

© 2019 г. Р.Г. СТРОНГИН, д-р физ.-мат. наук  
В.П. ГЕРГЕЛЬ, д-р техн. наук  
К.А. БАРКАЛОВ, канд. физ.-мат. наук  
(Нижегородский государственный университет им. Н.И. Лобачевского)

## АДАПТИВНАЯ ГЛОБАЛЬНАЯ ОПТИМИЗАЦИЯ НА ОСНОВЕ БЛОЧНО-РЕКУРСИВНОЙ СХЕМЫ РЕДУКЦИИ РАЗМЕРНОСТИ<sup>1</sup>

В статье рассматриваются задачи многомерной многоэкстремальной оптимизации и численные методы их решения. Об оптимизируемой функции делается лишь общее предположение, что она удовлетворяет условию Липшица с априори неизвестной константой (задачи такого типа часто встречаются в приложениях). Рассмотрено два способа редукции размерности в задачах многомерной оптимизации: использованные кривых Пеано (разверток) и рекурсивная многошаговая схема. Предложена обобщенная схема, комбинирующая эти два подхода. В новой схеме решение многомерной задачи сводится к решению семейства задач меньшей размерности, в которых, в свою очередь, используются развертки. Реализован адаптивный алгоритм, в котором все возникающие подзадачи решаются одновременно. Проведены численные эксперименты на нескольких сотнях тестовых задач, подтверждающие эффективность предложенной схемы редукции размерности.

### 1. Введение

В статье рассматриваются задачи глобальной оптимизации вида

$$(1) \quad \begin{aligned} \varphi(y^*) &= \min \{ \varphi(y) : y \in D \}, \\ D &= \{ y \in R^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N \}. \end{aligned}$$

Предполагается, что целевая функция может быть многоэкстремальной, задана неявно (функция вида «черный ящик»), а вычисление ее значений связано с решением задачи численного моделирования и является трудоемкой операцией.

Любая возможность достоверно оценить глобальный оптимум в многоэкстремальной задаче с функциями вида «черный ящик» принципиально основана на априорной информации, позволяющей связать возможные значения целевой функции с ее известными значениями в точках проведенных испытаний. Для многих прикладных задачах типичной является ситуация, когда ограниченное изменение вектора параметров  $y$  вызывает ограниченное изменение значений  $\varphi(y)$ . Математической моделью, описывающей указанное предположение, является условие Липшица

$$|\varphi(y') - \varphi(y'')| \leq L \|y' - y''\|, \quad y', y'' \in D, \quad 0 < L < \infty.$$

---

<sup>1</sup>Исследование выполнено за счет гранта Российского научного фонда (проект № 16-11-10150).

Предположение липшицевости целевой функции типично для многих подходов к разработке оптимизационных алгоритмов. Первые методы липшицевой оптимизации были предложены в начале 70-х годов [1, 2, 3]; с тех пор данное направление продолжает активно развиваться [4, 5, 6, 7, 8]. Например, многие известные методы основаны на различных способах разбиения области поиска на систему подобластей и последующего выбора наиболее перспективной подобласти для размещения очередного испытания (вычисления целевой функции). Результаты, полученные в данном направлении, представлены в работах [9, 10, 11, 12, 13].

В настоящее время для решения задач оптимизации с функциями вида «черный ящик» широко используются генетические и популяционные алгоритмы (см., например, [14, 15]), которые так или иначе основаны на идеях случайного поиска. В силу простоты реализации и использования они получили большую популярность, однако по качеству работы (численной оценкой которого может служить число корректно решенных задач из некоторого набора) они существенно уступают детерминированным алгоритмам [16, 17].

Одним из эффективных детерминированных методов решения задач многоэкстремальной оптимизации является *информационно-статистический алгоритм глобального поиска*. Основы информационно-статистического подхода были заложены Ю.И. Неймарком [18, 19] и развиты Р.Г. Стронгиным [20, 21]. Впоследствии метод, изначально предложенный для решения безусловных задач, был успешно обобщен для решения задач с невыпуклыми ограничениями [22, 23] и многокритериальных задач [24]. Для различных вариантов алгоритма были предложены способы распараллеливания, учитывающие особенности архитектуры современных вычислительных систем [25].

Разработанные методы основаны на редукции исходной многомерной задачи к эквивалентной одномерной или системе одномерных подзадач и последующим решением одномерных задач эффективными методами оптимизации функций одной переменной. Предложено две такие схемы: редукция на основе кривых, заполняющих пространство (кривых Пеано, или *разверток*) [26, 27], и схема рекурсивной вложенной оптимизации (*многошаговая схема*) [28, 29]. В статье [30] предложена адаптивная многошаговая схема, существенно повышающая эффективность оптимизации по сравнению с базовым прототипом. В данной работе предлагается обобщение адаптивной схемы редукции размерности, комбинирующее использование вложенной оптимизации и кривых Пеано. При таком подходе вложенные подзадачи в адаптивной схеме могут быть как одномерными, так и многомерными; в последнем случае для редукции размерности вложенных подзадач используются развертки.

