



УНИВЕРСИТЕТ
ЛОБАЧЕВСКОГО

Параллельный алгоритм для получения равномерного приближения решений множества задач глобальной оптимизации с нелинейными ограничениями

В.В. Соврасов К.А. Баркалов

Нижегородский государственный университет им. Н.И. Лобачевского

28 мая 2020 г.
Нижний Новгород

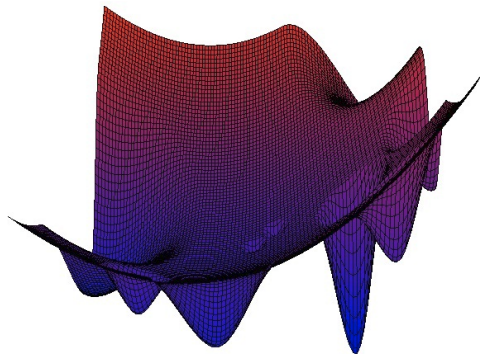
Постановка задачи

$$\begin{aligned}\varphi(y^*) &= \min\{\varphi(y) : y \in D\}, \\ D &= \{y \in \mathbb{R}^N : a_i \leq y_i \leq b_i, \\ &1 \leq i \leq N, g_j(y) \leq 0, 1 \leq j \leq m\}\end{aligned}$$

$\varphi(y)$, $g_j(y)$ – многоэкстремальные функции, удовлетворяющие условию Липшица:

$$\begin{aligned}|f(y_1) - f(y_2)| &\leq L\|y_1 - y_2\|, \\ y_1, y_2 &\in \mathbb{R}^N : a_i \leq y_i \leq b_i,\end{aligned}$$

где $L > 0$ константа Липшица и $\|\cdot\|$ обозначает l_2 норму в пространстве \mathbb{R}^N .



Постановка задачи

Далее будем интересоваться решением серии из q задач глобальной оптимизации с нелинейными ограничениями:

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

Подобные серии задач могут возникнуть, например, в следующих случаях:

- ▶ задача глобальной оптимизации с дискретным параметром;
- ▶ решение задачи многокритериальной оптимизации методом свертки критериев.

Возможные методы решения:

- ▶ решать каждую задачу независимо;
- ▶ разработать метод оптимизации, который решает все задачи из множества в совокупности, в каждый момент времени приоритизируя одну из них.

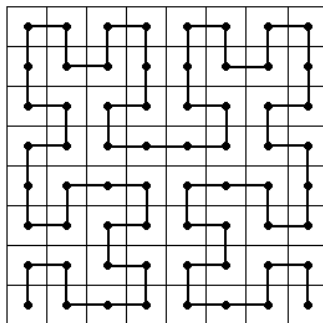
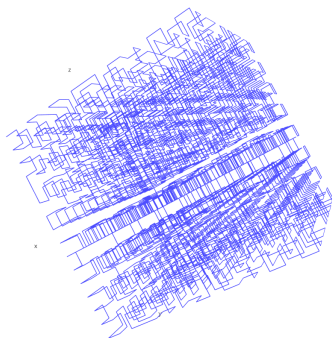
Редукция размерности

Кривая Пеано $y(x)$ позволяет уменьшить размерность многомерного пространства до 1:

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

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

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



Алгоритм глобальной оптимизации

Метод оптимизации генерирует последовательность точек $\{x_k : x_k \in [a, b]\}$ и состоит в выполнении следующих шагов:

- Шаг 1. Упорядочить поисковую информацию (одномерные точки) по возрастанию.
- Шаг 2. Для каждого интервала (x_{i-1}, x_i) вычислить величину $R(i)$, называемую характеристикой.
- Шаг 3. Выбрать интервал (x_{t-1}, x_t) с наибольшей характеристикой и провести испытание (вычислить органичения и целевую функцию) в точке x^{k+1} , выбранной с помощью решающего правила d :

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

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

Детальное описание метода: Strongin R.G., Sergeyev Ya.D.: Global optimization with non-convex constraints. Sequential and parallel algorithms (2000), Chapter 7

Алгоритм, решающий множество задач

Модификация метода для решения множества задач:

- ▶ Создать q копий характеристического АГО.
- ▶ Использовать q синхронно работающих копий АГО с тем лишь отличием, что на шаге 3 при выборе интервала с наилучшей характеристикой, выбор будет осуществляться из всех интервалов, которые породили на данный момент q копий АГО.
- ▶ Если наибольшая характеристика соответствует задаче i , то выполняется шаг 3 в копии метода с номером i , а остальные копии метода простаивают.
- ▶ Для реализации параллельности в предыдущем пункте можно выбрать p характеристик и провести испытания в p задачах одновременно.

Чтобы приведённая схема работала, требуется **сравнимость характеристик интервалов в разных задачах**.

Индексная схема учёта ограничений

Вместо исходной задачи рассмотрим следующую задачу без функциональных 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 $\nu = m + 1$ $g_\nu(x) = \varphi(x)$.

При использовании индексной схемы характеристики определяются следующим образом:

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

где $z_i = \psi(x_i)$. Можно показать, что при использовании таких характеристик метод будет выбирать интервалы из всех q задач, тем самым решая их все, а не только некоторое подмножество задач.

