

МРІ-реализация блочной многошаговой схемы параллельного решения задач глобальной оптимизации*

А.В. Сысоев, К.А. Баркалов, В.П. Гергель, И.Г. Лебедев

Нижегородский государственный университет им. Н.И. Лобачевского

Представлен новый подход к решению задач глобальной оптимизации, комбинирующий информационно-статистический алгоритм, разработанный в ННГУ им. Н.И. Лобачевского с блочной схемой редукции размерности. Предложен параллельный алгоритм, реализующий данный подход. Представлены синхронная и асинхронная схемы МРІ-реализации указанного алгоритма. Приведены результаты сравнения схем, показывающие преимущество асинхронного варианта.

1. Введение

Рассмотрим задачу многомерной многоэкстремальной оптимизации в виде

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

где $\varphi(y)$ – действительная функция, $a, b \in R^N$ есть заданные векторы.

Численное решение задачи (1) состоит в отыскании оценки

$$y_k^* \in D \quad (2)$$

близкой к точке y^* , где k – число вычислений значений оптимизируемой функции. Близость может быть определена, например, так

$$\|y^* - y_k^*\| \leq \varepsilon, \quad (3)$$

где $\varepsilon > 0$ есть заданная точность. При этом будем рассматривать задачи, где функция $\varphi(y)$ удовлетворяет условию Липшица

$$|\varphi(y_1) - \varphi(y_2)| \leq L \|y_1 - y_2\|, \quad y_1, y_2 \in D, \quad 0 < L < \infty, \quad (4)$$

и может быть задана некоторым алгоритмом вычисления ее значений в точках области D ; при этом *испытание* (вычисление одного значения) является вычислительно-трудоемкой операцией.

Как известно, вычислительные затраты на решение задачи (1) растут экспоненциально с ростом размерности, что делает актуальным распараллеливание процесса поиска оценки (2). В данной работе рассматривается подход, в котором схема распараллеливания основана на одновременном проведении нескольких испытаний в выбранных по некоторому правилу точках.

Данная статья продолжает серию исследований, начальные результаты которых были отражены в [1, 2].

2. Базовый параллельный алгоритм глобального поиска

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 за-

* Работа поддержана грантом МОН РФ (соглашение от 27 августа 2013 г. № 02.В.49.21.0003 между МОН РФ и ННГУ им. Н.И. Лобачевского).

дач одномерной оптимизации. Поэтому в качестве базовой задачи мы будем рассматривать одномерную задачу многоэкстремальной оптимизации

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

в которой целевая функция $\varphi(x)$ удовлетворяет условию Липшица.

Дадим краткое описание параллельного алгоритма глобального поиска (ПАГП), применяемого к решению задачи (5).

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

Предположим, что выполнено $n > 1$ итераций метода (в качестве точек x^1, \dots, x^p первой итерации выбираются произвольные различные точки отрезка $[0,1]$). Тогда точки x^{k+1}, \dots, x^{k+p} текущей $(n+1)$ -ой итерации определяются по правилам выбранного метода из класса характеристически представимых [3]. При этом характеристика интервала (x_{i-1}, x_i) , $2 \leq i \leq k$ рассматривается как некоторая мера вероятности нахождения в данном интервале точки глобального минимума, а испытания проводятся параллельно в первых p интервалах, имеющих наибольшие вероятности. Различные модификации одного из эффективных алгоритмов данного типа, информационно-статистического, и соответствующая теория сходимости представлены в [4].

3. Блочная многошаговая схема редукции размерности

Один из подходов к решению многомерных задач глобальной оптимизации состоит в их сведении к одномерным и использовании для решения редуцированной задачи эффективных одномерных алгоритмов глобального поиска. При этом редукция может применяться как к области D из (1), взаимно однозначно отображая гиперпараллелепипед D на отрезок $[0,1]$, так и к функции $\varphi(y)$, минимизацию которой можно выполнять на основе рекурсивной схемы (см. [5])

$$\min \{ \varphi(y) : y \in D \} = \min_{a_1 \leq y_1 \leq b_1} \min_{a_2 \leq y_2 \leq b_2} \dots \min_{a_N \leq y_N \leq b_N} \varphi(y). \quad (6)$$

В [3] подробно рассмотрены вопросы численного построения *развертки* на основе кривой Пеано $y(x)$, однозначно отображающей отрезок вещественной оси $[0,1]$ на n -мерный гиперкуб

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

Развертка является приближением к кривой Пеано с точностью порядка 2^{-m} , где m – параметр построения.

