



УНИВЕРСИТЕТ  
ЛОБАЧЕВСКОГО

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

**Соврасов В.В.**, научный руководитель: Баркалов К.А.

ННГУ им. Н.И. Лобачевского

14 сентября, 2021

Нижний Новгород

# Структура работы

Глава 1. Методы редукции размерности, их сравнение

Глава 2. Сравнение актуальных методов глобальной оптимизации

Глава 3. Метод для решения множества задач с ограничениями

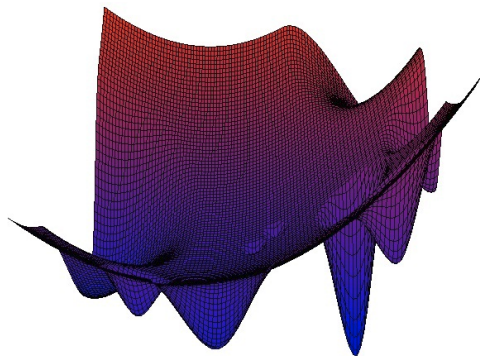
# Постановка задачи глобальной оптимизации

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

где  $D$  – прямоугольник в  $\mathbb{R}^N$ ,  $\varphi(y)$ ,  $g_j(y)$  – многоэкстремальные функции, удовлетворяющие условию Липшица в  $D$ :

$$|f(y_1) - f(y_2)| \leq L\|y_1 - y_2\|, y_1, y_2 \in D,$$

где  $L > 0$  – константа Липшица, а  $\|\cdot\|$  обозначает  $l_2$ -норму в  $\mathbb{R}^N$ .



# Постановка задачи глобальной оптимизации

Рассмотрим множество из  $q$  глобальной оптимизации с ограничениями-неравенствами:

$$\min \{ \varphi_1(y), y \in G_1 \}, \min \{ \varphi_2(y), y \in G_2 \}, \dots, \min \{ \varphi_q(y), y \in G_q \}. \quad (1)$$

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:

$$\exists \varepsilon > 0 : \forall s > 1, \forall i, j \in \{1, \dots, q\} \frac{\|\bar{y}_i^*(s) - y_i^*\|_\infty}{\|\bar{y}_j^*(s) - y_j^*\|_\infty} \leq \varepsilon, \quad (2)$$

где  $s$  это количество итераций метода оптимизации,  $\tilde{y}_i(s)^*$  это приближение к решению, полученное методом в задаче  $i$  из множества (1) на итерации  $s$ .

Упомянутые множества задач могут возникнуть, например, в следующих случаях:

- ▶ наличие у непрерывной задачи дискретного параметра, который принимает ограниченный диапазон значений;
- ▶ набор задач получен при скаляризации задачи многокритериальной оптимизации.

**Возможные подходы:**

- ▶ решать каждую задачу независимо;
- ▶ разработать метод оптимизации, решающий все задачи "одновременно", в каждый момент времени приоритизируя одну из них.

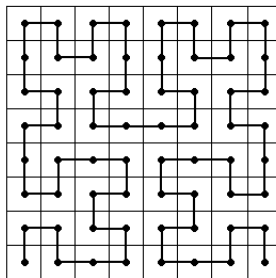
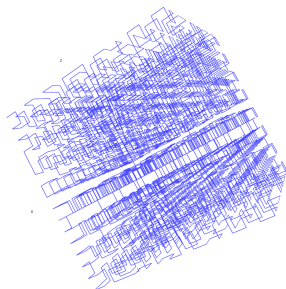
# Редукция размерности с помощью развёрток

Кривая типа Пеано  $y(x)$  позволяет снизить размерность исходной задачи оптимизации:

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

$$\min\{\varphi(y) : y \in D\} = \min\{\varphi(y(x)) : x \in [0, 1]\}$$

$y(x)$  является негладкой непрерывной функцией, отображающей отрезок  $[0, 1]$  на гиперкуб  $D_e$ .

