнения (9) достаточно потребовать неположительности свободного члена $c \leq 0$ при условии неотрицательности старшего коэффициента $a \geq 0$. Условие $a \geq 0$ достигается за счет построения разложения $W = A - B$ (см. раздел 2). В частности, всегда есть возможность выбрать матрицу B с неотрицательными элементами. Если в исходном разложении матрица B содержит отрицательные значения, достаточно прибавить ко всем элементам матриц A и B величину, равную модулю минимального отрицательного элемента. Такое преобразование не меняет разность $A - B$, но делает все элементы B неотрицательными, что, как легко заметить, обеспечивает неотрицательность коэффициента a. Будем считать, что построенное ранее разложение $W = A - B$ удовлетворяет указанному свойству.

Таким образом, требование существования вещественных корней (9) сводится к неравенству $c \leq 0$, что эквивалентно $\alpha_k \geq \dfrac{3\mu n}{4}$. Учитывая определение $\alpha_k = \gamma - \zeta_k$, приходим к неравенству $\gamma \geq \dfrac{3\mu n}{4} + \zeta_k$. Заметим, что последовательность $\{\zeta_k\}$ является невозрастающей. Значит, для гарантирования существования вещественных корней на всех итерациях алгоритма, достаточно выбрать $\gamma_- = \dfrac{3\mu n}{4} + \zeta_0$. Чтобы дополнительно не увеличивать количество точек в аппроксимации (и, соответственно, объём вычислений при глобальном поиске), при численной реализации использовался только один из корней уравнения (9).

Значения остальных параметров алгоритма были подобраны в ходе предварительных экспериментов с целью достижения баланса между качеством решений и временем вычислений (подробнее об этом — в следующем разделе). Эти настройки, наряду с ранее рассмотренными элементами СГП, определяют алгоритм глобального поиска (АГП).

В завершение раздела рассмотрим способ повышения вычислительной эффективности АГП. Напомним, что во всех предложенных вариантах аппроксимаций количество точек достаточно велико (n^3 и 2^{3n}). При росте n это приведет к значительному увеличению числа запусков локального поиска в процессе решения задачи, что существенно повлияет на время работы алгоритма. В связи с этим вместо полного перебора элементов множеств Dir_1 и Dir_2 предлагается выбирать из них несколько случайных элементов. Ранее вероятностные методы применялись для построения аппроксимации поверхности уровня функции $H(\cdot)$ в АГП в контексте генетических алгоритмов. Например, в работах [23, 24] построенные точки аппроксимации формировали начальную популяцию, к которой затем применялись операторы скрещивания и мутации. В данной работе предлагается альтернативный подход, основанный на простом выборе случайных элементов из множеств направлений.

Для выбора направления из множества Dir_1 случайным образом генерируются индексы $i_1, i_2, i_3 \in \{1, \dots, n\}$, $(e^{i_1}, e^{i_2}, e^{i_3})$ — выбранный элемент множества Dir_1. Процедура выбора направления из множества Dir_2 заключается в следующем:

- b^i инициализируется нулевыми значениями, $i = 1, 2, 3$;
- Генерируется случайное целое число $r \in \{1, \dots, n\}$ (общее количество единиц).
- Выполняется r итераций: на каждом шаге выбираются случайные индексы $q \in \{1, 2, 3\}$ и $i \in \{1, \dots, n\}$, i-я координата вектора b^q полагается равной 1.
- (b^1, b^2, b^3) — выбранный элемент множества Dir_2.

В дальнейших вычислительных экспериментах из каждого множества (Dir_1 и Dir_2) выбирается от 2 до 20 элементов, что существенно меньше их мощностей (n^3 и 2^{3n}). Предложенные способы сокращения количества точек аппроксимации обеспечивают гибкость АГП, поскольку в зависимости от требуемого времени работы можно варьировать требуемый в алгоритме объём вычислений.

