

Parallel Multi-objective Optimization on CPU Using Information Framework for Constructing Global Optimization Algorithms

Vladislav V. Sovrasov

State University of Nizhny Novgorod, Nizhny Novgorod, Russia
sovrasov.vlad@gmail.com

Abstract. В данной работе рассматривается параллельный алгоритм многокритериальной оптимизации. Рассматриваемый подход основан на применении информационно-статистического алгоритма к некоторой редуцированной однокритериальной задаче, множество глобальных оптимумов в которой совпадает с множеством слабоэффективных решений в исходной многокритериальной задаче. Последовательная версия данного метода была рассмотрена ранее. В данной работе к последовательному алгоритму многокритериальной оптимизации применяется схема распараллеливания по характеристикам, общая для всех информационно-статистических алгоритмов глобальной оптимизации. Также в работе впервые для многокритериального метода рассматривается одна из техник учёта локальных свойств оптимизируемой функции, позволяющая существенно ускорить сходимость.

Keywords: deterministic global optimization, multi-objective optimization, parallel numerical methods, derivative-free algorithms

1 Introduction

Задачи многокритериальной оптимизации привлекают в последнее время всё больший интерес, т.к. наиболее полно отражают противоречивость требований, возникающих к моделируемым объектам реального мира. В таких задачах ответом является одно или множество компромиссных решений. Как правило, эти компромиссные решения выбираются из множества, в котором все элементы не могут быть улучшены ни по одному критерию без ухудшения других (Pareto-optimal or efficient solutions) или из множества, в котором ни один элемент не может быть улучшен по всем критериям сразу (Slater-optimal or weak-efficient solutions). Последнее требование мягче и все Pareto-optimal solutions являются и Slater-optimal.

Найти всё множество efficient или weak-efficient точек является наиболее вычислительно сложной задачей, однако это даёт широкий выбор возможных компромиссных решений. Разработано множество подходов к поиску этих множеств: применение генетических алгоритмов непосредственно к многокритериальной задаче [1], использование параметрических свёрток, сводящих задачу к серии

скалярных задач глобальной оптимизации [], различные подходы для решения задач с выпуклыми или линейными критериями []. Основной проблемой при использовании большинства указанных методов является отсутствие теоретической гарантии равномерной сходимости ко всему искомому множеству на широком классе задач. Метод, обладающий этим свойством в случае, когда все критерии удовлетворяют условию Липшица, был разработан на основе информационного метода глобальной оптимизации []. В статье будет рассмотрена его параллельная и ускоренная с помощью техники local refinement [1] (chapter 3) модификация.

2 Problem Statement and Dimension Reduction

Задача многокритериальной оптимизации ставится следующим образом:

$$\min\{f(y) : y \in D\}, D = \{y \in \mathbb{R}^n : a_i \leq y_i \leq b_i, 1 \leq i \leq n\} \quad (1)$$

Будем считать, что компоненты вектор-функции (частные критерии) $f_i(y)$, $1 \leq i \leq m$, удовлетворяют в D условию Липшица с константами L_i :

$$|f_i(y_1) - f_i(y_2)| \leq L_i \|y_1 - y_2\|, y_1, y_2 \in D, 0 < L_i < \infty, 1 \leq i \leq m$$

As the solution to the problem (1) usually accepted the set $S(D) \subset D$ of strictly non-dominated points from the range of search, i. e.,

$$S(D) = \{y \in D : \nexists z \in D, f_i(z) < f_i(y), 1 \leq i \leq m\} \quad (2)$$

which is usually referred as the set of semi-effective (or weakly effective) solutions. The conditions in the right-hand side of the definition (2) are known as the principle of weak Pareto-optimality (or Slater's optimality principle).

The use of the evolvents $y(x)$ i.e. the curves filling the space are a classic dimension-reduction scheme for global optimization algorithms [2].

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

Such a mapping allows the reduction of a problem (1) stated in a multidimensional space to solving a one-dimensional problem at the expense of worsening its properties. In particular, the one-dimensional functions $f_i(y(x))$ are not Lipschitzian but Hölderian functions:

$$|f_i(y(x_1)) - f_i(y(x_2))| \leq H_i |x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0, 1] \quad (3)$$

where the Hölder constants H_i are related to the Lipschitz constant L_i by the relation

$$H_i = 4L_i d \sqrt{N}, d = \max\{b_i - a_i : 1 \leq i \leq n\}$$

Therefore, not limiting the generality, one can consider the solving of the one-dimensional problem $\min\{f(y(x)) : x \in [0, 1]\}$, satisfying Hölder condition. The issues of numerically building the mapping like a Peano curve and the corresponding theory have been considered in detail in [2]. Here we would note that an evolvent built numerically is an approximation to the theoretical Peano curve with a precision of the order 2^{-m} where m is the building parameter of the evolvent.

