ТЕХНИЧЕСКАЯ КИБЕРНЕТИКА
No.1 · 1987 · с. 119 – 127

УДК 519.854

МЕТОД ПОЛОВИННЫХ ДЕЛЕНИЙ ДЛЯ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ФУНКЦИИ МНОГИХ ПЕРЕМЕННЫХ

ЕВТУШЕНКО Ю.Г., РАТЬКИН В.А.

(Пересмотрена 12 марта 2003 г.)

Введение. Многочисленные практические задачи приводят к необходимости поиска глобальных решений. Укажем, например, задачи принятия решений в условиях неопределенности, многокритериальную оптимизацию, задачи нахождения гарантированных минимаксных оценок и т.п. Задачи нахождения глобальных решений являются наиболее трудоемкими в вычислительной математике. Поэтому при их создании следует в максимальной степени использовать все существующие возможности вычислительной техники, учитывать ее изменения. Важнейшей тенденцией развития вычислительной техники является резкое увеличение оперативной памяти ЭВМ. В данной работе предлагается численный метод глобальной оптимизации, который в значительно большей степени, чем известные методы, использует при расчетах оперативную память ЭВМ.

Одним из наиболее плодотворных направлений глобальной оптимизации является идея неравномерных покрытий допустимого множества. Интерес к этому направлению значительно возрос в последнее время в связи с разработкой новых высокопроизводительных ЭВМ, основанных на параллельной и конвейерной организации расчетов. Идея методов неравномерных покрытий, их программная реализация даны в [1, 2]. Основной схемой расчета в [1, 2] было так называемое послойное покрытие допустимого множества, в минимальной степени использующее машинную память. В этих работах был сформулирован следующий принцип: методы глобальной оптимизации должны допускать возможность использования вспомогательных методов локальной оптимизации для ускорения расчетов. Этот принцип сохранился в [3, 4], он также будет использован ниже. В [3] был предложен метод ветвления для реализации покрытия допустимого множества. В [4] был предложен алгоритм покрытия неравномерными параллелепипедами, активнее использующий память и благодаря этому более эффективный, чем послойные покрытия. Ниже будет описан метод, который можно условно назвать методом половинных делений. Этот термин не должен привести к путанице с методом половинного деления в задачах одномерной оптимизации. В нашем случае происходит половинное деление n-мерных параллелепипедов из допустимого множества. Метод объединяет идеи неравномерных покрытий с подходом, основанным на использовании оценок, получаемых с помощью интервального анализа [5, 6].

1. Постановка задачи, основные обозначения. Рассмотрим задачу отыскания глобального минимума функции $f(x)$, определенной на n-мерном параллелепипеде $P \subset \mathbb{R}^n$

$$f_* = \min_{x \in P} f(x), \qquad P = \{x \in \mathbb{R}^n : a \le x \le b\}. \tag{1.1}$$

Здесь и ниже векторное неравенство $x \le z$ означает, что $x^i \le z^i$ для всех $1 \le i \le n$. Через X_* обозначим множество решений задачи (1.1). Введем множество ε-оптимальных (или приближенных) решений задачи (1.1):

$$X_*^{\varepsilon} = \{x \in P : f(x) \le f_* + \varepsilon\}. \tag{1.2}$$

Очевидно, что $X_* \subset X_*^{\varepsilon} \subset P$. В большинстве практических задач достаточно найти по крайней мере одну точку $x_r \in X_*^{\varepsilon}$ и вычислить значение $f_r = f(x_r)$. Другими словами, надо с заданной точностью ε определить величину глобального минимума функции от n-мерного вектора x и найти хотя бы одну точку x_r, где это приближенное значение достигается.

