Journal of Siberian Federal University. Engineering & Technologies 3 (2010 3) 335-346
~ ~ ~

УДК 681.513.5; 517.977

Поиск главных минимумов многоэкстремальных функций при активном учёте ограничений неравенств

А.В. Кузнецов*, А.И. Рубан
Сибирский федеральный университет,
660041 Россия, Красноярск, Свободный, 79 [1]

Received 3.09.2010, received in revised form 10.09.2010, accepted 17.09.2010

Построены алгоритмы поиска глобального и близких к нему главных минимумов многоэкстремальных функций многих непрерывных переменных при наличии ограничений неравенств. В основе алгоритмов лежит разбиение заданной области поиска на подобласти, тяготеющие к требуемым главным минимумам, с последующим поиском в каждой подобласти глобального экстремума на основе алгоритмов метода усреднения координат. На численных примерах продемонстрирована эффективность работы алгоритмов.

Ключевые слова: глобальная оптимизация, главные минимумы.

Введение

При получении (изучении) законов природы и конструировании объектов применимы экстремальные принципы. При этом формулируется экстремальная задача (например, минимизации функции или функционала в допустимой области изменения варьируемых переменных; мы в дальнейшем рассматриваем только минимизацию функций) и находится её решение, например, глобальный минимум функции. Качество получаемого решения, как правило, определяется величиной минимизируемой функции; тогда под подозрение надо ставить не только точку глобального минимума, но и точки близлежащих (по величине минимизируемой функции) минимумов. При конструировании объектов эти точки могут соответствовать более эффективным вариантам за счёт многокритериального оценивания их эффективности. Глобальный и близлежащие к нему минимумы будем называть главными минимумами.

В работе [1] предложен алгоритм поиска главных минимумов при изменении независимых переменных в гиперпрямоугольной области. Фрагменты решения на основе его простейших численных примеров приведены в [2, с. 139–143]. Примеры демонстрируют хорошее качество работы алгоритма – высокое быстродействие и точность.

В реальных условиях допустимая область поиска (определяемая, как правило, ограничениями типа неравенств) может быть сколь угодно сложной и достаточно узкой (в виде жгута).

* Corresponding author E-mail address: kuaw26@mail.ru
[1] © Siberian Federal University. All rights reserved

Проблеме учёта этих ограничений при поиске главных минимумов посвящена предлагаемая работа. При этом, естественно, мы переносим опыт работы с ограничениями, полученный при конструировании алгоритмов поиска одного глобального минимума [2], на решение задачи поиска главных минимумов многоэкстремальных функций.

Постановка задачи

Необходимо отыскать заданное количество главных экстремумов функции многих переменных

$$I(x) = \text{гл}\min_{x \in X}, \quad x = (x_1, ..., x_m) . \tag{1}$$

Допустимая область X определяется системой ограничений неравенств

$$\varphi_j(x) \le 0, \; j = \overline{1, m_1} . \tag{2}$$

Ограничения сужают область поиска экстремума. Функция качества $I(x)$ может быть многоэкстремальной, разрывной, недифференцируемой, а также может быть искажена помехой. Функции ограничений тоже могут быть невыпуклыми и недифференцируемыми. Поиск экстремумов осуществляется только на основе измерений или вычислений указанных функций: $I(x)$; $\varphi_j(x)$, $j = \overline{1, m_1}$.

Алгоритмы

Все приводимые ниже алгоритмы поиска включают в себя два этапа: первый этап – это разбиение исходной области на подобласти, второй – поиск глобального минимума в каждой из найденных подобластей.

Исследователь задает: 1) исходную гиперпрямоугольную область $П^0$ (с центром в x^0 и интервалами Δx^0), которая должна содержать в себе искомые главные экстремумы; 2) количество исходных пробных точек n^0, равномерно распределённых внутри этой области; 3) количество главных минимумов $n_{\text{гл мин}}$; 4) количество пробных точек n, вид ядра и степень s его селективности, используемые при поиске глобального минимума в каждой подобласти.

Были разработаны три способа **активного** учета ограничений как при разбиении исходной области $П^0$ на подобласти, так и при дальнейшем поиске глобального минимума в каждой из подобластей.

Первый способ самый простой в вычислительном отношении, но он самый затратный – последовательно генерируются пробные точки (в $П^0$ или в подобластях, на которые она разбита) и из них оставляются только те точки, которые удовлетворяют исходным ограничениям неравенств. Чем меньше объём допустимой области, тем больше число безрезультатных проб. В этом основной недостаток данного способа учёта ограничений неравенств и соответственно первого алгоритма поиска главных минимумов.

Второй способ базируется на изменении минимизируемой функции в пробных точках. Во-первых, она берётся в относительных величинах, а во-вторых, к ней добавляется штрафная компонента (тоже в относительных величинах), построенная по функциям $\varphi_j(x)$, входящим в левую часть ограничений неравенств, в тех точках, где ограничения нарушены. При этом в расчётах участвуют только исходные пробные точки фиксированного объёма (n^0 или n). Более

тонкий учёт ограничений в пробных точках, не попадающих в допустимую область, приводит к существенному сокращению числа генерируемых пробных точек. Особенно эффективен этот способ учёта ограничений при стремлении к нулю объёма допустимой области.