5 Вычислительный эксперимент

Программная реализация алгоритмов выполнена на языке C++. Вычислительные эксперименты проводились на компьютере с процессором Intel Core i9-13900HK и 32 ГБ оперативной памяти. Тестирование проводилось на трех группах графов:

(1) **Группа 1.** Графы, полученные от компании, занимающейся проектированием и конфигурированием крупных беспроводных коммуникационных сетей. Представляют собой компоненты связности реальных графов мобильной связи. Данная группа содержит 11 примеров с числом вершин от 27 до 953 и числом рёбер от 120 до 9350.

(2) **Группа 2.** Синтетические графы, в которых рёбра между парами вершин добавлялись случайным образом до достижения заданного количества рёбер. Вес каждого ребра выбирался как случайная величина, равномерно распределённая на интервале $[0, 100]$. Данная группа содержит 5 примеров с числом вершин от 50 до 1000 и числом рёбер от 500 до 10000.

(3) **Группа 3.** Графы генерировались аналогично Группе 2, однако веса рёбер выбирались из нормального распределения с математическим ожиданием 50 и среднеквадратическим отклонением 20 с последующим усечением значений до диапазона $[0, 100]$. Данная группа содержит 10 примеров с числом вершин от 100 до 1000 и числом рёбер от 400 до 3000.

Все тестовые примеры, использованные в эксперименте, находятся в открытом доступе [25].

Результаты основного этапа численного эксперимента представлены в Таблицах 2–7 со следующими обозначениями:

- n и m — количество вершин и рёбер графа соответственно;
- VND и Gr — значения целевой функции, полученные методом спуска с чередующимися окрестностями (Variable Neighbor-hood Descent) и жадным алгоритмом соответственно;
- GSA — значение целевой функции, найденное АГП;
- t — время работы АГП в секундах;
- Q — количество случайных направлений, выбираемых из множеств Dir_1 и Dir_2;

- Δ_{VND} (Δ_{Gr}) — относительное улучшение решения по сравнению с алгоритмом VND (соответственно, жадным алгоритмом), вычисляемое как:

$$\Delta_{Alg} = \frac{Alg - GSA}{Alg} \times 100\%, \quad \text{где } Alg \in \{VND, Gr\};$$

- N_{loc} — количество запусков локального поиска, N_{lin} — количество решённых линеаризованных задач, N_{exit} — количество выходов из критических точек (итераций АГП $-$ 1).

Поскольку задача нахождения γ_+ сводится к решению задачи выпуклой максимизации, которая является вычислительно сложной [26, гл. 3], в работе использовано эмпирически подобранное значение $\gamma_+ = 1{,}5\,\gamma_-$.

Для сравнительного анализа использовались реализации жадного алгоритма и алгоритма VND, разработанные в рамках соревнования Challenge 5 международной конференции MOTOR 2025 [27]. Исходный код данных алгоритмов находится в открытом доступе [28]. Время работы жадного алгоритма и алгоритма VND для всех тестовых задач не превышало 1 секунды.

Анализ результатов для графов первой группы (таблицы 2–3) показывает, что АГП нашел решения с меньшим значением целевой функции по сравнению с жадным алгоритмом во всех рассмотренных задачах, а по сравнению с алгоритмом VND — в большинстве случаев. АГП с Dir_1 и АГП с Dir_2 показали близкие результаты.

Таблица 2. Результаты для группы 1 для Dir_1

n	m	VND	Gr	GSA	t, c	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
27	120	93.61	119.9	**89.4**	16	4.5	25.44	5	5	29	1051	3
53	293	471.13	602.87	**465.68**	24	1.16	22.76	5	5	29	1572	3
56	406	397.18	483.84	**346.37**	87	12.79	28.41	5	5	65	4199	7
108	566	**475.86**	923.76	487.82	267	-2.51	47.19	10	10	149	11308	8
114	891	2817.04	3092.86	**2346.75**	303	16.69	24.12	10	10	120	12725	5
283	2293	4664.86	5250.31	**4603.54**	2780	1.31	12.32	5	5	90	24361	6
313	2801	6506.26	8020.95	**6460.69**	1465	0.7	19.45	5	5	40	12144	4
360	3278	7082.93	8367.75	**6919.51**	1654	2.31	17.31	5	5	35	9953	3
688	7392	24801.2	29044.8	**24149.4**	4560	2.63	16.85	5	5	51	14648	3
689	6612	**13001.7**	16502.1	13615	2314	-4.72	17.5	5	2	33	7988	6
953	9350	**24616.3**	29898.5	24930	2052	-1.27	16.62	5	2	18	6323	4

