Распределение вычислительной нагрузки при параллельном решении серии задач оптимизации*

К.А. Баркалов, К.А. Николаев

Нижегородский государственный университет им. Н.И. Лобачевского

В ННГУ им. Н.И. Лобачевского развиваются эффективные алгоритмы решения задач глобальной оптимизации. В данной работе рассматривается вопрос о распределении вычислительной нагрузки между процессорами при решении серии оптимизационных задач. Предложен способ адаптивного распределения нагрузки, основанный на свойствах алгоритма оптимизации. Данный способ обеспечивает как равномерную загрузку процессоров, так и равномерную сходимость к решению всех задач серии.

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

1. Введение

Рассмотрим одномерную задачу условной глобальной оптимизации

$$\varphi(x^*) = \min\{\varphi(x) : x \in [a,b],\ \ g_j(x) \le 0,\ 1 \le j \le m\} \tag{1}$$

в предположении, что целевая функция $\varphi(x)$ и левые части ограничений $g_j(x)$, $1 \le j \le m$, являются определенными на отрезке $[a,b]$ липшицевыми функциями с соответствующими константами L_j, $1 \le j \le m$.

В ННГУ им. Н.И. Лобачевского был разработан эффективный подход к минимизации многоэкстремальных функций при невыпуклых ограничениях, развитый в [1–6] и получивший название *индексного метода* глобальной оптимизации.

Подход основан на раздельном учете каждого ограничения задачи и не связан с использованием штрафных функций. В соответствии с правилами индексного метода каждая итерация, называемая *испытанием* в соответствующей точке области поиска, включает последовательную проверку выполнимости ограничений задачи в этой точке. При этом обнаружение первого нарушенного ограничения прерывает испытание и инициирует переход к точке следующей итерации. Сказанное актуально для задач, в которых вычисление функций задачи требует заметных вычислительных ресурсов.

Данное исследование посвящено проблеме параллельного решения серии задач вида (1). Подобного рода проблема возникает, например, при использовании схемы вложенной оптимизации для решения многомерных задач [7]. В этом случае на каждом уровне вложенности возникает множество информационно-независимых одномерных подзадач, которые можно решить параллельно. Целью исследования является разработка новой схемы распределения задач по процессорам, которая будет обеспечивать как равномерную сходимость к решениям всех задач серии, так и равномерную загрузку используемых процессоров.

2. Параллельный алгоритм решения задач глобальной оптимизации

Введем классификацию точек x из области поиска $[a,b]$ с помощью *индекса* $\nu = \nu(x)$. Указанный индекс ν определяется условиями

$$g_j\big(y(x)\big) \le 0,\ 1 \le j \le \nu - 1,$$
$$g_\nu(x) < 0,$$

* Исследование выполнено за счет гранта Российского научного фонда (проект №16-11-10150)

где последнее неравенство несущественно, если $\nu = m + 1$. Введенный таким образом индекс точки удовлетворяет неравенствам $1 \le \nu = \nu(x) \le m + 1$.

Данная классификация порождает функцию

$$f(x) = g_\nu(x),\ \nu = \nu(x),$$

определенную и вычислимую на $[a, b]$. Ее значение в точке x есть либо значение левой части ограничения, нарушенного в этой точке (в случае $\nu \le m$), либо значение минимизируемой функции (в случае $\nu = m + 1$). Поэтому определение значения $f(x)$ сводится к последовательному вычислению величин $g_j(x),\ 1 \le j \le \nu = \nu(x)$, т.е. последующее значение $g_{j+1}(x)$ вычисляется лишь в том случае, когда $g_j(x) \le 0$. Процесс вычислений завершается либо в результате установления неравенства $g_j(x) > 0$, либо в результате достижения значения $\nu(x) = m + 1$.

Описанная процедура, называемая *испытанием* в точке y, автоматически приводит к определению индекса ν этой точки. Пара значений

$$z = g_\nu(x),\ \nu = \nu(x), \tag{2}$$

порожденная испытанием в точке $x \in [a, b]$, называется *результатом испытания*.

Последовательный индексный алгоритм для решения одномерных задач условной оптимизации вида (1) подробно описан в [6]. Для его распараллеливания можно применить подход, предложенный в [8] для решения безусловных задач глобальной оптимизации. Опишем кратко вычислительную схему параллельного алгоритма.

