ЭКОНОМИКА
И МАТЕМАТИЧЕСКИЕ МЕТОДЫ

1994
Том 30, вып. 2

ЧИСЛЕННЫЙ АНАЛИЗ ПРОГРАММ НЕГЛАДКОЙ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ*

© 1994 **Нестерова С.И., Скоков В.А.**

(*Москва*)

Приводятся результаты численного анализа программ негладкой оптимизации на основе использования генератора тестовых задач безусловной негладкой оптимизации и различных подходов к решению задач этого класса.

Программы негладкой оптимизации редко встречаются. Хотя время от времени и появляются работы об отдельных методах, реализованных в программах, но достаточно углубленного сравнительного анализа этих методов не проводилось.

Используя сложившуюся классификацию, в негладкой оптимизации можно выделить задачи: безусловные, с ограничениями двусторонними, линейными и нелинейными функциональными. Для задач безусловной оптимизации применимо большинство программ, реализующих методы первого порядка: простейший метод – субградиентного спуска, ортогонального спуска, с растяжением пространства, эллипсоидов (вписанных и описанных), уровней, ε-субградиентные, а также эвристический нулевого порядка – метод деформируемого многогранника. В некоторых частных случаях возможны также специальные приемы сглаживания исходных задач, когда используется специфика их аналитической структуры (например, когда целевая функция – максимум, сумма модулей или сумма срезок гладких функций). Для решения более сложных задач, например с общими ограничениями, применялись методы эллипсоидов, ортогонального спуска, ε-субградиентный и метод уровней; для задач с двусторонними ограничениями на переменные – методы с использованием негладкой замены переменных в сочетании с r-методом (растяжения пространства) и ортогонального спуска. Линейные ограничения наиболее эффективно могут быть использованы в программах метода уровней, ортогонального спуска и в ε-субградиентных методах.

Основываясь на опыте конкурса программ ПОИСК БМ-85 для гладкой безусловной оптимизации [1], мы предлагаем двухэтапный процесс тестирования программ негладкой безусловной минимизации. На первом программы проверяются на наборе негладких тестовых задач, наиболее часто встречающихся в литературе. На втором генератор тестовых задач (ГТЗ) формирует эти задачи с переменными характеристиками. Для каждой тестируемой программы генератор создает выходную таблицу результатов. Для сравнения разных программ применяется метод, близкий к [1].

В разд. 1 приводится краткое описание тестируемых алгоритмов: метода эллипсоидов [2, 3], уровней [4], r-алгоритма [5, 6], ортогонального спуска [7], деформируемых многогранников [8]. В разд. 2 даются набор стандартных тестовых функций и генератор тестовых задач, в разд. 3 – таблицы и анализ результатов численных экспериментов для всех описанных методов.

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта – 93-012-499).

1. АЛГОРИТМИЧЕСКИЕ СХЕМЫ НЕКОТОРЫХ МЕТОДОВ НЕГЛАДКОЙ ОПТИМИЗАЦИИ

Ниже приведены схемы наиболее популярных методов негладкой минимизации. Как мы уже отметили, сложность задачи, на которую рассчитан каждый метод, различна, поэтому его описанию всегда предшествует исходная постановка задачи.

1.1. Метод эллипсоидов [2, 3]. Он применим к решению задачи негладкой минимизации в наиболее общей постановке

$$f(x) \to \min | x \in R^n,$$

$$Ax = b, \ \ b_{(l)} \leqslant x \leqslant b_{(u)},$$

$$\|x\| \leqslant r,$$

$$f^{(i)}(x) \leqslant 0, \ \ i = 1, \ldots, m,$$

где A – невырожденная $p \times n$-матрица; f и $f^{(i)}$ – выпуклые, вообще говоря, недифференцируемые функции. Предполагается, что константа r в постановке задачи известна. Приведем общую схему метода эллипсоидов.