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

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

Для многошаговой схемы редукции (6) предложено обобщение [1], комбинирующее использование разверток с рекурсивной редукцией и позволяющее существенно повысить эффективность распараллеливания вычислений.

Рассмотрим вектор y как вектор блочных переменных

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

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

С использованием новых переменных основное соотношение многошаговой схемы (6) может быть переписано в виде

$$\min_{y \in D} \varphi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \varphi(y), \quad (7)$$

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

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

$$\varphi_i(u_1, \dots, u_i) = \min_{u_{i+1} \in D_{i+1}} \varphi_{i+1}(u_1, \dots, u_i, u_{i+1}), \quad 1 \leq i \leq M-1 \quad (8)$$

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

4. Параллельная блочная многошаговая схема

Параллельная модификация блочной многошаговой схемы может быть выполнена следующим образом. Введем вектор распараллеливания

$$\pi = (\pi_1, \pi_2, \dots, \pi_M), \quad (9)$$

где $\pi_i, 1 \leq i < M$ – число параллельно решаемых подзадач $(i+1)$ -го уровня вложенности, возникающих в результате выполнения параллельных итераций на i -м уровне. Для M -го уровня число π_M означает количество параллельных испытаний в процессе минимизации функции $\varphi_M(u_1, \dots, u_M) = \varphi(y_1, \dots, y_N)$ по переменной u_M при фиксированных значениях u_1, \dots, u_{M-1} , т.е. количество параллельно вычисляемых значений целевой функции $\varphi(y)$. В данной работе мы считаем, что компоненты вектора (9) не меняются в процессе решения задачи (7), а $\pi_M = 1$. Тогда общее число задействованных процессоров/ядер будет составлять

$$\Pi = 1 + \sum_{i=1}^{M-1} \prod_{j=1}^i \pi_j. \quad (10)$$

На рисунке 1 приведена общая схема организации вычислений при $M = 3$.

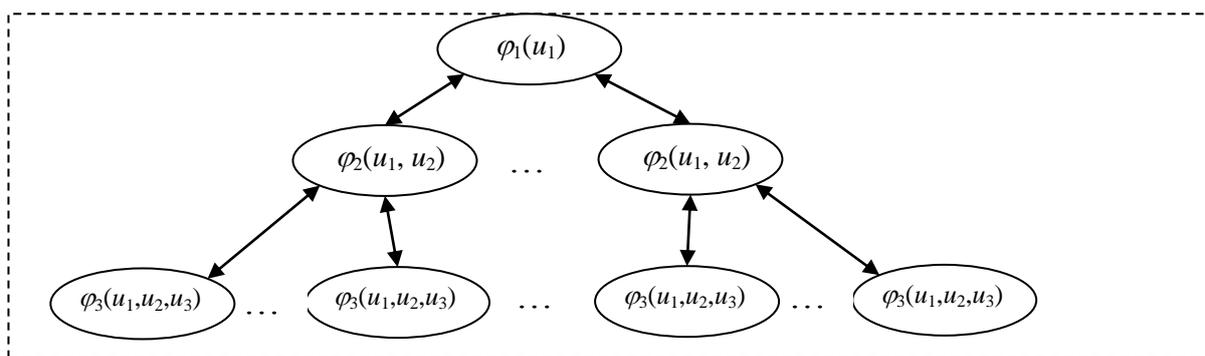


Рис. 1. Схема организации параллельных вычислений ($M = 3$)

5. Синхронная MPI-версия блочной многошаговой схемы

Соотношение (10) задает число MPI-процессов, которые должны быть созданы при запуске MPI-реализации блочной многошаговой схемы, а вектор (9) позволяет выстроить процессы в дерево, корневой процесс которого решает исходную задачу (1), процессы нижележащих уровней – подзадачи из (8), а процессы-листья кроме того вычисляют значения целевой функции $\varphi(y)$.