## 2. Базовый алгоритм глобального поиска

В качестве базовой задачи рассмотрим одномерную задачу многоэкстремальной оптимизации

$$(2) \quad \varphi^* = \varphi(x^*) = \min \{ \varphi(x) : x \in [0, 1] \}$$

с целевой функцией, удовлетворяющей условию Липшица.

Приведем описание алгоритма глобального поиска (АГП) для решения базовой задачи в соответствии с [27]. В процессе своей работы АГП порождает последова-

тельность точек  $x^i$ , в которых вычисляются значения целевой функции  $z^i = \varphi(x^i)$ . Будем называть процесс вычисления значения целевой функции *испытанием*.

В соответствии с алгоритмом, первые два испытания проводятся в граничных точках интервала  $[0, 1]$ , т.е.  $x^0 = 0$ ,  $x^1 = 1$ . В этих точка вычисляются значения целевой функции  $z^0 = \varphi(x^0)$ ,  $z^1 = \varphi(x^1)$ , и устанавливается значение счетчика  $k = 1$ . Точка очередного испытания  $x^{k+1}$ ,  $k \geq 1$ , выбирается в соответствии со следующими правилами.

**Шаг 1.** Перенумеровать нижним индексом (начиная с 0) точки  $x^i$ ,  $0 \leq i \leq k$ , проведенных испытаний в порядке возрастания координаты, т.е.,

$$0 = x_0 < x_1 < \dots < x_k = 1.$$

Сопоставить точкам  $x_i$ ,  $0 \leq i \leq k$ , вычисленные в них значения целевой функции  $z_i = \varphi(x_i)$ ,  $0 \leq i \leq k$ .

**Шаг 2.** Вычислить максимальное абсолютное значение относительной первой разности

$$(3) \quad \mu = \max_{1 \leq i \leq k} \frac{|z_i - z_{i-1}|}{\Delta_i}$$

где  $\Delta_i = x_i - x_{i-1}$ . Если вычисленное в соответствии с данной формулой значение равно 0, то положить  $\mu = 1$ .

**Шаг 3.** Для всех интервалов  $(x_{i-1}, x_i)$ ,  $1 \leq i \leq k$ , вычислить значение

$$(4) \quad R(i) = r\mu\Delta_i + \frac{(z_i - z_{i-1})^2}{r\mu\Delta_i} - 2(z_i + z_{i-1}),$$

называемое *характеристикой* интервала; величина  $r > 1$  является параметром алгоритма.

**Шаг 4.** Найти интервал  $(x_{t-1}, x_t)$  с максимальной характеристикой

$$(5) \quad R(t) = \max_{1 \leq i \leq k} R(i).$$

Если максимальная характеристика соответствует нескольким интервалам, то в качестве  $t$  выбрать минимальное число, удовлетворяющее (5).

**Шаг 5.** Провести новое испытание в точке

$$(6) \quad x^{k+1} = \frac{1}{2}(x_{t-1} + x_t) - \frac{z_t - z_{t-1}}{2r\mu}.$$

Алгоритм прекращает свою работу при выполнении условия  $\Delta_t < \epsilon$ ; здесь  $t$  из (5), а  $\epsilon > 0$  — заданная точность. В качестве оценки решения задачи выбираются значения

$$z_k^* = \min_{0 \leq i \leq k} \varphi(x^i), \quad x_k^* = \arg \min_{0 \leq i \leq k} \varphi(x^i).$$

Теоретические условия, определяющие сходимость алгоритма, представлены в [27].

### 3. Редукция размерности

#### 3.1. Редукция размерности с помощью кривых, заполняющих пространство

Рассмотрим схему редукции размерности, основанную на использовании кривых, заполняющих пространство (кривых Пеано). Известно, что подобного типа кривые позволяют однозначно отобразить одномерный отрезок  $[0, 1]$  на многомерную область, т.е.

$$(7) \quad \{y(x) : 0 \leq x \leq 1\} = \{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\}.$$

Отметим, что теоретическая кривая  $y(x)$  определяется как предельный объект. Поэтому при практической реализации может быть построено лишь некоторое приближение к истинной кривой. Методы построения подобных аппроксимаций (называемых *развертками*) рассмотрены в [21, 27]. При этом точность приближения развертки к истинной кривой  $y(x)$  зависит от *плотности* развертки  $m$  (являющейся параметром ее построения) и составляет величину порядка  $2^{-m}$  по каждой координате.