Для графов второй (таблицы 4–5) группы АГП показал лучшие значения целевой функции, чем жадный алгоритм и VND, во всех тестовых задачах. Однако АГП с Dir_2 в одном случае уступил VND. АГП с Dir_1 показал меньшие значения целевой функции, чем АГП с Dir_2, почти во всех примерах. Однако время работы АГП с Dir_1 было больше, чем АГП с Dir_2, за исключением одного случая.

Таблица 3. Результаты для группы 1 для Dir_2

n	m	VND	Gr	GSA	t, с	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
27	120	93.61	119.9	**89.45**	59	4.44	25.4	10	10	114	3934	7
53	293	471.13	602.87	**459.71**	104	2.42	23.75	10	10	127	6049	4
56	406	397.18	483.84	**352.15**	105	11.34	27.22	10	10	89	5032	5
108	566	**475.86**	923.76	484.43	476	-1.8	47.56	20	20	199	16439	4
114	891	2817.04	3092.86	**2419.56**	197	14.11	21.77	10	10	77	8206	8
283	2293	4664.86	5250.31	**4645.73**	891	0.41	11.52	10	10	34	8664	2
313	2801	6506.26	8020.95	**6316.71**	1492	2.91	21.25	10	10	51	12649	3
360	3278	7082.93	8367.75	**6777.88**	2139	4.31	19	10	10	50	12986	3
688	7392	24801.2	29044.8	**24327.3**	1734	1.91	16.24	5	5	19	5541	4
689	6612	**13001.7**	16502.1	13775.6	1081	-5.95	16.52	5	5	16	3990	3
953	9350	24616.3	29898.5	**24580.1**	3725	0.15	17.79	3	10	30	11078	4

Таблица 4. Результаты для группы 2 для Dir_1

n	m	VND	Gr	GSA	t, с	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
50	500	3916.85	4162.88	**3814.97**	277	2.6	8.36	10	10	139	14880	4
100	1000	7174.04	8928.79	**7068.16**	779	1.48	20.84	7	7	103	14783	5
250	2500	17402.8	21736.5	**17095.3**	634	1.77	21.35	5	5	28	5941	1
500	5000	34798.1	43709.7	**32939.4**	3139	5.34	24.64	5	5	26	6937	0
1000	10000	68460	84612.1	**67798.4**	7479	0.97	19.87	2	2	10	3853	3

Таблица 5. Результаты для группы 2 для Dir_2

n	m	VND	Gr	GSA	t, с	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
50	500	3916.85	4162.88	**3814.97**	460	2.6	8.36	10	10	148	16179	6
100	1000	7174.04	8928.79	**7110.17**	631	0.89	20.37	7	7	66	9829	4
250	2500	17402.8	21736.5	**16477.2**	504	5.32	24.2	5	5	18	4380	2
500	5000	34798.1	43709.7	**32939.4**	1640	5.34	24.64	5	5	11	3643	0
1000	10000	**68460**	84612.1	68786.4	2304	-0.48	18.7	2	2	3	1194	0

В третьей группе (таблицы 6–7) АГП продемонстрировал существенно лучшие результаты по сравнению с жадным алгоритмом и VND для всех графов. АГП с Dir_1 по качеству решений оказался не хуже АГП с Dir_2 во всех задачах. Однако время работы АГП с Dir_1 стабильно превышало время работы АГП с Dir_2.

