Параллельный алгоритм глобальной оптимизации в классах функций, определяемых кусочно-степенными мажорантами

Е.Е. Пестова, А.Г. Коротченко

Работа направлена на исследование параллельного диагонального алгоритма глобальной оптимизации многомерных функций из классов, определяемых кусочно-степенными мажорантами. В качестве характеристики гиперинтервала используется погрешность алгоритма для аналогичного класса функций, заданных на отрезке (вместо отрезка берется диагональ гиперинтервала). При построении параллельного алгоритма учтены свойства данной погрешности. Проведены вычислительные эксперименты, результаты которых демонстрируют, по меньшей мере, линейное ускорение и высокую эффективность параллельной реализации.

1. Введение

Задачи поиска экстремума функций, заданных на отрезке, входят как составные процедуры во многие методы решения задач оптимизации, а также возникают при решении ряда минимаксных задач. При этом указанные задачи являются внутренними по отношению к алгоритму, решающему внешнюю оптимизационную задачу, и их приходится решать многократно (на каждой итерации используемого алгоритма). Поэтому алгоритмы решения задач поиска экстремума функций, заданных на отрезке, должны быть эффективными: решать задачу быстро и точно.

Для оценки же точности решения указанных задач необходимо иметь априорную информацию об оптимизируемой функции или, другими словами, фиксировать класс, которому она принадлежит. При этом рассматриваемые классы функций должны, с одной стороны, отражать существенные свойства классов задач, встречающихся в приложениях, а с другой – допускать построение достаточно простых в реализации и эффективных алгоритмов.

Построению такого рода алгоритмов посвящены, например, работы [1–3]. В них рассматриваются функциональные классы, определяемые кусочно-степенными мажорантами, предлагаются алгоритмы поиска наибольшего значения функций из данных классов, устанавливаются свойства таких алгоритмов.

В данной статье предлагается использовать построенные алгоритмы для решения многомерных многоэкстремальных задач. Основной трудностью решения таких задач является рост вычислительных затрат при увеличении размерности задачи. Уменьшение числа испытаний функций при тех же требованиях к точности решения возможно за счет более полного использования априорных предположений об оптимизируемой функции.

Одним из адаптивных подходов к решению многомерной задачи является ее редукция к одномерной задаче (или к нескольким таким задачам) с последующим применением эффективных одномерных алгоритмов. В данной работе для поиска глобального экстремума применяется диагональный подход, предложенный Я. Пинтером [4]. Диагональные компонентные методы предназначены для решения задач без ограничений, в которых допустимым множеством является гиперпараллелепипед, последовательно разбиваемый в процессе решения задачи на компоненты, которые также являются гиперпараллелепипедами. Для оценки характеристики компоненты-параллелепипеда используется его главная диагональ, т.е. возможно применение алгоритмов [1–3] для оптимизации функций многих переменных из соответствующих классов.

2. Функциональные классы

При решении многокритериальных задач, а также задач оптимизации с использованием метода штрафов возникают классы функций, обладающие следующей структурой. Данные классы содержат так называемые базовые функции и замкнуты относительно ряда естественных операций. Указанные классы порождают соответствующие классы функций одной переменной.

В данной работе будем рассматривать класс функций Φ, в котором базовыми функциями являются непрерывные вогнутые, выпуклые, удовлетворяющие условию Липшица, функции, заданные на гиперпараллелепипеде, а в качестве операций, относительно которых класс замкнут, используются операции взятия минимума, максимума, суммирования с неотрицательными коэффициентами по конечному набору функций.

Порождаемый данным классом класс функций одной переменной формально может быть задан следующим образом.

Обозначим через $F(a, b, l)$ класс непрерывных функций, заданных на отрезке $[a, b]$ и удовлетворяющих условиям:

$$\frac{f(x_2)-l}{x_2-a} \le \frac{f(x_1)-l}{x_1-a}, \quad \frac{f(x_2)-l}{b-x_2} \ge \frac{f(x_1)-l}{b-x_1},$$

где $x_2 > x_1$, $x_1, x_2 \in (a,b)$, l – вещественная константа.

