

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»

На правах рукописи

СОВРАСОВ ВЛАДИСЛАВ ВАЛЕРЬЕВИЧ

**Разработка и исследование способов распределения ресурсов в
параллельных алгоритмах глобальной оптимизации**

Направление подготовки
09.06.01 – Информатика и вычислительная
техника

Направленность подготовки
05.13.18 – Математическое моделирование,
численные методы и комплексы программ

**Научный доклад об основных результатах
научно-квалификационной работы (диссертации)**

Научный руководитель:
Д. Т. Н., доцент,
Баркалов К. А.

Нижний Новгород
2021

Содержание

Введение	2
1 Постановки задачи глобальной оптимизации, обзор существующих методов решения	5
1.1 Задача глобальной оптимизации с функциональными ограничениями-неравенствами	5
1.2 Совместное решение серии задач глобальной оптимизации	6
1.3 Различные подходы к решению задач глобальной оптимизации	6
1.3.1 Стохастические методы	6
1.3.2 Детерминированные методы	7
2 Подходы к сравнению алгоритмов глобальной оптимизации	9
3 Методы редукции размерности	12
3.1 Инъективная развёртка	12
3.2 Сдвиговые развёртки	13
3.3 Вращаемые развёртки	15
3.4 Неинъективная развёртка	15
3.5 Гладкая развёртка	16
3.6 Сравнение развёрток	16
4 Сравнение методов глобальной оптимизации	17
4.1 Методы глобальной оптимизации для сравнения	17
4.1.1 Контроль параметра надёжности в AGS	18
4.2 Результаты численных экспериментов	19
4.3 Итоги сравнения методов	21
5 Алгоритм, решающий множество задач	22
5.1 Описание алгоритма	22
5.2 Условия сходимости	22
5.3 Результаты численных экспериментов	23
5.4 Итоги	25
Заключение	25
Список публикаций в изданиях, индексируемых Scopus	26

Введение

Актуальность темы исследования

Задачи нелинейной глобальной оптимизации встречаются в различных прикладных областях (например, поиск оптимального управления или гиперпараметров метода машинного обучения), и традиционно считаются одними из самых трудоёмких среди оптимизационных задач. Их сложность экспоненциально растёт в зависимости от размерности пространства поиска. Зачастую глобальная оптимизация при размерности пространства в 10 переменных сложнее, чем локальная оптимизация в существенно многомерном пространстве. Для последней может оказаться достаточно применения простейшего метода градиентного спуска или эвристических алгоритмов поиска по шаблону, в то время как чтобы *гарантированно* отыскать глобальный оптимум, методам оптимизации приходится накапливать информацию о поведении целевой функции во всей области поиска.

На практике в роли параметров для оптимизации могут выступать параметры математической модели, например, при решении задачи поиска оптимального управления или при настройке алгоритма машинного обучения под конкретные данные. В этих случаях о поведении целевой функции (выпуклость, наличие производных) ничего неизвестно, т.е. метод оптимизации работает с функцией, представляющей собой чёрный ящик. Кроме того, подобный чёрный ящик, как правило, имеет высокую вычислительную сложность, во много раз превосходящую вычислительную сложность решающих правил метода оптимизации.

Широко представленным классом методов, решающих задачи с трудоёмкой целевой функцией типа чёрный ящик, являются алгоритмы липшицевой глобальной оптимизации. Они требуют, чтобы целевая функция удовлетворяла условию Липшица в области поиска, что является более сильным свойством, чем непрерывность, но более слабым, чем предположение о наличии непрерывных частных производных. В то же время, этого достаточно, чтобы гарантировать глобальную сходимость при условии выбора подходящих параметров применяемого метода оптимизации.

В задачах глобальной оптимизации область допустимых значений целевой функции также может быть определена с помощью условий, например в виде функциональных ограничений типа неравенство. Это вносит дополнительную сложность в процесс решения задач, требуя создания схем учёта ограничений. Ещё более сложными являются задачи многокритериальной условной многокритериальной оптимизации и смешанного программирования. В последнем случае как целевая функция, так и функциональные ограничения зависят и от непрерывных, и от дискретных переменных. Существуют методы скаляризации многокритериальных задач. При этом порождается множество скалярных задач, каждую из которых необходимо решить. Задачи смешанного программирования в некоторых случаях тоже можно свести к серии задач, каждая из которых соответствует одному дискретному параметру. Решение серии задач услов-

ной глобальной задач при ограниченных вычислительных ресурсах налагает дополнительные требования на метод оптимизации: помимо поиска глобального экстремума необходимо распределять вычислительные ресурсы так, чтобы сразу во всех решаемых задачах положение глобального экстремума было оценено примерно с одинаковым качеством.

Многообразие постановок задач и методов их решения говорит о том, что глобальная оптимизация является развивающимся и актуальным направлением исследований в теории оптимизации, широко применяемым на практике.

Цель квалификационной работы

Целью квалификационной работы является исследование и разработка последовательных и параллельных алгоритмов для решения задач глобальной оптимизации, в частности разработка метода, который бы эффективно распределял ресурсы между задачами при решении множества задач условной оптимизации.

Для достижения поставленной цели, необходимо решить следующие задачи:

1. Сделать обзор современных методов глобальной оптимизации;
2. Сравнить актуальные методы между собой;
3. Сформулировать собственную модификацию алгоритма, решающего множество задач условной глобальной оптимизации, получить условия сходимости;
4. Оценить качество работы полученного метода на большом наборе разнообразных тестовых задач с ограничениями.

Объект исследования – программные системы для решения задач глобальной оптимизации в различных постановках.

Предмет исследования – методы решения задач условной глобальной оптимизации, теория их сходимости.

Обоснование специальности

Область исследования соответствует следующим пунктам паспорта специальности 05.13.18 – «Математическое моделирование, численные методы и комплексы программ»: 4. Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента; 3. Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий.

Научная новизна

В процессе работы была, разработана модификация метода для решения множества задач условной оптимизации, поддерживающая функциональные ограничения типа неравенство. Новизна работы заключается в следующем:

1. Предложена модификация алгоритма AGS, которая менее чувствительна к параметрам и сходится так же быстро, как AGS с заранее подобранными под класс задач параметрами;
2. Реализована поддержка нелинейных ограничений в алгоритме, решающем множество задач глобальной оптимизации в совокупности и распределяющего свои ресурсы так, чтобы обеспечивать равномерную сходимость во всех задачах. Доказано теорема о достаточных условиях сходимости полученного метода. Свойство равномерной сходимости проверено с помощью численного эксперимента.

Обоснованность и достоверность результатов

Достоверность результатов, полученных в ходе выполнения квалификационной работы, подтверждается тем, что различные части работы были опубликованы в рецензируемых сборниках трудов российских и международных конференций. Кроме того, программная реализация метода AGS-AR прошла процедуру ревью и была включена в состав популярной библиотеки алгоритмов нелинейной оптимизации NLOpt.

Публикации. Основные результаты по теме квалификационной работы изложены в трёх публикациях [1—3], каждая из которых входит в сборник трудов конференций, индексируемый в базе данных Scopus.

Структура и объемы работы

Квалификационная работа включает в себя введение, три главы, заключение и список литературы. Полный объем квалификационной работы составляет 38 страниц, включая 11 рисунков и 6 таблиц. Список литературы содержит 49 позиций.

Личный вклад автора

Личный вклад автора состоит в следующем:

1. Разработка методов, теоретический анализ их сходимости, их программная реализация.
2. Разработка плана вычислительных экспериментов, их проведение, анализ результатов.
3. Подготовка публикаций по проделанной работе.

1. Постановки задачи глобальной оптимизации, обзор существующих методов решения

1.1. Задача глобальной оптимизации с функциональными ограничениями-неравенствами

В рамках данной работы рассматривается следующая постановка задачи глобальной оптимизации: найти глобальный минимум N -мерной функции $\varphi(y)$ в гиперинтервале $D = \{y \in \mathbf{R}^N : a_i \leq x_i \leq b_i, 1 \leq i \leq N\}$. Для построения оценки глобального минимума по конечному количеству вычислений значения функции требуется, чтобы скорость изменения $\varphi(y)$ в D была ограничена. В качестве такого ограничения как правило принимается условие Липшица.

$$\varphi(y^*) = \min\{\varphi(y) : y \in D\} \quad (1)$$

$$|\varphi(y_1) - \varphi(y_2)| \leq L\|y_1 - y_2\|, y_1, y_2 \in D, 0 < L < \infty \quad (2)$$

Существуют различные методы, решающие рассмотренную многомерную задачу напрямую, а также эффективные методы решения одномерных задач. В данной работе рассматривается одномерный метод, который применяется совместно со схемой редукции размерности. Классической схемой редукции размерности исходной задачи для алгоритмов глобальной оптимизации является использование разверток — кривых, заполняющих пространство.

$$\{y \in \mathbf{R}^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\} \quad (3)$$

Отображение вида (3) позволяет свести задачу в многомерном пространстве к решению одномерной ценой ухудшения ее свойств. В частности, одномерная функция $\varphi(y(x))$ является не Липшицевой, а Гельдеровой:

$$|\varphi(y(x_1)) - \varphi(y(x_2))| \leq H|x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0; 1],$$

где константа Гельдера H связана с константой Липшица L соотношением

$$H \leq 2L\sqrt{N+3} \quad (4)$$

Область допустимых аргументов целевой функции также может быть задана с помощью функциональных ограничений, что значительно усложняет задачу. Постановка задачи глобальной оптимизации в этом случае будет иметь следующий вид:

$$\varphi(y^*) = \min\{\varphi(y) : y \in D, g_j(y) \leq 0, 1 \leq j \leq m\} \quad (5)$$

Обозначим $g_{m+1}(y) = \varphi(y)$. Далее будем предполагать, что все функции $g_k(y)$, $1 \leq k \leq m+1$ удовлетворяют условию Липшица на гиперинтервале D .