Инициализация. Выбираем $x_0 : Ax_0 = b$. Полагаем

$$h = r / (n+1), \ \ H_0 = I - A^T [AA^T]^{-1} A.$$

Итерация k-я ($k \geqslant 0$). Выбираем направление s_k по следующему правилу: если $\|x_k\| > r$, то $s_k = x_k$, в противном случае проверяем выполнение двусторонних ограничений на значения переменных. Если нарушено по крайней мере одно из нижних ограничений, то $s_k = e_i$, где i – номер нарушенного ограничения, а e_i – i-й координатный вектор. Если же нарушено какое-либо из верхних ограничений, то $s_k = -e_i$.

Когда предыдущие проверки не выявляют нарушенных ограничений, проверяем выполнение функциональных ограничений и полагаем $s_k = g^{(i)}(x_k)$, где $g^{(i)}(x_k)$ – произвольный субградиент нарушенного ограничения в точке x_k.

Если все ограничения задачи в точке x_k выполнены, то $s_k = g(x_k)$, где $g(x_k)$ – произвольный градиент целевого функционала в точке x_k.

Полагаем

$$x_{k+1} = x_k - h \frac{H_k s_k}{\langle H_k s_k, s_k \rangle^{1/2}},$$

$$H_{k+1} = \frac{n+1}{n} H_k - \frac{H_k s_k s_k^T H_k}{\langle H_k s_k, s_k \rangle}.$$

Итерация закончена.

1.2. Метод уровней. Приведем его описание [4] для задачи минимизации негладкой функции с линейными ограничениями

$$f(x) \to \min | x \in R^n,$$

$$Ax = b, \ \ b_{(l)} \leqslant x \leqslant b_{(u)},$$

где A – невырожденная $p \times n$-матрица; f – выпуклая, вообще говоря, недифференцируемая функция. Этот метод может применяться и к задачам общего вида (см. [4]), однако в этом случае его описание существенно усложняется. Заметим также, что для метода уровней все двусторонние ограничения на переменные должны быть конечны.

Инициализация. Выбираем в качестве стартовой точки x_0 любую допустимую точку задачи. Полагаем $r_0 = +\infty$, $f_0(x) = -\infty$; итерация k-я ($k \geqslant 0$): вычисляем в точке x_k значение целевой функции $f(x_k)$ и ее градиент $g(x_k)$. Полагаем

$$f_{k+1}(x) = \max\{f_k(x), f(x_k) + \langle g(x_k), x - x_k \rangle\},$$

$$r_{k+1} = \min\{r_k, f(x_k)\},$$

$$t_{k+1} = \min\{f_{k+1}(x) | Ax = b, \; b_{(l)} \leq x \leq b_{(u)}\},$$

$$l_{k+1} = \frac{t_{k+1} + r_{k+1}}{2},$$

$$x_{k+1} = pr\{x_k | f_{k+1}(x) \leqslant l_{k+1}\}.$$

Итерация закончена.

1.3. *r*-алгоритм. Простейшая версия этого метода [3, 5] в форме [6] рассчитана на решение задачи безусловной минимизации: $f(x) \to \min | x \in R^n$, где f – недифференцируемая функция. Приведем вычислительную схему этого метода.

Инициализация. Выбираем параметры метода

$$h > 0, \; q_0 \in (0,1), \; q_1 > 1, \; \gamma \in (2,3).$$

Берем в качестве стартовой точки x_0 произвольную точку пространства R^n и полагаем $h_0 = h$, $H_0 = I$, где I – единичная $n \times n$-матрица.

Вычисляем вектор $g_k = g(x_k)$ – градиент функции $f(x)$ в точке x_k.

Итерация k-я ($k \geqslant 0$). Полагаем $y_k = x_k - h_k H_k g_k$. Если $\langle g(y_k), y_k - x_k \rangle > 0$, то $x_{k+1} = y_k$, $h_{k+1} = q_0 h_k$; в противном случае полагаем

$$x_{k+1} = x_k - (q_1)^{i_k} h_k H_k' g_k, \; h_{k+1} = (q_1)^{i_k} h_k,$$

где i_k – наименьший из номеров $i = 0, 1, ...,$ для которого выполнится неравенство: $\langle q(y_k^i), y_k - x_k \rangle > 0$, где $y_k^i = x_k - (q_1)^i h_k H_k g_k$.