Таблица 6. Результаты для группы 3 для Dir_1

n	m	VND	Gr	GSA	t, с	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
100	400	1529.1	2095.83	**965.25**	242	36.87	53.94	5	5	63	409	6
200	700	2117.89	2888.01	**1815.36**	284	14.28	37.14	5	5	37	3693	5
300	1000	2780.97	4122.76	**2106.05**	874	24.27	48.92	5	5	42	4494	2
400	1500	4679.15	7153.89	**4212.53**	1382	9.97	41.12	5	5	47	6093	2
500	1500	2997.97	5255.46	**2326.25**	1464	22.41	55.74	5	5	49	5733	3
600	1800	3861.05	6004.01	**2954.31**	783	23.48	50.79	3	3	17	2169	1
700	2000	5069.25	7466.86	**2679.95**	6980	47.13	64.11	10	10	134	17248	1
800	2500	6049.47	9187.98	**4820.7**	1461	20.31	47.53	3	3	19	2637	2
900	3000	8579.41	12163.1	**6686.93**	751	22.06	45.02	2	2	7	997	2
1000	3000	6317.33	11030.3	**5880.63**	1107	6.91	46.69	2	2	6	908	1

Таблица 7. Результаты для группы 3 для Dir_2

n	m	VND	Gr	GSA	t, с	Δ_{VND}, %	Δ_{Gr}, %	M	Q	N_{loc}	N_{lin}	N_{exit}
100	400	1529.1	2095.83	**1104.99**	42	27.74	47.28	5	5	13	950	2
200	700	2117.89	2888.01	**1860.06**	78	12.17	35.59	5	5	13	1059	1
300	1000	2780.97	4122.76	**2247.24**	173	19.19	45.49	5	5	10	1145	2
400	1500	4679.15	7153.89	**4439.74**	415	5.12	37.94	5	5	13	1898	1
500	1500	2997.97	5255.46	**2620.76**	1484	12.58	55.74	8	20	50	5702	1
600	1800	3861.05	6004.01	**3116.24**	152	19.29	48.1	3	3	3	449	0
700	2000	5069.25	7466.86	**3141.27**	163	38.03	57.93	3	3	3	410	0
800	2500	6049.47	9187.98	**5165.81**	301	14.61	43.78	3	3	4	561	2
900	3000	8579.41	12163.1	**6820.98**	1606	20.5	43.92	7	7	16	2119	1
1000	3000	6317.33	11030.3	**6161.03**	448	2.47	44.14	2	2	3	374	0

Следует обратить внимание на параметр N_{exit}, который в некоторых случаях принимает нулевое значение. Это означает, что глобальный поиск не сработал: алгоритм остался в стартовой критической точке. Причиной может быть неудачный выбор множеств направлений Dir_1 и Dir_2 либо недостаточно большое значение параметра Q. С другой стороны, превосходство АГП над алгоритмом VND в этих случаях подтверждает высокую эффективность локального поиска. Фактически, итоговое качество решения здесь обеспечивается только процедурой локального поиска. А эффективнее всего ($N_{exit} \geq 3$ почти во всех случаях) АГП отработал для первой группы графов прикладной природы.

С целью проведения более глубокого сравнительного анализа предложенного подхода был осуществлён дополнительный вычислительный эксперимент. В нём алгоритм глобального поиска сопоставлялся с мультистарт версией алгоритма VND (в дальнейшем — MS-VND). Метод MS-VND заключается в многократном генерировании случайного начального решения с последующим применением к нему процедуры VND до тех пор, пока не будет исчерпан заданный лимит процессорного времени.

Для повышения итогового качества решений алгоритм глобального поиска также был модифицирован: после завершения специального метода локального поиска решение дополнительно улучшалось с помощью однократного запуска алгоритма VND. Данная модификация в дальнейшем обозначается как GSA+VND.

Оба алгоритма тестировались при одинаковом ограничении на время работы. В методе GSA+VND для каждого тестового примера использовалось то множество направлений (Dir_1 или Dir_2), которое показало наилучший результат по итогам предыдущего эксперимента.

