Повышение эффективности реализации индексного метода решения задач глобальной оптимизации*

А.В. Сысоев, Т.А. Сысоева

Нижегородский госуниверситет им. Н.И. Лобачевского

Рассматривается индексный метод решения задач многомерной многоэкстремальной условной глобальной оптимизации. Предложена новая схема редукции размерности на основе построения семейства множественных отображений с использованием кривых Пеано, позволяющая существенно повысить число используемых процессоров при параллельной реализации индексного метода. Предложена и реализована модификация индесного метода, более точно оценивающая константы Липшица в процессе численного решения задач глобальной оптимизации. Реализована эффективная схема работы с оценками констант Липшица на основе приоритетных очередей.

1. Постановка задачи многомерной глобальной оптимизации

Рассмотрим многомерную задачу глобальной оптимизации:

$$\varphi(y^*) = \min\{\varphi(y): y \in D, g_j(y) \le 0, 1 \le j \le m\},$$
$$D = \{y \in R^N: a_i \le y_i \le b_i, 1 \le i \le N\}.$$

Пусть $y(x)$, $x \in [0,1]$ есть развертка Пеано, однозначно отображающая отрезок $[0,1]$ на единичный N-мерный гиперкуб P, т.е.

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

Используя отображение $y(x)$, многомерная задача может быть сведена к одномерной

$$\varphi(y(x^*)) = \min\{\varphi(y(x)): x \in [0,1], g_j(y(x)) \le 0, 1 \le j \le m\}. \quad (2)$$

Введя области

$$q_1 = [0,1], q_{j+1} = \{x \in q_j: g_j(y(x)) \le 0\}, 1 \le j \le m,$$

и положив

$$1 \le M = \max\{v(y(x)): x \in [0,1]\} \le m+1,$$

получим следующее представление задачи (1)-(2):

$$g_M^* = g_M(x_M^*) = \min\{g_M(x): x \in Q_M\} = \min\{g_M(y(x)): x \in q_m\}.$$

При этом испытание в любой точке $x^k \in [0,1]$ позволяет определить пару

$$z^k = g_v\left(y(x^k)\right), v = v\left(y(x^k)\right).$$

Если $\varphi(y)$ удовлетворяет условию Липшица, то $\varphi(y(x))$ удовлетворяет равномерному условию Гельдера

$$\left|\varphi(y(x_1)) - \varphi(y(x_2))\right| \le 4L\sqrt{N}(|x_1 - x_2|)^{1/N}. \quad (3)$$

Рассмотренная схема сведения многомерной многоэкстремальной задачи условной оптимизации к эквивалентной ей одномерной задаче позволяет применить для ее решения индексный метод[1,2].

* Работа выполнена в лаборатории «Информационные технологии» ВМК ННГУ при поддержке проекта «Подготовка и переподготовка профильных специалистов на базе центров образования и разработок в сфере информационных технологий», госконтракт № 07.P20.11.0030, а такжеСовета по грантам Президента Российской Федерации (грант № НШ-64729.2010.9) и РФФИ (гранты №№ 11-01-00682-а, 11-07-97017-р_поволжье_а)

2. Индексный метод решения многомерных задач

Первое испытание осуществляется в произвольной внутренней точке $x^1 \in (a,b)$. Выбор точки x_{k+1}, $k \geq 1$, любого последующего испытания определяется следующими правилами.

Правило1. Перенумеровать точки $x^1,\ldots, x^k$ предшествующих испытаний нижними индексами в порядке увеличения значений координаты, т.е.

$$0 = x_0 < x_1 <\ldots< x_k < x_{k+1} = 1 \tag{4}$$

и сопоставить им значения $z_i = g_v(x_i)$, $v = v(x_i)$, $1\leq i \leq k$, вычисленные в этих точках; точки $x_0 = 0$ и $x_{k+1} = 1$ введены дополнительно (значения z_0 и z_{k+1} не определены) для удобства последующих обозначений.

Правило 2 . Провести классификацию номеров i , $1 \leq i \leq k$, точек из ряда (4) по числу ограничений задачи, выполняющихся в этих точках, путем построения множеств

$$I_v = \{i: 1 \leq i \leq k, v = v(x_i)\}, \qquad 1 \leq i \leq m+1.$$

содержащихномеравсехточекx_i, $1 \leq i \leq k$, имеющихиндексы, равныеодномуитомужезначениюv. Граничныеточкиx_0= a иx_{k+1}= b интерпретируются как имеющие нулевые индексы, иимсопоставляетсядополнительноемножество$I_0 = \{0, k+1\}$.