Простым вариантом взаимодействия процессов в дереве является синхронный, при котором каждый нетерминальный процесс раздает подчиненным процессам по одной подзадаче, дожидается, пока все пришлют найденные ими решения, после чего раздает им новые подзадачи.

Опишем общие схемы работы *корневого* процесса, *нетерминальных* и *терминальных* процессов в этом случае.

5.1. Схема работы корневого процесса

1. Если $N_1 > 1$ выполнить редукцию размерности, используя развертку Пеано, сведя задачу к одномерной.
2. Выбрать π_1 точек в интервалах с наилучшими характеристиками (на первой итерации выбрать произвольные π_1 точек).
3. Передать соответствующие зафиксированные N_1 координат вектора y (y_1, y_2, \dots, y_{N_1}) каждому из π_1 подчиненных процессов (далее потомков).
4. Дождаться решения порожденных подчиненными процессами подзадач, принять от каждого из них данные (оценку точки минимума, значение целевой функции).
5. Проверить условие остановки.
 - a. Если выполнено, СТОП.
 - b. Если не выполнено, перейти к шагу 2.

Указанная схема соответствует шаблону взаимодействия процессов «мастер-рабочий».

5.2. Схема работы нетерминального процесса

0. Принять от процесса-мастера (далее родителя) точку с фиксированными координатами u_1, \dots, u_{i-1} .
1. Если $N_i > 1$ выполнить редукцию размерности, используя развертку Пеано, сведя задачу к одномерной.
2. Выбрать π_i точек в интервалах с наилучшими характеристиками (на первой итерации выбрать произвольные π_i точек).
3. Передать соответствующие зафиксированные координаты (u_1, \dots, u_i) каждому из π_i потомков.
4. Дождаться решения порожденных потомками подзадач, принять от каждого из них данные (оценку точки минимума, значение целевой функции).
5. Проверить условие остановки.
 - a. Если выполнено, передать данные (оценку точки минимума, значение целевой функции) родителю. Перейти к шагу 0.
 - b. Если не выполнено, перейти к шагу 2.

5.3. Схема работы терминального процесса

0. Принять от родителя точку с фиксированными координатами u_1, \dots, u_{M-1} .
1. Если $N_M > 1$ выполнить редукцию размерности, используя развертку Пеано, сведя задачу к одномерной.
2. Решить редуцированную задачу оптимизации.
3. Передать данные (оценку точки минимума, значение целевой функции) родителю.
4. Перейти к шагу 0.

6. Асинхронная MPI-версия блочной многошаговой схемы

Синхронная версия блочной многошаговой схемы обладает существенным недостатком, связанным с возможными простоями всех узлов дерева процессов, кроме корневого. Простои могут возникать в том случае, если часть потомков некоторого процесса закончили решение своих подзадач и отправили данные родителю раньше остальных, поскольку родитель создаст для этих потомков новые подзадачи только после получения решений от всех своих потомков. С ростом числа уровней M в дереве процессов, указанные простои могут приводить к существенной потере эффективности, что подтверждается результатами экспериментов (раздел 7).

Для преодоления данной проблемы была разработана асинхронная версия блочной многошаговой схемы. Алгоритм работы терминальных процессов в этой схеме совпадает с алгоритмом в синхронном случае. Схемы для корневого и нетерминальных процессов представлены далее.