Использование подобного рода отображений позволяет свести решение многомерной задачи (1) к решению эквивалентной ей одномерной

$$(8) \quad \varphi(y^*) = \varphi(y(x^*)) = \min \{\varphi(y(x)) : x \in [0, 1]\}.$$

Важным свойством при этом является сохранение ограниченности относительных разностей функции (см. [27]). Если функция  $\varphi(y)$  в области  $D$  удовлетворяла условию Липшица, тогда редуцированная функция  $\varphi(y(x))$ ,  $x \in [0, 1]$  будет удовлетворять равномерному условию Гельдера

$$(9) \quad |\varphi(y(x_1)) - \varphi(y(x_2))| \leq H |x_1 - x_2|^{1/N},$$

где константа Гельдера  $H$  связана с константой Липшица  $L$  соотношением

$$(10) \quad H = 2L\sqrt{N+3}.$$

Условия (9), (10) позволяет легко обобщить «одномерный» алгоритм из раздела 2 для решения многомерных задач. Для этого достаточно заменить длины интервалов  $\Delta_i$  в формулах (3),(4) на длины

$$(11) \quad \Delta_i = (x_i - x_{i-1})^{1/N},$$

а также заменить формулу (6) для вычисления точки нового испытания на формулу

$$(12) \quad x^{k+1} = \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{1}{2r} \left[ \frac{|z_t - z_{t-1}|}{\mu} \right]^N.$$

#### 3.2. Многошаговая схема редукции размерности

Многошаговая схема редукции размерности (схема вложенной оптимизации) основана на известном соотношении [21, 25]

$$(13) \quad \min_{y \in D} \varphi(y) = \min_{y_1 \in [a_1, b_1]} \min_{y_2 \in [a_2, b_2]} \dots \min_{y_N \in [a_N, b_N]} \varphi(y),$$

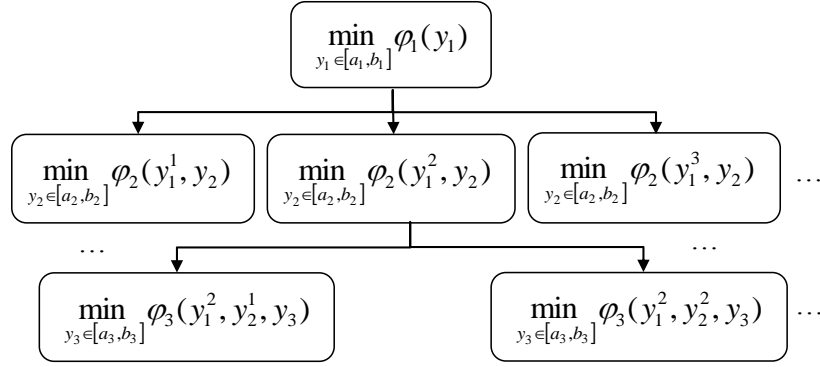


Рис. 1. Взаимосвязи между подзадачами в многошаговой схеме

которое позволяет свести решение исходной многомерной задачи (1) к решению семейства рекурсивно связанных одномерных подзадач.

Для формального описания многошаговой схемы введем семейство функций, определяемых в соответствии с соотношениями

$$(14) \quad \varphi_N(y_1, \dots, y_N) \equiv \varphi(y_1, \dots, y_N),$$

$$(15) \quad \varphi_i(y_1, \dots, y_i) = \min_{y_{i+1} \in [a_{i+1}, b_{i+1}]} \varphi_{i+1}(y_1, \dots, y_i, y_{i+1}), 1 \leq i \leq N-1.$$

Тогда, в соответствии с (13), для решения многомерной задачи (1) достаточно решить одномерную задачу

$$(16) \quad \varphi^* = \min_{y_1 \in [a_1, b_1]} \varphi_1(y_1).$$

Однако каждое вычисление значения функции  $\varphi_1$  в некоторой фиксированной точке  $y_1$  предполагает решение одномерной задачи оптимизации второго уровня

$$(17) \quad \varphi_1(y_1) = \min_{y_2 \in [a_2, b_2]} \varphi_2(y_1, y_2).$$

Вычисление значений функции  $\varphi_2$ , в свою очередь, требует одномерной минимизации функции  $\varphi_3$ , и т.д. вплоть до решения задачи

$$(18) \quad \varphi_{N-1}(y_1, \dots, y_{N-1}) = \min_{y_N \in [a_N, b_N]} \varphi_N(y_1, \dots, y_N),$$

на последнем уровне рекурсии.

Возникающие при этом подзадачи и взаимосвязи между ними отображены на рис. 1. Наглядно видно, что структура взаимосвязей имеет форму дерева, причем функции  $\varphi_N(y_1, \dots, y_N)$  на  $N$ -м уровне являются листьями дерева задач, их значения вычисляются непосредственно.

Для описанной выше многошаговой схемы было предложено обобщение — *блочная многошаговая схема*, которое комбинирует использование разверток и вложенной оптимизации [31].

Рассмотрим  $y$  как вектор, состоящий из векторов (блочных переменных)

$$(19) \quad y = (y_1, y_2, \dots, y_N) = (u_1, u_2, \dots, u_M),$$

где  $i$ -я блочная переменная, т.е. вектор  $u_i$ , состоит из взятых последовательно компонент вектора  $y$ , т.е.  $u_1 = (y_1, y_2, \dots, y_{N_1})$ ,  $u_2 = (y_{N_1+1}, y_{N_1+2}, \dots, y_{N_1+N_2})$ , ...,  $u_M = (y_{N-N_M+1}, y_{N-N_M+2}, \dots, y_N)$ , а  $N_1 + N_2 + \dots + N_M = N$ .