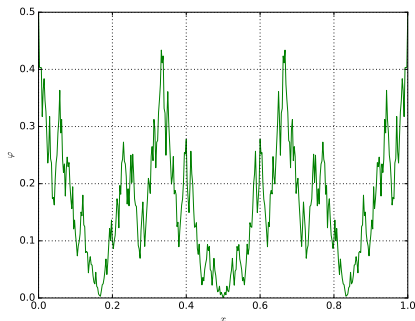
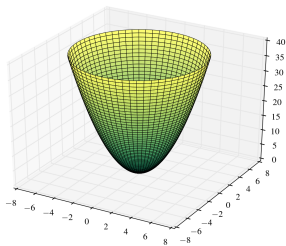


## Свойства задачи после редукции

После применения развёртки,  $\varphi(y(x))$  удовлетворяет условию Гёльдера:

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

$\varphi(y(x))$  является негладкой функцией с множеством локальных и глобальных экстремумов **global**, даже если  $\varphi(y)$  унимодальна. Появление дополнительных экстремумов связано с потерей информации о  $N$ -мерных окрестностях точек после применения отображения в одномерное пространство.



## Схема одномерного характеристического алгоритма оптимизации

Метод генерирует последовательность точек  $\{x_k : x_k \in [a, b]\}$ , выполняя следующие шаги:

- Шаг 1. Отсортировать поисковую информацию (одномерные точки) в порядке возрастания координаты.
- Шаг 2. Для каждого интервала  $(x_{i-1}, x_i)$  вычислить величину  $R(i)$ , называемую характеристика.
- Шаг 3. Выбрать интервал с наибольшей характеристикой,  $(x_{t-1}, x_t)$ , и произвести испытание (вычислить ограничения и целевую функцию) в точке  $x^{k+1}$ , выбранной в соответствии с решающим правилом  $d$ :

$$x^{k+1} = d(t) \in (x_{t-1}, x_t)$$

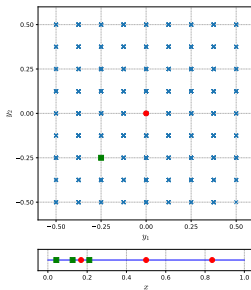
- Шаг 4. Если верно  $x_t - x_{t-1} < \varepsilon$ , остановить работу метода.

Описание характеристического метода AGS приведено в Strongin R.G., Sergeyev Ya.D.: *Global optimization with non-convex constraints. Sequential and parallel algorithms (2000), Chapter 7*



# Неинъективная развёртка

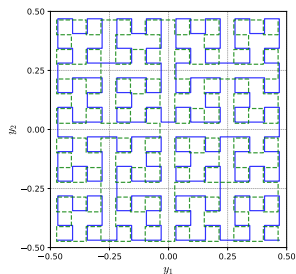
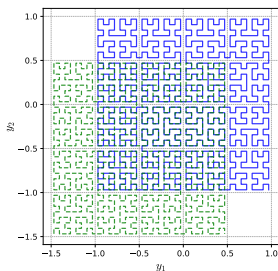
Одним из подходов к восстановлению потерянной информации является вычисление всех прообразов точки  $y \in \mathbb{R}^N$  и их учёт в методе оптимизации<sup>1</sup>. Это позволяет снизить влияние эффекта расщепления локальных экстремумов. Каждая  $N$ -мерная точка может иметь до  $2^N$  прообразов, поэтому для большого значения  $N$  вычисление и учёт их всех становится трудоёмким.



