

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

В.В. Соврасов

Нижегородский государственный университет им. Н.И. Лобачевского

В данной работе рассматривается построение параллельной версии алгоритма глобальной оптимизации, решающего одновременно множество задач с нелинейными ограничениями и получающего при этом равномерные оценки решений на этом множестве. Последнее свойство позволяет наиболее оптимально распределять вычислительные ресурсы, т. к. в процессе работы алгоритма погрешности численного решения во всех задачах убывают примерно с одинаковой скоростью. Подобные серии задач возникают, если задача глобальной оптимизации имеет дискретный параметр или при решении задачи многокритериальной оптимизации методом свёртки критериев. Рассматриваемый алгоритм использует отображения типа кривой Пеано для редукции многомерных задач оптимизации к одномерным. Эффективность реализованного алгоритма протестирована на наборах искусственно сгенерированных задач глобальной оптимизации.

Ключевые слова: Глобальная оптимизация, параллельные вычисления, алгоритмы прямой оптимизации, равномерная сходимость

1. Введение

Нелинейная глобальная оптимизация невыпуклых функций традиционно считается одной из самых трудных задач математического программирования. Отыскание глобального минимума функции от нескольких переменных зачастую оказывается сложнее, чем локальная оптимизация в тысячемерном пространстве. Для последней может оказаться достаточно применения простейшего метода градиентного спуска, в то время как чтобы *гарантированно* отыскать глобальный оптимум методами оптимизации приходится накапливать информацию о поведении целевой функции во всей области поиска [6, 10, 11, 15]. Решение серии таких задач при ограниченных вычислительных ресурсах является ещё более сложной задачей: помимо поиска глобального экстремума необходимо распределять вычислительные ресурсы так, чтобы сразу во всех решаемых задачах положение глобального экстремума было оценено примерно с одинаковым качеством. Обычно серию из q задач решают либо последовательно, либо параллельно порциями по p , $p \ll q$ задач, где p — количество параллельных вычислительных устройств. Такой подход ведёт к тому, что в каждый момент времени до окончания вычислений остаются задачи, в которых оценка глобального оптимума не получена вообще, в то время, как в задачах из начала списка оптимум может быть оценён даже с избыточной точностью.

В данной работе рассматривается обобщение ранее разработанного в ННГУ им. Н. И. Лобачевского параллельного метода глобальной оптимизации для одновременного решения множества задач [3] на случай задач с нелинейными ограничениями. Для учёта ограничений используется индексная схема [15], позволяющая работать с частично вычислимыми целевыми функциями и обладающая экономичностью, сравнимой с другими подходами [1]. Эффективность реализованного алгоритма показана на примере решения множеств задач, сгенерированных специализированным механизмом, порождающим наборы задач заданной

*Исследование выполнено при поддержке РФФИ, проект № 16-11-10150.

размерности с заданным количеством нелинейных ограничений [8]. Кроме искусственно сгенерированных задач, рассматриваемый метод протестирован также на множестве задач, возникающем при решении задачи многокритериальной оптимизации с нелинейными ограничениями методом свёртки критериев [5].

2. Постановка задачи глобальной оптимизации

В рамках данной работы будем рассматривать следующую постановку 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.

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

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

Существуют различные методы, решающие рассмотренную многомерную задачу напрямую [10, 13], а также эффективные методы решения одномерных задач [15, 17]. 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 [14].

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

Отображение вида (2) позволяет свести задачу в многомерном пространстве к решению одномерной ценой ухудшения её свойств. В частности, одномерная функция $\varphi(y(x))$ является не Липшицевой, а Гёльдеровой:

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

где константа Гельдера H связана с константой Липшица L соотношением

$$H = 4Ld\sqrt{N}, d = \max\{b_i - a_i : 1 \leq i \leq N\}.$$

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

$$\varphi(y^*) = \min\{\varphi(y) : g_j(y) \leq 0, 1 \leq j \leq m\} \quad (3)$$

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 в некотором гиперинтервале, включающем D .