3 Description of the Parallel Algorithm With Local Refinement

Рассмотрим схему скаляризации редуцированной задачи (1), представленную в [3]. Пусть

$$\varphi(x) = \max\{h(x, y) : y \in [0; 1]\}, x \in [0; 1]. \quad (4)$$

Рассмотрим скалярную задачу

$$\varphi^* = \min\{\varphi(x) : x \in [0; 1]\}. \quad (5)$$

Как показано в [3], множество слабо-эффективных решений редуцированной задачи (1) совпадает множеством глобально оптимальных решений задачи (5), т.е.

$$S([0; 1]) = \{x \in [0; 1] : \varphi(x) = \varphi^*\} \quad (6)$$

Также в [3] показано, что функция $\varphi(x)$ удовлетворяет условию Гёльдера при выполнении требований (3). Таким образом, к функции $\varphi(x)$ можно применить информационно-статистический алгоритм глобального поиска, чтобы решить задачу (5). Однако, $\varphi(x)$ задаётся через оператор $\max\{\dots\}$, поэтому непосредственно вычислить её затруднительно. В [3] приведена модификация классического информационно-статистического алгоритма [4], в которой значения $\varphi(x)$ вычисляются приближённо. Далее приведём модифицированную версию указанного алгоритма. Модификация заключается в использовании техники local refinement, описанной в [1](chapter 3), а также в распараллеливании по характеристикам [1](chapter 5).

Первые две итерации производятся в концевых точках $x^0 = 0$ и $x^1 = 1$ интервала $[0; 1]$. Выбор точек x^{k+j} , $1 \leq j \leq p$ осуществляется по правилам:

Step 1. Renumber the points in the set $X_k = \{x^1, \dots, x^k\} \cup \{0\} \cup \{1\}$, which includes the boundary points of the interval $[0, 1]$ as well as the points of preceding trials, by the lower indices in order of increasing coordinate values i.e.

$$0 = x_0 < x_1 < \dots < x_{k+1} = 1$$

Step 2. Compute the lower bound μ_ν for the Lipschitz constant for each objective function $f_\nu(x)$, $1 \leq \nu \leq m$:

$$\mu_\nu = \max_{1 \leq i \leq k} \frac{|f_\nu(x_i) - f_\nu(x_{i-1})|}{\Delta_i}, 1 \leq \nu \leq m \quad (7)$$

$$\Delta_i = (x_i - x_{i-1})^{\frac{1}{N}} \quad (8)$$

Step 3. To each point x_i , $0 \leq i \leq k$, juxtapose the value

$$z_i = \max\{h(x_i, x_j) : 0 \leq j \leq k\}, \quad (9)$$

where

$$h(x_i, x_j) = \min\left\{\frac{f_\nu(x_i) - f_\nu(x_j)}{\mu_\nu} : 1 \leq \nu \leq m\right\}, 0 \leq i, j \leq k \quad (10)$$

Step 4. For each interval (x_i, x_{i-1}) , $1 \leq i \leq k$ compute two characteristics:

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r^2 \Delta_i} - \frac{z_i + z_{i-1}}{2r} \quad (11)$$

$$R^*(i) = \frac{R(i)}{\sqrt{(z_i - z^*)(z_{i-1} - z^*) + 1.5^{-\alpha}}}, \quad (12)$$

where Δ_i is from (8), $z^* = \min\{z_i : 1 \leq i \leq k\}$. $r > 1$ and $\alpha \in [10; 30]$ are input parameters of the algorithm.

Step 5. Если $q \neq 0$ и $s \bmod q \neq 0$, то характеристики $R(i)$, $1 \leq i \leq k+1$, упорядочить в порядке убывания

$$R(t_1) \geq R(t_2) \geq \dots \geq R(t_k) \geq R(t_{k+1})$$

и выбрать p наибольших характеристик с номерами интервалов t_j , $1 \leq j \leq p$. Иначе то же самое сделать с характеристиками $R^*(i)$, $1 \leq i \leq k+1$. Здесь s — номер текущей итерации. q — параметр метода, отвечающий за степень интенсивности локального уточнения. Чем меньше q , тем чаще используются характеристики R^* , заставляющие метод выбирать следующие точки вблизи текущего найденного минимума.

