Использование параллельной системы глобальной оптимизации Globalizer для решения задач конкурса GenOpt*

И.Г. Лебедев, В.В. Сопрасов

Нижегородский государственный университет им. Н.И. Лобачевского

В работе рассматриваются параллельные алгоритмы решения задач многоэкстремальной оптимизации. Алгоритмы разработаны в рамках развиваемого в ННГУ им. Н.И. Лобачевского информационно-статистического подхода и реализованы в параллельной системе Globalizer. Обсуждаются результаты вычислительных экспериментов, которые проводились на тестовых функциях международного конкурса алгоритмов оптимизации GENOPT (www.genopt.org).

Ключевые слова: глобальная оптимизация, многоэкстремальные функции, редукция размерности, параллельные алгоритмы.

1. Введение

Один из известных подходов к исследованию и сравнению алгоритмов многоэкстремальной оптимизации основан на применении этих методов для решения множества тестовых задач, выбираемых случайным образом из некоторого сконструированного класса. При этом каждая тестовая задача может рассматриваться как конкретная реализация случайной функции, задаваемой с помощью специального генератора.

Указанный подход был использован организаторами международного конкурса алгоритмов глобальной оптимизации GENeralization-based challenge in global OPTimization (GENOPT), http://www.genopt.org . Участникам конкурса было предложено сравнивать эффективность алгоритмов при решении задач из трех семейств функций:

- функции, порождаемые генератором GKLS [1, **Ошибка! Источник ссылки не найден.**];
- классические тестовые функции (напр., функция Розенброка);
- сложные функции, представляющие комбинацию функций из предыдущих семейств.

Для решения конкурсных задач были использованы алгоритмы глобальной оптимизации, разработанные в ННГУ им. Н.И. Лобачевского [**Ошибка! Источник ссылки не найден.** – **Ошибка! Источник ссылки не найден.**]; данные алгоритмы реализованы в решателе Globalizer. В настоящей работе дано краткое описание использованного базового алгоритма глобальной оптимизации и его модификаций, а также приведены результаты численных экспериментов с конкурсными задачами. Эксперименты проводились на вычислительном кластере ННГУ в параллельном режиме, чтобы оценить эффективность распараллеливания алгоритма, т.к. для решения задач данных классов участникам конкурса предлагалось использовать свои алгоритмы в последовательном режиме.

2. Постановка задачи

Задача многомерной многоэкстремальной оптимизации может быть определена как проблема поиска наименьшего значения действительной функции $\varphi(y)$, имеющей несколько локальных и один глобальный минимум

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-31-00244 мол_а «Параллельные методы решения вычислительно трудоемких задач глобальной оптимизации на гибридных кластерных системах».

$$\varphi(y^*) = \min\{\varphi(y) : y \in D\},$$
$$D = \{y \in R^N : a_i \le y_i \le b_i, 1 \le i \le N\}, \quad (1)$$

где $a,b \in R^N$ есть заданные векторы.

Численное решение задачи (1) сводится к построению оценки $y_k^* \in D$, отвечающей некоторому понятию близости к точке y^* на основе конечного числа k вычислений значений оптимизируемой функции. Относительно класса рассматриваемых задач предполагается, что $\varphi(y)$ удовлетворяет условию Липшица

$$|\varphi(y_1) - \varphi(y_2)| \le L\|y_1 - y_2\|,\ y_1, y_2 \in D,\ 0 < L < \infty, \quad (2)$$

что соответствует ограниченности изменения значений функции при ограниченной вариации аргумента.

Таблица 1. Классы конкурсных задач

Семейство	Класс задач	Размерность	Название
GKLS	не дифференцируемые	10	GKLS_nd_10
		30	GKLS_nd_30
	дифференцируемые	10	GKLS_cd_10
		30	GKLS_cd_30
	дважды дифференцируемые	10	GKLS_td_10
		30	GKLS_td_30
Классические	Функция Розенброка	10	Rosenbrock_10
		30	Rosenbrock_30
	Функция Растригина	10	Rastrigin_10
		30	Rastrigin_30
	Функция Захарова	10	Zakharov_10
		30	Zakharov_30
Составные (по три класса каждой размерности)		10	Composite_10
		30	Composite_30