Полагаем

$$H_{k+1} = H_k - (1 - \gamma^{-2}) \frac{H_k g_k g_k^T H_k}{\langle H_k g_k, g_k \rangle}.$$

Итерация закончена.

Описанную версию r-алгоритма можно применять и для решения задачи с двусторонними ограничениями на переменные: $f(x) \to \min | x \in R^n$, $b_{(l)} \leqslant x \leqslant b_{(u)}$. Для этого достаточно сделать негладкую замену переменных: $x^i = \Phi(b_l^{(i)}, b_u^{(i)}, y^{(i)})$, $i = 1, \ldots, n$, где функция $\Phi(a, b, y)$ определена следующим образом. Если $y \in [a, b]$, то $\Phi(a, b, y) = y$. Если $y < a$, то находим минимальный номер $i = 0, 1, ...,$ при котором $y \leqslant a - i(b - a)$. Если такой номер i – четный, то полагаем $\Phi(a, b, y) = 2a - i(b - a) - y$; в противном случае полагаем $\Phi(a, b, y) = y + (i + 1)(b - a)$.

Далее, если $y > b$, то находим минимальный номер $i = 0, 1, ...,$ при котором $y \geqslant b + i(b - a)$. Если такой номер i – четный, то $\Phi(a, b, y) = 2b + i(b - a) - y$; в противном случае полагаем $\Phi(a, b, y) = y - (i + 1)(b - a)$.

1.4. Метод ортогонального спуска. Рассмотрим задачу $f(x) \to \min\limits_{x \in Q}$ при

$$A_1 x = b_1, \; A_2 x \leqslant b_2, \; Q = \{x \in R^n | a \leqslant x \leqslant b_0\},$$

где A_1 – $m_0 \times n$-матрица, а A_2 – $m_1 \times n$-матрица.

Предположим, что решение этой задачи x_* известно. Тогда ее можно переформулировать как задачу отыскания решения системы линейных уравнений и неравенств

$$A_1x = b_1,\ \ A_2x - b_2 \leqslant 0,\ \ f(x) - f(x_*) \leqslant 0,\ \ x \in Q.$$

Для решения можно применить метод ортогонального спуска [7]. Обозначим через $F(x)$ максимальную невязку в неравенствах. Тогда, чтобы решить систему, достаточно найти точку $x \in Q$, в которой $A_1x = b_1$ и $F(x) = 0$. Известно, что предельный вариант метода растяжения пространства в направлении субградиента является методом спуска по ортогональным направлениям [5, 6]. Чтобы применить его к данной задаче, мы должны были бы воспользоваться штрафной функцией и при этом не учитывать специфику линейных ограничений. В методе [7] движение к решению осуществляется в подпространстве, определяемом ограничениями – равенствами. Алгоритм состоит из двух этапов. На первом с помощью ортогонального спуска за m_0 итераций строится линейное многообразие L_{n-m_0} и выявляется сечение линейного многообразия, соответствующего первым $k-1$ гиперплоскостям системы уравнений, k-й гиперплоскостью этой системы. Данный этап может быть использован для исследования систем линейных уравнений, так как позволяет исключить линейно зависимые уравнения, обнаружить несовместность системы, провести ортогонализацию системы с одновременным поиском ближайшего к начальному приближению x_0 решения системы.

На втором этапе на каждой итерации строится сечение линейного многообразия L_{n-1} (которое задается точкой x_{k-1} и ортогональным дополнением Q_1, состоящим из построенных ортогональных векторов $q_i, i = 1,\ldots,m$) гиперплоскостью, проходящей через точку $x_{k-1} + \upsilon d$ и имеющую вектор-нормаль υ. На последующих итерациях ортогональное дополнение Q_i состоит из векторов $q_i, i = 1,\ldots,m_0$, и тех векторов $q_i, i = m_0 + 1,\ldots,r$, которые составляют с вектором υ тупой угол. По ходу итерационного процесса меняется состав и количество взаимоортогональных векторов q (за исключением первых m_0 векторов) и за счет ошибок округления все сильнее нарушается ортогональность векторов с номерами $m_0 + 1,\ldots,r$. Для устранения ошибок округления применяется специальная процедура восстановления. Она состоит в следующем.