1.2. Совместное решение серии задач глобальной оптимизации

Также будем интересоваться совместным решением серии из q задач вида (5):

$$\min \{ \varphi_1(y), y \in D_1 \}, \min \{ \varphi_2(y), y \in D_2 \}, \dots, \min \{ \varphi_q(y), y \in D_q \}. \quad (6)$$

Требуется получить приближённое решение во всех задачах их серии, потратив на это как можно меньшее количество испытаний. При решении серии задач (6) возникает проблема оптимального распределения ресурсов метода между ними. Будем считать, что метод оптимально распоряжается ресурсами, если при его остановке на некоторой итерации решение всех задач получено с одинаковой точностью. Такого свойства можно добиться, потребовав от метода равномерной сходимости на всём множестве задач:

$$\exists \varepsilon > 0 : \forall s > 1, \forall i, j \in \{1, \dots, q\} \frac{\|\tilde{y}_i(s)^* - y_i^*\|_\infty}{\|\tilde{y}_j(s)^* - y_j^*\|_\infty} \leq \varepsilon, \quad (7)$$

где s это количество итераций метода оптимизации, $\tilde{y}_i(s)^*$ это приближение к решению, полученное методом в задаче i из множества (6) на итерации s .

1.3. Различные подходы к решению задач глобальной оптимизации

Одним из возможных способов классификации алгоритмов глобальной оптимизации является разделение их на детерминированные и стохастические. Стохастические алгоритмы в том или ином виде используют в своей работе случайные переменные и поэтому результат их работы, вообще говоря, меняется от запуска к запуску. Детерминированные методы работают по заранее установленным решающим правилам, результат их работы определяется только задачей и, возможно, некоторыми настраиваемыми параметрами.

1.3.1. Стохастические методы

В последнее время стали популярны различные стохастические алгоритмы глобальной оптимизации, прежде всего эволюционные. Они имеют довольно простую структуру, позволяют решать задачи большой размерности, однако обеспечивают глобальную сходимость только в вероятностном смысле. Общим недостатком стохастических методов является их зависимость от случайных переменных и, в силу характера сходимости, начальной точки, с которой начинается оптимизация (она может также генерироваться случайно). Примерами неэволюционных стратегий стохастической оптимизации являются случайный поиск, метод имитации отжига и муравьиный алгоритм. Стоит отметить, что стохастические методы поиска не требуют дифференцируемости или даже непрерывности от целевой функции задачи оптимизации, могут работать с зашумлёнными функциями.

1.3.2. Детерминированные методы

Стохастические методы обладают одним концептуальным недостатком – нельзя гарантировать их сходимость при каждом конкретном запуске процесса оптимизации. Для детерминированных методов есть возможность построить теорию сходимости, наложив ограничения на скорость изменения целевой функции в задаче (1). Если этого не сделать, то по любому конечному количеству значений целевой функции невозможно построить её модель с гарантированной нижней оценкой. Известным и широко представленным классом детерминированных методов являются методы липшицевой оптимизации, предполагающие Липшиц-непрерывность целевой функции в задаче оптимизации. В одномерном случае к таким методам можно отнести семейство характеристических методов, в которое входит, например, метод Пиявского и AGS. Для их применения в случае многомерных задач используются схемы редукции размерности. Методы диагональных или симплексных разбиений не требуют использования таких схем, однако более сложны в реализации, т.к. требуется эффективно хранить и модифицировать множество многомерных гиперинтервалов или симплексов.

Индексный алгоритм глобального поиска В качестве примера детерминированного характеристического метода приведём индексный алгоритм глобального поиска IAGS.

Принимая во внимание схему редукции размерности (3), будем при описании метода считать, что требуется найти глобальный минимум функции $\varphi(x)$, $x \in [0; 1]$, удовлетворяющей условию Гёльдера, при ограничениях $g_j(x)$, также удовлетворяющих этому условию на интервале $[0; 1]$.

Рассматриваемый индексный алгоритм глобального поиска (IAGS) для решения одномерной задачи (5) предполагает построение последовательности точек x_k , в которых вычисляются значения минимизируемой функции или ограничений $z_k = g_s(x_k)$. Для учета последних используется индексная схема. Пусть $Q_0 = [0; 1]$. Ограничение, имеющее номер j , выполняется во всех точках области

$$Q_j = \{x \in [0; 1] : g_j(x) \leq 0\},$$

которая называется допустимой для этого ограничения. При этом допустимая область D исходной задачи определяется равенством: $D = \cap_{j=0}^m Q_j$. Испытание в точке $x \in [0; 1]$ состоит в последовательном вычислении значений величин $g_1(x), \dots, g_\nu(x)$, где значение индекса ν определяется условиями: $x \in Q_j, 0 \leq j < \nu, x \notin Q_\nu$. Выявление первого нарушенного ограничения прерывает испытание в точке x . В случае, когда точка x допустима, т.е. $x \in D$ испытание включает в себя вычисление всех функций задачи. При этом значение индекса принимается равным величине $\nu = m + 1$. Пара $\nu = \nu(x), z = g_\nu(x)$, где индекс ν лежит в границах $1 \leq \nu \leq m + 1$, называется результатом испытания в точке x .

Такой подход к проведению испытаний позволяет свести исходную задачу с функциональ-

ными ограничениями к безусловной задаче минимизации разрывной функции:

$$\psi(x^*) = \min_{x \in [0;1]} \psi(x),$$

$$\psi(x) = \begin{cases} g_\nu(x)/H_\nu & \nu < M \\ (g_M(x) - g_M^*)/H_M & \nu = M \end{cases}$$

Здесь $M = \max \{\nu(x) : x \in [0; 1]\}$, а $g_M^* = \min \{g_M(x) : x \in \cap_{i=0}^{M-1} Q_i\}$. В силу определения числа M , задача отыскания g_M^* всегда имеет решение, а если $M = m + 1$, то $g_M^* = \varphi(x^*)$. Дуги функции $\psi(x)$ гёльдеровы на множествах $\cap_{i=0}^j Q_i$, $0 \leq j \leq M - 1$ с константой 1, а сама $\psi(x)$ может иметь разрывы первого рода на границах этих множеств. Несмотря на то, что значения констант Гёльдера H_k и величина g_M^* заранее неизвестны, они могут быть оценены в процессе решения задачи.

Множество троек $\{(x_k, \nu_k, z_k)\}$, $1 \leq k \leq n$ составляет поисковую информацию, накопленную методом после проведения n шагов.

На первой итерации метода испытание проводится в произвольной внутренней точке x_1 интервала $[0; 1]$. Индексы точек 0 и 1 считаются нулевыми, значения z в них не определены. Пусть выполнено $k \geq 1$ итераций метода, в процессе которых были проведены испытания в k точках x_i , $1 \leq i \leq k$. Тогда точка x^{k+1} поисковых испытаний следующей $(k + 1)$ -ой итерации определяются в соответствии с правилами:

Шаг 1. Перенумеровать точки множества $X_k = \{x^1, \dots, x^k\} \cup \{0\} \cup \{1\}$, которое включает в себя граничные точки интервала $[0; 1]$, а также точки предшествующих испытаний, нижними индексами в порядке увеличения значений координаты, т.е.

$$0 = x_0 < x_1 < \dots < x_{k+1} = 1 \quad (8)$$

и сопоставить им значения $z_i = g_\nu(x_i)$, $\nu = \nu(x_i)$, $i = \overline{1, k}$.

Шаг 2. Для каждого целого числа ν , $1 \leq \nu \leq m + 1$ определить соответствующее ему множество I_ν нижних индексов точек, в которых вычислялись значения функций $g_\nu(x)$:

$$I_\nu = \{i : \nu(x_i) = \nu, 1 \leq i \leq k\}, 1 \leq \nu \leq m + 1,$$

определить максимальное значение индекса $M = \max\{\nu(x_i), 1 \leq i \leq k\}$.

Шаг 3. Вычислить текущие оценки для неизвестных констант Гёльдера:

$$\mu_\nu = \max \left\{ \frac{|g_\nu(x_i) - g_\nu(x_j)|}{(x_i - x_j)^{\frac{1}{N}}} : i, j \in I_\nu, i > j \right\}. \quad (9)$$

Если множество I_ν содержит менее двух элементов или если значение μ_ν оказывается равным нулю, то принять $\mu_\nu = 1$.

Шаг 4. Для всех непустых множеств I_ν , $\nu = \overline{1, M}$ вычислить оценки

$$z_\nu^* = \begin{cases} \min\{g_\nu(x_i) : x_i \in I_\nu\} & \nu = M \\ -\varepsilon_\nu & \nu < M \end{cases}, \quad (10)$$

где вектор с неотрицательными координатами $\varepsilon_R = (\varepsilon_1, \dots, \varepsilon_m)$ называется вектором резервов.