В процессе расчетов будут использованы векторы $a_i,\ b_i \in \mathbb{R}^n$ и порождаемые ими прямоугольные параллелепипеды P_i с гранями, параллельными координатным плоскостям: $P_i = \{x \in P : a_i \le x \le b_i\}$. Центр c_i параллелепипеда P_i и вектор его главной диагонали d_i определяются по формулам

$$c_i^j = 1/2(b_i^j + a_i^j), \quad d_i^j = b_i^j - a_i^j, \qquad 1 \le j \le n. \tag{1.3}$$

Запись

$$\lim_{\|d_i\|_\infty \to 0} (\ldots)$$

подразумевает предельный процесс для последовательности вложенных параллелепипедов $P \supset P_1 \supset \ldots \supset P_i \supset P_{i+1} \supset \ldots$ такой, что $\|d_i\|_\infty \to 0$. Здесь и всюду ниже

$$\|d_i\|_\infty = \max_{1 \le j \le n} |b_i^j - a_i^j|.$$

Предельную точку этой последовательности обозначим P_∞. Обозначим

$$\varphi(P_i) = \min_{x \in P_i} f(x). \tag{1.4}$$

Предположим, что есть способ для каждого $P_i \subset P$ определять нижнюю оценку значения функции $\varphi(P_i)$, т.е. задана функция $g(P_i)$, удовлетворяющая следующим двум условиям:

$$g(P_i) \le \varphi(P_i) \text{ для любого } P_i \subset P, \tag{1.5}$$

$$\lim_{\|d_i\|_\infty \to 0} [\varphi(P_i) - g(P_i)] = 0 \text{ равномерно по } P_\infty \in P. \tag{1.6}$$

В ряде случаев удается получить более точную оценку, для которой (1.6) заменяется условием

$$\varphi(P_i) - g(P_i) = O(\|d_i\|_\infty^{\beta}), \qquad \beta \ge 1, \tag{1.7}$$

равномерно по $P_\infty \in P$.

Требование, чтобы была известна функция $g(P_i)$ с $\beta > 1$, необязательно, однако наличие такого свойства ускоряет вычислительный процесс.

Значения функции $g(P_i)$ можно определять либо с помощью техники интервального анализа [5], либо используя какие-либо дополнительные предположения о классе, к которому принадлежит функция f. Пусть, например, функция f удовлетворяет условию Липшица с константой L, т.е. для любых x и z из P выполнено неравенство

$$|f(x) - f(z)| \le L\|x - z\|_\infty. \tag{1.8}$$

Тогда для всех $x \in P_i$ имеем

$$f(x) - f(c_i) \ge -L\|x - c_i\|_\infty \ge -(L/2)\|d_i\|_\infty. \tag{1.9}$$

Поэтому можно взять

$$g(P_i) = f(c_i) - (L/2)\|d_i\|_\infty. \tag{1.10}$$

Здесь, таким образом, $\beta = 1$.

Если помимо (1.8) выполнено условие вида $\|f_x(x) - f_x(z)\|_1 \le M\|x - z\|_\infty$, где $\|v\|_\alpha = \left(\sum_{i=1}^{n} |v^i|^\alpha\right)^{1/\alpha}$, то

$$\begin{aligned} f(x) - f(c_i) &\ge \langle f_x(c_i), x - c_i\rangle - \frac{M}{2}\|x - c_i\|_\infty \ge \\ &\ge -\frac{1}{2}\|d_i\|_\infty \cdot \|f_x(c_i)\|_1 - \frac{M}{8}\|d_i\|_\infty^2. \end{aligned}$$

Объединяя это неравенство с (1.9), получим, что можно взять

$$g(P_i) = f(c_i) - \frac{1}{2}\|d_i\|_\infty \cdot \min\left\{L, \|f_x(c_i)\|_1 + \frac{M}{4}\|d_i\|_\infty\right\}. \tag{1.11}$$

В этом случае $\beta = 2$.

2. Описание алгоритма. В процессе работы алгоритма будет строиться некоторая последовательность $B_m = \{P_1, P_2, \ldots, P_m\}$ параллелепипедов P_i, принадлежащих P. В центрах этих параллелепипедов c_i будут вычисляться значения минимизируемой функции. Пусть $N_m = \{c_1, c_2, \ldots, c_m\}$ — последовательность центров параллелепипедов, принадлежащих P. Назовем текущим рекордом величину

$$R_m = \min_{c_i \in N_m} f(c_i).$$

Любую точку c_s из N_m, удовлетворяющую условию $R_m = f(c_s)$, назовем рекордной точкой и обозначим x_r.