Результаты сравнительного эксперимента представлены в таблицах 8–10. В таблицах используются следующие обозначения:

- t — одинаковый для обоих алгоритмов бюджет процессорного времени (в секундах);
- $GSA + VND$ — значение целевой функции, полученное модифицированным алгоритмом глобального поиска;
- $Dir \in \{1, 2\}$ — номер множества направлений (Dir_1 или Dir_2);
- $MS\text{-}VND$ — значение целевой функции, полученное мультистарт версией алгоритма VND.

Таблица 8. Результаты сравнения АГП с мультистартом VND при одинаковом бюджете времени. Группа 1

n	m	t, с	$GSA + VND$	N_{loc}	N_{lin}	N_{out}	M	Q	Dir	$MS\text{-}VND$
27	120	20	88	27	986	1	5	5	1	**87.08**
53	293	163	435.93	224	10272	6	10	10	2	**432.98**
56	406	103	334.91	53	3427	5	5	5	1	**323.06**
108	566	354	437.15	192	16090	6	10	10	2	**417.44**
114	891	1180	2296.96	213	23081	7	10	10	1	**2241.55**
283	2293	1032	4305.68	29	8428	3	5	5	1	**4201.62**
313	2801	1040	6069.64	40	10025	6	5	5	2	**5668.51**
360	3278	1036	6559.4	29	6792	2	5	5	2	**6217.46**
688	7392	4216	22923.1	40	11410	4	5	5	1	**22698.5**
689	6612	2009	12696.1	22	5253	3	5	2	1	**12409.4**
953	9350	4467	22845.1	36	13209	5	3	5	2	**22305.8**

Таблица 9. Результаты сравнения АГП с мультистартом VND при одинаковом бюджете времени. Группа 2

n	m	t, с	$GSA + VND$	N_{loc}	N_{lin}	N_{out}	M	Q	Dir	$MS\text{-}VND$
50	500	239	**3814.97**	118	12927	2	10	10	1	**3814.97**
100	1000	790	6896.41	81	12393	5	7	7	1	**6775.33**
250	2500	1246	15924.9	51	13789	3	5	5	2	**15108.1**
500	5000	3670	32475.6	26	9274	0	5	5	2	**30771.9**
1000	10000	5482	65132	6	2662	1	2	2	1	**63833.8**

Анализ результатов, представленных в таблицах 8–10, показывает, что мультистарт версия алгоритма VND превосходит модифицированный АГП в подавляющем большинстве тестовых графов. На одном примере алгоритму GSA+VND удалось найти строго лучшее решение, и ещё в одном случае результаты совпали. Данный факт объясняется колоссальной разницей во времени выполнения одной итерации: за отведённый бюджет времени MS-VND успевает

Таблица 10. Результаты сравнения АГП с мультистартом VND при одинаковом бюджете времени. Группа 3

n	m	t, с	$GSA + VND$	N_{loc}	N_{lin}	N_{out}	M	Q	Dir	$MS\text{-}VND$
100	400	233	965.25	63	4884	5	5	5	1	**916.39**
200	700	425	1519.84	44	4337	4	5	5	1	**1220.13**
300	1000	521	1888.22	30	3267	3	5	5	1	**1424.34**
400	1500	1056	3899.03	32	4094	1	5	5	1	**3214.3**
500	1500	1330	2027.22	36	4314	2	5	5	1	**1691.59**
600	1800	935	2655.37	17	2170	2	3	3	1	**2391.27**
700	2000	4980	**2545.47**	73	9565	1	7	7	1	2641.66
800	2500	2040	4311.01	22	2958	4	3	3	1	**4284.32**
900	3000	808	5905.09	6	858	1	2	2	1	**5713.63**
1000	3000	2449	4907.23	10	1620	2	2	2	1	**4723.76**

совершить миллионы перезапусков из новых случайных точек, в то время как модифицированный АГП выполняет лишь несколько десятков итераций. Тем не менее, даже проиграв по качеству, в большинстве случаев модифицированный АГП продемонстрировал близкие к рекордным результаты.