6.1. Схема работы корневого процесса

1. Если $N_1 > 1$ выполнить редукцию размерности, используя развертку Пеано, сведя задачу к одномерной.
2. Выбрать π_1 точек на отрезке $[0,1]$.
3. Передать зафиксированные координаты y_1, y_2, \dots, y_{N_1} каждому из π_1 потомков.
4. Дождаться решения порожденных подчиненными процессами подзадач, принять от каждого из них данные (оценку точки минимума, значение целевой функции).
5. Выбрать π_1 точек в интервалах с наилучшими характеристиками.
6. Передать зафиксированные координаты y_1, y_2, \dots, y_{N_1} каждому из π_1 потомков.
7. Принять решение подзадачи от любого потомка, приславшего данные.
8. Проверить условие остановки. Если выполнено, СТОП.
9. Выбрать одну точку очередного испытания в интервале с наилучшей характеристикой.
10. Передать зафиксированные координаты y_1, y_2, \dots, y_{N_1} тому потомку, от которого приняли решение подзадачи на шаге 7.
11. Перейти к шагу 7.

Указанная схема соответствует шаблону взаимодействия процессов «мастер-рабочий».

6.2. Схема работы нетерминального процесса

0. Принять от родителя точку с фиксированными координатами u_1, \dots, u_{i-1} .
1. Если $N_i > 1$ выполнить редукцию размерности, используя развертку Пеано, сведя задачу к одномерной.
2. Выбрать π_i точек на отрезке $[0,1]$.
3. Передать зафиксированные координаты (u_1, \dots, u_i) каждому из π_i потомков.
4. Дождаться решения порожденных потомками подзадач, принять от каждого из них данные (оценку точки минимума, значение целевой функции).
5. Выбрать π_i точек в интервалах с наилучшими характеристиками.
6. Передать зафиксированные координаты (u_1, \dots, u_i) каждому из π_i потомков.
7. Принять решение подзадачи от любого потомка, приславшего данные.
8. Проверить условие остановки. Если выполнено, передать данные (оценку точки минимума, значение целевой функции) родителю. Перейти к шагу 0.
9. Выбрать одну точку очередного испытания в интервале с наилучшей характеристикой.
10. Передать зафиксированные координаты (u_1, \dots, u_i) тому потомку, от которого приняли решение подзадачи на шаге 7.

11. Перейти к шагу 7.

7. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на суперкомпьютере «Лобачевский» (операционная система – CentOS 6.4, система управления – SLURM). Один узел суперкомпьютера располагает двумя процессорами Intel Sandy Bridge E5-2660 2.2 GHz (8 ядер в каждом), 64 Gb RAM, сеть InfiniBand. Использовался компилятор Intel C++ 14.0.2.

В качестве тестовых задач были выбраны функции Растригина, Розенброка и Ньюмайера. Формульное задание функций представлено ниже.

Функция Растригина

$$\varphi(y_1, \dots, y_N) = 10N + \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i))$$

Функция Розенброка

$$\varphi(y_1, \dots, y_N) = \sum_{i=1}^{N-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

Функция Ньюмайера

$$\varphi(y_1, \dots, y_N) = \sum_{i=1}^N (x_i - 1)^2 - \sum_{i=2}^N x_i x_{i-1}$$

Эксперименты направлены на сравнение синхронной и асинхронной схем, а также на выяснение того, каким образом распределение размерностей по уровням влияет на время работы программы. Результаты представлены в таблицах 1, 2, 3. Эксперименты проводились при числе уровней в дереве процессов $M = 2$ и $M = 3$. Использовались вектора $\pi^1 = (3,1)$ и $\pi^2 = (3,3,1)$ соответственно. Размерность задачи N и распределение параметров по уровням блочной схемы указаны в первом столбце таблиц.