Step 6. Провести новые испытания в точках x^{k+j} , $1 \leq j \leq p$:

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - \text{sign}(z_{t_j} - z_{t_j-1}) \frac{|z_{t_j} - z_{t_j-1}|^n}{2r} \quad (13)$$

Все p испытаний на этом шаге могут быть произведены параллельно на p вычислительных устройствах.

The algorithm is terminated if the condition $\Delta_{t_j} \leq \varepsilon$ is fulfilled at least for one of the numbers t_j , $1 \leq j \leq p$; here $\varepsilon > 0$ is the predefined accuracy. After the search is terminated, the set $S(\{x^0, \dots, x^k\})$ of all non-dominated points of the truncated sequence $\{x^0, \dots, x^k\}$ is accepted as an estimation for S from (6).

The theoretical substantiation of this method when $p = 1$ and $q = 0$ is presented in [1](chapter 3). Sufficient condition of convergence is: exists an iteration such that $r\mu_\nu \geq 4H_\nu$, $1 \leq \nu \leq m$.

4 Test Problems

Для оценки степени ускорения сходимости модифицированного алгоритма из секции 3 использовались следующие задачи:

1. Markin-Strongin problem from [3]:

$$\begin{cases} f_1(y) = \min\{\sqrt{y_1^2 + y_2^2}, \sqrt{(y_1 - 1.5)^2 + (y_2 + 1.5)^2}\} \\ f_2(y) = \sqrt{(y_1 + 0.5)^2 + (y_2 - 0.5)^2} \end{cases} \quad y_1 \in [-1; 2], y_2 \in [-2; 1] \quad (14)$$

2. Fonseca and Fleming problem [5]:

$$\begin{cases} f_1(y) = 1 - \exp\left(-\sum_{i=1}^n \left(y_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ f_2(y) = 1 - \exp\left(-\sum_{i=1}^n \left(y_i + \frac{1}{\sqrt{n}}\right)^2\right) \end{cases} \quad y \in [-4; 4]^n \quad (15)$$

3. Schaffer function N. 2 [5]:

$$\begin{cases} f_1(y) = \begin{cases} -y, & \text{if } y \leq 1 \\ y - 2, & \text{if } 1 < y \leq 3 \\ 4 - y, & \text{if } 3 < y \leq 4 \\ y - 4, & \text{if } y > 4 \end{cases} \\ f_2(y) = (y - 5)^2 \end{cases} \quad y \in [-5; 10] \quad (16)$$

4. Poloni's two objective function [5]:

$$\begin{cases} f_1(y) = \left[1 + (A_1 - B_1(y))^2 + (A_2 - B_2(y))^2\right] \\ f_2(y) = (y_1 + 3)^2 + (y_2 + 1)^2 \end{cases} \quad y \in [-\pi; \pi]^2 \quad (17)$$

where

$$\begin{cases} A_1 = 0.5 \sin(1) - 2 \cos(1) + \sin(2) - 1.5 \cos(2) \\ A_2 = 1.5 \sin(1) - \cos(1) + 2 \sin(2) - 0.5 \cos(2) \\ B_1(y) = 0.5 \sin(y_1) - 2 \cos(y_1) + \sin(y_2) - 1.5 \cos(y_2) \\ B_2(y) = 1.5 \sin(y_1) - \cos(y_1) + 2 \sin(y_2) - 0.5 \cos(y_2) \end{cases}$$

5 Experimental Results

The computational experiments have been carried out on the Lobachevsky super-computer at Lobachevsky State University of Nizhni Novgorod. A computational node included 2 Intel Sandy Bridge E5-2660 2.2 GHz processors, 64 Gb RAM. The CPUs had 8 cores (i. e. total 16 cores were available per a node).

В данном разделе будем понимать под ускорением работы метода ускорение по числу выполненных итераций, а не по времени выполнения. Если вычисление критериев задачи (1) занимает достаточно много времени, то накладные затраты на реализацию решающих правил метода оптимизации невелики по сравнению с временем вычисления критериев $f_i(y)$, $1 \leq i \leq m$.

