

# МОДИФИКАЦИИ АДАПТИВНЫХ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ МНОГОМЕРНОЙ МНОГОЭКСТРЕМАЛЬНОЙ ОПТИМИЗАЦИИ

А.В. Гергель, Д.В. Гнатюк

Сложность решения многомерных многоэкстремальных задач оптимизации обусловлена несколькими причинами. В первую очередь стоит отметить экспоненциальный рост объема вычислений при увеличении размерности (числа варьируемых параметров). Во вторых вычисление значений функционалов оптимизационной задачи требует большого количества вычислений. Именно на такие вычислительно-трудоемкие задачи ориентирована разработка параллельных методов многомерной многоэкстремальной оптимизации в данной работе.

## 1. Задачи многомерной многоэкстремальной оптимизации и адаптивная многошаговая схема редукции размерности

Задача многомерной многоэкстремальной оптимизации может быть определена как проблема поиска наименьшего значения действительной функции  $\varphi(y)$

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

где  $D$  есть область поиска, представляющая собой некоторый гиперпараллелепипед  $N$ -мерного евклидова пространства.

Многие регулярные поисковые методы решения многомерных оптимизационных задач сводят многомерную задачу (явно или неявно) к системе одномерных подзадач. Один из наиболее общих методов редукции размерности состоит в применении *многошаговой схемы редукции размерности*, согласно которой решение многомерной задачи оптимизации может быть получено посредством решения последовательности «вложенных» одномерных задач (см., например, [1-3]):

$$\min_{y \in D} \varphi(y) = \min_{y_1 \in [a_1, b_1]} \dots \min_{y_N \in [a_N, b_N]} \varphi(y_1, \dots, y_N) \quad (2)$$

Использование этой схемы позволяет получить общий способ для применения одномерных алгоритмов оптимизации к решению многомерных оптимизационных задач. Согласно (2) решение многомерной задачи (1) сводится к решению одномерной задачи:

$$\varphi^* = \min_{y \in D} \varphi(y) = \min_{y_1 \in [a_1, b_1]} f_1(y_1), \quad (3)$$

где

$$f_i(y_i) = \varphi_i(y_1, \dots, y_i) = \min_{y_{i+1} \in [a_{i+1}, b_{i+1}]} \varphi_{i+1}(y_1, \dots, y_i, y_{i+1}) \quad 1 \leq i < N, \quad (4)$$

$$\varphi_N(y_1, \dots, y_N) = \varphi(y_1, \dots, y_N). \quad (5)$$

Правила (3) – (5) определяют множество задач:

$$F_l = \{ f_i(y_i), 1 \leq i \leq l \}. \quad (6)$$

порождаемых в соответствии с многошаговой схемой редукции. Количество задач в множестве  $F_l$  в процессе поиска может изменяться: увеличиваться при переходе к следующей переменной и уменьшаться при завершении решения какой-либо из задач (можно отметить, что при этом, количество задач не превышает размерность решаемой задачи  $N$ ). При этом активной – решаемой – в множестве  $F_l$  является только одна задача – это задача с максимальным номером варьируемой переменной.

В данной статье рассматривается обобщенная (адаптивная) многошаговая схема редукции размерности, в которой предлагается осуществлять одновременное решение всех задач множества  $F_l$ .

В работе предлагается обобщенная (адаптивная) многошаговой схемы редукции размерности. Такое решение может выполняться последовательно (при наличии единственного процессора), тогда для выполнения очередной итерации глобального поиска необходимо выбирать для решения одну из задач множества  $F_l$ . в соответствии с тем или иным правилом выбора задач. Но решение задач может выполняться и параллельно, если используемый вычислитель является многопроцессорным или многоядерным. Именно параллельный – более общий вариант – будет рассматриваться далее в работе.

Важно отметить, что выполнение итерации глобального поиска для любой задачи множества  $F_l$  с номером переменной меньшим, чем  $N$  ( $N$  есть размерность решаемой задачи), будет приводить к порождению новой одномерной задачи вида (4), и, тем самым, количество задач в семействе  $F_l$  может оказаться значительным (десятки и сотни тысяч для сложных задач оптимизации).

## 2. Централизованная схема параллельного глобального поиска

Характеристики интервалов, вычисляемые алгоритмами глобального поиска, рассматриваются как некоторая мера важности интервалов на предмет содержания в них искомого решения оптимизационной задачи. Следуя данному пониманию, в параллельных методах после выбора точки проведения испытания для первого процессора в точном соответствии с последовательным алгоритмом, для второго процессора точку испытания выбирается из следующего по важности интервала (т.е. из интервала со следующей по порядку максимальной характеристикой) и т.д.

Рассмотренная выше идея послужила основой для централизованной схемы параллельного глобального поиска. Пусть  $p > 1$  есть число процессоров, используемых для решения задачи оптимизации. Каждый из процессоров, получив точку  $y \in D$ , осуществляет независимо и параллельно с другими процессорами вычисление значения функции  $j(y)$  из (1). В процессе вычисления значения функции испытание считается незавершенным – тем самым, при выполнении алгоритма будут существовать точки завершенных и незавершенных испытаний. Центральный процессор осуществляет выбор очередной итерации, который заключается в поиске интервала с максимальной характеристикой из списка допустимых интервалов и вычисление по некоторому правилу точки нового испытания.