Правило3. Вычислить текущие нижние границы

$$\mu_v = max\left\{\frac{|z_i - z_j|}{(x_i - x_j)^{1/N}},\ i,j \epsilon I_v, i > j\right\} \tag{5}$$

для неизвестных констант Липшица L_vфункций g_v, $1 \leq v \leq m+1$. Если множество I_vсодержит менее двух элементов или если μ_vиз (5) оказывается равным нулю, то принять $\mu_v = 1$.

Правило 4 . Для всех непустых множеств I_v, $1 \leq v \leq m+1$, определить величины

$$z_v^* = \begin{cases} min\{z_i: i \in I_v\}, v = v^* \\ \quad 0,\ v < v^*. \end{cases} \tag{6}$$

Т.е. z_v^*= 0, если существуют точки x_i, $1 \leq i \leq k$, имеющие индекс, больший v.

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

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r_v^2 \mu_v^2 \Delta_i} - 2\frac{(z_i + z_{i-1} - 2z_v^*)}{r_v \mu_v}, v = v(x_{i-1}) = v(x_i), \tag{7}$$

$$R(i) = 2\Delta_i - 4\frac{(z_i - z_v^*)}{r_v \mu_v}, v(x_{i-1}) < v(x_i) = v, \tag{8}$$

$$R(i) = 2\Delta_i - 4\frac{(z_{i-1} - z_v^*)}{r_v \mu_v},\ v = v(x_{i-1}) > v(x_i), \tag{9}$$

$$\Delta_i = (x_i - x_{i-1})^{\frac{1}{N}}. \tag{10}$$

Величины r_v>1, $1 \leq v \leq m+1$, являются параметрами алгоритма. Подходящий выбор значений r_vпозволяет использовать произведение r_v и μ_vкак оценку константы Липшица L_v, $1 \leq v \leq m+1$.

Правило 6. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика

$$R(t) = max\{R(i): 1 \leq i \leq k+1\}. \tag{11}$$

Правило7. Провести очередное испытание в серединной точке интервала (x_{t-1}, x_t), если индексы его концевых точек не совпадают, т.е.

$$x^{k+1} = \frac{x_t + x_{t-1}}{2}, v(x_{t-1}) \neq v(x_t). \tag{12}$$

В противном случае провести испытание в точке

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - sign(z_t - z_{t-1})\frac{1}{2r_v}\left[\frac{|z_t - z_{t-1}|}{\mu_v}\right]^N, v = v(x_{t-1}) = v(x_t). \tag{13}$$

Описанные правила можно дополнить условием остановки, прекращающим испытания, если

$$x_t - x_{t-1} \leq \varepsilon, \tag{14}$$

где t из (3.4.8) и ε >0 есть заданная точность.

3. Улучшение работы с оценками констант Липшица в программной реализации индексного метода

В работе [3] высказано предположение относительно реализации индексного метода для многомерных задач, что оценки констант Липшица μ_v являются неубывающими. Это утверждение верно только для одномерного случая.

В многомерном случае оценки констант Липшица могут убывать, поскольку в процессе редукции размерности условие Липшица трансформируется в условие Гельдера, в котором модуль разности $|x_1 - x_2|$входит в степени $1/N$. Нетрудно показать, что при делении интервала (x_i, x_j) новой точкой испытания x^*оценки констант Липшица для двух новых интервалов (x_i, x^*) и (x^*, x_j) могут привести к уменьшению общего максимума по всем интервалам, а, значит, и уменьшить оценку μ_v.

Была выполнена реализация индексного метода, учитывающая возможность того, что оценки констант Липшица μ_v могут уменьшаться в процессе выполнения очередной итерации. В первоначальной версии это потребовало введения схемы пересчета оценок по всем интервала. Можно предположить, что более точное оценивание констант Липшица даст улучшение в характере сходимости, то есть позволит либо уменьшить число итераций метода поиска, либо находить лучшие оценки глобального оптимума.