Третий способ позволяет учесть нарушаемые ограничения за счёт расширения нормированного ядра в алгоритме расчёта рабочего шага и размеров прямоугольной области для пробных движений следующего шага. Появляются (в виде произведения) дополнительные ядра для относительных величин нарушаемых ограничений. Сила этого способа такая же, как и предыдущего, но он несколько сложней в настройке вида вводимых ядер и в подборе их коэффициентов селективности.

Первый алгоритм *(с использованием пробных точек внутри допустимой области)*

В заданной прямоугольной области поиска $П^0$ последовательно равномерно размещаются пробные точки, и из них формируется n^0 точек, попадающих в допустимую область (2). Разбиение пробных точек $x^{(i)}$, $i=\overline{1, n^0}$ на группы точек, тяготеющих к главным минимумам, производится следующим образом. В пробных точках вычисляется функция качества $I(x^{(i)})=I^{(i)}$, $i=\overline{1, n^0}$ и отыскивается такая точка $x_{\min}$, которая соответствует наименьшей величине $I_{\min}=\min\{I^{(i)}, i=\overline{1, n^0}\}$. Затем просматриваются все остальные пробные точки (кроме $x_{\min}$) и из них отбираются те точки $x^{(i)}$, которые расположены в окрестности $x_{\min}$, например, в соответствии с условием: $\left|x_v^{(i)}-x_{v\min}\right|\le \Delta x_v^0/c,\ v=\overline{1, m}$. Найденные точки образуют первую группу точек, множество номеров которых обозначим через M_{n1}, где n_1 – мощность множества M_{n1}. Если мощность множества меньше, чем некоторое $n_{\min}$ (в представленном далее примере $n_{\min}$=5), то это множество удаляется как малозначимое. Оставшиеся пробные точки по той же схеме разбиваем на группы точек, пока общее количество групп не станет равным заданному количеству $n_{\text{гл мин}}$ главных минимумов или в исходной выборке не останется пробных точек.

Параметр $c\ge 1$ определяет размеры подобластей и задается исследователем пока интуитивно на основе априорной информации о функции качества, требуемого количества $n_{\text{гл мин}}$ главных минимумов и объема начальной выборки.

После разбиения исходной области на подобласти можно использовать информацию, содержащуюся в пробных точках из каждой подобласти, для получения уточненных значений точки $x^0_{\text{подобл}}$ (центр подобласти), в среднем более тяготеющей к экстремуму, и интервалов варьирования $\Delta x^0_{\text{подобл}}$ (размер подобласти), необходимых для второго этапа поиска.

$x^0_{\text{подобл}}$ и $\Delta x^0_{\text{подобл}}$ при поиске **главного минимума** в каждой подобласти, например в первой из них, рассчитывают по формулам

$$x^0_{v,\text{подобл}}=\sum_{i\in M_{n_1}} x_v^{(i)}\overline{p}^{(i)}_{s,\min}, v=\overline{1,m}, \overline{p}^{(i)}_{s,\min}=\frac{p_s(g^{(i)}_{\min})}{\sum\limits_{j\in M_{n_1}} p_s(g^{(j)}_{\min})}, g^{(i)}_{\min}=\frac{I^{(i)}-\hat{I}_{\min}}{\hat{I}_{\max}-\hat{I}_{\min}}$$

$$\Delta x^0_{v,\text{подобл}}=\gamma_q\cdot\Delta x^0\cdot\left(\sum_{i\in M_{n_1}}^{n}|u_v^{(i)}|^q\overline{p}^{(i)}_{s,\min}\right)^{1/q}, v=\overline{1,m},\ q=1,2,\cdots .$$

Здесь: $\hat{I}_{\min}=\min\{I^{(i)}, i\in M_{n_1}\}$, $\hat{I}_{\max}=\max\{I^{(i)}, i\in M_{n_1}\}$; $p_s(g)$ – ядра; $u_v^{(i)}$ – равномерно распределённые в интервале [-1; 1] числа.

При поиске главных **максимумов** расчёт $x^0_{\nu,\text{подобл}}$ и $\Delta x^0_{\nu,\text{подобл}}$ ведётся аналогично вышеприведённому, только ядра $\overline{p}^{(i)}_{s,\min}$ заменяют на $\overline{p}^{(i)}_{s,\max} = \dfrac{p_s(g^{(i)}_{\max})}{\sum\limits_{j\in M_{\eta}} p_s(g^{(j)}_{\max})}$, $g^{(i)}_{\max} = \dfrac{\hat{I}_{\max} - I^{(i)}}{\hat{I}_{\max} - \hat{I}_{\min}}$.

На этом первый этап (разбиение исходной области на подобласти) считается завершенным.

На втором этапе производится отыскание экстремумов в каждой найденной подобласти. При этом к неравенствам (2), которые входят в исходную постановку задачи, добавляются неравенства, определяющие прямоугольные подобласти, найденные на этапе разбиения исходной области на подобласти. Добавление таких неравенств не дает алгоритму выходить за границы этих подобластей и исключает многократные попадания в один и тот же экстремум.