Далее будем интересоваться решением серии из q задач вида (3):

$$\min\{\varphi_1(y), y \in D_1\}, \min\{\varphi_2(y), y \in D_2\}, \dots, \min\{\varphi_q(y), y \in D_q\}. \quad (4)$$

3. Описание метода глобальной оптимизации

Принимая во внимание схему редукции размерности (2), 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 (ИАГП) для решения одномерной задачи (3) предполагает построение последовательности точек x_k , в которых вычисляются значения минимизируемой функции или ограничений $z_k = g_s(x_k)$. Для учёта последних используется индексная схема [15]. 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 функциональными ограничениями к безусловной задаче минимизации разрывной функции:

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

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.

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

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

Шаг 2. Для каждого целого числа $\nu, 1 \leq \nu \leq m + 1$ 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:

$$\mu_\nu = \max \left\{ \frac{|g_\nu(x_i) - g_\nu(x_j)|}{(x_i - x_j)^{\frac{1}{N}}} : i, j \in I_\nu, i > j \right\}. \quad (5)$$

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 $I_\nu, \nu = \overline{1, M}$ вычислить оценки

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

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

$$R(i) = \begin{cases} \Delta_i + \frac{(z_i - z_{i-1})^2}{(r_\nu \mu_\nu)^2 \Delta_i} - 2 \frac{z_i + z_{i-1} - 2z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_i) = \nu(x_{i-1}) \\ 2\Delta_i - 4 \frac{z_{i-1} - z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_{i-1}) > \nu(x_i) \\ 2\Delta_i - 4 \frac{z_i - z_\nu^*}{r_\nu \mu_\nu} & \nu = \nu(x_i) > \nu(x_{i-1}) \end{cases} \quad (6)$$

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 Гельдера.

Шаг 6. Выбрать наибольшую характеристику:

$$t = \arg \max_{1 \leq i \leq k+1} R(i) \quad (7)$$

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

$$\varphi_k^* = \min_{1 \leq i \leq k} \varphi(x_i), x_k^* = \arg \min_{1 \leq i \leq k} \varphi(x_i) \quad (8)$$

Далее следуя подходу, описанному в [3], для решения серии задач (4) будем использовать q синхронно работающих копий ИАГП 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 ИАГП. 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 (6), что позволяет динамически распределять ресурсы метода между задачами.

В [3] приведена теория сходимости такого подхода на случай решения задач без ограничений. При наличии ограничений характеристики интервалов, на концах которых нарушено разное количество ограничений, вычисляются в соответствии с нижними строчками из (6). Нетрудно заметить, что и в этом случае величины характеристик нормированы, а в случае точных оценок z_ν^* и μ_ν их масштаб не зависит от целевой функции и ограничений конкретной задачи. Таким образом, рассуждения из [3] можно провести и в случае использования индексной схемы.

Параллельная модификация метода не отличается от рассматриваемой в [3] 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).

4. Результаты численных экспериментов

Использование сгенерированных некоторыми случайными механизмами наборов тестовых задач с известными решениями является одним из общепринятых подходов к сравнению алгоритмов оптимизации [4]. В данной работе будем использовать два генератора тестовых задач, порождающих задачи различной природы [7,9]. Эти генераторы порождают задачи без нелинейных ограничений, поэтому в дополнение к ним использована система GCGen¹ [8], позволяющая генерировать задачи с ограничениями на основе произвольных нелинейных функций.

Вместе с системой GCGen распространяются примеры её использования и построения наборов задач, каждая из которых состоит из целевой функции и двух ограничений, порождённых генератором F_{GR} [9] или GKLS [7].

Генератор GKLS [7] позволяет получать функции заданной размерности и с заданным количеством экстремумов. В сочетании с GCGen были порождены два множества по 100 задач размерности 2 и 3. Каждая из задач имеет два ограничения. Также с целью демонстрации того, что свойства метода сохраняются при существенно разных свойствах задач был сгенерирован смешанный класс, состоящий из 50 задач с двухмерными функциями GKLS и 50 задач с функциями F_{GR} . На рис. 1 представлены примеры линий уровня рассматриваемых задач. Допустимая область закрашена.