Если система не содержит линейных уравнений ($m = 0$), то восстановление заключается в периодическом забывании векторов $q_i, i = 1,\ldots,r$; в противном случае восстановление происходит за счет периодического забывания векторов $q_i, i = m_0 + 1,\ldots,r$ и проектирования текущей точки x на L_{n-m_0}. Для этого всегда хранится точка, полученная после выполнения первого этапа проектирования. Восстановление проводится при достижении очередного рекордного значения по функции.

Вектор υ – ортонормированный вектор u, который вычисляется по одной из следующих формул:

$$u = -\varphi'_k(x)\varphi_k(x)/\|\varphi'_k(x)\|^2,\ \ k = 1,\ldots,m_0,$$

где $\varphi_k(x)$ – k-е линейное уравнение в ограничениях,

$$u = \begin{cases} u_1, & \|u_1\| \geqslant \|u_2\| \\ u_2, & \text{иначе.} \end{cases}$$

Здесь $u_1 = -F'(x)F(x)/\|F'(x)\|^2$,

$$u_2 = \begin{cases} 0, & a^i \leqslant x^i \leqslant b_0^i, \\ a^i - x^i, & a^i > x^i, \\ b_0^i - x^i, & b_0^i < x^i,\ \ i = 1,\ldots,n. \end{cases}$$

В процессе проектирования строится проекция вектора υ на ортогональное дополнение Q_1 к L_{n-1}: $s = \sum_{i=1}^{r}(\upsilon, q_i)q_i$.

Если $||s|| = 0$, то $\bar{x} = x + \upsilon d (d = ||u||)$ и новый ортогональный вектор $q_r = \upsilon$. Иначе вычисляется косинус угла между υ и s: $c = (s, \upsilon)/||s||$, и если $0 < c < 1$, то рассчитывается новый ортогональный вектор $q = (\upsilon - s)/||\upsilon - s||$ и точка $\bar{x} = x + q_r h$, где $h = d / (\upsilon, q_r)$.

1.5. Метод деформируемых многогранников [8]. Он широко применяется для решения безусловных инженерных задач дифференцируемой оптимизации, но может использоваться и в негладком случае. Опишем одну из его модификаций, которая была нами реализована в программе.

Итак, пусть требуется найти минимум $f(x)$, $x \in R^n$. Построим последовательность точек $x_i \in R^n$. Будем считать, что на итерации k для сохраняемых векторов $x_i \in R^n, i = 1,\dots, p$ (для симплексных методов $p = n + 1$, методов комплексов $p > n + 1$) уже вычислены значения целевой функции $f(x)$. Пусть x_h и x_l – вершины с наибольшим и наименьшим значениями функции $f(x_i), i = 1,\dots, p$, на k-й итерации. Определим, кроме того, центр многогранника (x_{p+1}), образованного вершинами $x_i, i = 1,\dots, p$. Для поиска лучшего значения целевой функции обычно используются операции отражения, растяжения, сжатия и редукции.

Отражение x_i относительно x_{p+1}:

$$x_{p+2} = x_{p+1} + \alpha(x_{p+1} - x_h).$$

Растяжение вдоль $(x_{p+2} - x_{p+1})$, если

$$f(x_{p+2}) \leqslant f(x_i), i = 1,\dots, p: x_{p+3} = x_{p+1} + \gamma(x_{p+2} - x_{p+1}).$$

Сжатие вдоль $(x_h - x_{p+1})$, если

$$f(x_{p+2}) > f(x_i), \;\; i \neq h: \;\; x_{p+4} = x_{p+1} + \beta(x_h - x_{p+1}).$$

Редукция многогранника, если

$$f(x_{p+2}) > f(x_h), \;\; x_i = x_l + \delta(x_i - x_l), \;\; i = 1,\dots, p.$$

Если в отраженной точке функция имеет значение не меньше минимального, но меньше наихудшего, то отраженная точка вводится вместо точки с наихудшим значением целевой функции.