Таблица 1. Сравнение синхронной и асинхронной версий, функция Растригина

| Размерность, распределение | Время (в секундах) | | Количество испытаний в корне | | Ускорение |
|----------------------------|--------------------|-------------------|------------------------------|-------------------|-----------|
| | Синхронная схема | Асинхронная схема | Синхронная схема | Асинхронная схема | |
| 3 | | | | | |
| (1, 2) | 0.063 | 0.064 | 33 | 29 | 0.994 |
| (2, 1) | 0.011 | 0.004 | 552 | 134 | 2.725 |
| (1, 1, 1) | 0.040 | 0.028 | 33 | 28 | 1.460 |
| 4 | | | | | |
| (1, 3) | 5.874 | 5.819 | 33 | 29 | 1.009 |
| (2, 2) | 0.156 | 0.041 | 522 | 144 | 3.771 |
| (3, 1) | 0.069 | 0.007 | 3429 | 307 | 10.105 |
| 5 | | | | | |
| (1, 4) | 98.925 | 102.799 | 33 | 31 | 0.962 |
| (2, 3) | 12.310 | 2.273 | 522 | 101 | 5.416 |
| (3, 2) | 1.335 | 0.096 | 4671 | 399 | 13.848 |
| (4, 1) | 0.465 | 0.091 | 22095 | 4908 | 5.103 |
| (1, 2, 2) | 16.968 | 9.799 | 33 | 29 | 1.732 |
| (2, 1, 2) | 7.362 | 1.316 | 552 | 122 | 5.594 |
| (2, 2, 1) | 11.756 | 0.922 | 552 | 129 | 12.745 |
| (3, 1, 1) | 21.227 | 1.506 | 3429 | 977 | 14.096 |
| (1, 3, 1) | 71.707 | 52.729 | 33 | 30 | 1.360 |

| | | | | | |
|-----------|--------|--------|----|----|-------|
| (1, 1, 3) | 15.900 | 14.588 | 33 | 25 | 1.090 |
|-----------|--------|--------|----|----|-------|

Таблица 2. Сравнение синхронной и асинхронной версий, функция Розенброка

| Размерность, распределение | Время (в секундах) | | Количество испытаний в корне | | Ускорение |
|----------------------------|--------------------|-------------------|------------------------------|-------------------|-----------|
| | Синхронная схема | Асинхронная схема | Синхронная схема | Асинхронная схема | |
| 3 | | | | | |
| (1, 2) | 0.818 | 0.868 | 90 | 97 | 0.942 |
| (2, 1) | 0.046 | 0.010 | 3087 | 710 | 4.513 |
| (1, 1, 1) | 0.117 | 0.055 | 111 | 58 | 2.122 |
| 4 | | | | | |
| (1, 3) | 68.473 | 69.708 | 69 | 70 | 0.982 |
| (2, 2) | 12.832 | 7.236 | 2691 | 1931 | 1.773 |
| (3, 1) | 1.391 | 0.535 | 88428 | 48930 | 2.599 |
| 5 | | | | | |
| (1, 4) | 179.227 | 182.465 | 69 | 68 | 0.982 |
| (2, 3) | 680.999 | 389.584 | 2283 | 1429 | 1.748 |
| (3, 2) | 42.797 | 371.367 | 7653 | 77807 | 0.115 |
| (4, 1) | 39.529 | 16.600 | 2332854 | 1380012 | 2.381 |
| (1, 2, 2) | 1807.803 | 1261.755 | 69 | 62 | 1.433 |
| (2, 1, 2) | 557.419 | 263.405 | 2337 | 1085 | 2.116 |
| (2, 2, 1) | 117.939 | 43.707 | 2337 | 1469 | 2.698 |
| (3, 1, 1) | 58.402 | 2.781 | 7911 | 1411 | 21.001 |
| (1, 3, 1) | 635.585 | 249,231 | 69 | 39 | 2.550 |
| (1, 1, 3) | 774.647 | 780.870 | 69 | 67 | 0.992 |