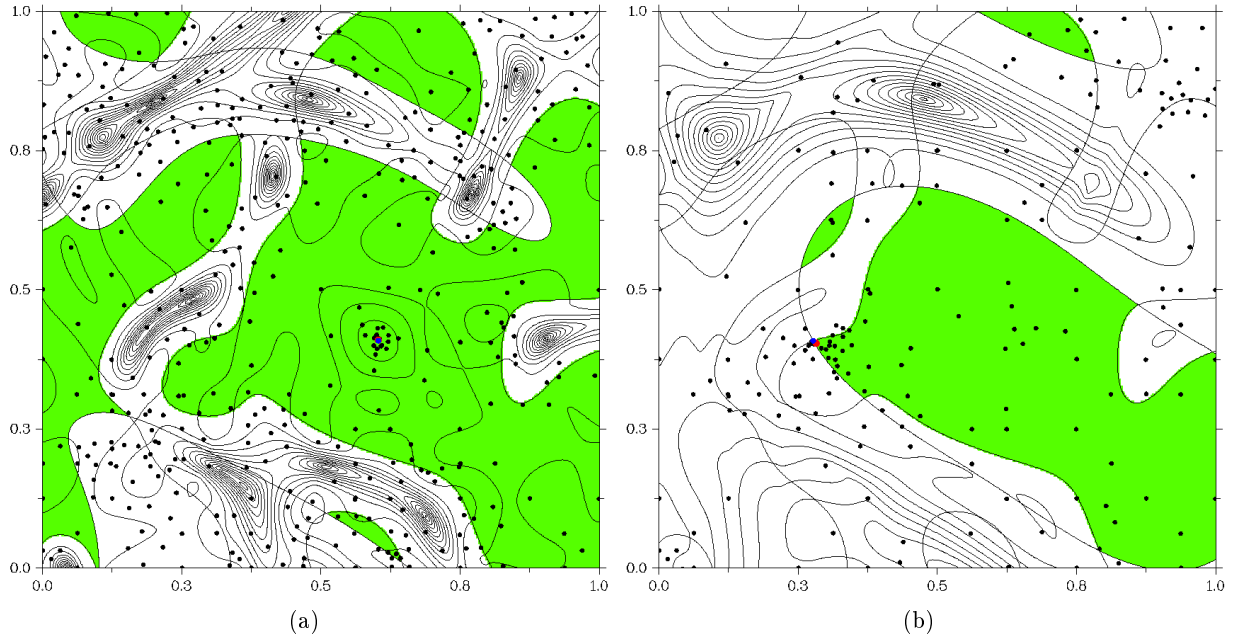


Рис. 1: Линии уровня и точки испытаний ИАГП в двух синтетических задачах.

Будем считать, что тестовая задача решена, если метод оптимизации провёл очередное испытание y^k в δ -окрестности глобального минимума y^* , т.е. $\|y^k - y^*\| \leq \delta = 0.01 \|b - a\|$, где a и b — левая и правая границы гиперкуба из (1). Если указанное соотношение не выполнено до истечения лимита на количество испытаний, то задача считается нерешённой.

При оценке качества метода и его реализации кроме ускорения от распараллеливания и времени выполнения также будем принимать во внимание среднее максимальное расстояния (в смысле l_{inf} -нормы) текущей оценки оптимума до его реального положения, вычисленное на множестве задач (4): D_{avg} и D_{max} . Динамика этих величин в процессе оптимизации показывает, насколько равномерно метод распределяет ресурсы между задачами.

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

Таблица 1: 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

Реализация параллельного метода была выполнена на языке C++ с использованием технологии OpenMP для распареллеливания процесса проведения испытаний на общей памяти. Все вычислительные эксперименты проведены на машине со следующей конфигурацией: Intel Core i7-7800X, 64GB RAM, Unubtu 16.04 OS, GCC 5.5 compiler.