Шаг 5. Для каждого интервала $(x_{i-1}; x_i)$, $1 \leq i \leq k$ вычислить характеристику

$$R(i) = \begin{cases} \Delta_i + \frac{(z_i - z_{i-1})^2}{(r_\nu \mu_\nu)^2 \Delta_i} - 2 \frac{z_i + z_{i-1} - 2z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_i) = \nu(x_{i-1}) \\ 2\Delta_i - 4 \frac{z_{i-1} - z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_{i-1}) > \nu(x_i) \\ 2\Delta_i - 4 \frac{z_i - z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_i) > \nu(x_{i-1}) \end{cases} \quad (11)$$

где $\Delta_i = (x_i - x_{i-1})^{\frac{1}{N}}$. Величины $r_\nu > 1$, $\nu = \overline{1, m}$ являются параметрами алгоритма. От них зависят произведения $r_\nu \mu_\nu$, используемые при вычислении характеристик в качестве оценок неизвестных констант Гёльдера.

Шаг 6. Выбрать наибольшую характеристику:

$$t = \arg \max_{1 \leq i \leq k+1} R(i) \quad (12)$$

Шаг 7. Провести очередное испытание в середине интервала $(x_{t-1}; x_t)$, если индексы его конечных точек не совпадают: $x^{k+1} = \frac{1}{2}(x_t + x_{t-1})$. В противном случае провести испытание в точке

$$x^{k+1} = \frac{1}{2}(x_t + x_{t-1}) - \operatorname{sgn}(z_t - z_{t-1}) \frac{|z_t - z_{t-1}|^n}{2r_\nu \mu_\nu^n}, \nu = \nu(x_t) = \nu(x_{t-1}),$$

а затем увеличить k на 1.

Алгоритм прекращает работу, если выполняется условие $\Delta_t \leq \varepsilon$, где $\varepsilon > 0$ есть заданная точность. В качестве оценки глобально-оптимального решения выбираются значения

$$\varphi_k^* = \min_{1 \leq i \leq k} \varphi(x_i), x_k^* = \arg \min_{1 \leq i \leq k} \varphi(x_i) \quad (13)$$

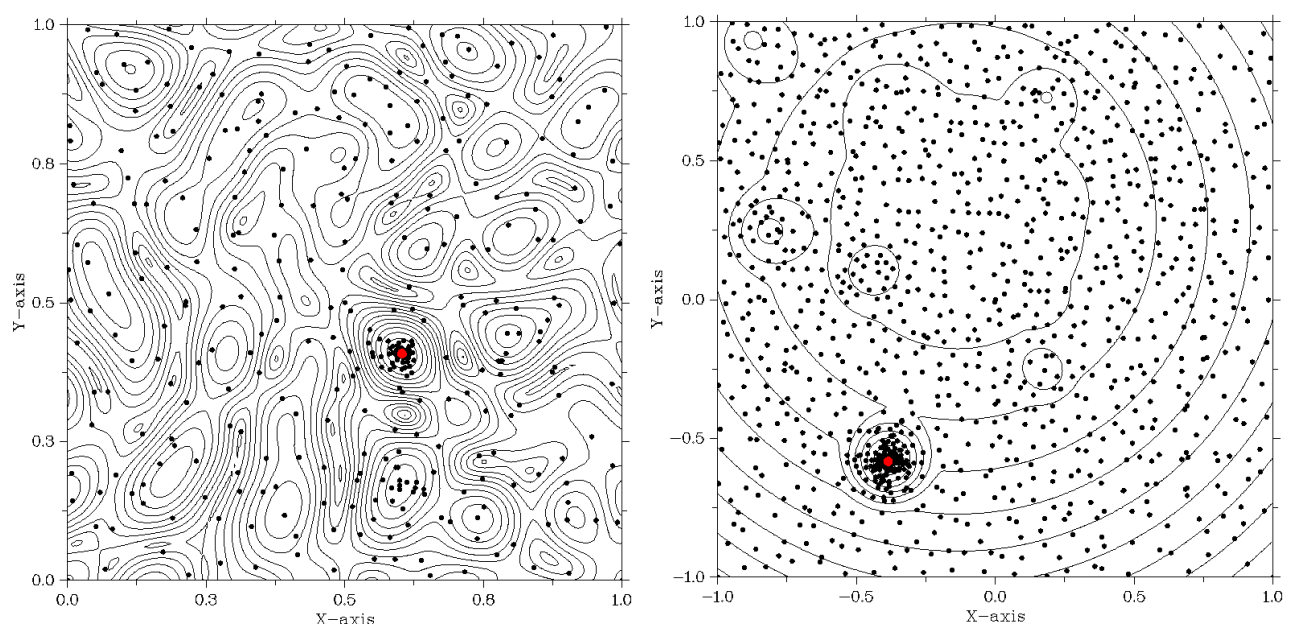
Отметим, что путём увеличения r и при $\varepsilon = 0$ можно гарантированно добиться сходимости, если все функции задачи (5) удовлетворяют условию Липшица на гиперкубе из (1).

2. Подходы к сравнению алгоритмов глобальной оптимизации

Основным подходом к сравнению между собой методов оптимизации является запуск этих методов на наборах тестовых задач и последующее сравнение собранных в процессе запуска метрик, характеризующие различные аспекты работы алгоритмов. Исключая некоторые специальные случаи, когда требуется определить метод, лучше других себя ведущий на небольшом наборе специфичных задач или вообще на одной задаче с параметром. Существуют два способа построить тестовый набор задач: собрать небольшое множество известных из литературы тестовых задач или использовать рандомизированные генераторы задач, позволяющие генерировать потенциально любое количество задач. Известных тестовых задач мало, поэтому используя только их не всегда можно достаточно полно выявить особенности работы различных методов. В то же время генераторы задач позволяют создать любое их количество, однако полученные задачи могут обладать скрытой общей структурой, благоприятной для некоторых

из тестируемых методов, либо обладать малой вариативностью. Поскольку в рамках работы рассматриваются алгоритмы для решения существенно многоэкстремальных задач, будем производить все сравнения на задачах, полученных с помощью нескольких генераторов достаточно таких задач. Это сгладит проблему потенциально низкой вариативности задач, полученных одним генератором. Также в одном специальном случае рассмотрим решение одной известной задачи из литературы.

В рамках данной работы будем использовать два механизма генерации тестовых задач. Тестовые задачи, получаемые данными механизмами, имеют абсолютно разную природу (см. линии уровня на Рис. 1), что увеличивает достоверность результатов сравнения алгоритмов оптимизации.



(а) Пример линий уровня функции, порождённой F_{GR} (б) Пример линий уровня функции, порождённой GKLS

Рис. 1. Линии уровня и точки испытаний ИАГП в двух синтетических задачах без ограничений

Обозначим набор двумерных задач, полученный с помощью ненастраиваемого генератора как F_{GR} . Генератор F_{GR} не позволяет контролировать размерность и сложность задач, а также количество локальных экстремумов. Каждая тестовая задача определяется формулой:

$$\varphi(y) = \sqrt{\left(\sum_{i=1}^7 \sum_{j=1}^7 A_{ij} g_{ij}(y) + B_{ij} h_{ij}(y)\right)^2 + \left(\sum_{i=1}^7 \sum_{j=1}^7 C_{ij} g_{ij}(y) - D_{ij} h_{ij}(y)\right)^2}$$

где

$$y \in [0; 1]^2,$$

$$g_{ij} = \sin(i\pi y_1) \sin(j\pi y_2),$$

$$h_{ij} = \cos(i\pi y_1) \cos(j\pi y_2),$$

где коэффициенты $A_{ij}, B_{ij}, C_{ij}, D_{ij}$ генерируются случайно с равномерным в интервале $[-1; 1]$ распределением. Получаемые функции являются существенно многоэкстремальными. В данной работе будем использовать 100 функций, сгенерированных с помощью F_{GR} .

Генератор GKLS позволяет получать тестовые задачи заданной размерности с заданным количеством локальных экстремумов. Также этот генератор позволяет варьировать сложность задачи, изменяя размер области притяжения глобального минимума. Это достигается за счёт модификации параболоида $g(x) = \|x - T\| + t$ в шаровых окрестностях некоторых случайно сгенерированных точек $M_i, i = \overline{1, m}$. В точках M_i располагаются локальные минимумы со значениями, превосходящими значение $g(T) = t$.

Перечисленные выше генераторы порождают задачи без нелинейных ограничений, поэтому в ходе работы была использована система GCGen¹, которая позволяет генерировать задачи с ограничениями на основе произвольных нелинейных функций. При использовании этой системы совместно с генераторами, таким как GKLS, порождённые ими функции выступают как в роли целевой функции, так и в роли ограничений.

В квалификационной работе рассматривается среднее количество испытаний, затраченное на решение одной задачи, и количество решённых задач как характеристики эффективности метода оптимизации на заданном классе задач. Чем меньше среднее количество испытаний на задачу, тем быстрее метод сходится к решению, а значит и меньше обращается к потенциально трудоёмкой процедуре вычисления ограничений и целевой функции задачи. Количество решённых задач характеризует надёжность метода. Чтобы сделать рассматриваемые характеристики независимыми друг от друга, будем вычислять среднее количество испытаний, принимая во внимание только решённые задачи.