Организаторами конкурса GENOPT были предложены 18 классов задач (см. таблицу 1). В каждом классе было задано 100 функций. Первое семейство состоит из функций полученных GKLS-генератором [1, **Ошибка! Источник ссылки не найден.**], позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами: количеством локальных минимумов, размерами их областей притяжения, точкой глобального минимума, значением функции в ней и т.п. Было выбрано три класса задач: не дифференцируемые, дифференцируемые и дважды дифференцируемые функции. Для всех функций было задано: пять локальных минимумов, радиус притяжения глобального минимума 1/3, расстояние от локального минимума до глобального r = 2/3; Задачи внутри класса отличаются расположением и величиной минимумов.

Второе семейство основано на классических тестовых функциях для методов оптимизации: Розенброка, Растригина и Захарова. Для усложнения генерируемых задач к данным функциям примялось нелинейное преобразование, трансформирующее функцию по отдельным направлениям.

Функции из третьего семейства получены с использованием случайного количества базовых непрерывных функций (функция Goldstein-Price размерности 2, функция Hartmann размерности 3 или 6, функции Rosenbrock, Rastrigin, Sphere, Zakharov, – размерность определяется случайно в диапазоне от 3 до 5 или 15). При этом сумма размерностей всех базовых функций равна размерности генерируемой задачи, каждая базовая функция принимает часть вектора координат генерируемой сложной функции, а вычисленные значения базовых функций – суммируются.

3. Используемые методы

В данном разделе описан базовый параллельный алгоритм глобального поиска(АГП) и его модификации, позволяющие решать многомерные задачи, использованные при решении различных конкурсных задач. Главной модификацией является использование редукция размерности с использованием кривых Пеано, позволяющая решать алгоритмом глобального поиска многомерные задачи. Следующей модификацией является использование локального метода. Обе модификации применяются для всех задач.

Еще одним способом редукции размерности является блочная многошаговая схема, эта модификация позволяет использовать множество процессов для решения многомерных задач. В конкурсе не использовалась, поскольку является параллельным алгоритмом.

Многие задачи из 10-мерных классов GKLS решаются с использованием первых двух модификаций, однако для 30-мерных задач этого оказалось недостаточно. Следующим вариантом модификации стала «мультистартовая» схема. Используя генератор псевдослучайных чисел Соболя, мы выбирали 900 начальных точек, в каждой из которых запускался локальный метод оптимизации не только с ограничением по точности и по числу итераций (не более 1000). Таким образом, в худшем случае на локальную фазу могло быть потрачено 900 тыс. испытаний. Все точки выполненных на локальной фазе испытаний помещались в базу глобального метода. Далее запускалась глобальная фаза, на которой оставшиеся до 1 миллиона испытания выполняет АГП.

Поскольку функция Растригина сепарабельна, ее глобальный оптимум можно искать, выполняя оптимизацию по каждой координате в отдельности. Таким образом, для решения задачи Растригина использовалась следующая схема:

- Сепарабельная фаза
  - Выбрать начальную точку.
  - Для каждой координаты выполнить:
    - Зафиксировать все координаты, кроме текущей.
    - Выполнить оптимизацию одномерным АГП.
  - Все точки испытаний локальной метода добавлять в базу точек АГП.
- Локальная фаза
  - Запустить из точки текущего рекорда локального метод Хука-Дживса с заданным условием выхода по точности.
  - Все точки испытаний локальной метода добавлять в базу точек АГП.
  - По достижении локальным методом заданной точности перейти к Глобальной фазе.
- Глобальная фаза
  - Выполнять итерации АГП, пока не будет обновлено текущее значение глобального минимума.
  - При обновлении глобального минимума перейти к Локальной фазе.

Рассмотренная схема позволила решить все задачи с функцией Растригина как 10-ти, так и 30-мерные.

Этот же подход был использован и для решения задач, построенных на унимодальных функциях (Розенброка и Захарова). В этих случая сепарабельная фаза дает хорошее начальное приближение для локального метода. Без использования сепарабельности локальный метод стартует из далекой от минимума точки и делает слишком много итераций до достижения условия остановки.