Опишем алгоритм движения к минимуму на (l+1)-м шаге, если на предыдущем шаге был получен вектор искомых переменных x^l, построенный на основе метода усреднения координат, он имеет вид:

$$
\begin{aligned}
&x^{l+1}_{\nu} = x^l_{\nu} + \Delta x^l_{\nu} u_{\nu,\min},\ u_{\nu,\min} = \sum_{i=1}^{n} u^{(i)}_{\nu} \overline{p}^{(i)}_{s,\min},\ \nu = \overline{1,\, m}\,,\\
&\overline{p}^{(i)}_{s,\min} = \frac{p_s(g^{(i)}_{\min})}{\sum\limits_{j=1}^{n} p_s(g^{(j)}_{\min})},\quad g^{(i)}_{\min} = \frac{I^{(i)} - \hat{I}_{\min}}{\hat{I}_{\max} - \hat{I}_{\min}},\\
&\Delta x^{l+1}_{\nu} = \gamma_q \cdot \Delta x^l_{\nu} \cdot \left(\sum_{i=1}^{n} |\, u^{(i)}_{\nu}\, |^q \overline{p}^{(i)}_{s,\min} \right)^{1/q},\ \nu = \overline{1,\, m}\,,\ l=0,\, 1,\, 2,\, \ldots;\\
&x^0 \equiv x^0_{\text{подобл}},\ \Delta x^0 \equiv \Delta x^0_{\text{подобл}};\ [0 < \gamma_q,\, q \in \{1, 2, \cdots\},\, 0 < s]\,.
\end{aligned}
\tag{3}
$$

Здесь: $\hat{I}_{\min} = \min\{I^{(i)},\, i = \overline{1,\, n}\}$, $\hat{I}_{\max} = \max\{I^{(i)},\, i = \overline{1,\, n}\}$; s – коэффициент селективности ядра $p_s(\cdot)$; в переменных $u^{(i)}_{\nu}$, $\overline{p}^{(i)}_{s,\min}$ для упрощения записи опущен номер итерации l; всегда $0 \le g^{(i)}_{\min} \le 1$. Ядра $\overline{p}^{(i)}_{s,\min}$ нормированы на системе n пробных точек: $\sum\limits_{i=1}^{n} \overline{p}^{(i)}_{s,\min} = 1$.

Перед совершением (l+1)-го рабочего шага при получении пробных точек $x^{(i)}$, $i = \overline{1,\, n}$, последовательно генерируются точки (равномерно распределённые) в прямоугольной области Π^l с центром в точке x^l:

$$
x^{(i)}_{\nu} = x^l_{\nu} + \Delta x^l_{\nu} \cdot u^{(i)}_{\nu},\ u_{\nu} \in [-1;\, 1],\ \nu = \overline{1,\, m},\ i = 1,\, 2,\, ..,
$$

и из них оставляется n точек, попадающих в допустимую область X. Пробные точки лежат в области $X^l = \Pi^l \bigcap X$. В пробных точках вычисляется минимизируемая функция $I^{(i)}$, $i = \overline{1,\, n}$, и на основе этих экспериментальных данных $[\, x^{(i)}_{\nu},\ \nu = \overline{1,\, m},\ I^{(i)},\ i = \overline{1,\, n}\,]$ совершается вышеприведённое рабочее движение и пересчитываются размеры области поиска.

Критерием останова процесса поиска обычно служит условие уменьшения размера области варьирования Δx^l до заданной величины: $\max\{|\, \Delta x^l_{\nu}\, |,\ \nu = \overline{1, m}\} \le \varepsilon$.

$p_s(g)$ на интервале изменения $0 \le g \le 1$ являются неотрицательными, убывающими ядрами. Выбор формы и коэффициента s селективности ядра существенно влияет на характеристики поиска глобального минимума. Чаще всего используется степенное ядро

$p_s(g) = (1-g^r)^s, r = 1,2,3,\ldots$. Коэффициент s для степенных ядер лежит в широком диапазоне целых положительных чисел и подбирается сравнительно просто.

Второй алгоритм *(с переходом к штрафной функции)*

В области П⁰ равномерно размещаются n^0 пробных точек. В них вычисляется штрафная функция:

$$\widetilde{I}(x^{(i)}) = \frac{I^{(i)} - \hat{I}_{\min}}{\hat{I}_{\max} - \hat{I}_{\min}} + \alpha \cdot \max\left\{ \frac{\varphi_j(x^{(i)}) - \varphi_{j\min}}{\varphi_{j\max} - \varphi_{j\min}},\ j \in J^{(i)} \in \overline{1, m_1} \right\}, \tag{4}$$

$i = \overline{1, n^0}$, $\alpha > 0$.