Выбор центра многогранника определяет направление спуска. В методе Нелдера–Мида центр выбирается по $(p - 1)$ наилучшим точкам

$$x_{p+1}^{(b)} = \frac{1}{p-1} \sum_{i \in I} x_i, \;\; I = \{i, i \neq h, i = 1,\dots, p\}.$$

Однако в этой формуле не учитываются значения фукнции, поэтому была предложена следующая модификация вычисления центра многогранника

$$x_{p+1}^{(c)} = \frac{\sum_{i \in I} f(x_i) x_i}{\sum_{i \in I} f(x_i)}.$$

Для $f(x_i) \leqslant 0 \;\; \forall x_i$ эта формула дает центр гравитации многогранника. Но если в некоторых точках функция принимает значения разных знаков, то результат будет иным. Введем небольшое изменение

$$x_{p+1}^{(c)} = \frac{\sum_{i \in I} (f(x_i) - C) x_i}{\sum_{i \in I} (f(x_i) - C)},$$

где C – достаточно большое число, например, равное $2f(x_h)$.

2. ТЕСТОВЫЕ ЗАДАЧИ НЕГЛАДКОЙ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ И ТЕСТИРУЮЩАЯ ПРОГРАММА-ГЕНЕРАТОР

Для проведения численных экспериментов с программами негладкой оптимизации был составлен набор тестовых задач, состоящий из двух групп. В первую включены наиболее известные задачи негладкой оптимизации небольшой размерности, во вторую – задачи безусловной негладкой оптимизации с переменной размерностью, в том числе с переменной обусловленностью.

При описании тестовых задач используются следующие обозначения: n – число переменных, x_0 – начальное, x_* – оптимальное приближение, f_* – оптимальное значение.

1. *CB*2 [9] – безусловная задача на минимум максимума из степенной (4-й степени), квадратичной и экспоненциальной функции; $n = 2$, $x_0 = (1;-0,1)$; $x_* = (1,1390;0,8996)$, $f_* = 1,952234$.

2. *CB3* [9] отличается от предыдущей задачи видом степенной функции; $n = 2$, $x_0 = (2,2)$; $x_* = (1,1)$; $f_* = 2$.

3. *Dem* [9] – безусловная на минимум максимума двух линейных и одной квадратичной функции; $n = 2$, $x_0 = (1;1)$; $x_* = (0;-3)$; $f_* = -3$.

4. *QL* [9] – безусловная задача на минимум максимума трех квадратичных функций; $n = 2$, $x_0 = (-1;5)$; $x_* = (1,2;2,4)$; $f_* = 7,2$.

5. *LQ* [9] – безусловная задача на минимум максимума из линейной и квадратичной функции; $n = 2$, $x_0 = (-0,5;-0,5)$, $x_* = (2^{(-0,5)}, 2^{(-0,5)})$, $f_* = -2^{(-0,5)}$.

6. *MIFL*1 [9] – безусловная задача на минимум суммы линейной функции и срезки квадратичной функции; $n = 2$, $x_0 = (0,8; 0,6)$, $x_* = (1; 0)$; $f_* = -2$.

7. *MIFL*2 [9] – безусловная задача на минимум суммы линейной, квадратичной и модуля от квадратичной функции; $n = 2$, $x_0 = (-1;1)$, $x_* = (1;0)$, $f_* = -1$.

8. Задача Розена–Сузуки [9, 10] известна как тестовая условной оптимизации. Со штрафным коэффициентом, равным 10, записывается негладкая штрафная функция как максимум четырех квадратичных функций; $n = 4$, $x_0 = (0;0;0;0)$, $x_* = (0;1;2;-1)$; $f_* = -44,0$.