Обобщая полученные результаты, можно заключить, что в большинстве рассмотренных задач АГП позволяет находить решения с меньшим значением целевой функции по сравнению с VND, однако требует больших вычислительных ресурсов. В то же время мультистарт версия алгоритма VND за счёт многократного повторения итераций компенсирует недостаточную интенсификацию поиска широкой диверсификацией в пространстве решений. При этом число запусков локального поиска сравнительно небольшое, однако количество решённых линеаризованных задач значительно. Следовательно, основная трудоёмкость алгоритма обусловлена вычислительной сложностью (медленной сходимостью) процедуры локального поиска. В связи с этим одним из направлений дальнейших исследований может стать разработка специального метода локального поиска, учитывающего изначальную комбинаторную природу задачи. Например, в [17, 18] использование специальной С-процедуры продемонстрировало высокую эффективность при решении задач о клике.

6 Заключение

В работе исследован подход на основе DC оптимизации для решения задачи взвешенной 3-раскраски графа. Сформулирована DC постановка задачи и на основе теории глобального поиска А. С. Стрекаловского разработаны специальные методы локального и глобального поисков.

Проведён сравнительный анализ эффективности предложенного алгоритма с жадным алгоритмом, VND и мультистарт версией VND. Численные эксперименты подтвердили, что алгоритм глобального поиска позволяет находить решения более высокого качества по сравнению с жадным алгоритмом и VND. Однако эксперименты с фиксированным бюджетом времени показали, что из-за высокой вычислительной сложности АГП в большинстве случаев уступает мультистарт версии VND. Ключевой задачей дальнейшего развития предложенного подхода является повышение скорости работы АГП. Будущие исследования будут направлены, в частности, на разработку специального метода локального поиска, который учитывает дискретную природу задачи, и на использование новой СГП с варьированием параметра штрафа.

References

[1] R.M. Karp, *Reducibility Among Combinatorial Problems*, Complexity of Computer Computations, (1972), 85–103. DOI: https://doi.org/10.1007/978-1-4684-2001-2_9

[2] R.L. Brooks, *On colouring the nodes of a network*, Mathematical Proceedings of the Cambridge Philosophical Society, **37**:2 (1941), 194–197. DOI: https://doi.org/10.1017/S030500410002168X

[3] P. Erdős, *Graph Theory and Probability*, Canadian Journal of Mathematics, **11** (1959), 34–38. DOI: https://doi.org/10.4153/CJM-1959-003-9

[4] L. Lovász, *Kneser's conjecture, chromatic number, and homotopy*, Journal of Combinatorial Theory, Series A, **25**:3 (1978), 319–324. DOI: https://doi.org/10.1016/0097-3165(78)90022-5

[5] M.R. Garey, D.S. Johnson, L. Stockmeyer, *Some simplified NP-complete graph problems*, Theoretical Computer Science, **1**:3 (1976), 237–267. DOI: https://doi.org/10.1016/0304-3975(76)90059-1

[6] D.P. Dailey, *Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete*, Discrete Mathematics, **30**:3 (1980), 289–293. DOI: https://doi.org/10.1016/0012-365X(80)90236-8

[7] B. Ghaddar, M.F. Anjos, F. Liers, *A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem*, Annals of Operations Research, **188** (2011), 155–174. DOI: https://doi.org/10.1007/s10479-008-0481-4

[8] R. Horst, H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed., Springer, Berlin, Heidelberg, 2013. DOI: https://doi.org/10.1007/978-3-662-03199-5

[9] A.S. Strekalovsky, *Elements of Nonconvex Optimization*, Nauka, Novosibirsk, 2003 (in Russian).