<sup>1</sup>Р.Г. Стронгин. Численные методы в многоэкстремальных задачах, 1978

## Сдвиговые и вращаемые развёртки

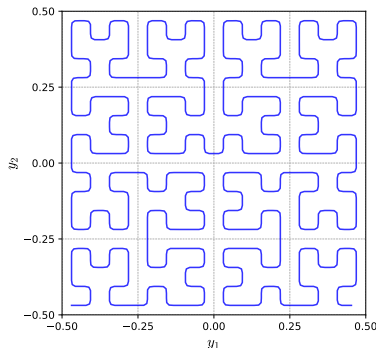
Получить фиксированное количество одномерных прообразов можно, например, используя заранее выбранный набор инъективных развёрток. Подобный набор может быть получен путём сдвига или вращения оригинальной развёртки. Набор сдвиговых развёрток<sup>2</sup> обладает свойством генерировать как минимум одну пару близких прообразов, для двух близких многомерных точек. Для вращаемых разветок подобного утверждения доказано не было, но на практике их использование также приносит ускорение сходимости.



<sup>2</sup>Strongin, R.G. Algorithms for multi-extremal mathematical programming problems employing the set

# Гладкая развёртка

Поскольку гладкие функции ведут себя более предсказуемо, гладкая аппроксимация кривой Пеано  $y(x)$  может ускорить сходимость процесса оптимизации <sup>3</sup>.



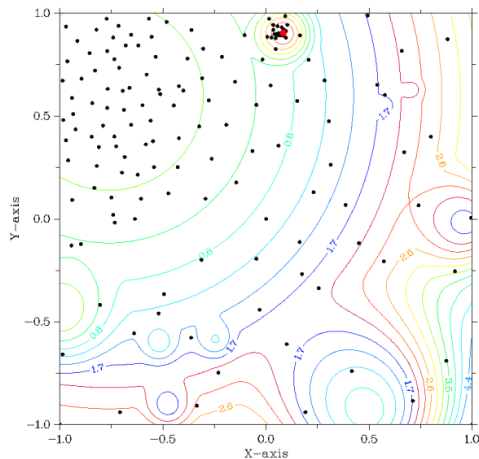
<sup>3</sup>Goryachih, A. A class of smooth modification of space-filling curves for global optimization problems, NET 2016

# Класс тестовых задач

Генератор GKLS (Gaviano, M. et al, 2003):

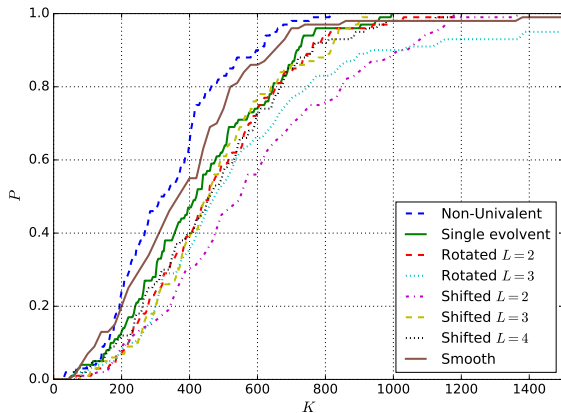
$$f(x) = \begin{cases} C_i(x), & x \in S_i, i \in 2, \dots, m \\ \|x - T\|^2 + t, & x \notin S_2, \dots, S_m \end{cases}$$

- ▶ варьируемое число локальных минимумов;
- ▶ варьируемый размер области притяжения глобального минимума;
- ▶ размерность функции также задаётся.

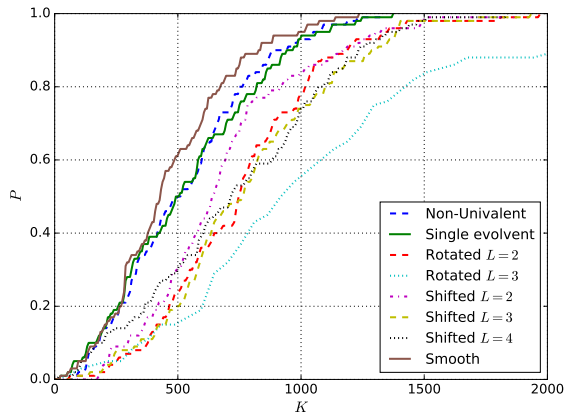


Также был использован один набор из 100 фиксированных многоэкстремальных функций (с 10-30 локальными минимумами). Обозначим его  $F_{GR}$ .