4.1. Результаты решения сгенерированных задач

Результаты решения тестовых задач последовательной и параллельной версией модифицированного ИАГП для решения множества задач представлены в таблице 1. Для всех двухмерных классов задач параметр $r = 4.7$. В случае трёхмерных задач $r = 4.7$, $\varepsilon_\nu = 0.1$. Во всех экспериментах в целевые функции и ограничения была внесена дополнительная вычислительная нагрузка так, чтобы время одного обращения к функции задачи было равно примерно 1мс.

Из таблицы видно, что ускорение по итерациям S_i растёт линейно с увеличением числа потоков p , в то время, как ускорение по времени S_p увеличивается не так быстро, что говорит о неидеальной реализации алгоритма. Увеличить реальное ускорение, верхней границей для которого является S_i , возможно путём оптимизации взаимодействий между копиями ИАГП и это планируется сделать в ходе будущей работы.

Для того, чтобы показать равномерную сходимость все тестовые задачи были также решены ИАГП в режиме решения отдельных задач. На рис. 2 указаны графики величин средних и максимальных расстояний от реальных оптимумов до их текущих оценок при решении серии из задач, порождённых двумя разными генераторами, по отдельности (сплошная кривая) и совместно (пунктирная кривая). Не смотря на значительную разницу в структуре задач, модифицированный ИАГП гораздо быстрее уменьшает как максимальное, так и среднее отклонение оценок от оптимумов. 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.

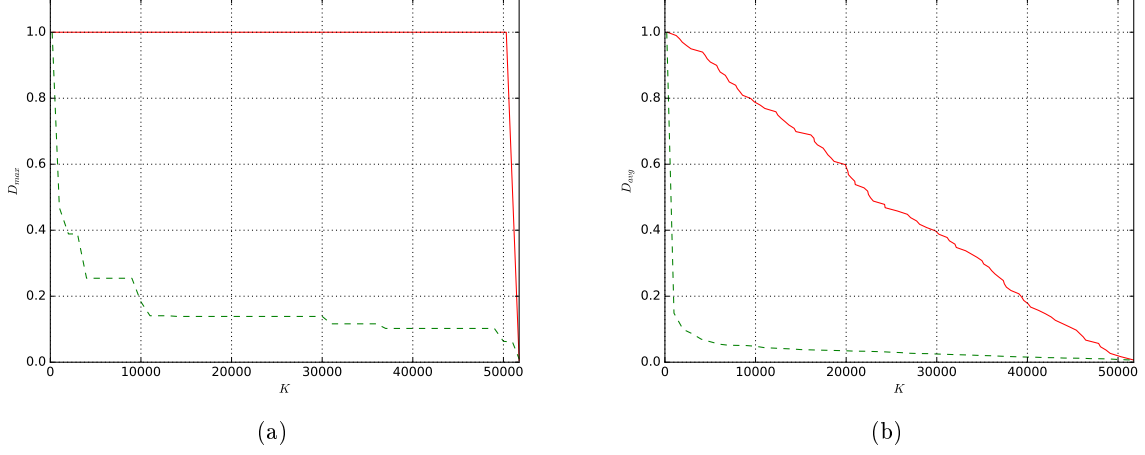


Рис. 2: Динамика величин D_{avg} и D_{max} в процессе решения множества задач

4.2. Пример решения многокритериальной задачи

Для демонстрации эффективности подхода к балансировке нагрузки рассмотрим пример, в котором множество задач вида (4) порождено в результате скаляризации многокритериальной задачи оптимизации с ограничениями.