3.1 Алгоритм глобального поиска

В качестве базовой задачи мы будет рассматривать одномерную задачу многоэкстремальной оптимизации $\varphi^* = \varphi(x^*) = \min\{\varphi(x) : x \in [0,1]\}$, в которой целевая функция $\varphi(x)$ удовлетворяет условию Липшица. Дадим детальное описание параллельного алгоритма глобального поиска (ПАГП), применяемого к ее решению.

Пусть в нашем распоряжении имеется $p \geq 1$ вычислительных элементов. Тогда на данной итерации можно провести одновременно p испытаний. Тогда общее число испытаний, выполненных после *n* параллельных итераций, составит $k = pn$.

Предположим, что выполнено $n > 1$ итераций метода (в качестве точек $x^1,..., x^p$ первой итерации выбираются произвольные различные точки отрезка [0,1]). Тогда точки $x^{k+1},..., x^{k+p}$ текущей (*n+1*)-ой итерации определяются по следующим правилам.

Правило 1. Перенумеровать точки множества $X_k = \{x^1,..., x^k\} \cup \{0\} \cup \{1\}$ так что

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

Правило 2. Полагая $z_i = \varphi(x_i), 1 \leq i \leq k$, вычислить величины

$$\mu = \max_{1 \leq i \leq k} \frac{|z_i - z_{i-1}|}{\Delta_i}, \; M = \begin{cases} r\mu, \mu > 0, \\ 1, \mu = 0, \end{cases} \tag{3}$$

где $r > 1$ является заданным параметром метода (параметр надежности), а $\Delta_i = x_i - x_{i-1}$.

Правило 3. Для каждого интервала $(x_{i-1}, x_i), 1 \leq i \leq k+1$, вычислить характеристику в соответствии с формулами

$$R(1) = 2\Delta_1 - 4\frac{z_1}{M}, \quad R(k+1) = 2\Delta_{k+1} - 4\frac{z_k}{M}, \tag{4}$$

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{M^2\Delta_i} - 2\frac{z_i + z_{i-1}}{M}, \;\; 1 < i < k+1, \tag{5}$$

Правило 4. Характеристики $R(i), 1 \leq i \leq k+1$, упорядочить в порядке убывания

$$R(t_1) \geq R(t_2) \geq \ldots \geq R(t_{k-1}) \geq R(t_{k+1}) \tag{6}$$

и выбрать p наибольших характеристик с номерами интервалов $t_j, 1 \leq j \leq p$.

Правило 5. Провести новые испытания в точках $x^{k+j}, 1 \leq j \leq p$, вычисленных по формулам

$$\begin{aligned} x^{k+j} &= \frac{x_{t_j} + x_{t_j - 1}}{2}, \; t_j = 1, \; t_j = k+1, \\ x^{k+j} &= \frac{x_{t_j} + x_{t_j - 1}}{2} - \frac{z_{t_j} - z_{t_j - 1}}{2M}, \;\; 1 < t_j < k+1. \end{aligned} \tag{7}$$

Алгоритм прекращает работу, если выполняется условие $\Delta_{t_j} \leq \varepsilon$ хотя бы для одного номера $t_j, 1 \leq j \leq p$; здесь $\varepsilon > 0$ есть заданная точность. В качестве оценки глобально-оптимального решения задачи (1) выбираются значения

$$\varphi_k^* = \min_{1 \leq i \leq k} \varphi(x^i), \; x_k^* = \arg \min_{1 \leq i \leq k} \varphi(x^i).$$

Данный способ организации параллельных вычислений имеет следующее обоснование [14, 15]. Используемые в алгоритме характеристики интервалов (5) могут рассматриваться как некоторые меры вероятности локализации в данных интервалах точки глобального минимума. Неравенства (6) упорядочивают интервалы по их характеристикам, и испытания проводятся параллельно в первых *p* интервалах, имеющих наибольшие вероятности. Различные модификации данного алгоритма и соответствующая теория сходимости представлены в [15].

3.2 Редукция размерности с использованием кривых Пеано

Для снижения сложности алгоритмов глобальной оптимизации, формирующих неравномерное покрытие области поиска, широко используются различные схемы редукции размерности, которые позволяют свести решение многомерных оптимизационных задач к семейству задач одномерной оптимизации.