# Сравнение развёрток



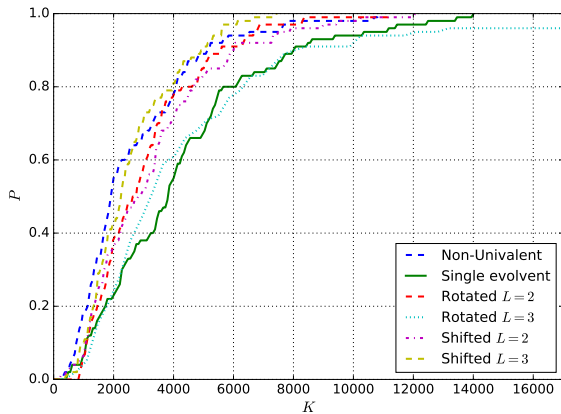
Minimal  $r$



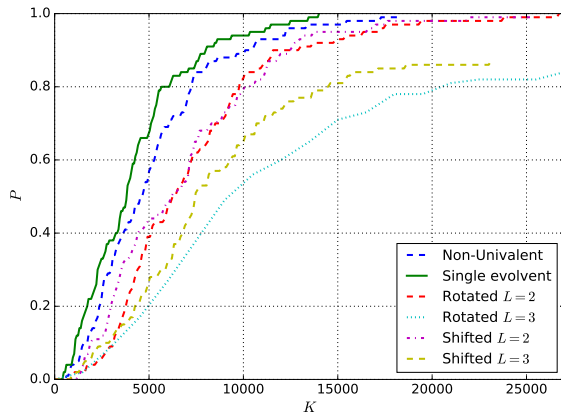
$r = 5.0$

Операционные характеристики на классе задач GKLS 2d Simple

# Сравнение развёрток



Minimal  $r$



$r = 4.5$

Операционные характеристики на классе задач GKLS 3d Simple

Рассмотрим следующие методы, реализации которых доступны в исходных кодах:

□ **Детерминированные:**

- ▶ DIRECT, DIRECTl (Gablonsky J. M. et al, 2001);
- ▶ AGS (Strongin R.G., 1978).

□ **Стохастические:**

- ▶ Multi Level Single Linkage (Alexander H. G. et al, 1987);
- ▶ StoGO (Madsen, Kaj at al, 1998);
- ▶ Differential Evolution (Storn, Rainer et al, 1997);
- ▶ Controlled Random Search (Price, W. L., 1983);
- ▶ Dual Simulated Annealing (Y Xiang et al, 1997);

## Настройка параметров в методе AGS

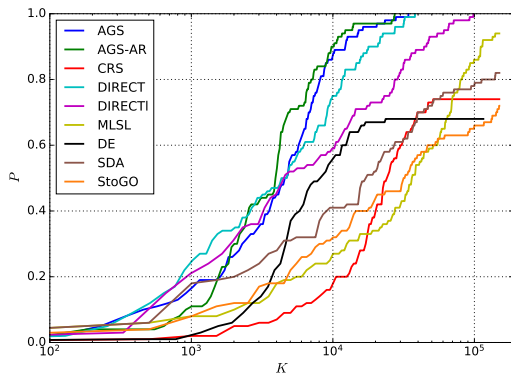
С целью снизить зависимость метода от множителя оценки константы Гёльдера  $r$ , рассмотрим следующую схему:

- ▶ выполнить  $q$  итераций AGS с  $r = r_{max}$ ;
- ▶ выполнить  $q$  итераций AGS с  $r = r_{min}$ ;
- ▶ выполнять указанные шаги до наступления сходимости или до исчерпания отведённого на вычисления времени.