Если функция $f(x) \in F(a,b,l)$, то $f(x) \ge l, x \in [a,b]$.

Перечислим некоторые свойства функций, принадлежащих классу $F(a, b, l)$.

Свойство 1. Если $f(x) \in F(a,b,l)$, то $f(x) \in F(a,b,l_1)$, где $l_1 \le l$.

Свойство 2. Классу $F(a, b, l)$ принадлежат все ограниченные снизу константой l вогнутые функции, определенные на $[a, b]$.

Свойство 3. Если $f_1(x) \in F(a,b,l_1)$, $f_2(x) \in F(a,b,l_2)$, то $\beta_1 \cdot f_1(x) + \beta_2 \cdot f_2(x) \in F(a,b,l)$, $l = \beta_1 \cdot l_1 + \beta_2 \cdot l_2$, $\beta_1 \ge 0, \beta_2 \ge 0$.

Свойство 4. Если $f_1(x),..., f_s(x) \in F(a,b,0)$, то $[f_1(x) \cdot ... \cdot f_s(x)]^{1/s} \in F(a,b,0)$.

Свойство 5. Если $f_i(x) \in F(a,b,l(i))$, $i=1,\ldots,s$, то $\min\limits_{1 \le i \le s} f_i(x) \in F(a,b,l)$, $\max\limits_{1 \le i \le s} f_i(x) \in F(a,b,l)$, $l = \min\limits_{1 \le i \le s} l(i)$.

Свойство 6. Выпуклая, дифференцируемая в точках a и b функция $f(x)$ принадлежит классу $F(a, b, l)$ при $l = \min[f(b) - f'(b) \cdot (b-a), f(a) + f'(b) \cdot (b-a)]$.

Свойство 7. Функция $f(x)$, удовлетворяющая на отрезке $[a, b]$ условию Липшица с константой L, принадлежит классу $F(a, b, l)$, где $l = \min[f(b) - L \cdot (b-a), \ f(a) - L \cdot (b-a)]$.

3. Описание алгоритма

Базовой процедурой алгоритма поиска наибольшего значения функций из класса Φ является алгоритм α поиска наибольшего значения функции $f(x) \in F(a,b,l)$. Опишем данный алгоритм.

Первое вычисление функции осуществляем в середине отрезка $[a, b]$. Пусть $f(x)$ вычислена в m ($m>0$) точках по алгоритму α, которые упорядочиваются таким образом, что $a < x_1 < x_2 < ... < x_m < b$ и $y_j = f(x_j)$, $j = 1,...,m$.

Параметр l может быть задан априори или уточняться на каждой итерации алгоритма с появлением новых точек вычисления функции (изначально l берется как минимум из значений функции на концах отрезка).

Подкласс класса $F(a, b, l)$, всех таких функций, которые в точках x_j принимают значения y_j, $j = 1,...,m$, обозначим через $F_m(a,b,l)$.

Пусть

$$f_j^1(x) = \frac{y_j - l}{x_j - a}(x-a) + l,$$

$$f_j^2(x) = \frac{y_j - l}{b - x_j}(b-x) + l,$$

$$f_j(x) = \max[f_j^1(x), f_j^2(x)], \ x \in D = [a,b], \ j = 1,...,m.$$

Введем в рассмотрение функцию

$$\varphi_m(x) = \min_{j=1,\dots,m} f_j(x),\ x \in D,$$

которая является точной верхней мажорантой всех функций из класса $F_m(a,b,l)$.

Положим

$x_m^* = \arg\max_{x\in D} \varphi_m(x),\ h(m) = \max_{j=1,\dots,m} y_j = y_\mu,\ 1 \le \mu \le m$, и пусть $h \ge h(m)$.

В силу задания функции $\varphi_m(x)$ значение x_m^* существует. Если $h \ge \varphi_m(x_m^*)$, то алгоритм α заканчивает свою работу. При этом если $h = \varphi_m(x_m^*)$, имеем $x_m^* = \arg\max_{x\in D} f(x)$ для $f(x) \in F_m(a,b,l)$.