Среднее число испытаний на задачу в некоторых случаях не позволяет получить полной картины о поведении численного метода оптимизации на рассматриваемом множестве задач. Например, если метод тратит много ресурсов на решение небольшого подмножества задач из выборки, то это невозможно понять, глядя только на среднее число испытаний. Как дополнительный критерий сравнения методов также использована операционная характеристика. Операционная характеристика представляет из себя кривую на плоскости (K, P) , где K это среднее количество испытаний, произведённое методом до выполнения критерия остановки, а P – доля задач из выборки, решённых не более чем за K испытаний. Если при заданном K операционная характеристика одного метода лежит выше, чем операционная характеристика другого, то это означает, что при фиксированных затратах на поиск, первый метод с большей вероятностью найдёт решение. При заданном P если характеристика одного метода лежит левее, чем характеристика другого, то при одинаковой надёжности, первый метод в среднем потратит меньше испытаний на поиск, чем второй.

¹Исходный код системы доступен по ссылке <https://github.com/UNN-ITMM-Software/GCGen>

3. Методы редукции размерности

В разделе 3 рассматриваются более подробно метод редукции размерности с помощью кривых Пеано, заполняющих пространство, и сравним между собой разные модификации различных кривых. Кроме прямой редукции размерности с помощью кривых типа Пеано, существует и способ, сводящий одну задачу оптимизации к множеству вложенных задач оптимизации: многошаговая схема:

$$\min_{y \in D} \varphi(y) = \min_{y_1 \in [a_1, b_1]} \dots \min_{y_N \in [a_N, b_N]} \varphi(y_1, \dots, y_N). \quad (14)$$

В рамках многошаговой схемы каждое вычисление целевой функции по переменной y_1 во внешней задаче оптимизации влечёт за собой проведение ещё $N - 1$ вложенной оптимизации. Недостатками данной схемы является её низкая экономичность в плане количества обращений к целевой функции и отсутствие теоретической гарантии сходимости IAGS при наличии функциональных ограничений (в некоторых случаях целевые функции во вложенных задачах перестают удовлетворять условию Липшица). В качестве обобщения многошаговой схемы было предложено вложенную оптимизацию вести не по одиночным переменным, а по блокам переменных. К многомерным вложенным задачам в таком случае можно применить редукцию размерности с помощью кривых типа Пеано, поэтому выявление наиболее эффективного типа кривой имеет смысл и для ускорения сходимости в случае блочной многошаговой схемы.

3.1. Инъективная развёртка

Для редукции размерности задач оптимизации в рамках информационно-статистического подхода применяются кривые типа Пеано, заполняющие пространство (развёртки). Такие кривые отображают отрезок $[0, 1]$ на N -мерный гиперкуб D .

После редукции размерности исходная многомерная задача (1) преобразуется в одномерную задачу следующего вида:

$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0, 1]\}. \quad (15)$$

Стоит отметить, что подстановка развёртки в Липшицеву функцию из (1) порождает одномерную функцию $\varphi(y(x))$, которая удовлетворяет условию Гёльдера:

$$|\varphi(y(x_1)) - \varphi(y(x_2))| \leq H|x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0, 1], \quad (16)$$

где константа H удовлетворяет неравенству $H \leq 2L\sqrt{N+3}$, L – константа Липшица из (2), и N – размерность задачи (1).

На Рис. 2 приведена одномерная функция, полученная после применения развёртки к параболоиду вращения $\varphi(y) = y_1^2 + y_2^2$. Точка с координатами $(0; 0)$ имеет несколько одномерных

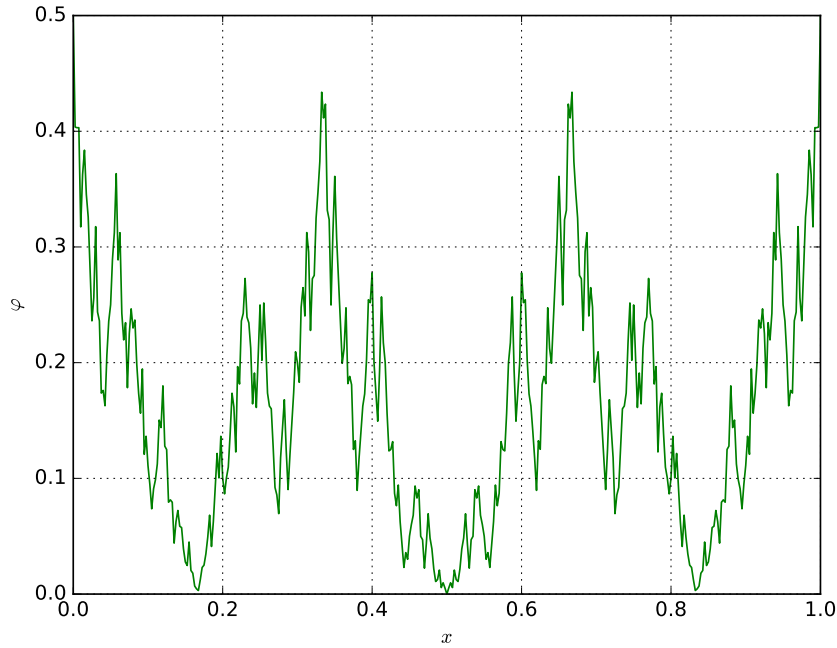


Рис. 2. Пример одномерной функции, порождённой развёрткой

прообразов и глобальный минимум целевой функции в ней расщепился на три глобальных минимума одномерной функции, полученной в результате редукции. Этот пример демонстрирует последствия потери информации об окрестности точки $(0; 0)$ при редукции размерности.

Вычислительная схема алгоритма, применяющего кривые Пеано для редукции размерности, выглядит следующим образом:

- Алгоритм оптимизации выполняет минимизацию редуцированной одномерной функции $\varphi(y(x))$ из (15),
- После нахождения точки следующего испытания x , её многомерный образ y вычисляется с использованием развёртки $y(x)$,
- Значение исходной многомерной функции $\varphi(y)$ вычисляется в точке $y \in D$,
- Вычисленное значение $z = \varphi(y)$ используется как значение одномерной редуцированной функции $\varphi(y(x))$ в точке x .

3.2. Сдвиговые развёртки

Применение кривой типа Пеано для редукции размерности приводит к потере локальной информации об окрестности многомерных точек в пространстве \mathbb{R}^N . Одним из способов частично решить эту проблему является использование множества развёрток:

$$Y_L(x) = \{y^0(x), y^1(x), \dots, y^L(x)\} \quad (17)$$

вместо одной кривой Пеано $y(x)$.

Такой набор отображений может быть получен путём сдвига исходной развёртки $y^0(x)$ на 2^{-l} , $0 \leq l \leq L$ по каждой из координат. Каждая из развёрток определена на своём гиперкубе $D_l = \{y \in R^N : -2^{-1} \leq y_i + 2^{-l} \leq 3 \cdot 2^{-1}, 1 \leq i \leq N\}$, $0 \leq l \leq L$.

На Рис. 3а пунктирной линией обозначен образ интервала $[0, 1]$, полученный с помощью $y^0(x)$, $x \in [0, 1]$. Поскольку гиперкуб D из (1) принадлежит пересечению семейства гиперкубов D_l , необходимо ввести дополнительное ограничение:

$$g_0(y) = \max \{|y_i| - 2^{-1} : 1 \leq i \leq N\}, \quad (18)$$

тогда гиперкуб D можно представить в виде

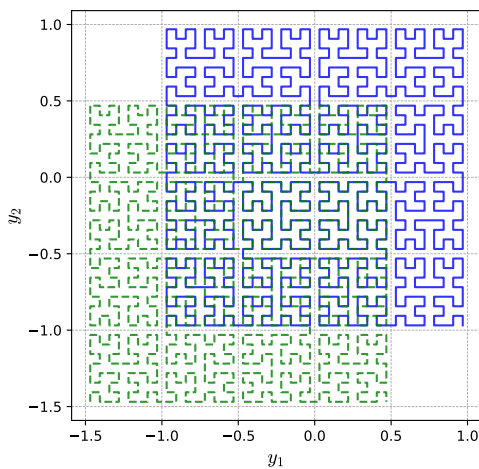
$$D = \{y^l(x) : x \in [0, 1], g_0(y^l(x)) \leq 0\}, \quad 0 \leq l \leq L,$$

т.е. $g_0(y) \leq 0$ если $y \in D$, иначе $g_0(y) > 0$. Следовательно, любая точка $y \in D$ имеет прообраз $x^l \in [0, 1]$, определяемый соответствующей развёрткой $y^l(x)$, $0 \leq l \leq L$.

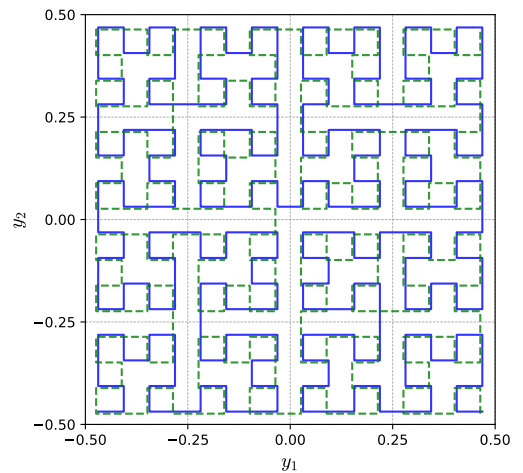
Таким образом, каждая развёртка $y^l(x)$, $0 \leq l \leq L$, порождает свою собственную задачу вида (1), имеющую расширенную (по сравнению с D) область поиска D_l и дополнительное ограничение-неравенство с левой частью вида (18)

$$\min \{\varphi(y^l(x)) : x \in [0, 1], g_j(y^l(x)) \leq 0, 0 \leq j \leq m\}, \quad 0 \leq l \leq L. \quad (19)$$

При этом, L копий метода AGS решают каждую из указанных задач, на каждой итерации обмениваясь многомерными точками следующего испытания и добавляя их прообразы в свою поисковую информацию. Такая схема имеет смысл только при наличии L параллельных вычислительных ядер или процессоров.