Проведеныэксперименты на 100 функциях Гришагина для сравненияреализации, построенной на предположении из работы [3], и новой, в которой учтено возможное уменьшение оценок констант Липшица. Результаты сравнения представлены в табл. 1.

Таблица 1.

	Количество итераций		
Значение функционала	Меньше	Такое же	Больше
Лучше	10	-	8
Такое же	42	31	6
Хуже	-	-	3

Как видим, на 18 функциях (первая строка) получено лучшее значение критерия, еще на 42 функциях найдена та же оценка глобального оптимума, что и в исходной версии, но за меньшее число итераций и на 31 функции результат не изменился.

4. Использование очередей для оценок констант Липшица

В индексном методе оценки констант Липшица строятся для каждого функционала задачи. Необходимость пересчета оценок ведет к замедлению выполнения каждой итерации, а значит, и всего индексного метода. Для ускорения работы была применена следующая идея:лучшие оценки констант Липшица запоминаются в очередях. Число очередей равно числу функционалов. При выполнении текущей итерации из соответствующей очереди удаляется оценка для интервала, в котором поставлена точка текущего испытания, вычисляются необходимые новые оценки и добавляются в очереди, если это возможно (оценку можно добавить в очередь, если она больше наименьшей из оценок, уже имеющихся в очереди). Таким образом, полный пересчет оценки константы Липшица по всем интервалам требуется только в случае, когда очередь соответствующего функционала опустела.

Выполнена реализация индексного метода с очередями для оценок констант Липшица. В табл. 2 представлены результаты сравнения времени работы метода с очередями и без очередей на некоторых функциях Гришагина(время в секундах).

Таблица 2.

№ функции	Реализация с очередями для констант Липшица	Реализация без очередей для констант Липшица
0	0,05853	0,1939
66	25,607	32,284

69	20,6606	26,0449
73	23,3715	29,0687
98	0,0716	0,1418

Таким образом, в среднем время работы метода уменьшилось на 30%.

5. Повышение эффективности схемы редукции размерности

Отображения, называемые *кривыми* (*развертками*) *Пеано*, сопоставляют любой липшицевой в гиперкубе P функции$\varphi(y)$одномерную функцию $\varphi(y(x))$, удовлетворяющую на отрезке [0, 1] равномерному условию Гельдера с показателем $1/N$, см. (3).

Алгоритм вычисления кривой Пеано с любой заданной точностью подробно описан, например, в [2]. Требуемая точность указывается целым числом M (*номер разбиения*), которое определяет допустимую максимальную погрешность оценки каждой координаты кривой $y(x)$ для любого заданного значения аргумента x, равную $2^{-(M+1)}$. Оценки точек кривой сопоставляют равномерной с шагом 2^{-NM} сетке на отрезке [0, 1] равномерную с шагом 2^{-M} сетку в гиперкубе P.

Редукция многомерных задач к одномерным с помощью разверток сохраняет непрерывность и равномерную ограниченность разностей при ограниченной вариации аргумента, однако теряет часть информации о близости точек в многомерном пространстве, поскольку точка x на отрезке [0, 1] имеет два соседа, тогда как соответствующая ей точка$y(x) \in P$имеет соседей по 2^N направлениям. Возможный способ учета этой информации состоит в следующем [4].

Вводится гиперкуб

$$P_0 = \{v \in R^N : -2^{-1} \le v_i \le 3 \cdot 2^{-1}, 1 \le i \le N\} \quad (15)$$

с длиной ребра, равной 2, и семейство гиперкубов

$$P_l = \{v \in R^N : -2^{-1} \le v_i + 2^{-l} \le 3 \cdot 2^{-1}, 1 \le i \le N\}, 1 \le l \le L, \quad (16)$$

где гиперкуб P_{l+1} получается путем сдвига гиперкуба P_l вдоль главной диагонали на шаг -2^{-l} по каждой координате, и для каждого гиперкуба P_l, $0 \le l \le L$ вводится своя развертка $y_l(x)$ типа кривой Пеано, отображающая отрезок [0, 1] на этот гиперкуб. Приближенное построение развертки $y_l(x)$ для точности, соответствующей разбиению с номером $m+1$ порождает в гиперкубе P_l равномерную сетку с шагом 2^{-m} по каждой координате. При этом на величину L накладывается ограничение $L<m$.