9. *Shor*1 [3, 5] – тестовая негладкая функция, представляющая собой максимум из 10 квадратичных функций с диагональными матрицами; $n = 5$, $x_0 = (0;0;0;0;1)$, $f_* = 22,60016$.

Следующие задачи имеют переменную размерность:

10. *MAXQ* [9] – безусловная задача на минимум максимума из квадрата координаты по всем координатам текущей точки; n – любое, $x_0 = (i, i = 1,\dots,[n/2]; \ -i, i = [n/2]+ +1,\dots,n)$, $x_* = (0,\dots,0)$, $f_* = 0$.

11. *MAXL* [9] – безусловная задача на минимум максимума модуля координаты по всем координатам текущей точки; n – любое, x_0, x_*, f_* – такие же, как в 10.

12. *GOFN* [9] – безусловная задача на минимум разности максимума координаты по всем координатам текущей точки, умноженного на n, и суммой всех координат этой точки; n – любое, $x_0 = \{i-(n+1)/2, i = 1,\dots,n\}$, $x_* = (0,\dots,0)$, $f_* = 0$.

13. *HILB* [3] – безусловная задача на минимум квадратичной функции с матрицей Гильберта. Квадратичная функция плохо обусловлена; n – любое, $x_0 = (0,\dots,0)$, $x_* = (1,\dots,1)$, $f_* = 0$.

14. *MXC* [3] – безусловная задача минимизации максимума модуля k-х координат, умноженных на k^3, по всем $k = 1, \dots, n$; n – любое, $x_0 = (t, t/2, \dots, t/n)$, $t = 10$, $x_* = (0,\dots,0)$, $f_* = 0$.

15. *SMD* [3] – безусловная задача минимизации суммы (по k) модулей k-х коор-

динат текущей точки в степени k, умноженных на k^3, $k = 1, \ldots, n$; n – любое, $x_0 = (t, t/2, \ldots, t/n)$, $t = 10$, $x_* = (0, \ldots, 0)$, $f_* = 0$.

16. *MXN* [3] – безусловная задача минимизации максимума линейной комбинации норм векторов; n – любое, $x_0 = (t, t/2, \ldots, t/n)$, $t = 10$, $x_* = (0, \ldots, 0)$, $f_* = 0$.

17. *PLN* [3, 11] – безусловная задача минимизации суммы модулей, наиболее трудоемкая. Решается задача приближения данных полиномом (n – 1)-й степени; n – любое, $x_0 = (0, \ldots, 0)$, $x_* = (1/n, \ldots, 1/n)$, $f_* = 0$.

На базе этих задач составлен генератор, позволяющий оценивать программу минимизации по надежности и эффективности. Выходная форма генератора включает информацию о тестируемом методе, список решаемых задач и таблицу результатов счета. По каждой задаче приводятся две строки результатов. Первая строка соответствует моменту останова программы, вторая – моменту достижения заданной погрешности решения. В каждой строке приводится размерность задачи, число вычислений функции и градиента, погрешность по функции, норма градиента, погрешность по переменным, время решения задачи и чистое время работы самого метода. Последнее позволяет оценивать методы, имеющие разную трудоемкость одной итерации.

3. ЧИСЛЕННЫЙ АНАЛИЗ ПОДПРОГРАММ НЕГЛАДКОЙ ОПТИМИЗАЦИИ

В табл. 1–5 приведены результаты счета задач 1–17 по методам: уровней [4], ортогонального спуска с использованием оптимального значения f_* (МОС) [7], Шора (r-алгоритму) [5, 6] (МШ), описанных эллипсоидов [2] (МЭ) и деформируемых многогранников [8] (МДМ). Все программы, за исключением метода уровней, являются собственными разработками и включены в пакет анализа оптимизационных экономических моделей на персональном компьютере (ПАОЭМ–НЛП ПК), созданный в ЦЭМИ [12]. Программа метода уровней любезно предоставлена нам авторами [4]. Она написана для исследовательских целей. При решении вспомогательных задач линейного и квадратичного программирования используют известную программу *ZQPCVX*, автором которой является Powell M.J.D. (см. [4]). Такой экономный подход сильно искажает картину сравнения методов по времени счета, но дает достоверные сведения о надежности и эффективности метода по числу вычислений функции и субградиента. Расчеты проводились в основном на *PC IBM*/286 без сопроцессора (частота 12 мГц). Всего решена 41 задача.