[10] A.S. Strekalovsky, *On Solving Optimization Problems with Hidden Nonconvex Structures*, Optimization in Science and Engineering (Ed. by T.M. Rassias, C.A. Floudas, S. Butenko), Springer, N.Y., (2014), 465–502. DOI: https://doi.org/10.1007/978-1-4939-0808-0_23

[11] A.S. Strekalovsky, *New global optimality conditions in a problem with d.c. constraints*, Trudy Instituta Matematiki i Mekhaniki UrO RAN, **25**:1 (2019), 245–261 (in Russian). DOI: https://doi.org/10.21538/0134-4889-2019-25-1-245-261

[12] A.S. Strekalovsky, *Minimizing sequences in a constrained DC optimization problem*, Proc. Steklov Inst. Math., **323**: Suppl 1 (2023), S255–S278. DOI: https://doi.org/10.1134/S0081543823060214

[13] A.S. Strekalovsky, A.V. Orlov, *Linear and Quadratic-Linear Bilevel Optimization Problems*, Izd-vo SO RAN, Novosibirsk, 2019 (in Russian).

[14] A.S. Strekalovsky, A.V. Orlov, *Global Search for Bilevel Optimization with Quadratic Data*, Bilevel optimization: advances and next challenges (Ed. by S. Dempe, A. Zemkoho), Springer, Cham, **161** (2020), 313–334. DOI: https://doi.org/10.1007/978-3-030-52119-6_11

[15] A.V. Orlov, *The Global Search Theory Approach to the Bilevel Pricing Problem in Telecommunication Networks*, Computational Aspects and Applications in Large Scale Networks (Ed. by V.A. Kalyagin et. al.), Springer, Cham, (2018), 57–73. DOI: https://doi.org/10.1007/978-3-319-96247-4_5

[16] A.V. Orlov, S. Batbileg, *Oligopolistic banking sector of Mongolia and three-person polymatrix games*, The Bulletin of Irkutsk State University. Series Mathematics, **11** (2015), 80–95 (in Russian).

[17] T.V. Gruzdeva, A.S. Strekalovsky, *Clique Problems and Nonconvex Optimization*, Nauka, Novosibirsk, 2014 (in Russian).

[18] T.V. Gruzdeva, *On a Continuous Approach for the Maximum Weighted Clique Problem*, Journal of Global Optimization, **56**:3 (2013), 971–981. DOI: https://doi.org/10.1007/s10898-012-9885-4

[19] J. Gui, Z. Jiang, S. Gao, *PCI Planning Based on Binary Quadratic Programming in LTE/LTE-A Networks*, IEEE Access, **7** (2019), 203–214. DOI: https://doi.org/10.1109/ACCESS.2018.2885313

[20] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New York, 2006. DOI: https://doi.org/10.1007/978-0-387-40065-5

[21] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004. DOI: https://doi.org/10.1017/CBO9780511804441

[22] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., McGraw-Hill, New York, 1976.

[23] A.V. Orlov, *Numerical study of a hybrid global search algorithm in hexamatrix games*, BSU Bulletin. Mathematics, Informatics, :2 (2023), 14–29 (in Russian). DOI: https://doi.org/10.18101/2304-5728-2023-2-14-29

[24] A.V. Orlov, *Hybrid Global Search Algorithm with Genetic Blocks for Solving Hexamatrix Games*, The Bulletin of Irkutsk State University. Series Mathematics, **41** (2022), 40–56. DOI: https://doi.org/10.26516/1997-7670.2022.41.40

[25] *A set of test graphs for a numerical experiment*, [Electronic resource]. URL: https://disk.yandex.ru/d/nLmRgg11N5bPlA

[26] R. Horst, P.M. Pardalos, N.V. Thoai, *Introduction to Global Optimization*, Springer, New York, 2000.

[27] *Mathematical Optimization Theory and Operations Research (MOTOR 2025). Challenge 5*, [Electronic resource]. URL: http://old.math.nsc.ru/conference/motor/2025/challenges.html

[28] *Repository of solutions for the 3-coloring problem (Challenge 5)*, [Electronic resource]. URL: https://github.com/prv-rus/3coloring

Georgii Alekseevich Zhukov
Novosibirsk State University,
Pirogova str., 1,
630090, Novosibirsk, Russia
Email address: g.zhukov@g.nsu.ru

Andrey Vasilievich Orlov
Matrosov Institute for System Dynamics and Control Theory SB RAS,
Lermontov str., 134,
664033, Irkutsk, Russia
Email address: anor@icc.ru