Пусть в нашем распоряжении имеется $p \ge 1$ вычислительных элементов (например, ядер на центральном процессоре), с использованием которых можно проводить p испытаний одновременно. На первой итерации метода параллельно выполняется p испытаний в произвольных различных точках $x^i \in (a, b),\ 1 \le i \le p$.

Предположим, что выполнено $n \ge 1$ итераций метода, в процессе которых были проведены испытания в $k = k(n)$ точках $x^i, 1 \le i \le k$. Тогда точки $x^{k+1}, \dots, x^{k+p}$ поисковых испытаний следующей $(n + 1)$-ой итерации определяются в соответствии с правилами.

Правило 1. Перенумеровать точки $x^1, \dots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т.е.

$$a = x_0 < x_1 < \dots < x_i < \dots < x_k < x_{k+1} = b, \tag{3}$$

и сопоставить им значения $z_i = g_\nu(x_i),\ \nu = \nu(x_i),\ 1 \le i \le k,$ из (2), вычисленные в этих точках; точки $x_0 = a$ и $x_{k+1} = b$ введены дополнительно, значения $z_0,\ z_{k+1}$ являются неопределенными.

Правило 2. Провести классификацию номеров $i,\ 1 \le i \le k,$ точек из ряда (3) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$I_\nu = \{i\colon 1 \le i \le k,\ \nu = \nu(x_i)\},\ 1 \le \nu \le m + 1,$$

содержащих номера всех точек $x_i,\ 1 \le i \le k,$ имеющих индексы, равные одному и тому же значению ν. Граничные точки $x_0 = a$ и $x_{k+1} = b$ интерпретируются как имеющие нулевой индекс, и им сопоставляется дополнительное множество $I_0 = \{0, k + 1\}$. Определить максимальное текущее значение индекса

$$M = max\{\nu = \nu(x_i),\ 1 \le i \le k\}.$$

Правило 3. Для всех значений $\nu, 1 \le \nu \le m + 1$, вычислить величины

$$\mu_\nu = \max\left\{\frac{|z_i - z_j|}{x_i - x_j} : i, j \in I_\nu,\ j < i\right\} \tag{4}$$

Если множество I_ν содержит менее двух элементов или μ_ν из (4) оказывается равным нулю, то принять $\mu_\nu = 1$.

Правило 4. Для всех непустых множеств $I_\nu, 1 \le \nu \le m + 1$, определить величины

$$z_\nu^* = \begin{cases} 0, & \nu < M, \\ \min\{q_\nu(x_i): i \in I_\nu\}, & \nu = M, \end{cases}$$

где M – максимальное текущее значение индекса.

Правило 5. Для каждого интервала $(x_{i-1}, x_i),\ 1 \le i \le k,$ вычислить *характеристику интервала* $R(i)$, где

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r_\nu^2 \mu_\nu^2 \Delta_i} - 2\frac{z_i + z_{i-1} - 2z_\nu^*}{r_\nu \ \mu_\nu}, \nu = \nu(x_{i-1}) = \nu(x_i),$$

$$R(i) = 2\Delta_i - 4\frac{z_i - z_\nu^*}{r_\nu \ \mu_\nu}, \nu(x_{i-1}) < \nu(x_i) = \nu, \tag{5}$$

$$R(i) = 2\Delta_i - 4\frac{z_{i-1} - z_\nu^*}{r_\nu \ \mu_\nu}, \nu = \nu(x_{i-1}) > \nu(x_i).$$

Здесь $\Delta_i = x_i - x_{i-1}$, а величины $r_\nu > 1,\ 1 \le \nu \le m+1$, являются параметрами алгоритма.

Правило 6. Характеристики $R(i), 1 \le i \le k+1$, упорядочить в порядке убывания

$$R(t_1) \ge R(t_2) \ge \dots \ge R(t_k) \ge R(t_{k+1}), \tag{6}$$

и выбрать из них p наибольших характеристик с номерами интервалов.