Local refinement advantages. В [6] приведены численные эксперименты, показывающие ускорение сходимости при применении техники local refinement в алгоритме, решающем скалярные задачи глобальной оптимизации. Подобных результатов стоит ожидать и от многокритериального алгоритма. Multi-objective algorithm with local refinement (MOALR) был применён к задаче Fonseca and Fleming (15) при $n = 2$. Параметры метода были следующие: $\varepsilon = 0.01$, $r =$

4, $q = 4$, $\alpha = 15$, $p = 1$. До остановки метода было произведено 1176 итераций, количество найденных слабо оптимальных точек — 90. При $q = 0$ (без локального уточнения) метод производит 1484 итерации, количество найденных слабо оптимальных точек — 93. На рис. 1 представлено множество S в задаче (15) и его численные оценки, полученные при $q = 0$ (fig. 1.a) и $q = 4$ (fig. 1.b). Из рис. 1 также видно, что приближённое решение покрывает всё множество weak-optimal solutions.

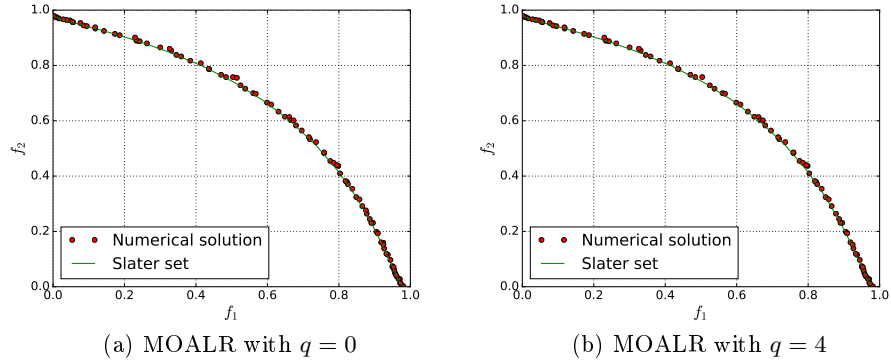


Fig. 1: Numerical estimation of S obtained by MOALR

В дальнейшем для всех экспериментов будем использовать MOALR при $q = 4$.

Parallel method results. Чтобы продемонстрировать ускорение по итерациям, которое даёт MOALR при $p > 1$, были решены все задачи из секции 4 при значениях $p = 1, 2, 4, 8, 16$. Параметры метода при этом были следующие: $r = 4.5$, $\varepsilon = 0.01$, $\alpha = 15$. Для задачи (16) $\varepsilon = 0.001$. В таблице 1 приведено количество итераций (в скобках мощность оценки множества S), а в таблице 2 — ускорение по итерациям.

Как видно из таблицы 1, мощность оценки множества S при изменении p меняется незначительно, т. е. качество полученной оценки сохраняется. В то же время, количество итераций сокращается пропорционально увеличению значения p (таблица 2). На рис. 2 приведены примеры численных решений рассматриваемых задач.

Parallel method time speedup. Целевые функции всех перечисленных в разделе задачи обладают невысокой вычислительной сложностью. Чтобы продемонстрировать возможность получения ускорения по времени для задач с трудновычислимыми критериями, в критерии были внесены дополнительные интенсивные вычисления с плавающей точкой, не влияющие на результирующие значения. Полученные

Table 1: Results of numerical experiments: number of iterations

Problem	p				
	1	2	4	8	16
Markin-Strongin	1041(198)	516(198)	256(185)	131(197)	68(191)
Fonseca and Fleming 2d	1181(93)	636(99)	386(111)	176(95)	106(97)
Fonseca and Fleming 3d	5346(160)	3551(183)	1186(143)	606(153)	351(142)
Schaffer function N. 2	271(158)	136(163)	64(146)	32(142)	16(141)
Poloni's function	3351(102)	1706(90)	856(88)	426(96)	201(99)

Table 2: Results of numerical experiments: speedup in iterations

Problem	p			
	2	4	8	16
Markin-Strongin	2.02	4.07	7.95	15.31
Fonseca and Fleming 2d	1.86	3.06	6.71	11.14
Fonseca and Fleming 3d	1.51	4.51	8.82	15.23
Schaffer problem N. 2	1.99	4.23	8.47	16.94
Poloni's problem	1.96	3.91	7.87	16.67

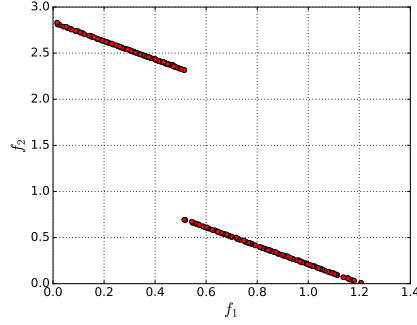
при этом ускорения приведены в таблице 3. Ускорение по времени во всех случаях меньше, чем ускорение по итерациям, т.к. часть времени выполнения занимает работа метода оптимизации, однако распараллеливание в большинстве случаев довольно эффективно и в случае 16 рабочих потоков. При ещё большей вычислительной сложности критериев стоит ожидать увеличения ускорения по времени для большого числа потоков.