Здесь множество $J^{(i)}$ включает только номера тех функций ограничений, для которых ограничения нарушены: при $j \in J^{(i)} \in \overline{1, m_1}$ всегда $0 < \varphi_j(x^{(i)})$; $\hat{I}_{\min} = \min\{I^{(i)},\ i = \overline{1, n}\}$, $\hat{I}_{\max} = \max\{I^{(i)},\ i = \overline{1, n}\}$; для каждого фиксированного j: $\varphi_{j\min} = \min\{\varphi_j(x^{(i)}), i : 0 < \varphi_j(x^{(i)})\}$, $\varphi_{j\min} = \min\{\varphi_j(x^{(i)}), i : 0 < \varphi_j(x^{(i)})\}$;

Если при некотором $j=l$ множество $\{i : 0 < \varphi_l(x^{(i)})\}$ пустое, то этот номер l не входит во множество $j^{(i)}$ (для каждого фиксированного i). Для пробных точек, в которых ни одно неравенство не нарушено (т.е. множество $j^{(i)}$ пустое), штрафная добавка берётся равной нулю.

Если при некотором $j=l$ множество $\{i : 0 < \varphi_l(x^{(i)})\}$ состоит из одного элемента, то соответствующий элемент в (4), входящий под оператор max{}, полагаем равным 1: $[\varphi_l(x^{(i)}) - \varphi_{l\min}] / [\varphi_{l\max} - \varphi_{l\min}] = 1$.

Далее, отыскивается точка $x_{\min}$, которая соответствует наименьшей величине $\widetilde{I}_{\min} = \min\{\widetilde{I}^{(i)}, i = \overline{1, n^0}\}$ штрафной функции. Дальнейший отбор точек, находящихся в окрестности $x_{\min}$, и разбиение на группы точек проводится так же, как и в первом алгоритме.

После разбиения П⁰ на подобласти поиск минимума в каждой из подобластей ведется так же, как и в первом алгоритме, но пробные точки размещаются равномерно по всей подобласти (а не в её части, как в предыдущем алгоритме), и вместо исходной функции $I^{(i)}$ используется штрафная функция $\widetilde{I}^{(i)}$ (4).

Третий алгоритм *(с переходом к многомерным ядрам)*

В области П⁰ равномерно размещаются n^0 пробных точек и в них вычисляются многоэкстремальная функция и функции φ_j ограничений: $I(x^{(i)}) = I^{(i)}, \varphi_j(x^{(i)}),\ j = \overline{1, m}, i = \overline{1, n^0}$. Далее формируются нормированные ядра $\overline{p}_s^{(i)}$:

$$\overline{p}_s^{(i)} = \frac{p_{1,s}(g^{(i)}) \prod_{j \in J^{(i)}} p_{2,s}(g_j^{(i)})}{\sum_{\mu=1}^{n} p_{1,s}(g^{(\mu)}) \prod_{j \in J^{(\mu)}} p_{2,s}(g_j^{(\mu)})} \tag{5}$$

Здесь при минимизации $g^{(i)} = g_{\min}^{(i)} = \dfrac{I^{(i)} - \hat{I}_{\min}}{\hat{I}_{\max} - \hat{I}_{\min}}$, при максимизации $g^{(i)} = g_{\max}^{(i)} = \dfrac{\hat{I}_{\max} - I^{(i)}}{\hat{I}_{\max} - \hat{I}_{\min}}$; а также $g_j^{(i)} = \dfrac{\varphi_j(x^{(i)}) - \varphi_{j,\min}}{\varphi_{j,\max} - \varphi_{j,\min}}$, $j = \overline{1, m_1}$; $p_{1,S}$ – ядро по оптимизируемой функции; $p_{2,S}$ – ядро по

ограничениям; $\hat{I}_{\max}$, $\hat{I}_{\min}$, $\varphi_{j,\max}$, $\varphi_{j,\min}$ для каждого фиксированного j, а также множество $J^{(i)}$ те же, что и во втором алгоритме.

Если при некотором $j=l$ множество $\{i:\ 0<\varphi_j(x^{(i)})\}$ состоит из одного элемента, то аргумент ядра $p_{2,s}$ полагаем равным 0,75. Это значение было получено экспертным путем. Если во всех пробных точках неравенства не нарушены, то ядра становятся одномерными и зависят только от оптимизируемой функции: $\overline{p}_s^{(i)} = p_{1,s}(g^{(i)}) \Big/ \sum_{\mu=1}^{n} p_{1,s}(g^{(\mu)})$.

Далее отыскивается точка $x_{\min}$, которая соответствует наибольшей величине нормированного ядра $\overline{p}_s^{(i)}$. Отбор точек, находящихся в окрестности $x_{\min}$, и разбиение на группы точек проводятся так же, как и ранее.

После разбиения Π^0 на подобласти поиск минимума в каждой из них ведется так же, как и для первого алгоритма, но вместо размещения точек только в допустимой области точки размещаются равномерно по всей подобласти поиска, и расчет нормированных ядер $\overline{p}_{s,\min}^{(i)}$ производится по формуле (5).