Правило 7. Провести новые испытания в точках $x^{k+j}, 1 \le j \le p$, вычисленных по формулам

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2},\ \nu\left(x_{t_j-1}\right) \ne \nu\left(x_{t_j}\right)$$

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - \frac{1}{2r_\nu}\frac{|z_{t_j} - z_{t_j-1}|}{\mu_\nu},\ \nu = \nu\left(x_{t_j-1}\right) = \nu\left(x_{t_j}\right) \tag{7}$$

Алгоритм прекращает работу, если выполняется условие $\Delta_{t_j} \le \epsilon$ хотя бы для одного номера $t_j, 1 \le j \le p$; здесь $\epsilon > 0$ соответствует точности поиска минимума в решаемой задаче.

Условия сходимости данного параллельного алгоритма следуют из теоремы о сходимости индексного алгоритма и теоремы о сходимости параллельного характеристического алгоритма [9]. Способ использования данного алгоритма на кластерных системах описан в [10].

3. Распределение задач по процессорам при решении серии задач

Пусть в вычислительной системе имеется p процессоров. Рассмотрим различные варианты организации параллельных вычислений для решения серии из Q задач, где $Q > p$ (случай $Q \le p$ является тривиальным).

Традиционно в параллельных вычислениях используются два способа распределения задач по процессорам – статический и динамический.

Статический способ.

1. Каждая задача решается независимо на отдельном процессоре.
2. Все множество задач делится на p фрагментов длины Q/p.
3. Полученные порции задач распределяются между процессорами для решения.

Очевидным недостатком данного способа распределения задач является дисбаланс нагрузки процессоров. Разные задачи требуют для решения разного числа поисковых испытаний, поэтому решение i-й порции из Q/p задач на i-м процессоре может потребовать существенно меньше испытаний, чем решение j-й порции на j-м процессоре; i-й процессор будет простаивать.

Динамический способ.

1. Каждая задача решается независимо на отдельном процессоре.
2. В начальный момент запускается решение p задач с номерами $1, 2, \dots, p$.
3. Как только какая-либо i-я задача решилась на j-м процессоре, на данном процессоре запускается решение следующей $(p+1)$-й задачи.

При динамическом способе распределения задач будет обеспечена почти полная загрузка процессоров; дисбаланс возможен только при решении последних p задач из серии. Однако оба этих способа (и статический, и динамический) позволяют получать оценки экстремумов одновременно не более p задач, оценка решения в остальных задачах серии не производится.

Использование параллельного индексного алгоритма позволяет организовать одновременное решение серии задач новым способом, который будет обеспечивать равномерную сходимость ко всем глобальным экстремумам во всех задачах.

Все задачи объединяются в единое множество, каждой задаче присваивается порядковый номер. Для каждой задачи вычисляются характеристики (5), и строится упорядоченный набор характеристик всех интервалов всех задач (6). После этого выбираются p наибольших характеристик с номерами t_j, $1 \le j \le p$, и в интервалах, соответствующих этим характеристикам, выполняются поисковые испытания в точках $x^{k+j} \in \left(x_{t_j-1}, x_{t_j}\right)$, $1 \le j \le p$.

Очевидно, что интервалы с наибольшими характеристиками (6), и, следовательно, точки (7) могут принадлежать разным задачам. Этим обеспечивается равномерная сходимость последовательности точек испытаний к глобальным минимумам во всех задачах.

4. Результаты численных экспериментов

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. Применение алгоритмов многоэкстремальной оптимизации на больших выборках таких реализаций позволяет оценивать эффективность каждого конкретного алгоритма. К числу таких генераторов для одномерных задач относятся функции вида

$$f(x) = \sum_{j=1}^{14} \left(A_j \sin(2j\pi x) + B_j \cos(2j\pi x)\right),\ x \in [0,1] \tag{8}$$

предложенные Хиллом [11]. Функции Хилла были успешно использованы в конструкции генератора одномерных задач с ограничениями с управляемой мерой допустимой области [6].

Для проведения численных экспериментов была сгенерирована серия из 100 тестовых задач, состоящих из двух ограничений и целевой функции вида (8). Для решения данной серии задач было задействовано 10 ядер на одном из узлов вычислительного кластера ННГУ; использовались две стратегии распределения задач:

1. динамическое распределение задач;
2. одновременное (совместное) решение всех задач.