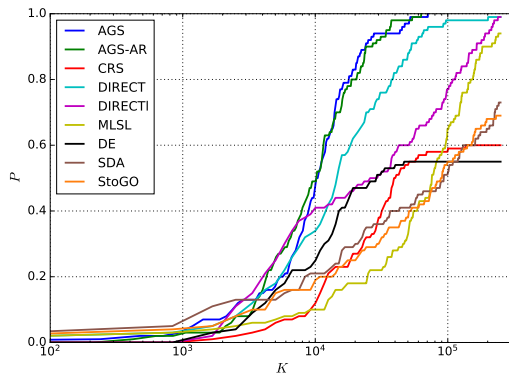
В результате появляются три параметра вместо одного, но метод не столь чувствителен к ним. Примем  $q = 50 \cdot \log_2(N - 1) \cdot N^2$ ,  $r_{min} = 3$ ,  $r_{max} = 2 \cdot r_{min}$ .



# Результаты сравнения методов

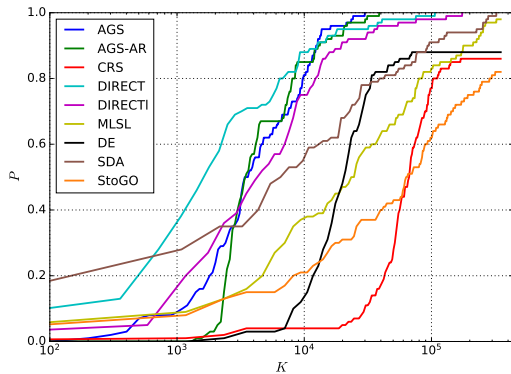


4d Simple

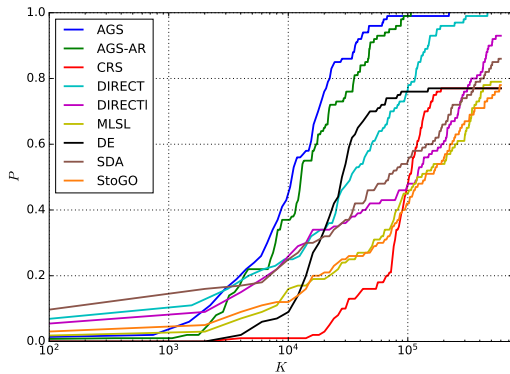


4d Hard

# Результаты сравнения методов



5d Simple



5d Hard

## Метод, решающий множество задач

Метод опирается на ранее упомянутый алгоритм AGS (BarkalovK., StronginR., 2018):

- ▶ Создать  $q$  копий AGS, каждая из которых получает по одной задаче.
- ▶ Запустить  $q$  синхронно работающих копий AGS, но вместо выполнения Шага 3, остановить все копии и выбрать интервал с наибольшей характеристикой среди интервалов во всех задачах.
- ▶ Если выбранный на предыдущем шаге интервал принадлежит задаче  $i$ , выполнить Шаг 3 в  $i^{th}$  копии AGS. Остальные копии пропускают Шаг 3.
- ▶ Чтобы обеспечить параллельность вычислений, метод выбирает  $p$  интервалов на предыдущих двух шагах и выполняет  $p$  испытаний параллельно.

Описанный алгоритм предполагает **сравнимость характеристик во всех задачах**.

# Индексная схема учёта ограничений

Вместо исходной задачи с функциональными ограничениями, индексная схема рассматривает следующую задачу без ограничений:

$$\begin{aligned}\psi(x^*) &= \min_{x \in [0;1]} \psi(x), \\ \psi(x) &= \begin{cases} g_\nu(x)/H_\nu & \nu < M \\ (g_M(x) - g_M^*)/H_M & \nu = M \end{cases},\end{aligned}$$

где при  $\nu = m + 1$ ,  $g_\nu(x) = \varphi(x)$ .

Характеристики алгоритма глобального поиска при использовании индексной схемы (IAGS) имеют следующий вид:

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

где  $z_i = \psi(x_i)$ .