Численный пример

Исследуем поведение алгоритмов при сужении допустимой области. Рассмотрим четырёхэкстремальную потенциальную функцию, построенную за счёт применения операции min к четырём элементарным экспоненциальным потенциалам:

$$\begin{aligned} I(x) = \min\Big\{ & -3\exp\{-3[|x_1-3|^{1.5}+|x_2|^{1.5}]\}, \\ & -5\exp\{-2.5[|x_1+3|^{2.5}+|x_2|^{2.5}]\}, \\ & -7\exp\{-[|x_1|^{1.2}+|x_2-3|^{1.2}]\},\ -10\exp\{-2[|x_1|^{2}+|x_2+3|^{2}]\}\Big\} \end{aligned} \tag{6}$$

Допустимая область имеет вид кольца шириной Δ:

$$x_1^2+x_2^2 \le (3+\frac{\Delta}{2}) \text{ и } x_1^2+x_2^2 \ge (3-\frac{\Delta}{2}). \tag{7}$$

На рис. 1 представлен пространственный вид функции (6). Она имеет минимумы в точках (–3; –0), (0; 3), (3; 0), (0; –3) и принимает в них соответственно значения (–3), (–5), (–7), (–10). На рис. 2 приведены линии равных уровней функции (6) и допустимая область (7) при Δ=0,4.

Все три вышеописанных алгоритма хорошо справляются с поиском условного глобального экстремума при достаточно широкой допустимой области (большом Δ).

На рис. 3-5 отражено изменение по итерациям минимизируемой функции и координат при поиске двух главных минимумов функции (6) при ограничениях (7) и Δ=1,01.

Как видно из рис. 3, для первого алгоритма исходная точка поиска находится в малой окрестности истинного решения. Данный факт обусловлен тем, что пробные точки размещаются только в допустимой области, она для данного примера имеет малую площадь.

Как видно из рис. 4, второй алгоритм достаточно быстро отыскивает главные экстремумы, совершая при этом на два-три шага больше чем первый алгоритм. Это обусловлено тем, что

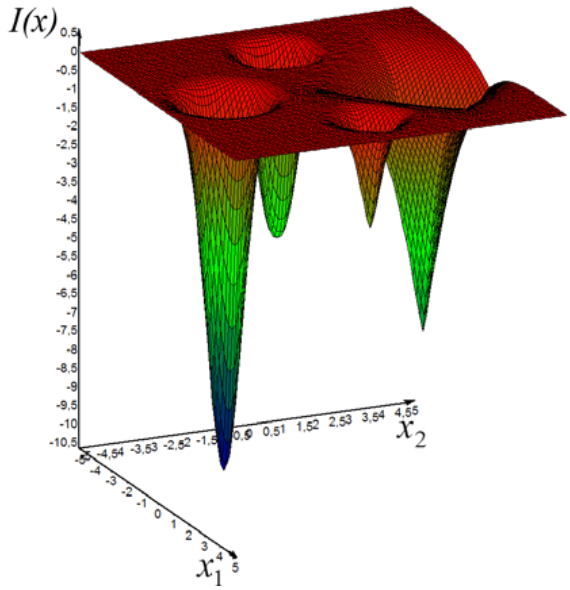

Рис. 1. Потенциальная функция (6)