Используя введенные обозначения, основное соотношение многошаговой схемы (13) может быть переписано в форме

$$(20) \quad \min_{y \in D} \varphi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \varphi(y),$$

где подобласти  $D_i$ ,  $1 \leq i \leq M$ , являются проекциями исходной области поиска  $D$  на подпространства, соответствующие переменным  $u_i$ ,  $1 \leq i \leq M$ .

Способ решения задачи (1) на основе соотношения (20) будет в целом повторять (14)–(16). Необходимо лишь заменить исходные переменные  $y_i$ ,  $1 \leq i \leq N$ , на блочные переменные  $u_i$ ,  $1 \leq i \leq M$ .

При этом вложенные подзадачи

$$(21) \quad \varphi_i(u_1, \dots, u_i) = \min_{u_{i+1} \in D_{i+1}} \varphi_{i+1}(u_1, \dots, u_i, u_{i+1}), 1 \leq i \leq M-1.$$

будут многомерными, что является основным отличием от исходной многошаговой схемы. Для решения многомерных подзадач может быть использована схема редукции размерности на основе кривых Пеано.

### 3.3. Блочная адаптивная схема редукции размерности

Решение возникающего множества подзадач вида (15) (для многошаговой схемы) или вида (21) (для блочной многошаговой схемы) может быть организовано различными способами. Очевидный способ (детально проработанный в [28, 32, 33] для многошаговой схемы и в [31, 34] для блочной многошаговой схемы) основан на решении подзадач в соответствии с рекурсивным порядком их порождения. Однако здесь возникает потеря значительной части информации о целевой функции.

Иным подходом является адаптивная схема, в которой все подзадачи решаются одновременно, что позволяет более полно учитывать информацию о многомерной задаче и за счет этого ускорять процесс ее решения. Для случая одномерных подзадач данный подход был теоретически обоснован и апробирован в [30, 35, 36]. Настоящая работа предлагает обобщение адаптивной схемы для случая многомерных подзадач.

Перед изложением новой схемы организации вычислений еще раз отметим, что в рамках исходной многошаговой схемы (или же ее блочного варианта) порождаемые подзадачи решаются строго последовательно; получаемая в результате иерархическая схема порождения и решения подзадач (в виде дерева) представлена на рис. 1. Построение этого дерева происходит динамически в процессе решения исходной задачи (1). При этом вычисление одного значения функции  $\varphi_i(y_1, y_2, \dots, y_i)$  на  $i$ -м уровне требует полного решения всех задач одного из поддеревьев уровня  $i+1$ .

Адаптивная многошаговая схема редукции размерности изменяет порядок решения подзадач: они будут решаться не по одной (в соответствии с их иерархией в дереве задач), а одновременно, т.е. будет существовать некоторое множество подзадач, находящихся в процессе решения. В рамках новой схемы:

- для вычисления значения функции  $i$ -го уровня из (21) порождается новая задача уровня  $i + 1$ , в которой проводится только одно испытание, после чего новая порожденная задача включается в множество уже имеющихся задач, подлежащих решению;
- итерация глобального поиска состоит в выборе одной (наиболее перспективной) задачи из множества имеющихся задач, в которой проводится одно испытание; точка проведения нового испытания определяется в соответствии с алгоритмом из раздела 3.1;
- в качестве минимальных значений функций из (21) используются их текущие оценки, полученные на основе накопленной поисковой информации.

Краткое описание основных шагов блочной адаптивной схемы редукции размерности состоит в следующем. Пусть вложенные подзадачи вида (21) решаются с помощью алгоритма глобального поиска, описанного в разделе 3.1. Тогда каждой подзадаче (21) можно присвоить некоторое числовое значение, называемое характеристикой этой задачи. В качестве такой характеристики можно взять значение  $R(t)$  из (5), т.е. максимальную характеристику интервалов, сформированных в данной задаче. В соответствии с правилом вычисления характеристик (4), чем выше значение характеристики, тем более перспективной является подзадача для продолжения поиска в ней глобального минимума исходной задачи (1). Поэтому на каждой итерации выбирается подзадача с максимальной характеристикой для проведения в ней очередного испытания. Это испытание либо приводит к вычислению значения целевой функции  $\varphi(y)$  (если выбранная подзадача принадлежала уровню  $j = M$ ), либо порождает новые подзадачи согласно (21) при  $j \leq M - 1$ . В последнем случае новые порожденные задачи добавляются к текущему множеству задач, вычисляются их характеристики и процесс повторяется. Завершение процесса оптимизации происходит, когда в корневой задаче выполняется условие остановки алгоритма, решающего эту задачу.