Первым из рассматриваемых способов редукции размерности является использование кривой Пеано $y(x)$, однозначно отображающей отрезок вещественной оси [0,1] на n-мерный куб

$$\left\{y \in R^N : -2^{-1} \le y_i \le 2^{-1}, 1 \le i \le N\right\} = \left\{y(x) : 0 \le x \le 1\right\}.$$

Вопросы численного построения отображений типа кривой Пеано и соответствующая теория подробно рассмотрены в [14]. Здесь же отметим, что численно построенная развертка является приближением к теоретической кривой Пеано с точностью порядка 2^{-m}, где m – параметр построения развертки.

Использование подобного рода отображений позволяет свести многомерную задачу (1) к одномерной задаче

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

Важным свойством является сохранение ограниченности относительных разностей функции: если функция $\varphi(y)$ в области D удовлетворяла условию Липшица (2) с константой L, то функция $\varphi(y(x))$ на интервале [0,1] будет удовлетворять равномерному условию Гельдера

$$\left|\varphi(y(x_1)) - \varphi(y(x_2))\right| \le H\left|x_1 - x_2\right|^{1/N},\ x_1, x_2 \in [0,1], \tag{8}$$

где константа Гельдера H связана с константой Липшица L соотношением

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

Соотношение (8) позволяет модифицировать приведенный в разделе 2 алгоритм решения одномерных задач для решения многомерных задач, редуцированных к одномерным. Для этого длины интервалов Δ_i, участвующие в правилах (3)–(5) алгоритма, заменяются на длины в новой метрике $\Delta_i = (x_i - x_{i-1})^{1/N}$,а вместо формулы (7) вводится выражение

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - \mathrm{sign}(z_{t_j} - z_{t_j-1})\frac{1}{2r}\left[\frac{|z_{t_j} - z_{t_j-1}|}{\mu}\right]^N,\ 1 < t_j < k+1.$$

3.3 Рекурсивная схема редукции размерности

Схема рекурсивной оптимизации основана на известном (см. [16]) соотношении

$$\min\{\varphi(y) : y \in D\} = \min_{a_1 \le y_1 \le b_1} \min_{a_2 \le y_2 \le b_2} \ldots \min_{a_N \le y_N \le b_N} \varphi(y), \tag{9}$$

которое позволяет заменить решение многомерной задачи (1) решением семейства одномерных подзадач, рекурсивно связанных между собой.

Введем в рассмотрение множество функций

$$\varphi_N(y_1, ..., y_N) = \varphi(y_1, ..., y_N), \tag{10}$$

$$\varphi_i(y_1, ..., y_i) = \min_{a_{i+1} \le y_{i+1} \le b_{i+1}} \varphi_{i+1}(y_1, ..., y_i, y_{i+1}),\ 1 \le i \le N-1. \tag{11}$$

Тогда, в соответствии с соотношением (9), решение исходной задачи (1) сводится к решению одномерной задачи

$$\varphi_1(y_1^*) = \min\{\varphi_1(y_1) : y_1 \in [a_1, b_1]\}. \tag{12}$$

Однако при этом каждое вычисление значения одномерной функции $\varphi_1(y_1)$ в некоторой фиксированной точке предполагает решение одномерной задачи минимизации

$$\varphi_2(y_1, y_2^*) = \min\{\varphi_2(y_1, y_2) : y_2 \in [a_2, b_2]\},$$

и так далее до вычисления φ_N согласно (10).

Для изложенной выше рекурсивной схемы предложено обобщение (блочная рекурсивная схема), которое комбинирует использование разверток и рекурсивной схемы с целью эффективного распараллеливания вычислений.

Рассмотрим вектор y как вектор блочных переменных

$$y = (y_1, y_2, ..., y_N) = (u_1, u_2, ..., u_M),$$

где *i*-я блочная переменная u_i представляет собой вектор размерности N_i из последовательно взятых компонент вектора *y*, т.е. $u_1 = (y_1, y_2, ..., y_{N_1})$, $u_2 = (y_{N_1+1}, y_{N_1+2}, ..., y_{N_1+N_2}), \ldots$, $u_2 = (y_{N-N_M+1}, y_{N-N_M+2}, ..., y_N)$, причем $N_1 + N_2 + ... + N_M = N$.