С каждым параллелепипедом P_i свяжем набор $S_i = (c_i, d_i, g_i)$, где $g_i = g(P_i)$. Совокупность наборов S_i для всего набора параллелепипедов B_m будем называть списком наборов и обозначать $S = \{S_1, S_2, \ldots, S_m\}$.

Начальные операции.

1) Положить $P_1 = P$.

2) Задать $\varepsilon > 0$ и некоторую точку $x_0 \in P$.

3) Вычислить c_1, d_1, $f(c_1)$, $f(x_0)$, $R^{(1)} = \min\{f(c_1), f(x_0)\}$.

4) Вычислить $g_1 = g(P_1)$.

5) Положить $N_1^{(1)} = \{c_1\}$, $B_1^{(1)} = \{P_1\}$, $S_1 = (c_1, d_1, g_1)$, $S^{(1)} = \{S_1\}$. Взять $x_r = c_1$, если $f(c_1) = R^{(1)}$, и $x_r = x_0$, если $f(x_0) = R^{(1)}$.

6) Если $g_1 \ge R^{(1)} - \varepsilon$, то закончить работу алгоритма и перейти к п. 17.

Основной цикл (k-й шаг).

7) Из текущего набора параллелепипедов $B_m^{(k)}$ выбрать тот параллелепипед P_s, для которого

$$g_s = \min_{1 \le i \le m} g_i.$$

8) В параллелепипеде P_s определить номер наибольшего ребра t:

$$d_s^t = \max_{1 \le j \le n} d_s^j.$$

9) Разделить параллелепипед P_s пополам по t-й координате, породив тем самым два новых параллелепипеда P' и P''. Их центры и главные диагонали обозначим c', d' и c'', d'' соответственно.

10) Вычислить

$$\tilde{R} = \min\{f(c'), f(c'')\}. \tag{2.1}$$

11) Если $\tilde{R} < R^{(k)}$, то положить $R^{(k+1)} = \tilde{R}$. В качестве рекордной точки x_r взять ту из точек c', c'', в которой достигается минимум (2.1).

12) Если $\tilde{R} \geq R^{(k)}$, то положить $R^{(k+1)} = R^{(k)}$.

13) Определить величины $g' = g(P')$ и $g'' = g(P'')$.

14) Исключить параллелепипед P_s из набора $B_m^{(k)}$, т.е. удалить набор S_s из списка $S^{(k)}$. Включить в список два набора $S' = (c', d', g')$ и $S'' = (c'', d'', g'')$, положив $S_s = S'$ и $S_{m+1} = S''$.

15) В полученном списке наборов $\{S_i\}_{1 \leq i \leq m+1}$ провести для всех S_i следующую проверку: если выполнено

$$g_i \geq R^{(k+1)} - \varepsilon, \tag{2.2}$$

то S_i исключить из списка. Новый список наборов $\{S_{i_1}, S_{i_2}, \ldots, S_{i_p}\}$ перенумеровать и обозначить $S^{(k+1)} = \{S_j\}_{1 \leq j \leq p}$.

16) Положить $m = p$. Если $m = 0$, т.е. $S^{(k+1)} \neq \emptyset$, то закончить работу алгоритма и перейти к п. 17. В противном случае перейти к п. 7, положив $k = k + 1$.

Окончание работы.

17) Точка x_r является рекордной, т.е. $x_r \in X_*^\varepsilon$ есть ε-приближенное решение задачи (1.1).

3. Обоснование алгоритма.