Пусть $h(m) < \varphi_m(x_m^*)$. Тогда в качестве аргумента, реализующего приближенное решение задачи отыскания наибольшего значения функции $f(x) \in F_m(a,b,l)$ можно принять значение x_μ, а погрешность в определении наибольшего значения $f(x)$ равна

$$Q(m) = \varphi_m(x_m^*) - h(m). \tag{1}$$

Величина $Q(m)$ является максимальной возможной разностью между найденным после m вычислений функции $f(x)$ по алгоритму α приближенным значением максимума $f(x)$ и его неизвестным точным значением и выражает собой критерий, оценивающий качество работы алгоритма α.

Пусть

$$x_0 = t_0 = a,\ x_{m+1} = t_m = b,$$

$$y_0 = l + \left(\frac{b-a}{b-x_1}\right)(y_1 - l),\ y_{m+1} = l + \left(\frac{b-a}{x_m - a}\right)(y_m - l).$$

Величины y_0 и y_{m+1} носят вспомогательный характер и введены для простоты дальнейшего изложения.

При $m \ge 2$ введем в рассмотрение величины:

$$t_j = \frac{A_j a + B_j b}{A_j + B_j},\ A_j = \frac{y_j - l}{x_j - a},\ B_j = \frac{y_{j+1} - l}{b - x_{j+1}},$$

$$j = 1,\dots,m-1.$$

Точки $t_0 < t_1 < \dots < t_m$ доставляют локальные максимумы функции $\varphi_m(x)$.

Через $J(m)$ обозначим множество индексов j таких, что при $j \in J(m)$ отрезок $[x_j, x_{j+1}]$ содержит точку t_j, доставляющую функции $\varphi_m(x)$ значение, большее $h(m)$. Множество всех таких отрезков обозначим через $S(m)$. Здесь $J(m) \subseteq \{0, 1, \dots, m\}$ и пусть $q(m) = |J(m)|$, а $|J(m)|$ означает число элементов множества $J(m)$.

Для каждого $i=1,\dots,q(m)$ пусть $s(i)$ такой индекс множества $J(m)$, что число элементов множества $J(m)$, не превосходящих $s(i)$, равно i.

Введем в рассмотрение функции:

$$u(x,y,\omega) = a + \frac{\omega - l}{y - l}(x - a),\ v(x,y,\omega) = b - \frac{\omega - l}{y - l}(b - x),$$

$$x \in D,\ y > l,\ \omega > l.$$

Для $i=1,\dots,q(m)$ определим на отрезке $[x_{s(i)}, x_{s(i)+1}]$ величины u_i и v_i $(u_i < v_i)$ следующим образом:

$$u_i = u(x_{s(i)}, y_{s(i)}, h(m)),\ v_i = (x_{s(i)+1}, y_{s(i)+1}, h(m)).$$

В силу задания величин u_i, v_i имеем: $\varphi_m(x) \ge h(m)$ при $x \in [u_i, v_i]$.

Через $\mathcal{F}(m)$ обозначим подмножество множества $J(m)$ такое, что для каждого $i \in \mathcal{F}(m)$ отрезок $[u_i, v_i]$ содержит точку t_i, доставляющую функции $\varphi_m(x)$ значение глобального максимума. Тогда, с учетом (1),

$$Q(m) = Q_i(m),$$
$$Q_i(m) = \frac{(h(m)-l)(b-a)}{(b-a+u_i-v_i)} - (h(m)-l), \; i \in \mathcal{J}(m).$$

Отметим, что пока не будут вычислены значения функции $f(x)$ по крайней мере в одной точке на каждом интервале (u_i, v_i), $i \in \mathcal{J}(m)$, нельзя гарантировать уменьшение значения критерия $Q(m)$.

Будем считать, что на каждом шаге алгоритма α задается подмножество $\tilde{S}(m)$ множества $S(m)$, и пусть $\tilde{J}(m)$ – подмножество множества индексов $J(m)$, соответствующее подмножеству отрезков $\tilde{S}(m)$, а $r(m) = |\tilde{S}(m)|$.