С использованием новых переменных основное соотношение многошаговой схемы (9) может быть переписано в виде

$$\min_{y \in D} \varphi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} ... \min_{u_M \in D_M} \varphi(y), \tag{13}$$

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

Формулы, определяющие способ решения задачи (1) на основе соотношений (13) в целом совпадают с рекурсивной схемой (10)–(12). Требуется лишь заменить исходные переменные $y_i, 1 \le i \le N$, на блочные переменные $u_i, 1 \le i \le M$.

При этом принципиальным отличием от исходной схемы является тот факт, что в блочной схеме вложенные подзадачи

$$\varphi_i(u_1, ..., u_i) = \min_{u_{i+1} \in D_{i+1}} \varphi_{i+1}(u_1, ..., u_i, u_{i+1}),\ 1 \le i \le M-1, \tag{14}$$

являются многомерными, и для их решения может быть применен способ редукции размерности на основе кривых Пеано.

Число векторов и количество компонент в каждом векторе являются параметрами блочной многошаговой схемы и могут быть использованы для формирования подзадач с нужными свойствами. Например, если $M = N$, т.е. $u_i = y_i, 1 \le i \le N$, то блочная схема идентична исходной; каждая из вложенных подзадач является одномерной. А если $M = 1$, т.е. $u = u_1 = y$, то решение задачи эквивалентно ее решению с использованием единственной развертки, отображающей [0,1] в *D*; вложенные подзадачи отсутствуют.

3.4 Метод Хука Дживса

Еще одна реализованная модификация базового метода состояла в прямом использовании метода локальной оптимизации, а именно метода Хука-Дживса [17]). Схематически работа комбинированного метода такова:

- Глобальная фаза
  - Выполнять итерации АГП, пока не будет обновлен текущий «рекорд» (минимальное значение целевой функции в уже посчитанных точках испытаний).
- Локальная фаза
  - Запустить из точки текущего рекорда локального метод Хука-Дживса с заданным условием выхода по точности.
  - Все точки испытаний локальной метода добавлять в базу точек АГП.
  - По достижении локальным методом заданной точности перейти к Глобальной фазе.

Точность локального метода (условие остановки) для всех классов задач GKLS, а также Растригина и композитных была взята равной 0.00001, для функции Розенброка – 0.0000005, для функции Захарова – 0.000001.

4. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на одном из узлов высокопроизводительного кластера ННГУ им. Н.И. Лобачевского. Узел кластера располагает 2-я процессорами Intel Xeon L5630 2.13 GHz, 24 Gb RAM. Центральный процессор является 4-х ядерным

Рассмотренные в разделе 3 методы и их модификации реализованы в решателе Globalizer, предназначенном для параллельного решения многомерных многоэкстремальных задач глобальной оптимизации, разрабатываемом в ННГУ им. Н.И. Лобачевского. Алгоритмическую основу решателя Globalizer составляют алгоритм глобального поиска и блочная многошаговая схема редукции размерности. По условиям конкурса при решении задач использовался только последовательный режим работы решателя. Однако Globalizer поддерживает работу как на системах с распределенной памятью (используя MPI), так и с общей памятью (использую OpenMP). Кроме того, поддерживаются графические процессоры NVIDIA и сопроцессоры Intel Xeon Phi.

В финальной части конкурса первое место в общем зачете занял коллектив Эдуардо Сегредо, Бен Пичтер, Эмма Харт из Школа вычислительной техники, Университет Напьер в Эдинбурге, Великобритания и Эдуардо Лалла-Руис, Стефан Восс из Института информационных систем, Университет Гамбурга, Германия. Они использовали алгоритмы дифференциальной эволюции и генетический алгоритм. Коллектив ННГУ Н.И. Лобачевского в который входили: А.В. Сысоев, К.А. Баркалов, В.П. Гергель, И.Г. Лебедев, В.В. Соврасов, занял 3-е место в общем зачете и 1-е по общему числу решенных задач (http://forum.genopt.org/genopt_final.html). Использовался последовательный алгоритм глобального поиска и его модификации.