В процессе решения серии задач были построены функции $D_{av}(K)$ и $D_{max}\ (K)$, характеризующие среднее и максимальное отклонение текущего приближения от точного решения по всем задачам в зависимости от общего числа итераций. Данные функции представлены на рис. 1 и рис. 2. Зеленая кривая характеризует первую стратегию распределения задач по процессорам, а синяя – вторую. Очевидно, что обе стратегии распределения задач обеспечивают решение всех задач выборки за примерно одинаковое число итераций. Нагрузка на ядра в данном случае также является одинаковой.

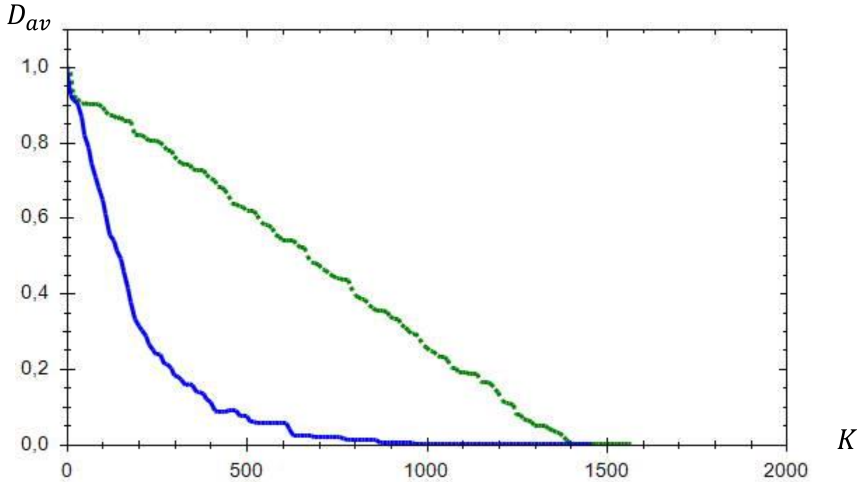

Рис. 1. Среднее отклонение текущего приближения от точного решения

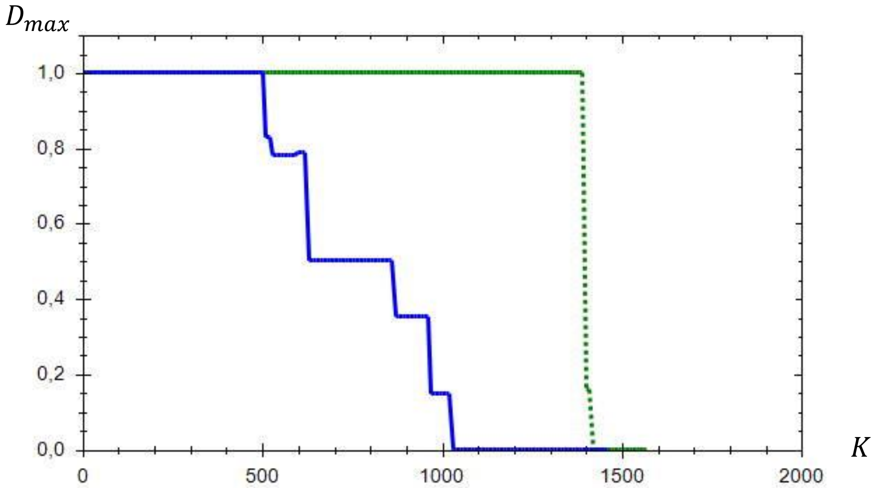

Рис. 2. Максимальное отклонение текущего приближения от точного решения

Одновременно с этим стратегия №2 – одновременное решение всех задач – обеспечивает равномерную сходимость к решению задач серии. Расположение кривых на рис. 1 показывает, что при совместном решении задач уже за 500 испытаний можно получить хорошее (в среднем) приближение к решению во всех задачах. А за 1000 испытаний малое значение примет и максимальное отклонение. Динамическое распределение задач значительно проигрывает в обоих случаях.

5. Заключение

В данной статье представлены предварительные результаты, полученные при исследовании проблемы параллельного решения серии задач оптимизации. Была предложена новая схема распределения задач по процессорам, которая одновременно обеспечивает и равномерную сходимость к решениям всех задач серии, и равномерную загрузку используемых процессоров.