Результаты выполненных численных экспериментов для оценки эффективности централизованной схемы подтвердили следующий факт - при превышении определенного числа процессоров-исполнителей центральный процессор не обеспечивает нагрузку всех имеющихся процессоров-исполнителей. Таким образом, дальнейшее повышение числа процессоров выше некоторого порога не приводит к желаемому росту ускорения.

### 3. Модификация централизованной схемы параллельного глобального поиска

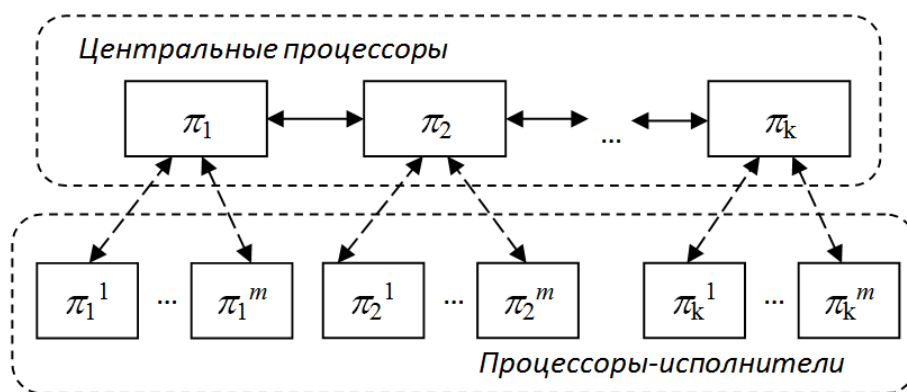


Рис. 1

Идея модификации централизованной схемы основана на устранении «узкого» места оригинальной схемы. Это достигается за счет увеличения числа процессоров, выполняющих роль центрального. В роли менеджера, выдающего задания процессорам-исполнителям, в модифицированной схеме выступает не один, а  $k$  процессоров. При условии сохранения единства поисковой информации между этими  $k$  процессорами, выбор очередной итерации каждым из них может осуществляться следующим образом:

- обозначим множество центральных процессоров  $P = \{p_1, p_2, \dots, p_k\}$ ;
- выберем  $k$  интервалов с максимальными характеристиками  $L = \{l_1, l_2, \dots, l_k\}$ ;
- тогда точка очередного испытания для процесса  $p_i$  выбирается в интервале  $l_i$ .
- Для обеспечения единства поисковой информации для всех центральных процессоров:
- перед каждой новой итерацией производится синхронизация очередей характеристик каждого из процессов;
- процессор-исполнитель высылает результаты своих вычислений каждому из центральных процессоров.

Работу модифицированной централизованной схемы можно схематично представить в следующем виде (см. рис. 1).

Таким образом, данная модификация централизованной схемы позволяет достаточно хорошо масштабировать алгоритм параллельного глобального поиска на большое число процессоров.

### 4. Результаты вычислительных экспериментов

На рисунке 2 приведены результаты выполненных численных экспериментов для оценки

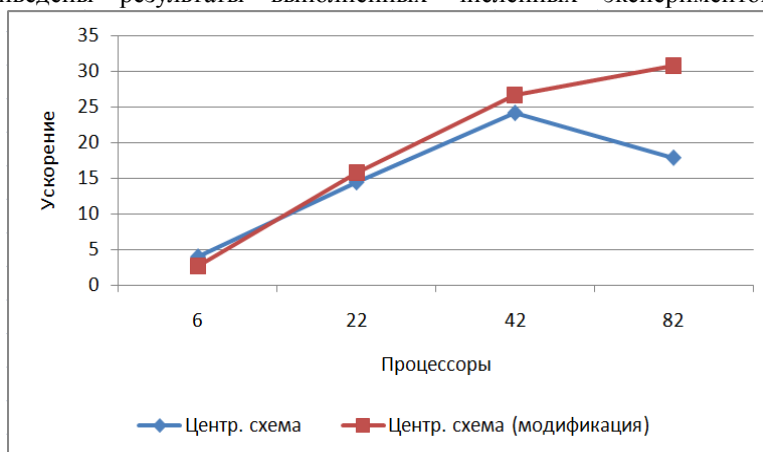


Рис. 2

эффективности предложенной модификации централизованной схемы параллельного глобального поиска для адаптивной многошаговой схемы редукции размерности на классе тестовых задач многоэкстремальной оптимизации Растригина.

#### ЛИТЕРАТУРА:

1. А.В. Гергель Д.В. Гнатюк Адаптивные параллельные алгоритмы для многомерной многоэкстремальной оптимизации // Материалы конференции «Высокопроизводительные параллельные вычисления на кластерных системах». 2009. С. 92-96
2. Р.Г. Стронгин. Численные методы в многоэкстремальных задачах. М.: Наука, 1978
3. С.Ю. Городецкий, В.А. Гришагин Нелинейное программирование и многоэкстремальная оптимизация. – Н.Новгород: Изд-во ННГУ, 2007
4. В.П. Гергель Теория и практика параллельных вычислений. – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2007.