Теорема 1. *Пусть в задаче* (1.1) *функция f полунепрерывна снизу на n-мерном прямоугольном параллелепипеде P. Пусть для всякого параллелепипеда $P_i \subset P$ определена функция $g(P_i)$, удовлетворяющая условиям* (1.5) *и* (1.6). *Тогда описанный алгоритм за конечное число вычислений значений функции f определяет рекордную точку $x_r \in X_*^\varepsilon$.*

Доказательство. Покажем, что условие остановки метода будет выполнено при конечном номере шага q основного цикла. Из компактности P следует равномерная полунепрерывность снизу функции f на P. Поэтому существует такое $\delta_1 > 0$, что

$$f(c_i) - \varphi(P_i) \leq \frac{\varepsilon}{2}, \tag{3.1}$$

для любого $P_i \subset P$ такого, что $\|d_i\|_\infty \leq \delta_1$. Из условий (1.5) – (1.6) следует существование такого $\delta_2 > 0$, что

$$\varphi(P_i) - g(P_i) \leq \frac{\varepsilon}{2}, \tag{3.2}$$

для любого $P_i \subset P$ такого, что $\|d_i\|_\infty \leq \delta_2$. Складывая неравенства (3.1) и (3.2), получаем, что

$$f(c_i) - g(P_i) \leq \varepsilon, \tag{3.3}$$

как только

$$\|d_i\|_\infty \leq \delta, \tag{3.4}$$

где $\delta = \min\{\delta_1, \delta_2\}$. Из (3.3) и неравенства $R^{(q)} \leq f(c_i)$ следует, что для любого P_i, для которого выполнено (3.4), имеет место условие (2.2) исключения S_i из списка.

Следовательно, никакой список S_i, полученный в процессе работы алгоритма, заведомо не будет содержать параллелепипедов настолько малых, что для них выполнено условие (3.4). Утверждение о конечности числа шагов алгоритма следует теперь из того, что делению пополам подвергается наибольшее ребро рассматриваемого параллелепипеда. Поэтому за конечное число делений он будет разбит на параллелепипеды, удовлетворяющие условию (3.4), а так как число параллелепипедов в списке конечно, то и остальные его представители за конечное число шагов алгоритма будут разделены на такие параллелепипеды, что для них будет выполнено (3.4).

Покажем теперь, что рекордная точка x_r принадлежит X_*^ε. Из полунепрерывности f на компакте P следует, что X_* непусто. Последовательность $B^{(1)}, B^{(2)}, \ldots, B^{(k)}, \ldots$ получалась путем половинного деления исходного параллелепипеда P. Поэтому каждая точка x из P исключалась только тогда, когда находился некоторый параллелепипед P_i, содержащий точку x и удовлетворяющий условию (2.2). Возьмем произвольную точку x_* из множества X_*. Эта точка была исключена вместе с некоторым содержащим ее параллелепипедом P_α. При исключении имели место условия:

$$x_* \in P_\alpha, \qquad f(x_*) \geq g(P_\alpha) \geq R^{(k_\alpha)} - \varepsilon.$$

Так как последовательность значений $R^{(k)}$ монотонно убывает, то $R^{(k_\alpha)} \geq R^{(q)}$ и поэтому выполнено $f_* = f(x_*) \geq R^{(q)} - \varepsilon$. Таким образом, $R^{(q)} \leq f_* + \varepsilon$, т.е. $x_r \in X_*^\varepsilon$, где $f(x_r) = R^{(q)}$. $\square$

4. Некоторые модификации. Для более эффективной работы предлагаемого алгоритма целесообразно учитывать следующие дополнения и замечания по организации вычислительного процесса.

1) Для того чтобы не проводить на каждом шаге основного цикла поиск набора с наименьшим g_i в текущем списке наборов в п. 7 и сократить количество проверок в п. 15, удобно хранить список наборов в порядке возрастания g_i. Тогда в п. 1 следует всегда выбирать P_1, а в п. 15 проверки будут осуществляться последовательно, начиная с элементов в конце списка. Как только найдется g_s такое, что неравенство (2.2) не выполнено, то проверки необходимо прекратить, так как для всех остальных наборов g_i, $1 \leq i \leq s$, это неравенство тоже не будет выполнено. При таком упорядочении наборов нет необходимости осуществлять перенумерацию элементов списка после его "сортировки" в п. 15. Отметим, что в описанном случае несколько усложнится включение наборов S' и S'' в список. Их необходимо включать в список таким образом, чтобы сохранялась упорядоченность наборов по g_i.