(а) Две сдвиговые развёртки, определённые на гиперкубах D_0 и D_1



(б) Две вращаемые развёртки на одной плоскости

Рис. 3. Различные развёртки, построенные с низкой плотностью

3.3. Вращаемые развёртки

Схема построения множества развёрток (здесь и далее будем называть, развёртки, порождённые ей сдвиговыми или S -развёртками), описанная в Секции 3.2, позволяет сохранять часть информации о близости точке в многомерном пространстве и, таким образом, обеспечивает более точную (по сравнению с одной развёрткой) оценку константы Гёльдера в процессе оптимизации. Однако, однако, этот подход имеет недостаток в виде наличия дополнительного ограничения что затрудняет построение эффективных реализаций алгоритма оптимизации на базе S -развёрток (см. конец Секции 3.6).

Чтобы обойти сложности, возникающие при работе с S -развёртками и одновременно сохранить часть информации об окрестностях точек N -мерном вещественном пространстве, была предложена ещё одна схема построения множества развёрток. Эта схема предполагает вместо получения группы развёрток делать сдвиг не сдвиг кривой Пеано по диагонали гиперкуба, а её вращение относительно центра координат. На Рис. 3b представлены две развёртки, аппроксимирующие кривую Пеано при $N = 2$. Получая новые развёртки путём отражения относительно осей координат, можно породить множество развёрток мощностью до 2^N . При этом дополнительное ограничение $g_0(y)$ из (18), необходимое для получения набора S -развёрток, отсутствует.

3.4. Неинъективная развёртка

Как уже было сказано в секции 3.2, потеря информации о близости точек в многомерном пространстве может быть частично компенсирована использованием множественных отображений $Y_L(x) = \{y^1(x), \dots, y^L(x)\}$. Однако, сама по себе кривая типа Пеано сохраняет в себе часть этой информации: она не является инъективным отображением, поэтому имея один образ $y(x) \in \mathbb{R}^N$, можно получить несколько несколько отличных x прообразов $t_j \in [0, 1], t_j \neq x$, которые затем могут быть добавлены в поисковую информацию индексного метода.

Кривая типа Пеано, используемая в (15) для редукции размерности, определяется через предельный переход, поэтому не может быть вычислена непосредственно. При численной оптимизации используется некоторое её приближение, являющееся инъективной кусочно-линейной кривой. Позднее было предложено неинъективное отображение равномерной сетки на отрезка $[0, 1]$ на равномерную сетку в гиперкубе D . Каждый многомерный узел может иметь до 2^N одномерных прообразов. На рис. 4b крестиками обозначена сетка в пространстве \mathbb{R}^2 , для двух узлов которой указаны соответствующие им одномерные прообразы из $[0, 1]$ (отмечены квадратами и кругами). Каждый указанный узел имеет по 3 прообраза.

Недостатком неинъективной развёртки является потенциально большое количество прообразов (до 2^N).

3.5. Гладкая развёртка

Рассмотренные в предыдущих пунктах способы построения развёртки строят кривую $y(x)$, которая не является гладкой (см. рис. 3а). Отсутствие гладкости может негативно сказаться на свойствах редуцированной одномерной функции $\varphi(y(x))$, т.к. гладкая кривая более качественно передает информации о возрастании/убывании исходной функции. На основе исходного алгоритма построения негладкой развёртки было предложен обобщенный алгоритм, позволяющий строить гладкую развёртку. Для иллюстрации на рис. 4а изображена гладкая развёртка в двумерном случае. Недостатком гладкой развёртки является в несколько раз большая сложность вычисления по сравнению с кусочно-линейными кривыми (требуется вычислять нелинейные гладкие функции). Причём с ростом точности аппроксимации и размерности количество интервалов гладкости увеличивается и сложность вычисления кривой нарастает.

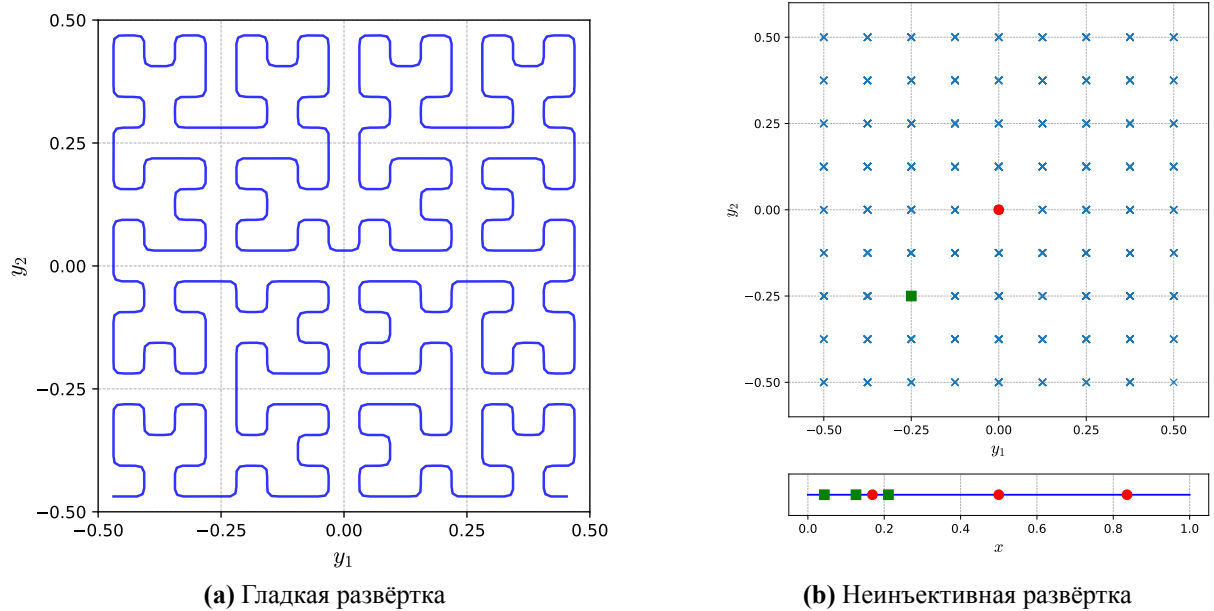


Рис. 4. Различные развёртки, построенные с низкой плотностью

3.6. Сравнение развёрток

В рамках квалификационной работы произведено сравнение скорости сходимости IAGS с различными типами развёткок проводилось в соответствии с методикой, описанной в Секции 2. С целью понять, обладает ли какой-либо из перечисленных ранее типов развёрток существенным преимуществом над другими, были построены операционные характеристики индексного метода с различными типами развёрток на наборах задач GKLS 2d Simple и GKLS 3d Simple.

В результате проведённого сравнения различных способов редукции размерности, основанных на отображениях типа кривой Пеано, можно сделать вывод о том, что для построения базовой версии AGS с практической стороны выгоднее использовать единственную кривую Пеано, кроме случаев низкой размерности или чрезвычайно трудоёмкого вычисления целевой

функции. Из множественных отображений оптимальным выбором будет использование вращаемых развёрток. Поскольку результаты, полученные для алгоритма с одной развёрткой непосредственно распространяются на алгоритмы с вращаемыми развёртками, далее будем рассматривать только случай единственной кривой Пеано.

4. Сравнение методов глобальной оптимизации

В разделе 3 квалификационной работы был рассмотрен вопрос о выборе базовой схемы редукции размерности размерности задач оптимизации вида (1) для алгоритма IAGS. Далее проведено сравнение алгоритма глобального поиска с одной развёрткой типа кривой Пеано и других алгоритмов глобальной оптимизации, основанных на кардинально различных принципах. Такое сравнение позволило выяснить, перспективен ли базовый алгоритм глобального поиска для дальнейших модификаций. В качестве тестовых задач для сравнения, рассматривались задачи без ограничений, поскольку большинство сравниваемых методов не предусматривают специально разработанных схем для их эффективного учёта. Алгоритм глобального поиска без индексной схемы обозначен как AGS. Также рассматривается схема контроля параметра надёжности r в AGS, позволяющую в некоторых случаях снизить зависимость метода от его выбора.