Подробное описание индексной схемы приведено в Strongin R.G., Sergeyev Ya.D.: *Global optimization with non-convex constraints. Sequential and parallel algorithms (2000), Chapter 7*

## Достаточные условия сходимости AGS:

1.  $G \neq \emptyset$ , исходная задача имеет решение.
2. Функции  $g_j(y) \leq 0, 1 \leq j \leq m+1$ , липшицевы с константами  $L_i$  в гиперинтервале  $D$  (здесь  $g_{m+1}(y) = \varphi(y)$ ).
3. Начиная с некоторого момента значения оценок констант Гёльдара  $i$  становятся достаточно большими.

**Достаточные условия сходимости MIAGS:** пусть указанные выше условия выполнены для каждой задачи  $i, 1 \leq i \leq q$  из (1), 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

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

## Пример решения многокритериальной задачи

Рассмотрим следующую задачу:

$$\begin{aligned} & \text{Minimize } \begin{cases} f_1(y) = 4y_1^2 + 4y_2^2 \\ f_2(y) = (y_1 - 5)^2 + (y_2 - 5)^2 \end{cases} \quad y_1 \in [-1; 2], y_2 \in [-2; 1] \\ & \text{s.t.} \\ & \begin{cases} g_1(y) = (y_1 - 5)^2 + y_2^2 - 25 \leq 0 \\ g_2(y) = -(y_1 - 8)^2 - (y_2 + 3)^2 + 7.7 \leq 0 \end{cases} \end{aligned}$$

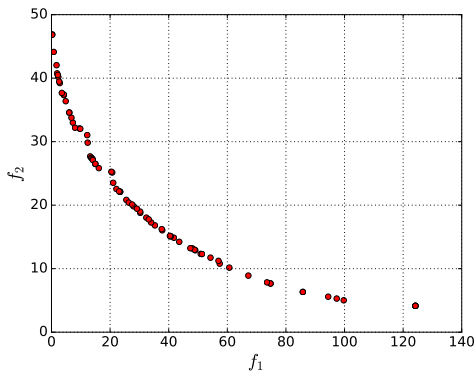
После скаляризации задача примет следующую форму:

$$\varphi(y^*(\lambda_1, \lambda_2)) = \min_{y \in D} \max\{\lambda_1 f_1(y), \lambda_2 f_2(y)\}; \lambda_1, \lambda_2 \in [0, 1], \lambda_1 + \lambda_2 = 1$$

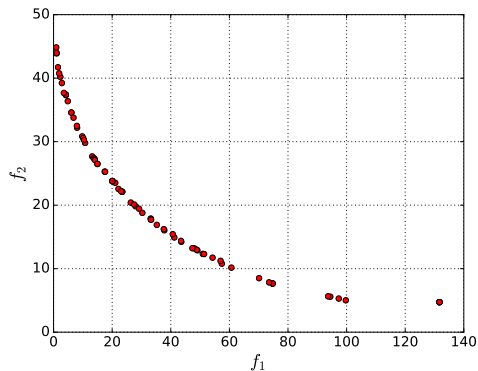
Для численного построения Парето фронта рассмотрим 100 пар  $(\lambda_1, \lambda_2)$  таких, что  $\lambda_1^i = ih$ ,  $\lambda_2^i = 1 - \lambda_1^i$ ,  $h = 10^{-2}$ ,  $i = \overline{1, 100}$ .

## Пример решения многокритериальной задачи

$$SP(S) = \sqrt{\frac{1}{|S|-1} \sum_{i=1}^{|S|} (\bar{d} - d_i)^2}, \quad \bar{d} = \text{mean}\{d_i\},$$
$$d_i = \min_{s_i, s_j \in S: s_i \neq s_j} \|F(s_i) - F(s_j)\|_1, \quad F = (f_1, f_2)$$