Таблица 3. Сравнение синхронной и асинхронной версий, функция Ньюмайера

| Размерность, распределение | Время (в секундах) | | Количество испытаний в корне | | Ускорение |
|----------------------------|--------------------|-------------------|------------------------------|-------------------|-----------|
| | Синхронная схема | Асинхронная схема | Синхронная схема | Асинхронная схема | |
| 3 | | | | | |
| (1, 2) | 0.115 | 0.119 | 39 | 39 | 0,965 |
| (2, 1) | 0.007 | 0.004 | 366 | 136 | 1,999 |
| (1, 1, 1) | 0.046 | 0.024 | 39 | 31 | 1,927 |
| 4 | | | | | |
| (1, 3) | 9.054 | 8.445 | 39 | 35 | 1,072 |
| (2, 2) | 0.241 | 0.163 | 396 | 258 | 1,480 |
| (3, 1) | 0.061 | 0.032 | 3705 | 2274 | 1,878 |
| 5 | | | | | |
| (1, 4) | 103.608 | 109.979 | 39 | 38 | 0,942 |
| (2, 3) | 33.260 | 19.939 | 465 | 351 | 1,668 |
| (3, 2) | 1.952 | 1.232 | 3060 | 2072 | 1,584 |
| (4, 1) | 0.633 | 0.108 | 37182 | 8089 | 5,884 |
| (1, 2, 2) | 55.894 | 33.081 | 39 | 33 | 1,690 |
| (2, 1, 2) | 9.822 | 5.910 | 435 | 246 | 1,662 |
| (2, 2, 1) | 7.634 | 0.485 | 432 | 72 | 15,741 |
| (3, 1, 1) | 20.166 | 3.679 | 3141 | 2138 | 5,482 |
| (1, 3, 1) | 85.942 | 49.078 | 39 | 35 | 1,751 |
| (1, 1, 3) | 50,658 | 58.330 | 39 | 38 | 0,868 |

По результатам проведенных экспериментов можно видеть, что асинхронная версия в большинстве случаев работает быстрее. Максимальное достигнутое ускорение ~21 на функции Розенброка и распределении размерностей (3, 1, 1). При этом в некоторых случаях асинхрон-

ный алгоритм работает примерно так же, как синхронный. Также можно заметить, что корень дерева в асинхронной схеме делает меньше или столько же итераций, сколько в синхронной.

Что касается распределения числа параметров по уровням дерева, из таблиц 1-3 видно, что при $M = 2$ наилучший результат достигается при распределении вида $(N-1, 1)$. При $M = 3$ этот эффект заметен не так явно, но в целом видна тенденция уменьшения времени при увеличении значений первых элементов в векторе (N_1, N_2, N_3) .

8. Заключение

Основным результатом работы являются синхронная и асинхронная реализации параллельной блочной многошаговой схемы решения многомерных задач глобальной оптимизации. Данная схема сочетает свойства двух подходов к редукции многомерных задач: редукции размерности на основе кривых Пеано и рекурсивной (многошаговой) редукции целевой функции. На каждом уровне предложенной блочной схемы распараллеливание выполняется «по характеристикам». В работе показано преимущество асинхронной схемы взаимодействия процессов над синхронным аналогом.

В дальнейшем планируется дополнить разработанную схему возможностью использования множественных разверток при решении многомерных подзадач.

Литература

1. K. Barkalov, V. Gergel. Multilevel scheme of dimensionality reduction for parallel global search algorithms // OPT-i 2014. An International Conference on Engineering and Applied Sciences Optimization (Kos Island, Greece, 4–6 June 2014). 2014. pp. 2111–2124.
2. А.В. Сысоев, К.А. Баркалов, В.П. Гергель. Блочная многошаговая схема параллельного решения задач многомерной глобальной оптимизации // Материалы XIV Международной конференции "Высокопроизводительные параллельные вычисления на кластерных системах". 10-12 ноября, ПНИПУ, Пермь. 2014. С. 425–432.
3. R.G. Strongin, Ya.D. Sergeyev, Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000.
4. 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.
5. Городецкий С.Ю., Гришагин В.А. Нелинейное программирование и многоэкстремальная оптимизация. Н.Новгород: Изд-во ННГУ, 2007.

MPI implementation of dimension reduction multilevel scheme for parallel solving the global optimization problems

Alexander Sysoyev, Konstantin Barkalov, Victor Gergel and Ilya Lebedev

Keywords: global optimization, information-statistical algorithm, dimension reduction, multilevel scheme, synchronous scheme, asynchronous scheme, cluster

The paper presents the new approach to solve the global optimization problems that combines the information-statistical algorithm developed in the University of Nizhni Novgorod with multilevel scheme of dimension reduction. A parallel algorithm that implements such approach is suggested and its synchronous and asynchronous MPI implementation is presented. The advantage of asynchronous scheme is shown by comparing with synchronous one.