Распределение решенных задач по классам представлено в таблице 2. Во всех запусках точность построения развертки m равнялась 10. Параметр метода r был равен 2.5 для всех задач, кроме классов GKLS. Для GKLS r варьировался от 2.5 до 200.

Таблица 2. Процент решенных задач по классам

Класс	Решилось
GKLS_nd_10	99
GKLS_nd_30	15
GKLS_cd_10	96
GKLS_cd_30	1
GKLS_td_10	94
GKLS_td_30	0
Rosenbrock_10	100
Rosenbrock_30	100
Rastrigin_10	100
Rastrigin_30	100
Zakharov_10	100
Zakharov_30	100
Composite_10	100
Composite_30	100

Для всех классов, кроме GKLS использовалась модификация с сепарабельной фазой, с точностью 0.02.

При решении задач GKLS сначала использовались базовая версия алгоритма глобального поиска с редукцией размерности с использованием кривой Пеано и локальное уточнение. Затем для нерешенных задач вначале запускается случайный поиск. Распределение числа решенных задач без использования и с использованием случайного поиска указано в таблице 3.

Таблица 3. Использование случайного поиска при решении задач классов GKLS

Класс	Решилось без случайного поиска	Решилось со случайным поиском
GKLS_nd_10	78	99
GKLS_nd_30	0	15
GKLS_cd_10	67	96
GKLS_cd_30	0	1
GKLS_td_10	65	94
GKLS_td_30	0	0

Ниже приведены результаты численного сравнения двух последовательных алгоритмов – DIRECT [1] и алгоритма глобального поиска (АГП без случайного поиска). Численное сравнение проводилось на классах функций GKLS_nd, GKLS_cd и GKLS_td размерности 10. Максимально допустимое число итераций составляло K_{max} = 1 000 000. В таблице 4 отражено среднее число итераций и количество решенных задач данного класса.

Таблица 4. Использование случайного поиска при решении задач классов GKLS

Класс	Direct		АГП	
	Решилось задач	Среднее число итераций	Решилось задач	Среднее число итераций
GKLS_nd_10	42	182439	78	118377
GKLS_nd_10	43	397291	67	269450
GKLS_nd_10	45	450101	65	351866

Как видно из таблицы 4, АГП превосходит метод DIRECT на рассмотренных классах задач как по среднему числу итераций так и по числу решенных задач.

Оценим теперь ускорение, которое достигается при использовании параллельного алгоритма глобального поиска, в зависимости от числа P используемых ядер. Численное сравнение проводилось на классах функций GKLS_nd, GKLS_cd и GKLS_td размерности 10. Максимально допустимое число итераций составляло K_{max} = 1 000 000. В таблице 5 приведено число решившихся задач, в таблице 6 – среднее время решения одной задачи, а в таблице 7 – ускорение по времени относительно последовательного запуска.

Таблица 5. Количество решенных задач

Класс	P = 2	P = 4	P = 8	P = 16
GKLS_nd_10	81	88	84	82
GKLS_cd_10	78	85	85	78
GKLS_td_10	73	81	82	78

Таблица 6. Среднее время решения задачи

Класс	P = 2	P = 4	P = 8	P = 16
GKLS_nd_10	15,80	16,30	15,01	12,55
GKLS_cd_10	38,09	50,89	46,23	38,42
GKLS_td_10	57,99	65,29	62,35	51,79

Таблица 7. Ускорение по времени

Класс	P = 2	P = 4	P = 8	P = 16
GKLS_nd_10	1,48	1,43	1,56	1,86
GKLS_cd_10	1,40	1,04	1,15	1,38
GKLS_td_10	1,20	1,06	1,11	1,34

Результаты экспериментов показывают преимущество параллельного алгоритма по сравнению с последовательным и по числу решенных задач, и по времени решения. Но поскольку для десятимерных задач необходимо использовать расширенную точность при вычислениях, то параллельное проведение испытаний не дает значительного ускорения.

Далее приведены результаты для многопроцессорного запуска. Численное сравнение проводилось на классах функций GKLS_nd, GKLS_cd и GKLS_td размерности 10. В соответствии с блочной рекурсивной схемой было использовано два уровня подзадач с размерностями N_1 = 8(один процесс), N_2 = 2(четыре процесса), r принято 2,5 и 3,5. Эксперименты проводились на одном узле кластера. В таблице 8 приведено число решившихся задач, среднее время решения и ускорение по времени относительно последовательного запуска.