Для оценки методов применялись два критерия: надежность и эффективность. Надежность вычислялась отношением числа решенных с заданной погрешностью задач к числу решаемых. Эффективность метода измерялась временем решения задач и числом вычислений функции (*nf*) и ее субградиента (*ng*). Для задач с трудоемкими расчетами функции и субградиента время счета является хорошей оценкой эффективности метода. Наряду с этим его трудоемкость на малых задачах может исказить оценку эффективности, поэтому в таких случаях методы лучше оценивать по числу вычислений функции и субградиента.

По результатам табл. 1 самым надежным является метод уравнений – 97%, хотя фактически он дает 100%, так как одна из задач не была решена с заданной погрешностью только из-за отсутствия необходимого объема оперативной памяти компьютера. Методы МОС и МШ обеспечивают приблизительно одинаковую надежность (92–95%). По ним для некоторых задач не достигнута заданная погрешность. МЭ имеет надежность менее 76% и МДМ хорошо работает лишь на малоразмерных задачах при поиске грубых приближений к оптимальному решению (по результатам решаемых задач надежность МДМ меньше 47%).

Для оценки эффективности метода используем суммарные затраты по группам решенных задач. В табл. 2 приведены суммарные затраты *nf, ng* и число решенных задач по группам. В рассматриваемых методах в каждой точке функция и субградиент

Таблица 1

Надежность метода
(число решенных задач из 41 с погрешностью 0,0001)

Число задач (размерность)	Метод				
	МУ	МШ	МОС	МЭ	МДМ
7 (2)	7	7	6	7	5
2 (4–5)	2	2	2	2	0
8 (5)	8	8	7	8	3
8 (10)	8	7	8	4	–
8 (15)	7	8	8	4	–
8 (50)	8	6	8	–	–
Средняя надежность, %	40/41	38/41	39/41	25/33	8/17
	97	92,7	95,1	< 75,8	< 47,0

Таблица 2

Эффективность по (*nf,ng*)

Число задач (размерность)	Метод			
	МУ	МШ	МОС	МЭ
7 (2)	154/7	385/7	125/6	579/7
2 (4–5)	74/7	268/2	231/2	936/2
8 (5)	258/8	1090/8	202/7	4156/8
8 (10)	346/8	1720/7	326/8	2361/8
8 (15)	443/7	3448/8	599/8	3009/4
8 (50)	955/8	5435/6	2456/8	–
Суммарные затраты	2230	12346	3944	11041
Число решенных задач	40	38	39	25

Таблица 3

Эффективность по времени

Число задач (размерность)	Метод			
	МУ	МШ	МОС	МЭ
7 (2)	96,1/7	7,3/7	3,0/6	8,4/7
2 (4–5)	247,5/2	15,7/2	22,7/2	57,3/2
8 (5)	535,8/8	287,0/8	155,7/7	1364,5/8
8 (10)	2031,6/8	735,5/7	223,4/8	293,3/4
8 (15)	5665,2/7	1843,5/8	1197,3/8	797,9/4
8 (50)	7191,0/8	734,0/6	287,3/8	–

вычислялись одновременно, поэтому в табл. 2 приводится одна цифра вычисления функции и субградиента.

По числу вычислений функции и субградиента самыми эффективными являются методы: уровней, ортогонального спуска, Шора и эллипсоидов. Обратим внимание на то, что в схеме МОС существенно знание оптимального значения функции. Установленное ранжирование будет верным, если вычисление функции и субградиента превышают по затратам трудоемкость итерации. Для рассматриваемых задач это соотношение не выполняется (см. табл. 3).