На очередной итерации алгоритма α будем вычислять функцию $f(x)$ в $r(m)$ точках множества $\tilde{S}(m)$ следующим образом:

$$x(i) = \frac{u_i + v_i}{2}, \; i \in \tilde{J}(m).$$

Выбор множества $\tilde{S}(m)$ на каждом шаге алгоритма α определяет одну из его реализаций.

Пусть $\mathrm{r}(m)=1$ для каждого шага алгоритма α и

$$j_0 = \min_{j \in \mathcal{J}(m)} j, \; u = u_{j_0}, \; v = v_{j_0}.$$

Тогда $\tilde{J}(m) = \{j_0\}$, $\tilde{S}(m) = \{[u, v]\}$.

Положим

$$Q = Q(m), \; h = h(m).$$

Тогда критерий Q примет вид:

$$Q = \frac{(h-l)(b-a)}{(b-a+u-v)} - (h-l),$$

а функцию $f(x)$ на очередном шаге следует вычислять в точке

$$\bar{x} = \frac{u+v}{2}.$$

Отметим, что критерий Q зависит от длины отрезка $[a,b]$, а также от диапазона принимаемых значений оптимизируемой функции на данном отрезке. Данный факт используется при разработке параллельной реализации алгоритма. Если в ходе работы алгоритма функция вычислена в $(n+1)$ точках на исходном отрезке и полученные при этом отрезки оказываются одинаковыми или близкими по длине, что соответствует наименее благоприятному случаю, то время работы алгоритма α на каждом из полученных отрезков будет в n раз меньше времени работы на всем отрезке. Причем описанный случай имеет место для функций с узким диапазоном значений (чем ближе функция к константе, тем медленнее работает алгоритм).

Для функций из класса Φ данный алгоритм применяется в диагональном методе при вычислении характеристики гиперпараллелепипеда и определении точек вычисления функции для следующей итерации, используется стандартная схема разбиения (диагональные методы подробно рассмотрены в [5]). При этом вершины главной диагонали гиперпараллелепипеда рассматриваются как конечные точки одномерного интервала, а в качестве характеристики выбирается критерий, фигурирующий при описании алгоритма α. Функция на диагонали вычисляется в заданном числе точек, что является параметром метода.

Поведение функции на различных участках может быть различным, поэтому деление исходной области по количеству процессоров не самый лучший вариант, ведь время работы на каждом процессоре может сильно варьироваться. Необходимо осуществить балансировку нагрузки, т.е. отдать на обработку каждому процессору несколько участков, причем число таких участков должно быть больше числа процессоров. Как только последний процессор завершает обработку, из полученных результатов выбирается рекордное значение функции. Если значение рекорда получается больше оценки гиперпараллелепипеда, то в дальнейших итерациях данный гиперпараллелепипед не рассматривается (принятая стратегия, в общем случае, может не гарантировать отыскание глобального решения). Указанная стратегия начинает применяться с некоторой итерации, которая определяется в ходе решения в зависимости от размеров текущих гиперпараллелепипедов.

4. Результаты экспериментов

На рис.1 представлены результаты вычислительных экспериментов для пяти функций в области [-3;3]×[-3;3], для которых оптимальное значение было известно заранее. Для рассматриваемых примеров поиск решения осуществлялся с точностью 10^{-8}. Вычисления производились с использованием процессора Intel Core 2 Quad Q8200 (2.33GHz, 4Mb, 1333MHz). Число участков разбиения вдвое больше числа задействованных ядер. Параметр l не задается априори, а уточняется на каждой итерации алгоритма.