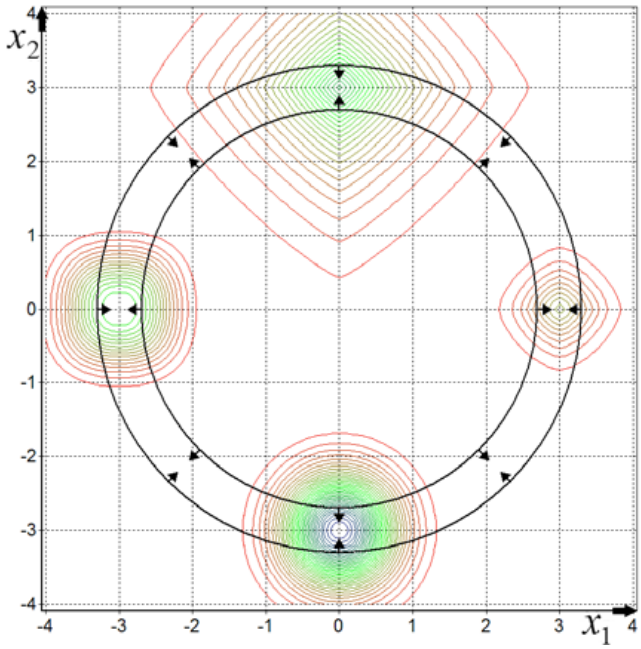

Рис. 2. Равные уровни функции (6) и допустимая область (7) при Δ=0,4

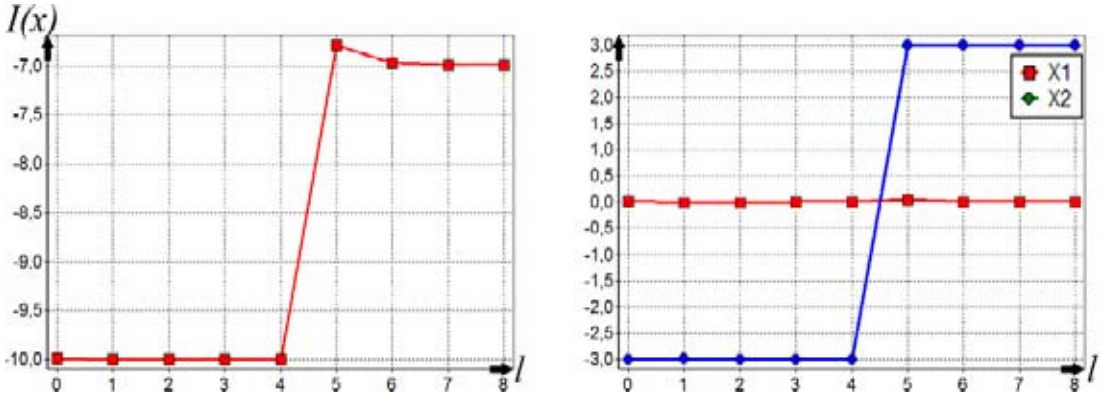

Рис. 3. Процесс поиска двух главных минимумов **первым** алгоритмом

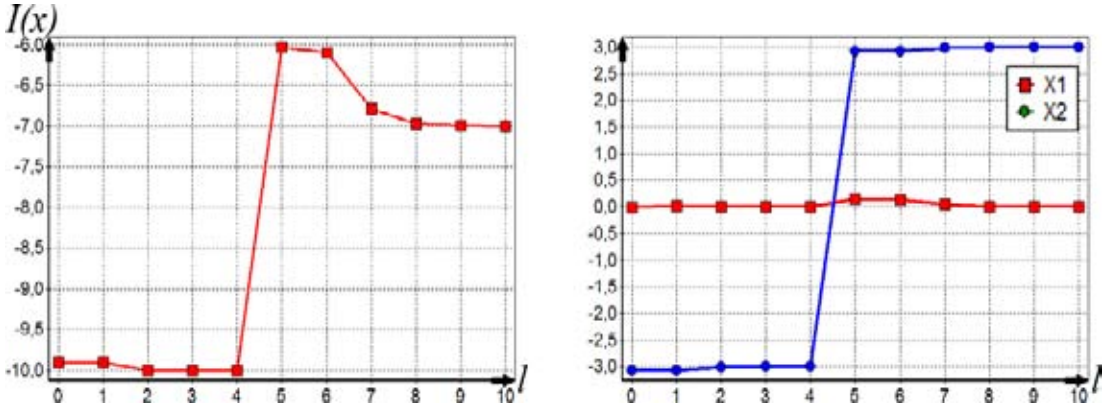

Рис. 4. Процесс поиска двух главных минимумов **вторым** алгоритмом

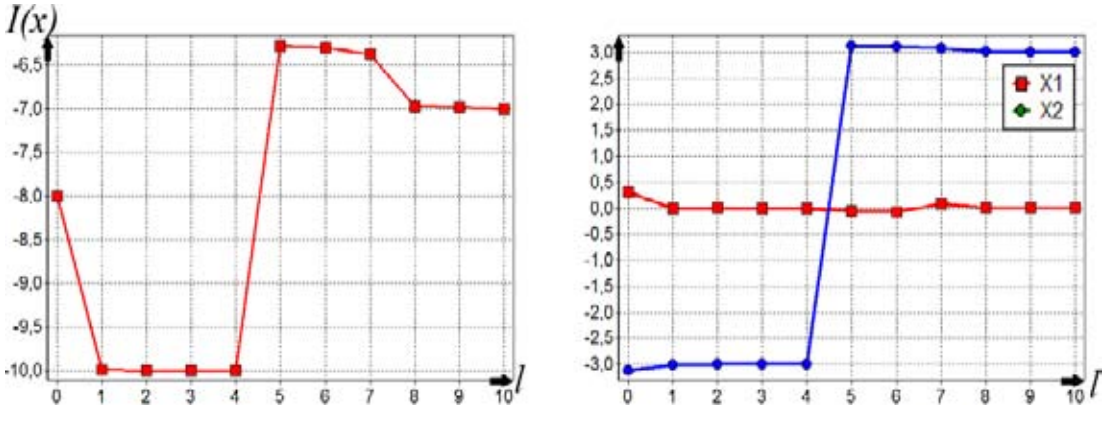

Рис. 5. Процесс поиска двух главных минимумов **третьим** алгоритмом

второй алгоритм производит поиск по области, в то время как первый алгоритм уже на первой стадии нашел хорошее приближенное решение.

Как видно из рис. 5, третий алгоритм ведет себя примерно так же, как и второй.

Проведем исследование поведения алгоритмов для узкой допустимой области. Для этого будем менять ширину допустимой области Δ от 0,001 до 0,01 с шагом 0,001 и усреднять полученные значения на каждом шаге по 101 запуску. Используем следующие показатели качества работы алгоритмов: N – количество попыток размещения пробных точек и оценки вероятности отыскания «истинных» главных экстремумов – $\hat{P}(\left|x^* - x\right| \le \varepsilon)$.