2) В целях экономного использования памяти ЭВМ при программной реализации алгоритма можно в каждом списке S_i из текущего набора хранить не n-мерный массив чисел d_i, а компактно закодированную информацию о целых числах ℓ_j, показывающих, сколько раз делилось пополам ребро d^j исходного параллелепипеда P для получения d_i^j. Тогда диагональ d_i вычисляется по формулам вида $d_i^j = d^j \cdot 2^{\ell_j}$, $1 \leq j \leq n$, что в машинной арифметике реализуется весьма эффективно.

3) Для метода несущественно, что исследуемый параллелепипед P_i делится на каждом шаге основного цикла именно на два параллелепипеда. Его можно делить и на большее число частей. Благодаря этому можно повысить эффективность работы алгоритма. Пусть для P_i выполнено неравенство

$$\max_{1 \leq j \leq n} d_i^j < 2 \min_{1 \leq j \leq n} d_i^j, \tag{4.1}$$

означающее, что ребра текущего параллелепипеда стали одного порядка. Тогда рекомендуется разбивать этот параллелепипед на 2^n частей, разделив P_i пополам по каждой координате. Если бы все эти параллелепипеды были получены в результате работы исходного алгоритма, то при этом было бы произведено на

$$\sum_{i=1}^{n-1} 2^i = 2^n - 2$$

больше вычислений минимизируемой функции по сравнению с данной модификацией.

Заметим, что такая стратегия деления обоснована только тогда, когда функция f промасштабирована так, что величины

$$L_i = \max_{x \in P} \left| \frac{\partial f}{\partial x^i}(x) \right|, \qquad 1 \le i \le n,$$

одного порядка.

4) Существенное ускорение работы алгоритма можно получить, если в процессе вычислений использовать процедуры локальной минимизации для задачи (1.1). Если после очередного вычисления значения функции f в точке c_s был уменьшен рекорд, то следует обратиться к локальному поиску минимума f на P, взяв в качестве начальной точки c_s. Пусть в результате получена некоторая точка $\bar{x}$, в которой $f(\bar{x}) < f(\bar{c}_s)$. Тогда полагаем $\tilde{R} = f(\bar{x})$, $x_r = \bar{x}$. Уменьшение рекорда дает возможность в п. 15 исключить большее число параллелепипедов P_i из списка.

5) Если в постановке задачи (1.1) есть ограничения: x_j, $j \in J \subseteq \{1, \ldots, n\}$, — целые, то можно уменьшить каждый параллелепипед P_i в списке, заменяя его максимальным параллелепипедом с целочисленными координатами x^j, $j \in J$, содержащимся в P_i.

5. Результаты расчетов. Для сравнительного анализа эффективности работы предлагаемого метода и методов из [1]–[4] решалась следующая тестовая многоэкстремальная задача:

$$f_* = \min_{x \in P} \left(\prod_{i=1}^{n} \cos(c^i x^i + c^{n+i}) + \prod_{i=1}^{n} \cos(c^{2n+i} x^i + c^{3n+i} \right), \tag{5.1}$$

где $P = \{x \in \mathbb{R}^n : -1 \le x^i \le 1,\ 1 \le i \le n\}$. Задача решалась для размерностей $n = 1, 2, 3, 4$. В табл. 1 представлены коэффициенты c^i для этих значений n.

Таблица 1

Коэффициенты для функции (5.1)

n	1	2	3	4
c^1	-6.47314	0.94775	-5.22607	7.22412
c^2	-1.08618	5.19019	-0.58521	0.18140
c^3	4.93066	-0.07813	-7.98633	-6.32520
c^4	-5.012245	4.74048	3.54321	-4.76050
c^5		7.44678	-3.22080	7.36377
c^6		5.10718	-7.13696	-4.44949
c^7		6.36621	-7.38574	5.63477
c^8		4.00903	2.09302	7.33618
c^9			2.91260	5.08154
c^{10}			-5.79810	1.06226
c^{11}			-0.08203	7.54785
c^{12}			2.40845	5.44849
c^{13}				6.12744
c^{14}				-6.53638
c^{15}				7.16406
c^{16}				-7.67358

Расчеты проводились следующими методами: ГЛОБ1 – метод послойных покрытий (см. [1, 2]), ГЛОБ2 – метод двоичного ветвления без стратегии выбора направления ветвления (см. [3]), ГЛОБ2М – метод двоичного ветвления со стратегией полного перебора возможных направлений ветвления, ГЛОБ3 – описанный выше метод половинных делений. Всюду предполагалось, что функция (5.1) удовлетворяет условию (1.8), функция $g(P_i)$ определялась по формуле (1.10). Константы Липшица L для минимизируемой функции и приближенные значения минимума f_* для различных n даны в табл. 2.