4.1. Методы глобальной оптимизации для сравнения

- **Multi Level Single Linkage.** MLSL является улучшенным вариантом мултистартовой схемы. Алгоритм сэмплирует стартовые точки, равномерно распределённые по области поиска, и производит локальную оптимизацию из них. В сравнении со стандартной мултистартовой схемой MLSL использует кластеризацию и набор эвристических правил для избежания повторных локальных спусков в уже обнаруженные локальные минимумы.
- **DIRECT.** Метод является детерминированным и рекурсивно разделяет область поиска, формируя дерево гиперпрямоугольников. DIRECT использует значения целевой функции и оценку константы Липшица (2) для получения оценки перспективности дальнейшего разбиения каждого из гиперинтервалов.
- **Locally-biased DIRECT (DIRECT/).** Вариация метода DIRECT, которая меньше уделяет внимания гиперинтервалам с низкой оценкой перспективности. Эта вариация сходится быстрее на задачах с небольшим количеством локальных минимумов, но в сложных случаях может не найти глобальный.
- **Dual Simulated Annealing.** Этот стохастический метода является комбинацией классического метода имитации отжига (CSA) и быстрого метода имитации отжига (FSA), совмещенной с применением локальной оптимизации. Метод сходится гораздо быстрее, чем CSA и FSA.

- **Differential Evolution.** DE является адаптацией оригинального генетического алгоритма к непрерывному пространству поиска.
- **Controlled Random Search.** CRS производит испытания в случайно сгенерированных начальных точках и определяет следующую точку для испытания на основе симплекса, построенного из случайно выбранных точек более ранних испытаний. CRS не считается эволюционным алгоритмом, хотя хранит точки, похожие на популяцию, и выполняет случайные трансформации над ними, по аналогии с применением оператора мутации.
- **StoGO.** StoGO разделяет пространство поиска на гиперпрямоугольники и использует метод ветвей и границ, вычисляя верхние оценки минимумов в подобластях с помощью локальной оптимизации.

Все перечисленные алгоритмы доступны в исходных кодах как компоненты широко распространённых программных пакетов. DIRECT, DIRECT l , CRS, MLSL и StoGO являются компонентами библиотеки NLOpt. Реализации DE и DSA распространяются в пакете SciPy для языка Python.

4.1.1. Контроль параметра надёжности в AGS

Параметр r из (9) непосредственно влияет на глобальную сходимость AGS: при достаточно большом значении r метод гарантированно сходится ко всем глобальным минимумам целевой функции. В то же время, согласно (11), при бесконечно большом значении r AGS превращается в полный перебор по равномерной сетке.

В идеальном случае, для обеспечения наибольшей скорости сходимости оценка константы Липшица из (9) не должна быть слишком завышенной, но на практике реальное значение L из (2) неизвестно и приходится либо брать заведомо большое значение r , либо проводить несколько запусков AGS с разными параметрами. Чтобы в некоторой степени решить обозначенную проблему выбора r в квалификационной работе предложена следующая схема:

- совершить q итераций AGS при $r = r_{max}$;
- совершить q итераций AGS при $r = r_{min}$;
- повторять предыдущие шаги до сходимости или исчерпания лимита итераций.

В указанном алгоритме $r_{min} < r_{max}$, $q > 1$. Вместо одного параметра r теперь необходимо выбирать 3, однако, согласно численным экспериментам, сделать это проще, чем найти оптимальное значение r . Интуитивно практическую эффективность предложенной схемы можно объяснить тем, что теперь работа метода происходит в двух режимах: глобальный поиск при $r = r_{max}$ и локальный при $r = r_{min}$. Если во время фазы глобального поиска метод приблизился к глобальному минимуму, то во время следующей фазы оценка глобального минимума будет

Таблица 1. Среднее количество испытаний, затраченное методами оптимизации при решении задач различных классов

	AGS	AGS-AR	CRS	DIRECT	DIRECT/	MLSL	SDA	DE	StoGO
F_{GR}	193.1	248.3	400.3	182.2	214.9	947.2	691.2	1257.3	1336.8
GKLS 2d Simple	254.9	221.6	510.6	189.0	255.2	556.8	356.3	952.2	1251.5
GKLS 2d Hard	728.7	785.0	844.7	985.4	1126.7	1042.5	1637.9	1041.1	2532.2
GKLS 3d Simple	1372.1	1169.5	4145.8	973.6	1477.8	4609.2	2706.5	5956.9	3856.1
GKLS 3d Hard	3636.1	1952.1	6787.0	2298.7	3553.3	5640.1	4708.4	6914.3	7843.2
GKLS 4d Simple	5729.8	4919.1	19883.6	7328.8	15010.0	41484.8	22066.0	6271.2	29359.2
GKLS 4d Hard	13113.4	12860.1	27137.4	22884.4	55596.1	80220.1	68048.0	12487.6	58925.5
GKLS 5d Simple	5821.5	6241.3	62921.7	5966.1	10795.5	52609.2	34208.8	20859.4	69206.8
GKLS 5d Hard	17008.6	21555.1	87563.9	61657.3	148637.8	138011.8	115634.6	26850.0	141886.5

Таблица 2. Количество задач оптимизации, решённых методами при заданном лимите испытаний (общее количество задач в каждом наборе равно 100)

	AGS	AGS-AR	CRS	DIRECT	DIRECT/	MLSL	SDA	DE	StoGO
F_{GR}	100	100	76	100	100	97	96	96	67
GKLS 2d Simple	100	100	85	100	100	100	100	98	90
GKLS 2d Hard	100	97	74	100	100	100	93	85	77
GKLS 3d Simple	100	100	75	100	100	100	89	86	44
GKLS 3d Hard	100	100	72	100	99	100	88	77	43
GKLS 4d Simple	100	100	74	100	100	94	82	68	72
GKLS 4d Hard	100	100	60	99	99	94	73	55	69
GKLS 5d Simple	100	100	86	100	100	98	100	88	82
GKLS 5d Hard	100	100	77	100	93	79	86	77	78

быстро уточнена. Если двух фаз не хватило, то процесс продолжается. Таким образом выбирается более оптимальный компромисс между локальной и глобальной оптимизацией. Далее будем обозначать метод, использующий описанную схему как AGS-AR.

4.2. Результаты численных экспериментов

Сравнение алгоритмов оптимизации было проведено по методике, описанной в Секции 2 и на описанных там же классах тестовых задач.

Результаты запуска рассматриваемых методов оптимизации на наборах тестовых задач представлены в таблицах 1, 2. DIRECT, AGS, AGS-AR продемонстрировали лучшую скорость сходимости на всех классах, при этом AGS-AR уступает DIRECT на двухмерных задачах класса Simple, но имеет преимущество на задачах класса Hard. Как можно видеть из Table 2, детерминированные методы (AGS, AGS-AR, DIRECT и DIRECT/) оказались более надёжными, т.е. решили больше тестовых задач. Среди стохастических методов самую высокую надёжность показали MLSL и SDA.

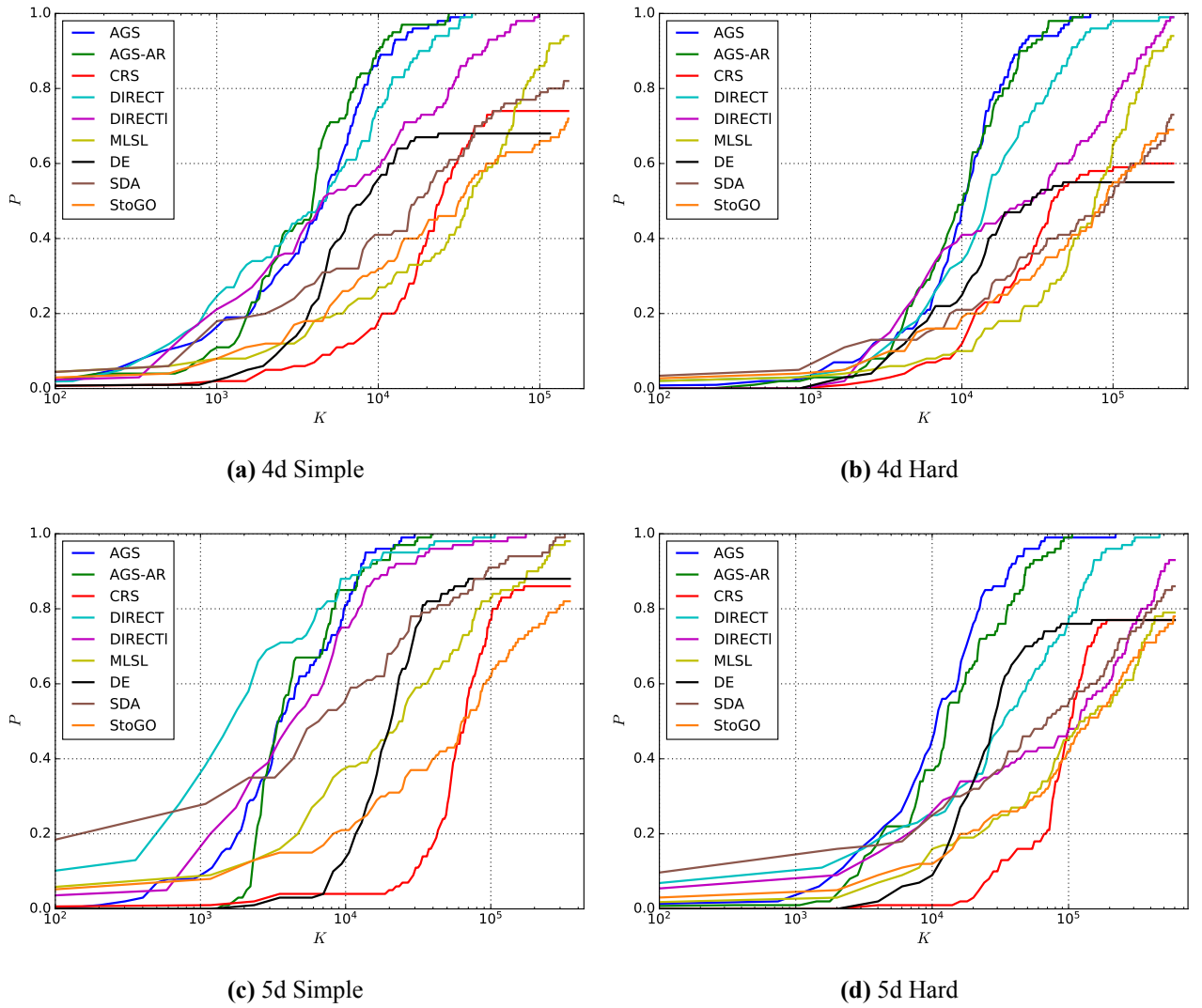


Рис. 5. Операционные характеристики методов при решении задач из классов GKLS 4d и 5d.