Таблица 8. Многопроцессорный запуск

Класс	Решилось задач	среднее время	ускорение
GKLS_nd_10	87	10,75	2,18
GKLS_cd_10	86	19,49	2,73
GKLS_td_10	80	21,37	3,25

В многопроцессорном режиме при использовании того же оборудования, как и в многопоточном, решилось немного больше задач в каждом классе, при этом уменьшилось среднее время решения.

5. Заключение

В работе представлены результаты исследования параллельного алгоритма глобальной оптимизации при решении задач конкурса GenOpt 2016. Описаны использованные методы оптимизации и их модификации, направленные на ускорение времени решения задачи на современных вычислительных системах. Все рассмотренные модификации реализованы в системе Globalizer, использовавшейся при проведении экспериментов. С помощью данной программной системы решались10-ти и 30-тимерные задачи из класса GKLS; проведено сравнение результатов, полученных при последовательном и параллельном решении. Использование параллельного алгоритма позволило как сократить среднее время решения, так и увеличить процент корректно решившихся задач.

Наибольший интерес для дальнейшего исследования представляют 30-тимерные задачи семейства GKLS. Для их решения в планируется использовать большую степень параллелизма: увеличить число уровней в блочной рекурсивной схеме редукции размерности, использовать множественные отображения на основе кривых Пеано вместо единственного отображения.

Литература

1. Gaviano M., Lera D., Kvasov D. E., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // ACM Transactions on Mathematical Software. 2003. Vol. 29. P. 469-480.
2. Сергеев Я.Д., Квасов Д.Е. Диагональные методы глобальной оптимизации. М.: Физматлит, 2008. 352 с.
3. Sergeyev, Ya.D., Grishagin, V.A.: Sequential and parallel global optimization algorithms // Optimization Methods and Software. 1994. Vol. 3, No. 3. P. 111–124.
4. В.П. Гергель. Об одном способе учета значений производных при минимизации многоэкстремальных функций //Журнал вычислительной математики и математической физики. 1996. Т. 36, № 6. С. 51–67.
5. Gergel V.P. A global optimization algorithm for multivariate functions with Lipschitzian first derivatives // Journal of Global Optimization. 1997. Vol. 10 No. 3. P. 257–281.
6. Grishagin V.A., Sergeyev Ya.D., Strongin R.G. Parallel characteristical algorithms for solving problems of global optimization // Journal of Global Optimization. 1997. Vol. 10 No. 2. P. 185–206.
7. Gergel V.P., Sergeyev Ya.D. Sequential and parallel algorithms for global minimizing functions with Lipschitzian derivatives // Computers and Mathematics with Applications 1999. Vol. 37 No. 4-5. P. 163–179.
8. Sergeyev Ya.D, Grishagin V.A. Parallel asynchronous global search and the nested optimization scheme // Journal of Computational Analysis and Applications. 2001. Vol. 3 No. 2. P. 123–145.

9. Баркалов К. А., Стронгин Р. Г. Метод глобальной оптимизации с адаптивным порядком проверки ограничений//Журнал вычислительной математики и математической физики. 2002 Т. 42, № 9, С. 1338–1350.
10. Strongin R.G., Sergeyev Ya.D. Global optimization: fractal approach and non redundant parallelism // Journal of Global Optimization. 2003. Vol 27 No. 1, P. 25-50.
11. Gergel V.P., Strongin R.G. Parallel computing for globally optimal decision making on cluster systems // Future Generation Computer Systems 2005. Vol. 21 No. 5. P. 673-678.
12. Barkalov K.A., Gergel V.P. Multilevel scheme of dimensionality reduction for parallel global search algorithms. // OPT-i: Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization 2014. P. 2111-2124.
13. Gergel V., Grishagin V., Israfilov R. Local tuning in nested scheme of global optimization. // Procedia Computer Science, 2015. Vol. 51, P. 865-874.
14. Strongin R.G., Sergeyev Ya.D. Global optimization with non-convex constraints. Sequential and parallel algorithms. Dordrecht, Kluwer Academic Publishers, 2000.
15. Стронгин Р.Г., Гергель В.П., Гришагин В.А., Баркалов К.А. Параллельные вычисления в задачах глобальной оптимизации. М.: Издательство Московского университета. 2013. 280 с.
16. Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация. Н.Новгород: Изд-во ННГУ, 2007.
17. Химмельблау Д. Прикладное нелинейное программирование. М.: Мир, 1975. 536 с.