Рассмотрим тестовую задачу, предложенную в [16]:

$$\begin{aligned}
 & \text{Minimize} \begin{cases} f_1(y) = 4y_1^2 + 4y_2^2 \\ f_2(y) = (y_1 - 5)^2 + (y_2 - 5)^2 \end{cases} & y_1 \in [-1; 2], y_2 \in [-2; 1] \\
 & s.t. \\
 & \begin{cases} g_1(y) = (y_1 - 5)^2 + y_2^2 - 25 \leq 0 \\ g_2(y) = -(y_1 - 8)^2 - (y_2 + 3)^2 + 7.7 \leq 0 \end{cases}
 \end{aligned} \tag{9}$$

Будем использовать свёртку Гермейера для скаляризации задачи (9). После свёртки скалярная целевая функция имеет вид:

$$\varphi(y, \lambda_1, \lambda_2) = \max\{\lambda_1 f_1(y), \lambda_2 f_2(y)\}, \tag{10}$$

где $\lambda_1, \lambda_2 \in [0, 1]$, $\lambda_1 + \lambda_2 = 1$. Перебирая все возможные коэффициенты свёртки, можно найти всё множество парето-оптимальных решений в задаче (9). Для численного построения множества Парето выберем 100 наборов коэффициентов (λ_1, λ_2) таких, что $\lambda_1^i = ih$, $\lambda_2^i = 1 - \lambda_1^i$, $h = 10^{-2}$, $i = \overline{1, 100}$.

В качестве ограничения на вычислительные ресурсы был выбран лимит в 2500 испытаний. Множество вспомогательных скалярных задач решалось двумя способами:

- каждая задача решается отдельно с помощью ИАГП с установленным лимитом в 25 испытаний. Таким образом, вычислительные ресурсы равномерно распределены между задачами;

- все задачи решаются одновременно с помощью обобщённого ИАГП с установленным лимитом в 2500 испытаний.

В обоих случаях параметр $r = 4$.

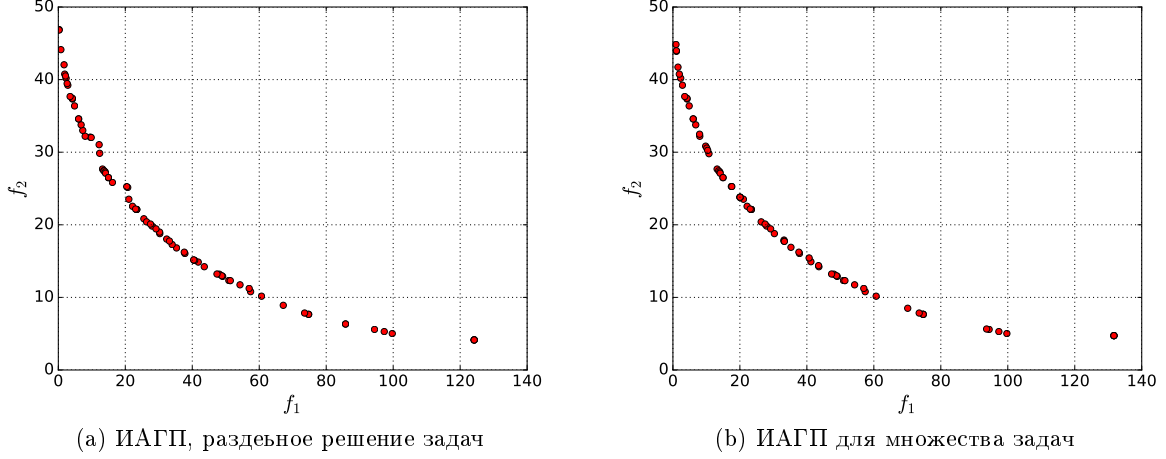


Рис. 3: Численные оценки множества Парето в задаче (9), полученные после 2500 испытаний

На рис. 3 представлены графики решений, полученных каждым из методов. Все графики качественно совпадают с указанным в [16] (авторы не предоставили другой информации и решениях для сравнения). Видно, что на рис. 3а кривая Парето имеет вогнутости, что не соответствует решению, указанному в [16] и означает нехватку ресурсов для решения некоторых из вспомогательных задач. Для оценки качества решения также был вычислен показатель $Spacing(SP)$ [12], характеризующий плотность точек аппроксимации множества Парето.