Дальнейшие исследования по данной теме будут направлены:

- на углубление теоретических основ предложенного подхода (строгое доказательство равномерной сходимости алгоритма);
- на накопление более представительной экспериментальной базы. В частности, будут проведены вычислительные эксперименты, связанные с решением серии задач оптимизации разной трудоемкости (с различным числом ограничений; с различным временем вычисления значений функций в разных задачах, и т.п.). Решение серии задач разной трудоемкости будет более наглядно демонстрировать равномерное распределение вычислительной нагрузки при совместном решении всех задач.

Литература

1. Стронгин Р.Г., Маркин Д.Л. Минимизация многоэкстремальных функций при невыпуклых ограничениях // Кибернетика. 1986. №4. С. 63–69.

2. Маркин Д.Л., Стронгин Р.Г. Метод решения многоэкстремальных задач с невыпуклыми ограничениями, использующий априорную информацию об оценках оптимума// Журнал вычислительной математики и математической физики. 1987. Т. 27, № 1. С. 52–62.

3. Стронгин Р.Г. Параллельная многоэкстремальная оптимизация с использованием множества разверток // Журнал вычислительной математики и математической физики. 1991. Т. 31. № 8. С. 1173–1185.

4. Strongin R.G. Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves// Journal of Global Optimization. 1992. Vol. 2. P. 357–378.

5. Sergeyev Ya.D., Markin D.L. An algorithm for solving global optimization problems with nonlinear constraints // Journal of Global Optimization. 1995. Vol. 4, No. 4. P. 407–419.
6. Стронгин Р.Г., Баркалов К.А. Метод глобальной оптимизации с адаптивным порядком проверки ограничений // Журнал вычислительной математики и математической физики. 2002. Т. 42, № 9. С. 1338–1350.
7. Gergel V., Grishagin V., Israfilov R. Local tuning in nested scheme of global optimization // Procedia Computer Science. 2015. Vol. 51. P. 865–874.
8. Grishagin V.A., Sergeyev Ya.D., Strongin R.G. Parallel characteristical global optimization algorithms // Journal of Global Optimization. 1997. Vol. 10. P. 185–206.
9. Strongin R.G., Sergeyev Ya.D. Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000. 704 p.
10. 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.
11. Hill J.D. A search technique for multimodal surfaces // IEEE Transactions on Systems Science and Cybernetics. 1969. Vol. 5, No. 1. P. 2–8.

Load balancing during parallel solving a set of optimization problems*

K.A. Barkalov, K.A. Nikolaev

Lobachevsky State University of Nizhny Novgorod

Efficient parallel algorithms for solving global optimization problems have been developed in Lobachevsky State University of Nizhny Novgorod. In this study a new approach for parallel solving a series of global optimization problems on computing cluster is investigated. An adaptive method of load balancing based on algorithm properties is proposed. This method provides uniform load as well as uniform convergence to solutions of all problems in the series.

Keywords: parallel algorithms, global optimization, muliextremal functions.

References

1. Strongin R.G., Markin, D.L. Minimization of multiextremal functions with nonconvex constraints // Cybernetics. 1986. Vol. 22, No. 4, P. 486–493.
2. 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 // Journal of Computational Mathematics and Mathematical Physics. 1987. Vol. 27, No. 1. P. 33–39.
3. Strongin R.G. Parallel multi-extremal optimization using a set of evolvents // Computational Mathematics and Mathematical Physics. 1991. Vol. 31, No. 8. P. 37–46.
4. Strongin R.G. Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves// Journal of Global Optimization. 1992. Vol. 2. P. 357–378.
5. Sergeyev Ya.D., Markin D.L. An algorithm for solving global optimization problems with nonlinear constraints // Journal of Global Optimization. 1995. Vol. 4, No. 4. P. 407–419.
6. 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. 1289–1300.
7. Gergel V., Grishagin V., Israfilov R. Local tuning in nested scheme of global optimization // Procedia Computer Science. 2015. Vol. 51. P. 865–874.
8. Grishagin V.A., Sergeyev Ya.D., Strongin R.G. Parallel characteristical global optimization algorithms // Journal of Global Optimization. 1997. Vol. 10. P. 185–206.
9. Strongin R.G., Sergeyev Ya.D. Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht, 2000. 704 p.
10. 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.
11. Hill J.D. A search technique for multimodal surfaces // IEEE Transactions on Systems Science and Cybernetics. 1969. Vol. 5, No. 1. P. 2–8.

* This research was supported by Russian Science Foundation (project No 16-11-10150)