Use of parallel Globalizer solver for solving global optimization problems from GenOpt competition

I.G. Lebedev, V.V. Sovrasov

Lobachevsky State University of Nizhny Novgorod

In this work parallel algorithms for solving multiextremal optimization problems are considered. These algorithms have been developed in the information-statistical approach proposed in Lobachevsky State University of Nizhny Novgorod and have been implemented in parallel software Globalizer. Results of numerical experiments carried out with test function from international competition of optimization algorithms GenOpt (www.genopt.org) are discussed.

Keywords: global optimization, multiextremal functions, dimension reduction, parallel algorithms

References

1. Gaviano M., Lera D., Kvasov D. E., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // ACM Transactions on Mathematical Software. 2003. Vol. 29. P. 469-480.
2. Sergeyev Y.D., Kvasov D.E. Diagonal'nyye metody global'noy optimizatsii. Fizmatlit, 2008. P. 352.
3. Sergeyev, Ya.D., Grishagin, V.A.: Sequential and parallel global optimization algorithms // Optimization Methods and Software. 1994. Vol. 3, No. 3. P. 111–124.
4. Gergel V.P. A method of using derivatives in the minimization of multiextremum functions // Computational Mathematics and Mathematical Physics. 1996. Vol. 36 No. 6. P. 729–742.
5. Gergel V.P. A global optimization algorithm for multivariate functions with Lipschitzian first derivatives // Journal of Global Optimization. 1997. Vol. 10 No. 3. P. 257–281.
6. Grishagin V.A., Sergeyev Ya.D., Strongin R.G. Parallel characteristical algorithms for solving problems of global optimization // Journal of Global Optimization. 1997. Vol. 10 No. 2. P. 185–206.
7. Gergel V.P., Sergeyev Ya.D. Sequential and parallel algorithms for global minimizing functions with Lipschitzian derivatives // Computers and Mathematics with Applications 1999. Vol. 37 No. 4-5. P. 163–179.
8. Sergeyev Ya.D, Grishagin V.A. Parallel asynchronous global search and the nested optimization scheme // Journal of Computational Analysis and Applications. 2001. Vol. 3 No. 2. P. 123–145.
9. Barkalov K.A., Strongin R.G. A global optimization technique with an adaptive order of checking for constraints. // Computational Mathematics and Mathematical Physics. 2002 Vol. 42. No 9. P. 1338–1350.
10. Strongin R.G., Sergeyev Ya.D. Global optimization: fractal approach and non redundant parallelism // Journal of Global Optimization. 2003. Vol 27 No. 1, P. 25-50.
11. Gergel V.P., Strongin R.G. Parallel computing for globally optimal decision making on cluster systems // Future Generation Computer Systems 2005. Vol. 21 No. 5. P. 673-678.
12. Barkalov K.A., Gergel V.P. Multilevel scheme of dimensionality reduction for parallel global search algorithms. // OPT-i: Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization 2014. P. 2111-2124.

13. Gergel V., Grishagin V., Israfilov R. Local tuning in nested scheme of global optimization. // Procedia Computer Science, 2015. Vol. 51, P. 865-874.
14. Strongin R.G., Sergeyev Ya.D. Global optimization with non-convex constraints. Sequential and parallel algorithms. Dordrecht, Kluwer Academic Publishers, 2000.
15. Strongin R., Gergel V., Grishagin V., Barkalov K. Parallel calculations in global optimization. Moscow, Publishing of the Moscow State University. 2013. P. 280.
16. Gorodetskiy S., Grishagin V. Nonlinear programming and multiextremal optimization. Nizhni Novgorod, Publishing of the Nizhni Novgorod State University, 2007.
17. Himmelblau D.M. Applied Nonlinear Programming. New York, McGraw-Hill, 1972. 498 P.