Отметим новые элементы предлагаемой схемы редукции размерности.

- В адаптивной многошаговой схеме все порождаемые задачи решаются совместно – на каждой итерации выбирается одна (наиболее перспективная) подзадача из множества подзадач, в которой проводится одно испытание. Такой подход, с одной стороны, требует одновременного хранения поисковой информации во всех решаемых подзадачах. С другой стороны, он позволяет (при необходимости) увеличивать необходимую точность решения задачи (1) без проведения повторного решения всех подзадач (21). Например, начальная стадия оптимизации может быть выполнена с достаточно грубой точностью, что позволит быстро получить некоторую оценку решения; далее (после анализа результатов) точность может быть повышена и процесс глобального поиска может быть продолжен.
- В исходной многошаговой схемой редукции размерности решение каждой возникающей подзадачи проводится «до конца», т.е. порождения новой подзадачи не произойдет, пока не будет решена текущая подзадача. Но точка  $\bar{u}_i$  ее решения, вообще говоря, не будет совпадать с  $i$ -й компонентой вектора решения

Таблица 1. Среднее число испытаний на классах GKLS *Simple* и *Hard*,  $N = 2$

	GKLS <i>Simple</i>	GKLS <i>Hard</i>
$K_e$	252	674
$K_n$	697	1252
$K_a$	279	815

$y^* = (u_1^*, \dots, u_i^*, \dots, u_M^*)$  исходной многомерной задачи. В адаптивной схеме подобные бесперспективные подзадачи могут быть отбракованы уже на начальном этапе их решения.

- Наличие правила выбора подзадачи при выполнении очередной итерации в рамках адаптивной схемы позволяет получать различные ее варианты, соответствующие разным правилам выбора подзадачи. Например, как исходная многошаговая схема редукции размерности, так и схема редукции размерности с помощью единственной кривой Пеано могут быть получены как частные случаи нового подхода.
- Существование множества одновременно решаемых подзадач позволяет организовать их параллельное решение и эффективно задействовать ресурсы современных вычислительных систем. Некоторые результаты в данном направлении представлены в [37].

#### 4. Результаты экспериментов

Стандартный подход к сравнению алгоритмов глобальной оптимизации основан на решении указанными методами серии задач, выбранных случайно из некоторого класса. Будем использовать два класса многоэкстремальных тестовых функций (GKLS *Simple*, GKLS *Hard*), описанные в [38].

В работах [36, 39] было экспериментально установлено, что алгоритм глобального поиска (АГП) как с использованием разверток, так и в сочетании с адаптивной схемой редукции размерности превосходит многие известные алгоритмы глобальной оптимизации, включая методы DIRECT и DIRECT $l$ . Поэтому в данном исследовании мы ограничимся сравнением вариантов АГП в сочетании с различными схемами редукции размерности.

Для сравнения эффективности работы алгоритмов будем использовать два критерия: среднее число испытаний и операционные характеристики. Операционной характеристикой алгоритма называется функция  $P(k)$ , определяемая как доля задач из рассматриваемой серии, для решения которых потребовалось не более чем  $k$  испытаний. Задачу будем считать решенной, если алгоритм сгенерировал точку  $y^k$  очередного испытания в окрестности решения  $y^*$ , т.е.  $\|y^k - y^*\| < \delta \|b - a\|$ , где  $\delta = 0.01$ ,  $a$  и  $b$  — границы области поиска  $D$ .

Первая серия экспериментов была проведена на двумерных задачах классов GKLS *Simple* и GKLS *Hard* (100 функций каждого класса). В табл. 1 представлено среднее число испытаний, выполненных АГП с использованием разверток ( $K_e$ ), многошаговой схемы ( $K_n$ ) и адаптивной многошаговой схемы ( $K_a$ ). На рис. 2(а,б) приведены операционные характеристики алгоритмов, полученные на указанных классах