Первый алгоритм. Считаем, что n^0=500, область поиска с центром x^0=(0; 0) и размером Δx^0= (4; 4), q=2, y_q=1,2, ядро по функции качества параболическое с коэффициентом селективности s=100, размер выборки при поиске экстремума в подобласти n=150. На рис. 6 представлены графики описывающие поведение алгоритма. Первый график показывает усредненное значение N для каждого Δ, второй – оценку вероятности отыскания истинного экстремума.

Как и ожидалось, в первом алгоритме уменьшение ширины допустимой области Δ приводит к резкому увеличению количества попыток генерации пробных точек. Как видно из рис. 6, для отыскания экстремумов при ширине Δ=0,001 было предпринято $2{,}2 \cdot 10^6$ попыток размещения пробных точек. Оценка вероятности отыскания истинного экстремума очень высока и равна единице, это объясняется тем, что узкая допустимая область практически не оставляет «свободы» поиска для алгоритма. С ростом Δ уменьшается количество попыток размещения пробных точек, так как допустимая область становится шире и в нее легче попасть. Следует отметить, что такое большое число попыток размещения пробных точек при наличии узкой допустимой области делает этот алгоритм практически непригодным для использования при решении практических задач, так как в практических задачах размещение пробной точки эквивалентно проведению дорогостоящего эксперимента.

Второй алгоритм. Зададим следующие параметры алгоритма: размер выборки для выделения подобластей n^0=500, область поиска с центром x^0=(0; 0) и размером Δx^0= (4; 4), q=2, y_q=1,2, ядро по функции качества параболическое с коэффициентом селективности s=200, размер выборки при поиске экстремума в подобласти n=150, штрафной коэффициент α=1. На рис. 7 представлены графики, описывающие поведение алгоритма. Первый график показывает коли-