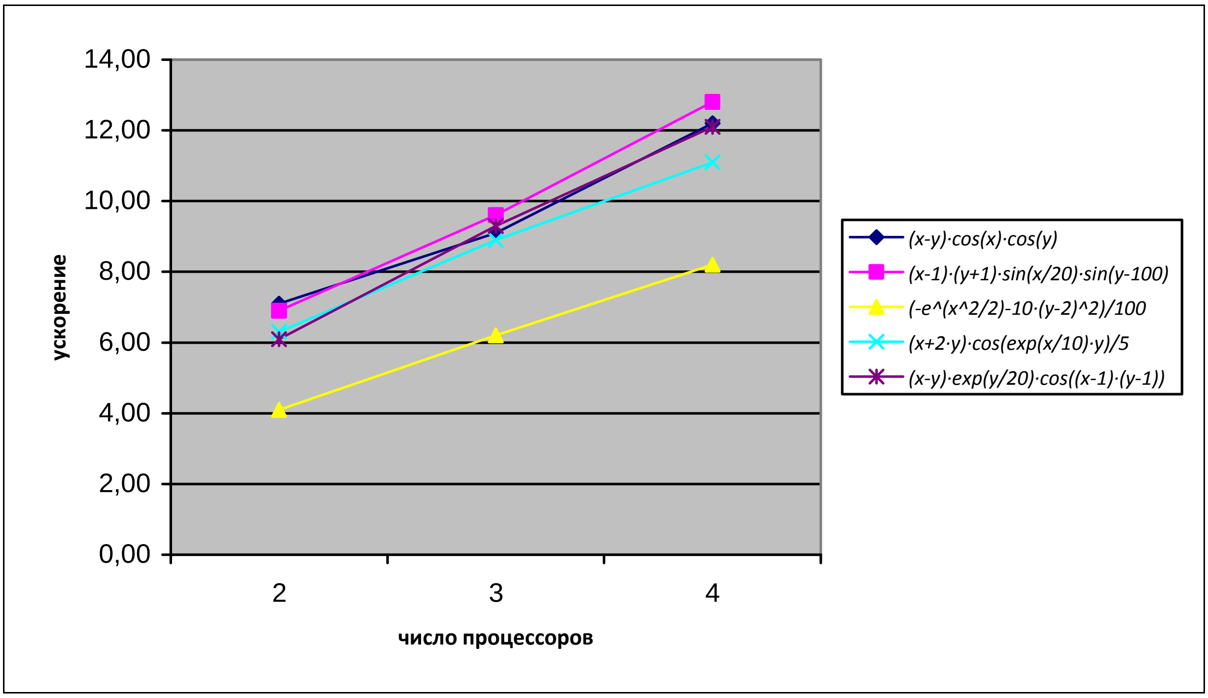

Рис. 1. Графики зависимости ускорения параллельного алгоритма от числа используемых процессоров

Графики показывают, что для функций с широким диапазоном значений параллельный алгоритм демонстрирует более чем линейное ускорение на приведенных примерах. Это связано, как указано выше, со свойствами используемого алгоритма. Предложенный алгоритм хорошо поддается распараллеливанию по данным.

5. Заключение

В работе рассмотрен параллельный диагональный алгоритм для вычисления глобального максимума многомерных функций из классов, определяемых кусочно-степенными мажорантами. На основе свойств погрешности алгоритма предложена эффективная параллельная схема, которая обеспечивает, по меньшей мере, линейное ускорение для функций из указанных классов.

Литература

1. Коротченко А.Г. О приближенно-оптимальных алгоритмах поиска экстремума в одном классе функций // Журнал вычислительной математики и математической физики. 1990. Т. 30. № 3. С. 355-365.

2. Коротченко А.Г., Бобков А.Н. О последовательном алгоритме поиска максимума в классах многоэкстремальных функций, определяемых кусочно-степенными мажорантами. Вестник ННГУ, серия Математика, 2004, вып. 1(2), с. 116-125.
3. Коротченко А.Г., Бобков А.Н. Об одном алгоритме поиска экстремума в классах функций, определяемых кусочно-степенными мажорантами. Вестник ННГУ, серия Математическое моделирование и оптимальное управление, 2004, вып. 1(27), с. 194-202.
4. Pinter J.D. Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). – Dordrecht: Kluwer Academic Publishers, 1996.
5. Сергеев Я. Д., Квасов Д. Е. Диагональные методы глобальной оптимизации. – М.: Физматлит, 2008, 352 стр.