Поскольку гиперкуб P из (1) является общей частью всех гиперкубов семейства (15), (16), т.е.

$$P \cap P_0 \cap P_1 \cap \ldots \cap P_L = P \quad (17)$$

то имеют место включения

$$P \subset \{y_l(x) : x \in [0,1]\}, 0 \le l \le L. \quad (18)$$

Наконец, вводя ограничение

$$p(v) = max\{|v_i| - 2^{-1}, 1 \le i \le N\} \le 0, \quad (19)$$

можно задать гиперкуб P как множество вида

$$P = \{y_l(x), x \in [0,1] : p(y_l(x)) \le 0\}, 0 \le l \le L. \quad (20)$$

Введенные гиперкубы позволяют более полно отражать информацию о близости точек в многомерном пространстве. Если точки $v^1 = y_l(x_l^1) \in P, v^2 = y_l(x_l^2) \in P$ близки в гиперкубе P, но соответствующие им прообразы x_l^1, x_l^2 не являются близкими на отрезке [0, 1], то найдется такое соответствие $y_t(x), 0 \le t \le L$, что прообразы x_l^1, x_l^2, где $y^1 = y_t(x_t^1), y^2 = y_t(x_t^2)$, будут близкими на отрезке [0, 1] (см. [5]). Рассмотренная схема приведения исходной многомерной задачи к семейству одномерных называется *множественной разверткой*.

Множественная развертка позволяет естественным образом построить параллельную модификацию индексного метода. Действительно, достаточно раздать каждому процессу по одной развертке и обмениваться точками испытаний, чтобы последовательность точек на каждом процессе росла практически в p раз быстрее (где p – число процессов), чем в случае использования одного процесса.

Однако параллелизм на основе множественной развертки ограничен условием, которое накладывает приближенное построение каждой отдельной развертки, – $L+1<=m$. Это означает, что при точности построения развертки $m = 10$, возможно использование только десяти разверток, а, значит, только десяти процессоров для параллельного поиска глобального оптимума. Это ограничение существенно сужает возможность применения параллельных вычислений. Для его преодоления можно использовать *модификацию исходной схемы*, позволяющую строить *семейство множественных разверток*.

Рассмотрим первоначально двумерный случай. На рис. 1 представлены гиперкубы P_0-P_3 исходной множественной развертки для $L = 3$.

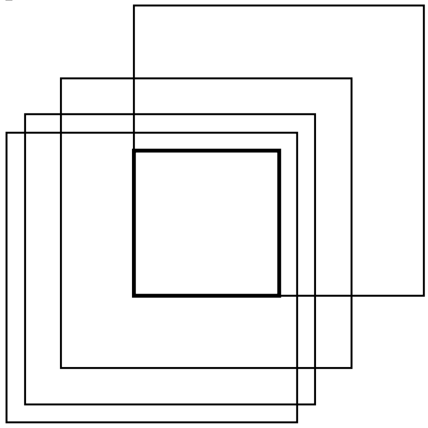

Рис. 1. Гиперкубы исходной множественной развертки ($N = 2$, $L = 3$)

При этом левый нижний угол гиперкуба P_0 совпадает с левым нижним углом гиперкуба P. Построим еще три множественных развертки, в каждой из которых положение базового гиперкуба P_0 выбирается так, чтобы у него с гиперкубом P совпадал один из трех оставшихся углов.

На рис.2 представлена множественная развертка, в которой у базового гиперкуба P_0 и гиперкуба P совпадают правые нижние углы. Нетрудно показать, что все четыре множественных развертки будут обладать одной и только одной общей разверткой, задаваемой в каждой из них гиперкубом

$$P_1 = \{v \in R^N : -1 \le v_i \le 1, 1 \le i \le N\} \quad (21)$$

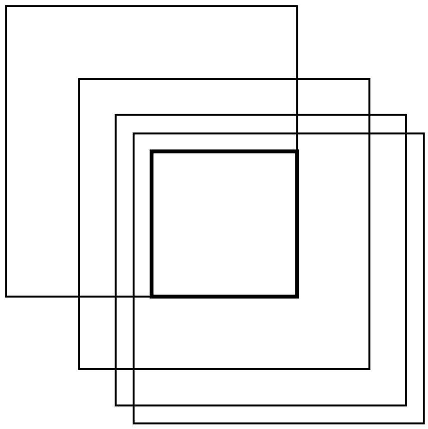

Рис. 2. Гиперкубы модифицированной множественной развертки

Таким образом, общее число разверток построенного семейства будет равно $4 \cdot 4 - 4 + 1 = 9$ для рассмотренного примера.