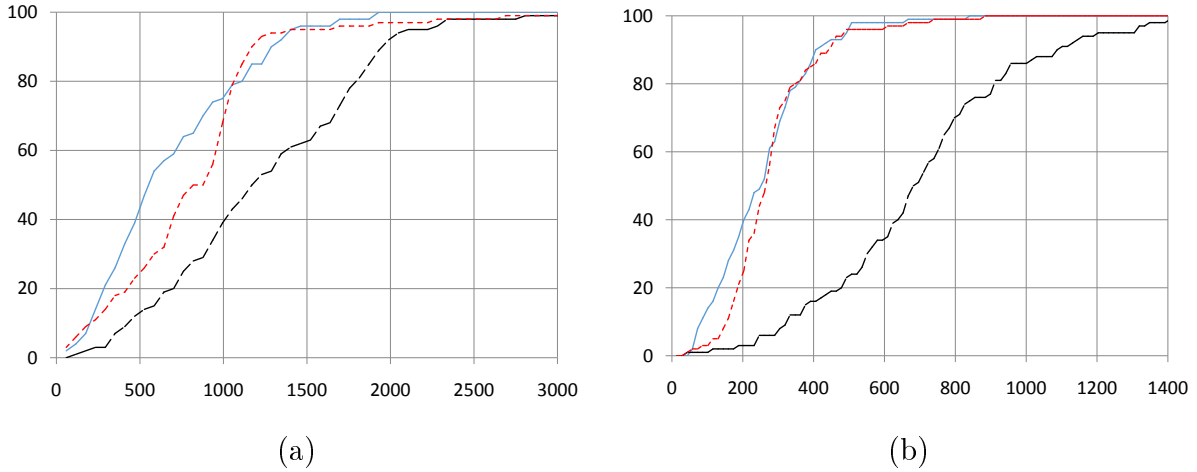


Рис. 2. Операционные характеристики на классах GKLS *Simple* (a) и *Hard* (b),  $N = 2$

Таблица 2. Среднее число испытаний на классах GKLS *Simple* и *Hard*,  $N = 4$

	GKLS <i>Simple</i>	GKLS <i>Hard</i>
$K_a$	21 747	35 633
$K_{ba}$	15 942	33 206

задач. Непрерывная линия соответствует алгоритму с использованием разверток, короткий пунктир — адаптивной многошаговой схеме, длинный пунктир — многошаговой схеме. Результаты экспериментов показывают, что АГП с использованием адаптивной многошаговой схемы демонстрирует примерно одинаковое быстроедействие по сравнению с развертками, и оба они значительно превосходят алгоритм, использующий многошаговую схему. Поэтому в дальнейших экспериментах мы ограничимся сравнением различных вариантов адаптивной схемы редукции размерности.

Вторая серия экспериментов была проведена на четырехмерных задачах классов GKLS *Simple* и GKLS *Hard* (100 функций каждого класса). В табл. 2 представлено среднее число испытаний, выполненных АГП с использованием адаптивной многошаговой схемы ( $K_a$ ) и блочной адаптивной схемы ( $K_{ba}$ ) с формированием двух уровней подзадач одинаковой размерности  $N_1 = N_2 = 2$ . Отметим, что при использовании исходного варианта адаптивной многошаговой схемы при решении задачи размерности  $N = 4$  формируется четыре уровня одномерных подзадач, что усложняет их обработку.

На рис. 3(a,b) приведены операционные характеристики алгоритмов, полученные на классах GKLS *Simple* и GKLS *Hard*. Пунктирная линия соответствует алгоритму с использованием адаптивной, а непрерывная — блочной адаптивной схемы. Результаты экспериментов показывают, что использование блочной адаптивной схемы дает ощутимый выигрыш по числу испытаний (до 35%) по сравнению с исходной схемой редукции размерности.

Последняя серия экспериментов была проведена на шестимерных задачах класса GKLS *Simple* (100 функций). Сравнивалась работа АГП с использованием разверток и блочной адаптивной схемы с формированием двух уровней подзадач одинаковой размерности  $N_1 = N_2 = 3$ . Среднее число испытаний, выполненных АГП с

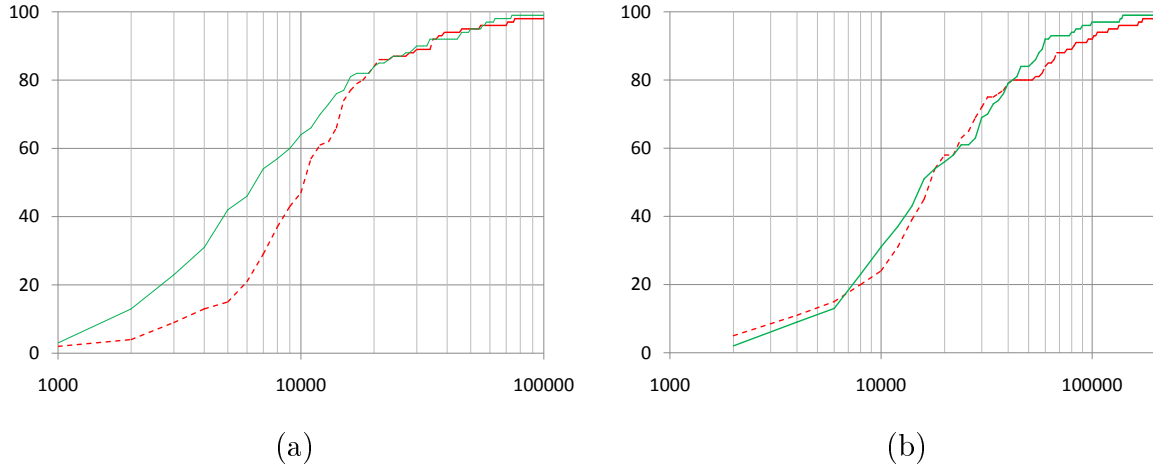


Рис. 3. Операционные характеристики на классах GKLS *Simple* (a) и *Hard* (b),  $N = 4$