Таблица 2

Константы Липшица и приближенные значения минимума

n	1	2	3	4
L	11.404	18.692	26.189	37.632
f_*	-1.971	-1.943	-1.338	-1.722

Для различных значений n и ε в табл. 3 приводится необходимое число вычислений значения минимизируемой функции, потребовавшееся при решении задачи (5.1) различными методами. В крайнем правом столбце табл. 3 приводится количество вычислений функции, которое потребовал бы полный перебор с постоянным шагом $2\varepsilon/L$.

Таблица 3

Результаты решения задачи (5.1)

n	ε	ГЛОБ1	ГЛОБ2	ГЛОБ2М	ГЛОБ3	Полн. пер.
	0.001	2290	703	303	303	11405
1	0.0001	21773	2139	887	689	114050
	0.00001	214377	7083	2933	737	1140500
	0.1	4304	3128	2093	425	34040
2	0.03	21126	10541	7013	981	388213
	0.01	87061	36305	20741	2841	3494000
	0.003	201845	123769	61245	8633	38821300
	1.0	9607	7209	6513	5641	17963
3	0.5	52482	21297	18153	12537	143697
	0.1	330659	162289	96817	63545	17965000
	2.0	73150	73145	83009	70545	125346
4	1.5	175047	115361	229553	104097	396116

Приведенные результаты показывают, что при $n \geq 1$ описанный выше метод требует заметно меньше вычислений функции по сравнению со всеми другими методами. Это достигается благодаря более активному использованию машинной памяти.

В расчетах не использовались процедуры локальной оптимизации, что несколько снижало эффективность работы методов. Все методы были запрограммированы на языке Си, расчеты велись на персональном компьютере IBM PC-XT.

6. Решение задач нелинейного программирования (НЛП). Рассмотрим следующую задачу:

$$f_* = \min_{x \in Z} f(x), \tag{6.1}$$

где $Z = P \cap X$, а $X = \{x \in \mathbb{R}^n : F(x) = 0\}$. Функция $F(x) \geq 0$ всюду на P. Множество решений задачи (6.1) обозначим через Z_*. Определим множества

$$X^\varepsilon = \{x \in \mathbb{R}^n : F(x) \leq \varepsilon\}, \quad Z^\varepsilon = X^\varepsilon \cap P, \quad Z_*^\varepsilon = \{x \in Z^\varepsilon : f(x) \leq f_* + \varepsilon\}.$$

Предполагаем, что для функции $F(x)$ определена функция $q(P_i)$, аналогичная $g(P_i)$ для $f(x)$, т.е.

$$q(P_i) \leq \min_{x \in P_i} F(x) = \psi(P_i) \ \text{ для любого } \ P_i \subset P, \tag{6.2}$$

$$\lim_{\|d_i\|_\infty \to 0} [\psi(P_i) - q(P_i)] = 0 \ \text{ равномерно по } \ P_\infty \in P. \tag{6.3}$$