IAGS,  $SP_{single} = 0.984$



MIAGS,  $SP_{multi} = 0.749$

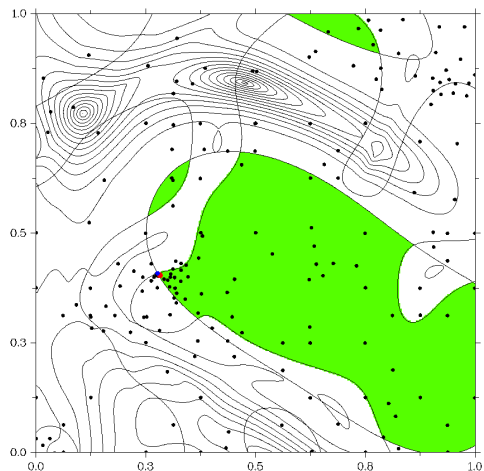
Численные оценки множества Парето, полученные после 2500 испытаний

# Решение тестовых задач с ограничениями

Наборы задач с ограничениями были получены путём применения системы GCGen к функциям, порождённым генераторами задач без ограничений. Основные свойства полученных наборов задач:

- ▶  $q=100$ ;
- ▶ Размерность 2, 3 или 4;
- ▶ Базовые функции GKLS или  $F_{GR}$ , или их комбинация;
- ▶ Количество нелинейных ограничений: 2.

GCGen is available here: <https://github.com/UNN-ITMM-Software/GCGen>



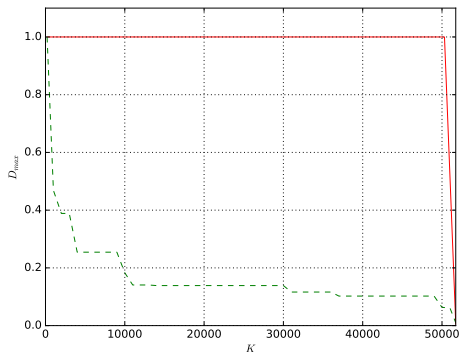


Параллельный метод реализован на языке C++ с использованием технологии OpenMP для параллельных вычислений на общей памяти.

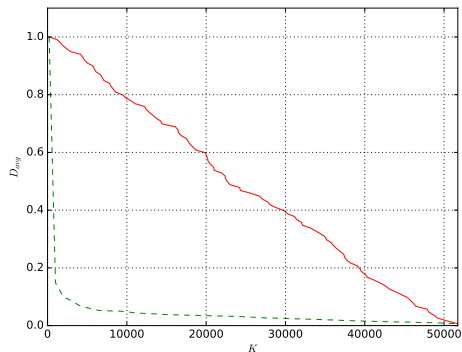
Все численные эксперименты были выполнены на машине со следующей конфигурацией: Intel Core i7-7800X (6 cores) CPU, 64GB RAM, Unubtu 16.04 OS, GCC 5.5 compiler.

В целевые функции и ограничения во всех задачах вносились задержки так, чтобы вычисление каждой функции занимало 1 мс.

# Результаты решения тестовых задач



$D_{max}$



$D_{avg}$

Изменение средней и максимальной ошибки, при решении смешанного класса задач, полученного с помощью двух генераторов: GKLS и  $F_{GR}$

## Результаты решения тестовых задач

Класс задач	$p$	Количество итераций	Время, с	$S_i$	$S_t$
GKLS & $F_{GR}$ -based	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
GKLS-based 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

## Результаты решения тестовых задач

Класс задач	$p$	Количество итераций	Время, с	$S_i$	$S_t$
GKLS-based 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
GKLS-based 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

## Заключение

В ходе выполнения работы были получены следующие результаты:

1. Произведено сравнение различных способов редукции размерности, основанных на отображениях типа кривой Пеано;
2. Предложена модификация AGS, AGS-AR, менее чувствительная к параметрам; 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. Программная реализация метода AGS-AR прошла процедуру ревью и была включена в состав популярной библиотеки алгоритмов нелинейной оптимизации NLOpt;
3. 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.

## Q&A

sovrasov.vladislav@itmm.unn.ru  
<https://github.com/sovrasov>