Операционные характеристики методов (Рис. 5a, 5b, 5c, 5d) показывают, что AGS и AGS-AR быстрее других методов достигают 100% надёжности. Также на классе GKLS 5d Simple DIRECT имеет скорость сходимости выше, чем у других методов, но есть несколько сложных задач, решение которые решаются существенно дольше других, что влияет на среднее количество испытаний.

Чувствительность AGS и AGS-AR к выбору параметров. Чтобы оценить степень влияния настроек методов на скорость сходимости AGS и AGS-AR, были проведены эксперименты на классе задач GKLS 5d Simple со следующими параметрами:

- AGS при $r = 4$ (как в Таблице ??);
- AGS при $r = 6$;
- AGS-AR с параметрами, указанными в начале Секции 4.2 ($q = 50 \cdot \log_2(4) \cdot 25 = 2500$, $r_{min} = 3$, $r_{max} = 2 \cdot r_{min}$);

- AGS-AR при $r_{max} = 8$ и остальными параметрами как в предыдущем эксперименте;
- AGS-AR при $q = 1000$ и другими параметрами как в начале Секции 4.2.

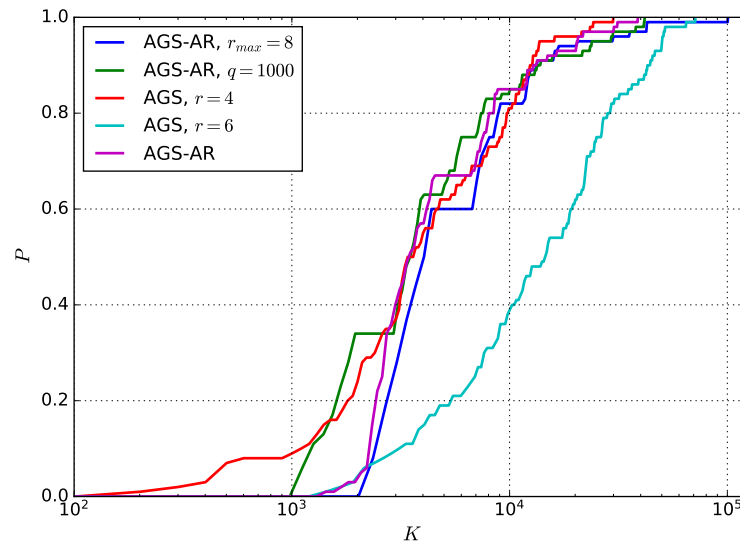


Рис. 6. Операционные характеристики AGS и AGS-AR на классе GKLS 5d Simple с различными параметрами.

Операционные характеристики, полученные в экспериментах, описанных выше, представлены на Рис. 6. AGS при $r = 6$ (голубая кривая) показывает самую медленную сходимость, что говорит о высокой чувствительности AGS к выбору значения r . Поскольку AGS-AR использует то же самое значение r как AGS при $r = 6$, операционные характеристики этих методов идентичны до точки $K = 2500$. После этой границы AGS-AR переключается на $r = 3$ и количество решённых задач начинает быстро возрастать, вплоть до начала следующей фазы глобального поиска при $K = 5000$. Интервалы, на которых AGS-AR работает с $r = r_{max}$ видны на операционных характеристиках как плато. Изменение r и q влияет на операционную характеристику AGS-AR незначительно. Это наблюдение подтверждает надёжность предложенной модификации AGS с переменным параметром r .

4.3. Итоги сравнения методов

В результате сравнения различных методов оптимизации, были сделаны следующие выводы:

- предложенная модификация AGS, AGS-AR менее чувствительна к параметрам и сходится так же быстро, как AGS с заранее подобранными под класс задач параметрами;
- AGS-AR продемонстрировал надёжность и скорость сходимости на уровне другого детерминированного метода, DIRECT, и превзошёл на рассмотренных тестовых задачах многие другие методы, реализации которых так же доступны в исходных кодах;

- на рассмотренных существенно многоэкстремальных задачах малой размерности стохастические методы оптимизации существенно уступают в скорости сходимости и надёжности детерминированным.

5. Алгоритм, решающий множество задач

В главе 5 квалификационной работы рассмотрено построение алгоритма, решающего серию задач с нелинейными ограничениями из (6). Задачи, подобные (6), могут возникать, например, при скаляризации задачи многокритериальной оптимизации методом свёртки критериев или при решении задач смешанного целочисленного программирования путём перебора всех возможных значений целочисленных параметров и дальнейшей оптимизации по вещественным. Как уже было отмечено ранее, желаемым свойством метода, решающего совокупность задач, является равномерная сходимость (7).

5.1. Описание алгоритма

По аналогии с подходом, описанном в [4], для решения серии задач (6) будем использовать q синхронно работающих копий IAGS с тем лишь отличием, что на шаге 6 при выборе интервала с наилучшей характеристикой, выбор будет осуществляться из всех интервалов, которые породили на данный момент q копий IAGS. Если наибольшая характеристика соответствует задаче i , то выполняется шаг 7 в копии метода с номером i , а остальные копии метода простаивают. Таким образом, на каждой итерации испытание проводится в задаче, наиболее перспективной с точки зрения характеристик (11), что позволяет динамически распределять ресурсы метода между задачами. Обозначим метод, решающий множество задач с ограничениями, как MIAGS.

Параллельная модификация метода не отличается от рассматриваемой в [4] и заключается в выборе p интервалов на шаге 6 и выполнения p испытаний параллельно на следующем шаге. При этом все ресурсы метода в рамках итерации могут быть направлены как на одну, так и на $l \leq p$ задач одновременно (в зависимости от того, какой из задач принадлежат выбранные методам интервалы).

5.2. Условия сходимости

Достаточные условия сходимости метода IAGS в случае $q = 1$ приведены в [5], рассмотрим их подробнее:

Теорема 1. *(Достаточные условия сходимости IAGS) Предположим, что следующие утверждения справедливы:*

1. $D \neq \emptyset$, задача (5) имеет решение.
2. Функции $g_j(y) \leq 0, 1 \leq j \leq m + 1$, липшицевы в области D с соответствующими константами L_i (здесь $g_{m+1}(y) = \varphi(y)$).

3. Для достаточно больших k из (8), значения μ_ν из (9) удовлетворяют неравенствам:

$$r_\nu \mu_\nu > 2^{3-1/N} L_\nu \sqrt{N+3}, \quad 1 \leq \nu \leq m+1. \quad (20)$$

Тогда любая предельная точка \bar{y} последовательности $\{y_k\} = \{y(x_k)\}$, сгенерированная IAGS при решении задачи (5), является допустимой и удовлетворяет условиям

$$\varphi(\bar{y}) = \inf\{\varphi(y^k) : g_i(y^k) \leq 0, 1 \leq i \leq m, k = 1, 2, \dots\} = \varphi(y^*). \quad (21)$$

Замечание 1. Из соотношения между константами Гёльдера и Липшица из неравенства (4) следует, что параметр r_ν из (11) должен удовлетворять условию

$$r_\nu > 2^{2-1/N}. \quad (22)$$

Теорема 2. (О сходимости MIAGS) Пусть условия 1-3 Теоремы 1 верны для каждой задачи i , $1 \leq i \leq q$ из (6), т.е. каждая из задач может быть решена AGS. Тогда в процессе решения q задач MIAGS сгенерирует q бесконечных последовательностей $\{y_i^k\}$, $1 \leq i \leq q$, таких, что

$$\varphi_i(\bar{y}_i) = \inf\{\varphi(y_i^k) : g_j^i(y_i^k) \leq 0, 1 \leq j \leq m_i, k = 1, 2, \dots\} = \varphi_i(y_i^*).$$

Доказательство Теоремы 2 приведено в квалификационной работе. Теорема устанавливает только достаточные условия сходимости IAGS, наличие равномерной сходимости проверено численно.

5.3. Результаты численных экспериментов

В качестве тестовых задач для MIAGS в рамках работы были рассмотрены наборы задач с ограничениями, которые описаны в Секции 2. Кроме того, рассматривалась одна многокритериальную задачу, которая сводится к решению совокупности задач вида (6) с помощью свёртки критериев.

При оценке качества метода и его реализации кроме ускорения от распараллеливания по итерациям и по времени выполнения, также было принято во внимание среднее максимальное расстояние (в смысле l_{\inf} -нормы) текущей оценки оптимума до его реального положения, вычисленное на множестве задач (6): D_{avg} и D_{max} . Динамика этих величин в процессе оптимизации показывает, насколько равномерно метод распределяет ресурсы между задачами. Если кривые D_{avg} и D_{max} ведут себя одинаково, то метод идеально распределяет вычислительные ресурсы между задачами, в противном случае, чем больше разница между ними, тем более неравномерно происходит распределение.