Приведенный выше алгоритм после незначительных изменений можно использовать для решения задачи (6.1). В данном случае набор S_i будет иметь вид $S_i = (c_i, d_i, g_i, q_i)$, $g_i = g(P_i)$, $q_i = q(P_i)$. Если в п. 10 основного цикла алгоритма оказалось, что $c' \in Z^\varepsilon$, то вычислить значение $f(c')$, если $c'' \in Z^\varepsilon$, то вычислить $f(c'')$ и определить по формуле (2.1) величину $\tilde{R}$. Если одна из этих точек, скажем c', принадлежит Z^ε, а $c'' \notin Z^\varepsilon$, то положить $\tilde{R} = f(c')$. Если обе точки недопустимые, то положить $\tilde{R} = R^{(k)}$. В п. 14 помимо g_i вычисляется q_i. В п. 15 набор S_i исключается из списка, если выполнено хотя бы одно из следующих двух условий:

$$q(P_i) > \delta, \tag{6.4}$$

$$g(P_i) \geq R^{(k+1)} - \varepsilon. \tag{6.5}$$

Здесь δ задается в п. 2 алгоритма таким образом, чтобы $0 < \delta < \varepsilon$.

Теорема 2. *Пусть в задаче* (6.1) *множество* $Z \neq \emptyset$, *функции* f, F *полунепрерывны снизу на* P, *и для всякого параллелепипеда* $P_i \subset P$ *определены функции* $g(P_i)$, $q(P_i)$, *удовлетворяющие условиям* (1.5) – (1.6), (6.2) – (6.3). *Тогда алгоритм за конечное число вычислений функций* f, F *отыскивает точку* $x_r \in Z_*^\varepsilon$.

Доказательство. Покажем, что для любых $\varepsilon > \delta > 0$ существует такой конечный номер шага q основного цикла, что список наборов $S^{(q)}$ станет пустым множеством. Из полунепрерывности снизу функций f и F на компакте P следует, что для любых $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ существуют такие $\delta_1 > 0$ и $\delta_2 > 0$, что

$$f(c_i) - \varphi(P_i) \leq \varepsilon_1, \tag{6.6}$$

$$F(c_i) - \psi(P_i) \leq \varepsilon_2, \tag{6.7}$$

как только $\|d_i\|_\infty \leq \delta_1$, $\|d_i\|_\infty \leq \delta_2$. Из условий (1.5), (1.6), (6.2), (6.3) следует, что для любых $\varepsilon_3 > 0$ и $\varepsilon_4 > 0$ существуют такие $\delta_3 > 0$ и $\delta_4 > 0$, что

$$\varphi(P_i) - g(P_i) \leq \varepsilon_3, \tag{6.8}$$

$$\psi(P_i) - q(P_i) \leq \varepsilon_4, \tag{6.9}$$

как только $\|d_i\|_\infty \le \delta_3$, $\|d_i\|_\infty \le \delta_4$. Существует такое конечное q, что начиная с q-го шага для всех P_i выполнено условие $\|d_i\|_\infty \le \Delta$, где $\Delta = \min\{\delta_1, \delta_2, \delta_3, \delta_4\}$ (см. теорему 1). Складывая почленно неравенства (6.6), (6.8) и (6.7), (6.9), получим, что для P_i выполнены неравенства

$$f(c_i) - g(P_i) \le \varepsilon_1 + \varepsilon_3, \tag{6.10}$$
$$F(c_i) - q(P_i) \le \varepsilon_2 + \varepsilon_4. \tag{6.11}$$

Если $c_i \in Z^\varepsilon$, то, положив $\varepsilon_1 + \varepsilon_3 = \varepsilon$ и учитывая, что $R^{(q)} \le f(c_i)$, из (6.10) получим $R^{(q)} - g(P_i) \le \varepsilon$, т.е. выполнено неравенство (6.5) и S_i исключается из списка. Если $c_i \notin Z^\varepsilon$, то $F(c_i) > \varepsilon$ и из (6.11) получаем $\varepsilon - q(P_i) < \varepsilon_2 + \varepsilon_4$. Положив $\delta = \varepsilon - \varepsilon_2 - \varepsilon_4$ (можно, очевидно, считать, что $0 < \delta < \varepsilon$), приходим к выполнению условия (6.4) и S_i исключается из списка.