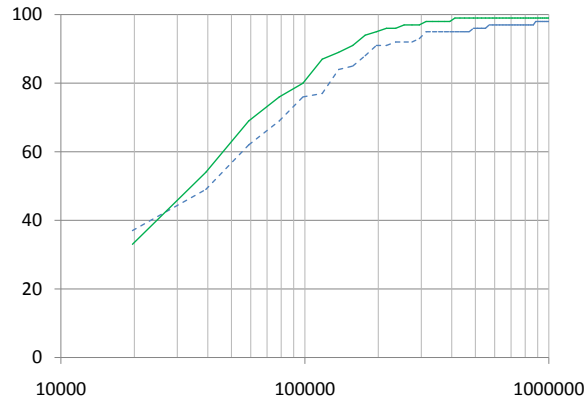


Рис. 4. Операционные характеристики на классе GKLS *Simple*,  $N = 6$

использованием разверток, составило 102 987, тогда как с использованием блочной адаптивной схемы — 75 390. На рис. 4 приведены операционные характеристики алгоритмов. Пунктирная линия соответствует алгоритму с использованием разверток, а непрерывная — блочной адаптивной схемы.

## 5. Заключение

В данной работе предложена обобщенная адаптивная схема редукции размерности для задач глобальной оптимизации, комбинирующая использование кривых Пеано и схему вложенной (рекурсивной) оптимизации. Для решения редуцированных подзадач используется алгоритм глобального поиска. Приведена вычислительная схема алгоритма, рассмотрены основные вопросы, связанные с использованием адаптивной схемы редукции размерности. Проведены вычислительные эксперименты на серии тестовых задач с целью сравнения эффективности различных схем редукции размерности. Результаты экспериментов показывают, что использование блочной адаптивной схемы редукции размерности может значительно сократить число испытаний, необходимое для решения задачи с заданной точностью.

## СПИСОК ЛИТЕРАТУРЫ

1. *Евтушенко Ю.Г.* Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Ж. вычисл. матем. и матем. физ. 1971. Т. 11, №6, С. 1390–1403.
2. *Пуявский С.А.* Один алгоритм отыскания абсолютного экстремума функций // Ж. вычисл. матем. и матем. физ. 1972. Т. 12, № 4. С. 888–896.
3. *Shubert B.O.* A sequential method seeking the global maximum of a function // SIAM J. Numer. Anal. 1972. V. 9. P. 379–388.
4. *Jones D.R., Perttunen C.D., Stuckman B.E.* Lipschitzian optimization without the Lipschitz constant // J. Optim. Theory Appl. 1993. V. 79. No. 1. P. 157–181.
5. *Pinter J. D.* Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). Dordrecht: Kluwer Academic Publishers, 1996.
6. *Žilinskas J.* Branch and bound with simplicial partitions for global optimization // Math. Model. Anal. 2008. V. 13. No. 1. P. 145–159.
7. *Евтушенко Ю.Г., Малкова В.У., Станевичюс А.-И. А.* Распараллеливание процесса поиска глобального экстремума // Автомат. и телемех. 2007. № 5. С. 46–58
8. *Евтушенко Ю.Г., Малкова В.У., Станевичюс А.-И. А.* Параллельный поиск глобального экстремума функций многих переменных // Ж. вычисл. матем. и матем. физ. 2009. Т. 49, №2. С. 255–269.
9. *Jones D. R.* The DIRECT global optimization algorithm. In: Floudas C. A., Pardalos P. M. (eds.) The Encyclopedia of Optimization, Second Edition. Springer. 2009. P. 725–735.
10. *Paulavičius R., Žilinskas J., Grothey A.*: Investigation of selection strategies in branch and bound algorithm with simplicial partitions and combination of Lipschitz bounds // Optim. Lett. 2010. V. 4(1). P. 173–83
11. *Evtushenko Y. G., Posypkin M. A.* A deterministic approach to global box-constrained optimization // Optim. Lett. 2013. V. 7(4). P. 819–829
12. *Квасов Д.Е., Сергеев Я.Д.* Методы липшицевой глобальной оптимизации в задачах управления // Автомат. и телемех. 2013. № 9. С. 3–19.
13. *Paulavičius R., Žilinskas J.* Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints // Optim. Lett. 2016. V. 10(2). P. 237–246.
14. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. Учебное пособие. М.: ФИЗМАТЛИТ, 2004.

15. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие. М.: Издательство МГТУ им. Н. Э. Баумана, 2014.
16. *Kvasov D.E., Mukhametzhanov M.S.* Metaheuristic vs. deterministic global optimization algorithms: The univariate case // Appl. Math. Comput. 2018. V. 318. P. 245–259.
17. *Sergeyev Y., Kvasov D., Mukhametzhanov M.* On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // Sci. Rep. V. 8(1). Art. No. 435.
18. *Неймарк Ю.И., Стронгин Р.Г.* Поиск экстремума функций по принципу максимума информации // Автомат. и телемех. 1966. № 11. С. 113–118.
19. *Неймарк Ю.И., Стронгин Р.Г.* Информационный подход к задаче поиска экстремума функций // Изв. АН СССР, Техническая кибернетика. 1966. № 1. С. 17–26.
20. *Стронгин Р.Г.* Многоэкстремальная минимизация // Автомат. и телемех. 1970. № 7. С. 63–67.
21. *Стронгин Р.Г.* Численные методы в многоэкстремальных задачах (информационно-статистические алгоритмы). М.: Наука, 1978.
22. *Стронгин Р.Г., Маркин Д.Л.* Минимизация многоэкстремальных функций при невыпуклых ограничениях // Кибернетика. 1986. №4. С.63–69.
23. *Маркин Д.Л., Стронгин Р.Г.* Метод решения многоэкстремальных задач с невыпуклыми ограничениями, использующий априорную информацию об оценках оптимума // Ж. вычисл. матем. и матем. физ. 1987. Т.27, № 1. С.52–61.
24. *Маркин Д.Л., Стронгин Р.Г.* О равномерной оценке множества слабоэффективных точек в многоэкстремальных многокритериальных задачах оптимизации // Ж. вычисл. матем. и матем. физ. 1993 Т. 33, № 2. С. 195–205.
25. *Стронгин Р.Г., Гегель В.П., Гришагин В.А., Баркалов К.А.* Параллельные вычисления в задачах глобальной оптимизации. М.: Издательство Московского университета, 2013.
26. *Стронгин Р.Г.* Параллельная многоэкстремальная оптимизация с использованием множества разветок // Ж. вычисл. матем. и матем. физ. 1991. Т. 31, № 8. С. 1173–1185.
27. *Strongin R.G., Sergeyev Ya.D.* Global Optimization with Non-Convex Constraints. Sequential and Parallel Algorithms. Dordrecht: Kluwer Academic Publishers, 2000.
28. *Grishagin V.A., Sergeyev Y.D., Strongin R.G.* Parallel characteristical algorithms for solving problems of global optimization // J. Glob. Optim. 1997. V. 10(2). P. 185–206.
29. *Sergeyev Y., Grishagin V.* Parallel asynchronous global search and the nested optimization scheme // J. Comput. Anal. Appl. 2001. V. 3(2). P. 123–145.

30. *Gergel V., Grishagin V., Gergel A.* Adaptive nested optimization scheme for multidimensional global search // J. Glob. Optim. 2016. V. 66(1). P. 35–51.
31. *Barkalov K., Gergel V.* Multilevel scheme of dimensionality reduction for parallel global search algorithms // In: OPT-i 2014 – 1st International Conference on Engineering and Applied Sciences Optimization, Proceedings. 2014. P. 2111–2124.
32. *Sergeyev Y., Grishagin V.* Parallel asynchronous global search and the nested optimization scheme // J. Comput. Anal. Appl. 2001. V. 3(2). P. 123–145.
33. *Gergel V., Grishagin V., Israfilov R.* Local tuning in nested scheme of global optimization // Procedia Computer Science. 2015. V. 51(1). P. 865–874
34. *Barkalov K., Lebedev I.* Solving multidimensional global optimization problems using graphics accelerators // CCIS. 2016. V. 687. P. 224–235.
35. *Grishagin V., Israfilov R., Sergeyev Y.* Comparative efficiency of dimensionality reduction schemes in global optimization // AIP Conference Proceedings. 2016. V. 1776, art. no. 060011.
36. *Grishagin V., Israfilov R., Sergeyev Y.* Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes // Appl. Math. Comput. 2018. V. 318. P. 270–280.
37. *Gergel V., Grishagin V., Israfilov R.* Parallel dimensionality reduction for multiextremal optimization problems // LNCS. 2019. V. 11657. P. 166–178.
38. *Gaviano M., Lera D., Kvasov D.E., Sergeyev Ya.D.* Software for generation of classes of test functions with known local and global minima for global optimization // ACM Trans. Math. Software. 2003. V. 29. P. 469–480.
39. *Sovrasov V.* Comparison of several stochastic and deterministic derivative-free global optimization algorithms // LNCS. 2019. V. 11548. P. 70–81.

Стронгин Р.Г., *Нижегородский государственный университет им. Н.И. Лобачевского*, президент, *Нижний Новгород*, [strongin@unn.ru](mailto:strongin@unn.ru)

Гергель В.П., *Нижегородский государственный университет им. Н.И. Лобачевского*, директор института информационных технологий, математики и механики, *Нижний Новгород*, [gergel@unn.ru](mailto:gergel@unn.ru)

Баркалов К.А., *Нижегородский государственный университет им. Н.И. Лобачевского*, доцент кафедры математического обеспечения и суперкомпьютерных технологий института информационных технологий, математики и механики, *Нижний Новгород*, [konstantin.barkalov@itmm.unn.ru](mailto:konstantin.barkalov@itmm.unn.ru)