Table 3: Results of numerical experiments: speedup in time

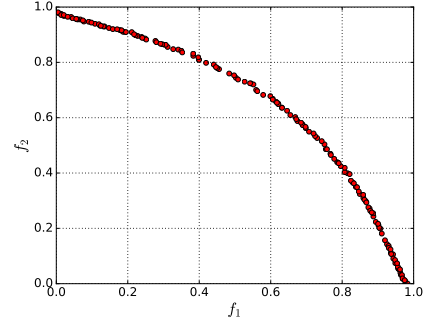
Problem	p				
	1(time, s)	2	4	8	16
Markin-Strongin	104.47	1.97	3.65	6.79	9.90
Fonseca and Fleming 2d	118.95	1.85	2.81	5.79	6.40
Fonseca and Fleming 3d	554.45	1.51	4.14	8.05	10.69
Schaffer problem N. 2	27.21	1.91	3.79	6.39	6.37
Poloni's problem	336.74	1.82	3.64	6.98	10.60

6 Conclusion

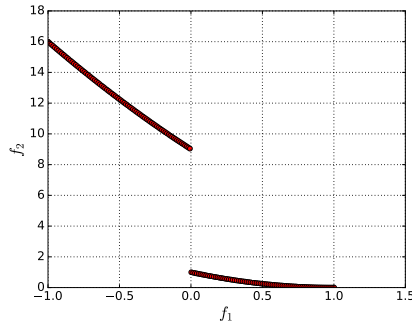
В работе был рассмотрен метод решения многокритериальных задач оптимизации, позволяющий находить равномерную оценку множества слабо-оптимальных точек. К нему применены техники распараллеливания и ускорения сходимости, общие для всех алгоритмов подобной структуры. Проведённые численные



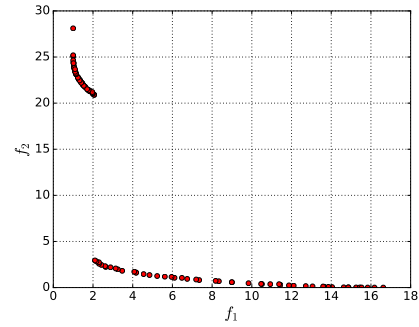
(a) Markin-Strongin problem



(b) Fonseca and Fleming 3d problem



(c) Schaffer problem N. 2



(d) Poloni's problem

Fig. 2: Numerical estimation of S obtained by MOALR

эксперименты показали наличие ускорения сходимости при использовании local refinement. Также для данного метода оказалось эффективным распараллеливание по характеристикам. Как и для информационных методов скалярной оптимизации, ускорение по итерациям при распараллеливании по характеристикам оказывается линейным в большинстве случаев (см. результаты распараллеливания метода скалярной оптимизации напр., в [7]). Исходя из результатов экспериментов, параллельный MOALR целесообразно применять в задачах низкой размерности (1—3) с трудновычислимыми критериями. Кроме того, на численных примерах продемонстрировано свойство равномерной сходимости рассматриваемого алгоритма ко всему множеству weak-optimal solutions.

References

1. Strongin R.G., Sergeyev Ya.D. *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Kluwer Academic Publishers, Dordrecht, 2000.

2. Sergeyev, Y.D., Strongin, R.G., Lera, D. *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer Briefs in Optimization, Springer, New York, 2013.
3. Markin, D.L., Strongin, R.G. Uniform estimates for the set of weakly effective points in multi-extremum multicriterion optimization problems. *Computational Mathematics and Mathematical Physics*, 33(2):171–179, 1993.
4. Markin, D.L., Strongin, R.G. A method for solving multi-extremal problems with non-convex constraints, that uses a priori information about estimates of the optimum. *Computational Mathematics and Mathematical Physics*, 27(1):33–39, 1987.
5. S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *Trans. Evol. Comp*, 10(5):477–506, October 2006.
6. Konstantin Barkalov and Ilya Lebedev. Local tuning in multilevel scheme of parallel global optimization. *AIP Conference Proceedings*, 1776(1):060006, 2016.
7. Konstantin Barkalov, Ilya Lebedev, Vladislav Sovrasov, and Aleksandr Sysoyev. Implementation of a parallel algorithm for searching the global extremum of a function on intel xeon phi. In *Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies*, pages 68–80, 2016.