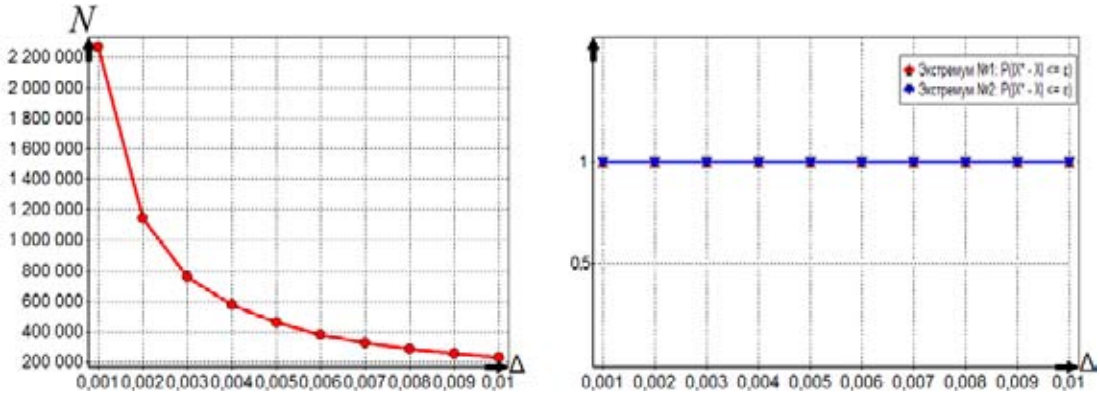

Рис. 6. Зависимость показателей качества работы **первого** алгоритма от Δ

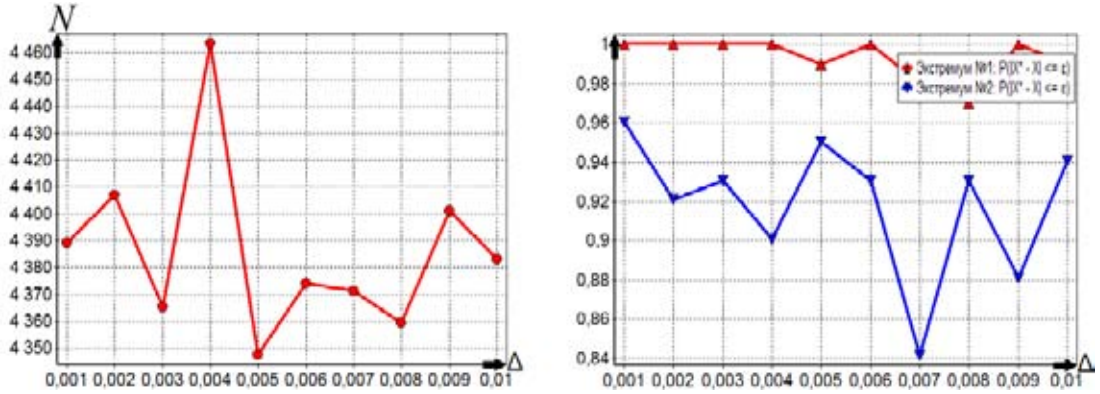

Рис. 7. Зависимость показателей качества работы **второго** алгоритма от Δ

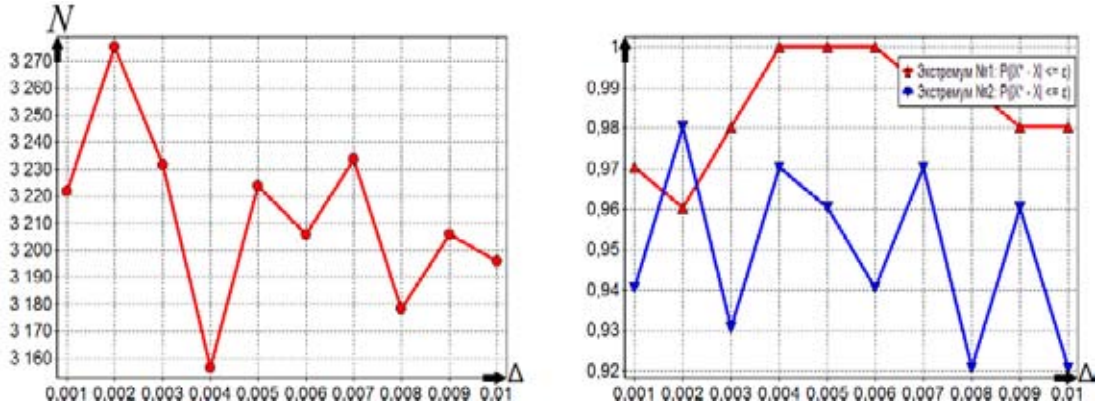

Рис. 8. Зависимость показателей качества работы **третьего** алгоритма от Δ

чество попыток генерации пробных точек, второй – оценку вероятности отыскания истинного экстремума.

Из графиков наглядно видно, что количество попыток размещения пробных точек в четыреста раз меньше, чем для первого алгоритма (для Δ =0,001). Оценка вероятности отыскания истинного решения достаточно высока и в среднем равна 0,95. Качество работы алгоритма не зависит от ширины допустимой области. Этот алгоритм более прост в настройке, чем третий алгоритм, так как необходимо настроить одно ядро, а у третьего алгоритма – два.

Третий алгоритм. Зададим следующие параметры алгоритма: размер выборки для выделения подобластей n^0=400, область поиска с центром x^0=(0; 0) и размером Δx^0= (4; 4), q=2, y_q=1,2, ядро по функции качества параболическое с коэффициентом селективности s=50, ядро по ограничениям параболическое с коэффициентом селективности s=100, размер выборки при поиске экстремума в подобласти n=100. На рис. 8 представлены графики, описывающие поведение алгоритма. Первый график показывает количество попыток генерации пробных точек, второй – оценку вероятности отыскания истинного экстремума.

Из графиков видно, что так же, как и для второго алгоритма, количество попыток и оценки вероятности отыскания истинного решения не зависит от ширины допустимой области. При этом количество попыток в шестьсот раз меньше, чем для первого алгоритма (для Δ=0,001). Оценка вероятности отыскания истинного решения достаточно высока и в

среднем равна 0,95. Третий алгоритм работает немного лучше второго, но он более сложен в настройке.

Программная реализация алгоритмов

Продемонстрированные алгоритмы входят в состав пакета глобальной оптимизации **Global Optimizer v2.0,** в котором также представлены основные алгоритмы глобальной оптимизации, базирующиеся на методе усреднения координат. Пакет реализован на языке Delphi с использованием среды визуального программирования Delphi 2010 фирмы Embarcadero Technologies и функционирует в операционной системе из семейства Windows 2000/XP/Vista/7.

При проектировании программных алгоритмов поиска главных минимумов был использован объектно-ориентированный подход. Алгоритмы реализованы как надстройка над алгоритмами поиска глобального экстремума, описанными в монографии [2]. Этот подход позволяет распараллелить поиск главных экстремумов и тем самым значительно повысить скорость работы алгоритма. Этот вариант требует специальных исследований.

Заключение

Алгоритмы поиска главных экстремумов привлекают своей простотой, сравнительно небольшими вычислительными затратами, высокими точностными показателями.

Проведенные исследования выявили, что все три алгоритма примерно одинаково эффективны, если допустимая область достаточно большая и количество ограничений сравнительно невелико. При сужении допустимой области и увеличении количества ограничений первый алгоритм становится неэффективным, так как требует значительного количества попыток размещения пробных точек и вычислений в них функций ограничений, тогда как второй и третий алгоритмы от этого свободны.

Статья публикуется при поддержке Программы развития Сибирского федерального университета

Список литературы

1. Рубан А.И. Метод непараметрической поисковой оптимизации // Изв. вузов. Физика. 1995. Т. 38. № 9. С. 65–73.
2. Рубан А.И. Глобальная оптимизация методом усреднения координат: Монография. Красноярск: ИПЦ КГТУ, 2004. 302 с.

Search of the Main Minima of Multiextreme Functions with Active Account of Inequalities

Alexey V. Kuznetsov and Anatoly I. Rouban
Siberian Federal University
79 Svobodny, Krasnoyarsk, 660041 Russia

Three algorithms for searching main minima proposed. These algorithms are based on the splitting of search region into subregions around demanded main minima, with the subsequent search in each subregion. On numerical examples the overall effectiveness of algorithms is shown.

Keywords: global optimization, main minima.