Дадим общее описание правила построения каждой множественной развертки для произвольного N.

Введем двоичную нумерацию множественных разверток, состоящую из N разрядов:

$$\alpha = (\alpha_0, \dots, \alpha_{N-1}),\ \alpha_i = 0, 1 \quad (22)$$

Пусть исходная множественная развертка имеет двоичный номер, в котором все разряды равны нулю.

Для построения каждой множественной развертки достаточно задать описание базового гиперкуба P_0 и правила сдвига гиперкубов.

Базовый гиперкуб P_0 для множественной развертки с двоичным номером α будет задаваться как

$$P_0 = \{v \in R^N: -2^{-1} - \alpha_i \le v_i \le 3 \cdot 2^{-1} - \alpha_i, 1 \le i \le N\} \quad (23)$$

То есть по тем разрядам, в которых в двоичном номере множественной развертки стоит единица, базовый гиперкуб будет иметь по соответствующим переменным границы $-3 \cdot 2^{-1}..2^{-1}$, а по разрядам равным нулю границы по переменным будут $-2^{-1}..3 \cdot 2^{-1}$.

Смещение же остальных гиперкубов P_1, …, P_Lв множественной развертке с номером α будет описываться вектором

$$\beta = (-2^{-l} \cdot (-1)^{\alpha_0}, \dots, -2^{-l} \cdot (-1)^{\alpha_{N-1}}) \quad (24)$$

То есть по переменным, которым в двоичном номере множественной развертки соответствует единица, сдвиг будет выполняться на 2^{-l}, а по остальным на -2^{-l}.

Модифицированная множественная развертка значительно повышает потенциал использования параллелизма в процессе поиска глобально оптимума. Так уже для трехмерных задач при точности построения приближения развертки $m = 10$, общее число доступных разверток в семействе множественных разверток составит $m \cdot 2^3 - 2^3 + 1 = 73$. В общем же случае число разверток может быть получено согласно следующему правилу.

Утверждение 1. Для задач размерности N при точности приближенного построения кривых Пеано m максимальное число разверток в семействе множественных разверток равно

$$(m - 1) \cdot 2^N + 1. \quad (25)$$

6. Программная реализация модифицированной множественной развертки

Введем «сквозную» нумерацию гиперкубов во всех семействах модифицированной множественной развертки от 0 до $Lmax = (m - 1) \cdot 2^N$. По номеру гиперкуба k можно восстановить номер семейства и номер развертки внутри семейства.

Опишем общую схему использования разверток в процессе выполнения итерации индексного метода.

Пусть точка очередного испытания принадлежит интервалу [k, k+1), который соответствует гиперкубу P_k. По номеру k определяем номер семейства разверток, номер гиперкуба в нем и, следовательно, положение этого гиперкуба в пространстве. С помощью развертки необходимо отобразить точку x в данный гиперкуб P_k.

Для полученной точки проверяется ограничение из (19). Если оно выполняется, то необходимо вычислить значения функционалов, предварительно отобразив точку из гиперкуба P_k в область поиска D.

Проведя вычисления функционалов, необходимо построить все прообразы точки испытания по всем используемым разверткам. Происходит обратный процесс. Сначала точка «переводится» из области поиска D в очередной гиперкуб P_l.

Затем точка из гиперкуба P_l отображается в единичный гиперкуб P.

Далее полученную точку с помощью развертки отображаем на отрезок $[l, l+1)$.

Литература

1. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. М.: Наука, 1978.
2. Стронгин Р.Г. Поиск глобального оптимума. М.: Знание, 1990.
3. Баркалов К.А. Разработка и исследование методов ускорения сходимости алгоритмов глобальной условной оптимизации // Дисс. на соискание ученой степени канд. физ.-мат. наук. 2006.
4. Strongin R.G., SergeyevYa.D. Global optimization: fractal approach and non-redundant parallelism, Journal of Global Optimization, 2003, 27(1), 25-50.
5. Гергель В.П. Решение одного класса многомерных многоэкстремальных многокритериальных задач со сложными ограничениями // Дисс. на соискание ученой степени канд. техн. наук. 1984.