Итак, алгоритм за конечное число шагов q основного цикла заканчивает работу. Предположим, что при этом никакая из точек

$$c_i \in \{N_{m_k}^{(k)}\}_{1\le k\le q}$$

не принадлежит Z^ε. Тогда все параллелепипеды $P_i^{(k)}$ таковы, что

$$q(P_i^{(\ell)}) > \delta > 0, \qquad P = \bigcup_{k=1}^{q} \bigcup_{i=1}^{m_k} P_i^{(k)}.$$

Но это противоречит условию теоремы о том, что $Z \ne \emptyset$. Поэтому среди точек из $N_m^{(q)}$ найдется хотя бы одна точка, принадлежащая Z^ε.

Возьмем какую-нибудь точку $x_* \in Z_*$. Для нее существует такой параллелепипед P_α, что $x_* \in P_\alpha$. В процессе работы алгоритма параллелепипед P_α был исключен из списка, т.е. выполнялось какое-то из двух условий $q(P_\alpha) > \delta$, $g(P_\alpha) \ge R^{(q)} - \varepsilon$. Первый случай невозможен, так как $x_* \in P_\alpha$ и $F(x_*) = 0$. Поэтому имел место второй случай, но тогда $f(x_*) \ge q(P_\alpha) \ge R^{(q)} - \varepsilon$. А так как $f(x_r) = R^{(q)}$ и $x_r \in X^\varepsilon$, то из последнего неравенства следует, что найденная алгоритмом рекордная точка x_r принадлежит Z_*^ε. $\square$

Как видно из приведенного изложения, задача (6.1) с вычислительной точки зрения оказывается проще, чем задача (1.1) (по числу обращений к вычислению значений минимизируемой функции). Действительно, здесь в п. 14 основного цикла алгоритма появляются дополнительные возможности сократить список благодаря тому, что рассматриваемый параллелепипед P_i может находиться в недопустимой области, т.е. для него выполнено условие (6.4). Здесь так же, как и выше, существенное ускорение работы алгоритма можно получить, используя методы поиска локальных решений задачи (6.1).

Заключение. Описанный подход можно применять к решению задач многокритериальной оптимизации. Наиболее простой способ состоит в использовании результатов работы [7]. Предложенный метод можно перенести на отыскание решений задачи поиска глобального минимакса аналогично тому, как было сделано в [8].

В данной статье использовалось предположение о существовании оценки минимума минимизируемой функции f на произвольном параллелепипеде P_i — функции $g(P_i)$. В разделе 1 указывались формулы, определяющие $g(P_i)$ для функций, удовлетворяющих условию Липшица. Более перспективным представляется вычисление значений функции $g(P_i)$ на основе техники интервального анализа, не использующего знания констант Липшица. С одной стороны, оценки констант Липшица, как правило, сильно завышены, с другой стороны, даже если известны точные константы Липшица минимизируемой функции на исходном параллелепипеде, то на произвольном параллелепипеде $P_i \subset P$ они могут не достигаться и даже быть сильно завышенными. Техника интервального анализа дает возможность более тонко оценивать характер поведения функций на рассматриваемом параллелепипеде P_i, что позволяет проводить более эконономные расчеты.

ЛИТЕРАТУРА

1. *Евтушенко Ю.Г.* Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке). ЖВМ и МФ, 1971. 11, N° 6.

2. *Евтушенко Ю.Г.* Методы поиска глобального экстремума. — В кн.: Исследование операций. М.: ВЦ АН СССР, 1974, вып. 4.

3. *Потапов М.А.* Методы неравномерных покрытий и их применение для решения задач глобальной оптимизации в диалоговом режиме. Автореф. на соиск. учен. ст. канд. физ.-матем. наук. М.: ВЦ АН СССР, 1984.

4. *Evtushenko Yu.* Numerical optimization techniques. — Optimization software, Inc. New-York, 1986.

5. *Moore R.E.* Interval analysis. Prentice-Hall, Englewood Clitts, 1966.

6. *Ratschek H.* Inclusion functions and global optimization. — Math. Programming, 1985, 33, N° 3.

7. *Evtushenko Yu., Potapov M.* A nondifferentiable approach to multicriteria optimization. — Lecture Notes, 255, Springer-Verlag, 1984.

8. *Евтушенко Ю.Г.* Численный метод отыскания наилучших гарантированных оценок. ЖВМ и МФ, 1972, 12, N° 1.