Пример решения многокритериальной задачи

Рассматриваемая задача:

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

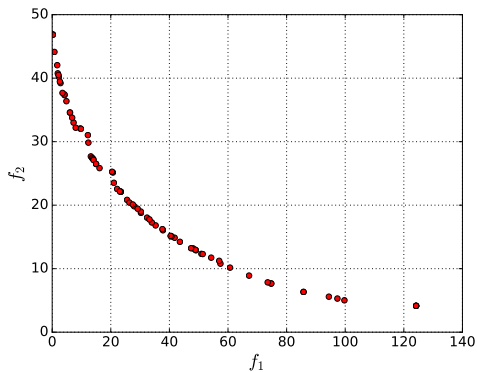
После использования свёртки Гермейера для скаляризации задача примет вид:

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

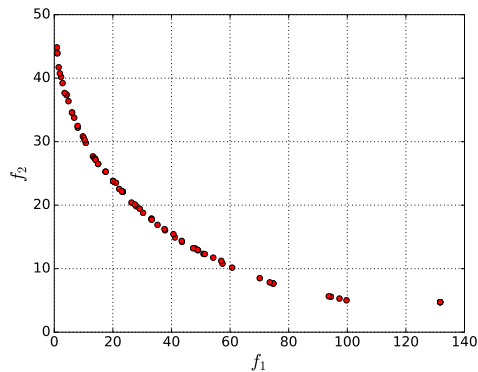
Для численного построения множества Парето выберем 100 наборов коэффициентов (λ_1, λ_2) таких, что $\lambda_1^i = ih$, $\lambda_2^i = 1 - \lambda_1^i$, $h = 10^{-2}$, $i = \overline{1, 100}$.

Пример решения многокритериальной задачи

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



Раздельное решение задач, $SP_{single} = 0.984$



Решение множества задач, $SP_{multi} = 0.749$

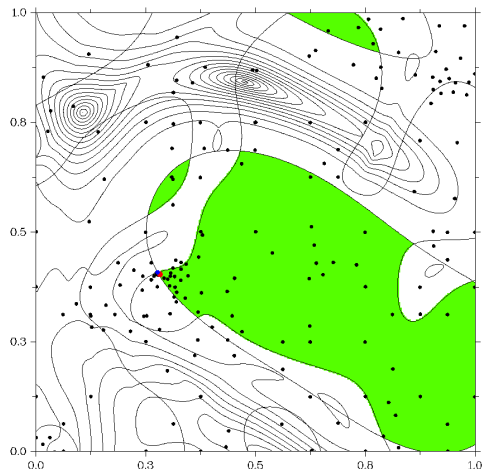
Численные оценки множества Парето, полученные после 2500 испытаний

Тестовые многомерные задачи с ограничениями

Тестовые наборы были получены с помощью системы GCGen, позволяющей строить многомерные задачи с ограничениями из заданных функций, и нескольких известных генераторов тестовых задач (GKLS, F_{GR}). Характеристики каждого из сгенерированных наборов задач:

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

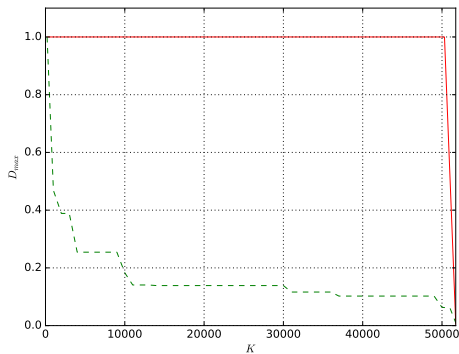
Система GCGen доступна по ссылке <https://github.com/UNN-ITMM-Software/GCGen>



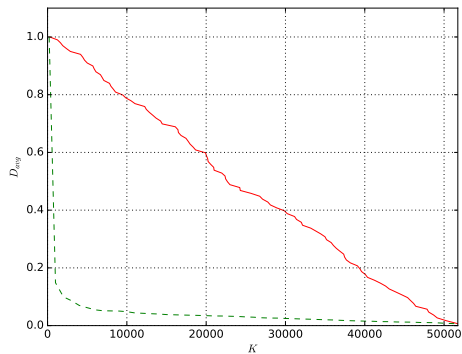
Реализация параллельного метода была выполнена на языке C++ с использованием технологии OpenMP для распарелливания процесса проведения испытаний на общей памяти.

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

Результаты экспериментов на наборах синтетических задач



D_{max}



D_{avg}

Динамика величин средней по задачам и максимальной точности в процессе решения множества двумерных задач, порождённых двумя разными генераторами GKLS и F_{GR}

Результаты экспериментов на наборах синтетических 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
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

Заключение

В ходе работы были достигнуты следующие результаты:

- ▶ реализована поддержка нелинейных ограничений в алгоритме, решающем множество задач глобальной оптимизации;
- ▶ проведены численные эксперименты, демонстрирующие преимущество рассматриваемого подхода в скорости сходимости на всём множестве задач в среднем над решением задач по отдельности;
- ▶ показана эффективность совместного решения множества задач на примере решения многокритериальной задачи с нелинейными ограничениями.

В ходе дальнейшей работы планируется:

- ▶ улучшить текущую реализацию алгоритма, сократив расходы на содержание поисковой информации для множества задач и тем самым улучшив показатели параллельного ускорения по времени;
- ▶ реализовать версию рассматриваемого алгоритма, работающего на распределенной памяти.

Контакты:

`sovrasov.vlad@gmail.com`

`https://github.com/sovrasov`