Результаты решения тестовых задач последовательной и параллельной версией модифицированного IAGS для решения множества задач представлены в таблице 3. Для всех двухмерных классов задач параметр $r = 4.7$. В случае трехмерных задач $r = 4.7$, $\varepsilon_\nu = 0.1$. Кроме того, для

Таблица 3. Результаты экспериментов на наборах синтетических задач

Класс задач	p	Количество итераций	Время, с	S_i	S_t
GCGen GKLS Simple 2d & F_{GR}	1	51434	90.20	-	-
	2	25698	56.96	2.00	1.58
	4	13015	36.67	3.95	2.46
	6	8332	26.85	6.17	3.36
GCGen GKLS Simple 2d	1	59066	97.53	-	-
	2	29060	60.56	2.04	1.61
	4	14266	38.92	4.14	2.51
	6	9436	29.53	6.26	3.30
GCGen GKLS Simple 3d	1	782544	1117.55	-	-
	2	397565	752.92	1.97	1.48
	4	208073	526.67	3.76	2.12
	6	142089	445.45	5.50	2.51
GCGen GKLS Simple 4d	1	14021720	15806.6	-	-
	2	6313070	7254.85	2.22	2.18
	4	3479344	4932.55	4.03	3.20
	6	2783339	3955.38	5.04	3.99

трёх- и четырёх- мерных задач была применена техника ε -резервирования из [5] Глава 8.3 с $\varepsilon = 0.1$. Это позволило сократить время экспериментов, ускорив сходимость метода. Следуя изначальному предположению о высокой трудоёмкости проведения испытаний, во всех экспериментах в целевые функции и ограничения была внесена дополнительная вычислительная нагрузка так, чтобы время одного обращения к функции задачи было равно примерно 1 мс.

Из Таблицы 3 видно, что ускорение по итерациям $S_i = \frac{iters(p=1)}{iters(p=i)}$ растет линейно с увеличением числа потоков p , в то время, как ускорение по времени $S_t = \frac{time(p=1)}{time(p=i)}$ увеличивается не так быстро, что говорит о неидеальной реализации алгоритма. Увеличить реальное ускорение, верхней границей для которого является S_i , возможно путем оптимизации взаимодействий между копиями IAGS и это планируется сделать в ходе будущей работы.

Для того, чтобы показать равномерную сходимость все тестовые задачи были также решены IAGS в режиме решения отдельных задач. На рис. 7 указаны графики величин средних и максимальных расстояний от реальных оптимумов до их текущих оценок при решении серии из задач, порожденных двумя разными генераторами, по отдельности (сплошная кривая) и совместно (пунктирная кривая). Несмотря на значительную разницу в структуре задач, MIAGS гораздо быстрее уменьшает максимальное и среднее отклонения оценок от действительных оптимумов, чем это происходит при раздельном решении задач. Это говорит о наличии равномерной сходимости по всему множеству совместно решаемых задач. При этом в случае последовательного решения задач величина D_{max} имеет наибольшее значение вплоть до решения последней задачи.

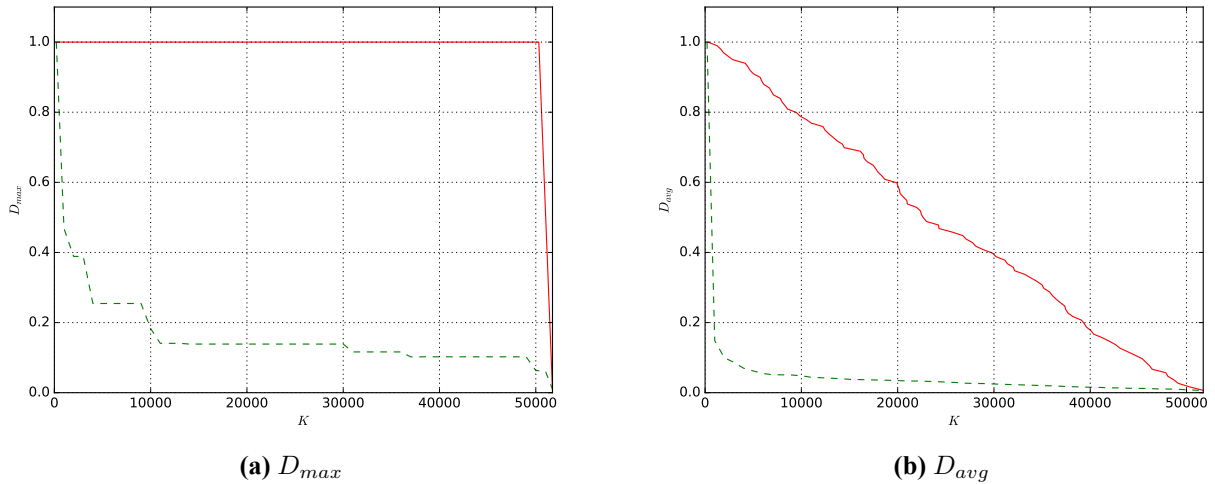


Рис. 7. Динамика величин D_{avg} и D_{max} в процессе решения множества двухмерных задач, порождённых двумя разными генераторами GKLS и F_{GR}

5.4. Итоги

В разделе 5 квалификационной работы была реализована поддержка нелинейных ограничений в алгоритме, решающем множество задач глобальной оптимизации в совокупности и распределяющего свои ресурсы так, чтобы обеспечивать равномерную сходимость во всех задачах. Доказано теорема о достаточных условиях сходимости полученного метода. Свойство равномерной сходимости проверено с помощью численного эксперимента. Также в ходе численных экспериментов была оценена эффективность выполненной параллельной реализации метода и показана эффективность совместного решения множества задач на примере получения Парето фронта в многокритериальной задаче с нелинейными ограничениями.

Заключение

В рамках квалификационной работы была поставлена цель выбрать оптимальную модификацию алгоритма глобального поиска (AGS) и на его основе построить метод для одновременного решения множества задач глобальной оптимизации с нелинейными ограничениями.

В ходе работы получены следующие результаты:

1. Произведено сравнение различных способов редукции размерности, основанных на отображениях типа кривой Пеано. В результате был сделан вывод о том, что для построения базовой многомерной версии метода AGS с практической стороны наиболее выгодно использовать единственную кривую Пеано;
2. Предложена модификация AGS, AGS-AR, менее чувствительная к параметрам, и сходящаяся так же быстро, как AGS с заранее подобранными под класс задач параметрами;

3. AGS-AR продемонстрировал надёжность и скорость сходимости на уровне другого детерминированного метода, DIRECT, и превзошёл на рассмотренных тестовых задачах многие другие методы, реализации которых так же доступны в исходных кодах;
4. Программная реализация метода AGS-AR прошла процедуру ревью и была включена в состав популярной библиотеки алгоритмов нелинейной оптимизации NLOpt;
5. Реализована поддержка нелинейных ограничений в алгоритме, решающем множество задач глобальной оптимизации в совокупности и распределяющего свои ресурсы так, чтобы обеспечивать равномерную сходимость во всех задачах. Доказано теорема о достаточных условиях сходимости полученного метода. Свойство равномерной сходимости проверено с помощью численного эксперимента.
6. В ходе численных экспериментов была оценена эффективность выполненной параллельной реализации метода, решающего множество задач, и показана эффективность такого подхода над решением задач по отдельности на примере получения Парето фронта в многокритериальной задаче с нелинейными ограничениями.

Список публикаций в изданиях, индексируемых Scopus

1. *Barkalov K., Sovrasov V., Lebedev I.* Comparison of Dimensionality Reduction Schemes for Parallel Global Optimization Algorithms // *Supercomputing* / под ред. V. Voevodin, S. Sobolev. — Cham : Springer International Publishing, 2019. — С. 50—62. — ISBN 978-3-030-05807-4.
2. *Sovrasov V.* Comparison of Several Stochastic and Deterministic Derivative-Free Global Optimization Algorithms // *Mathematical Optimization Theory and Operations Research* / под ред. M. Khachay, Y. Kochetov, P. Pardalos. — Cham : Springer International Publishing, 2019. — С. 70—81. — ISBN 978-3-030-22629-9.
3. *Sovrasov V., Barkalov K.* Parallel Global Optimization Algorithm with Uniform Convergence for Solving a Set of Constrained Global Optimization Problems // *Advances in Optimization and Applications* / под ред. N. Olenov, Y. Evtushenko, M. Khachay, V. Malkova. — Cham : Springer International Publishing, 2020. — С. 38—52. — ISBN 978-3-030-65739-0.
4. *Barkalov K., Strongin R.* Solving a Set of Global Optimization Problems by the Parallel Technique with Uniform Convergence // *J. of Global Optimization*. — Norwell, MA, USA, 2018. — Май. — Т. 71, № 1. — С. 21—36. — ISSN 0925-5001. — DOI: 10.1007/s10898-017-0555-4.
5. *Strongin R.G., Sergeyev Ya.D.* Global optimization with non-convex constraints. Sequential and parallel algorithms. — Dordrecht : Kluwer Academic Publishers, 2000. — DOI: 10.1007/978-1-4615-4677-1.