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

В случае отдельного решения задач $SP_{single} = 0.984$, а при решении задач методом с балансировкой нагрузки $SP_{multi} = 0.749$, что говорит о более качественном приближении решения.

5. Заключение

В ходе работы была реализована поддержка нелинейных ограничений в алгоритме, решающем множество задач глобальной оптимизации. Проведены численные эксперименты, демонстрирующие преимущество рассматриваемого подхода 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. Также планируется реализовать версию рассматриваемого алгоритма, работающего на распределённой памяти по схеме, описанной в [2].

Список литературы

1. Barkalov, K., Lebedev, I.: Comparing two approaches for solving constrained global optimization problems. In: R. Battiti, D.E. Kvasov, Y.D. Sergeyev (eds.) *Learning and Intelligent Optimization*, pp. 301–306. Springer International Publishing, Cham (2017)
2. Barkalov, K., Lebedev, I.: Parallel algorithm for solving constrained global optimization problems. pp. 396–404 (2017). doi: 10.1007/978-3-319-62932-2_38
3. Barkalov, K., Strongin, R.: Solving a set of global optimization problems by the parallel technique with uniform convergence. *J. of Global Optimization* **71**(1), 21–36 (2018). doi: 10.1007/s10898-017-0555-4. URL <https://doi.org/10.1007/s10898-017-0555-4>
4. Beiranvand, V., Hare, W., Lucet, Y.: Best practices for comparing optimization algorithms. *Optimization and Engineering* **18**(4), 815–848 (2017). doi: 10.1007/s11081-017-9366-1
5. Ehrgott, M.: *Multicriteria Optimization*. Springer-Verlag, Berlin, Heidelberg (2005)
6. Evtushenko, Y., Posypkin, M.: A deterministic approach to global box-constrained optimization. *Optim. Lett.* **7**, 819–829 (2013). doi: 10.1007/s11590-012-0452-1
7. Gaviano, M., Kvasov, D.E., Lera, D., Sergeev, Ya.D.: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Transactions on Mathematical Software* **29**(4), 469–480 (2003). doi: 10.1145/962437.962444
8. Gergel, V., Barkalov, K., Lebedev, I., Rachinskaya, M., Sysoyev, A.: A flexible generator of constrained global optimization test problems. p. 020009 (2019). doi: 10.1063/1.5089976
9. Grishagin, V.: Operating characteristics of some global search algorithms. *Problems of Statistical Optimization* **7**, 198–206 (1978 (In Russian))
10. Jones, D.R.: The direct global optimization algorithm. In: *The Encyclopedia of Optimization*, pp. 725–735. Springer, Heidelberg (2009). doi: 10.1007/978-0-387-74759-0_128
11. Paulavivcius, R., Zilinskas, J., Grothey, A.: Parallel branch and bound for global optimization with combination of lipschitz bounds. *Optim. Method. Softw.* **26**(3), 487–498 (1997). doi: 10.1080/10556788.2010.551537
12. Riquelme, N., Von Lucken, C., Baran, B.: Performance metrics in multi-objective optimization. In: *2015 Latin American Computing Conference (CLEI)*, pp. 1–11 (2015). doi: 10.1109/CLEI.2015.7360024
13. Sergeyev, Y., Kvasov, D.: *Deterministic Global Optimization* (2017). doi: 10.1007/978-1-4939-7199-2
14. Sergeyev, Y.D., Strongin, R.G., Lera, D.: *Introduction to global optimization exploiting space-filling curves*. Springer Briefs in Optimization, Springer, New York (2013). doi: 10.1007/978-1-4614-8042-6
15. Strongin R.G., Sergeyev Ya.D.: *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Kluwer Academic Publishers, Dordrecht (2000). doi: 10.1007/978-1-4615-4677-1
16. To, T.B., Korn, B.: *Mobes: A multiobjective evolution strategy for constrained optimization problems* (1999)
17. Норкин В. И.: О методе Пиявского для решения общей задачи глобальной оптимизации. *Ж. вычисл. матем. и матем. физ.* **32**, 992–1006 (1992)