Таблица 4

Средняя эффективность по (*nf*/*ng*), %

Число задач (размерность)	Метод			
	МУ	МШ	МОС	МЭ
7 (2)	100	40	75,9	26,6
2 (4–5)	100	27,6	32	8
8 (5)	100	23,7	63,1	6,2
8 (15)	94,2	15,9	100	7,6
8 (15)	41,5	17,4	100	8,5
8 (50)	100	16,6	38,9	–
Средняя эффективность	89,3	23,5	68,3	11,4

Таблица 5

Средняя эффективность по времени, %

Число задач (размерность)	Метод			
	МУ	МШ	МОС	МЭ
7 (2)	7,6	100	27,5	86,9
2 (4–5)	6,3	100	69,2	27,4
8 (5)	53,6	100	95	21
8 (10)	11	26,7	100	21,5
8 (15)	19,9	64,9	100	40,5
8 (50)	4	14,4	100	–
Средняя эффективность	17,1	67,7	82,0	39,5

В табл. 3 в нижней строке время решения приведено для другого компьютера, быстродействие которого в 25 раз больше, чем у *IBM PC*/286. Очевидно, что числа этой строки, приведенные к масштабу *IBM PC*/286, будут подавлять в сумме все другие слагаемые, но они не нарушают общей закономерности. Наименьшая трудоемкость оказывается у МОС, а наибольшая у МУ, где на каждой итерации требуется решить по одной задаче линейного и квадратичного программирования. Системы ограничений этих задач на двух соседних итерациях отличаются одним ограничением. Эту специфику метода, вообще говоря, можно использовать для сокращения трудоемкости, но в данном эксперименте такая возможность не была реализована.

В табл. 2, 3 трудно привести ранжирование методов из-за разного числа решенных задач, поэтому дополним эти таблицы. За каждую нерешенную задачу введем штраф [10], равный самым большим затратам на ее решение среди всех методов, соответственно по числу вычислений функции и субградиента или по времени. Затем в каждой i-й строке найдем числа $a_i = 100\min\{j = 1,\dots,5| a_{ij}\}$, $i = 1, \dots, 6$, и положим $a_{ij} += a_i / a_{ij}$.

Анализ приведенных результатов свидетельствует о том, что лучшая средняя эффективность по числу вычислений функции субградиента у МУ (89,3%) и у МОС (68,3%). Если трудоемкость одной итерации того или иного метода выше затрат на вычисление функций субградиента, то ранжирование методов меняется. По времени счета лучшая средняя эффективность у МОС (82%) и метода Шора (67,7%). Следует иметь в виду, что в МОС используется оптимальное значение функции, и это резко снижает его практическое применение. В МУ, как уже отмечалось, есть резерв для его усовершенствования.

Кроме указанного, для оценки программ оптимизации существенной характеристикой является объем рабочей памяти. В МУ он зависит от числа итераций, но можно использовать и фиксированный объем памяти. В МОС нужно хранить систему ортогональных векторов, поэтому максимально необходимая память не превышает $n \times n$. Однако в силу принятого в методе правила отбора ортогональных векторов их число в системе было, как правило, меньше n. В МШ и МЭ рабочая память должна быть порядка $n \times n/2$.

ЛИТЕРАТУРА

1. *Надольская Т.А., Нестеров Ю.Е., Пурмаль Е.И., Скоков В.А.* Об итогах Всесоюзного конкурса программ "Поиск БМ-85" // Экономика и мат. методы. 1989. Т. XXIV. Вып. 4.
2. *Немировский А.С., Юдин Д.Б.* Сложность задач и эффективность методов оптимизации. М., 1979.
3. *Нестеров Ю.Е., Пурмаль Е.И.* Анализ эффективности методов негладкой оптимизации. М.: ЦЭМИ АН СССР, 1984.
4. *Lemarechal C., Nemirovski A., Nesterov Yu.* New Variants of Bundle Methods // Rapports de recherche N 1508. Paris, 1991.
5. *Шор Н.З.* Методы минимизации недифференцируемых функций и их приложения. Киев, 1979.
6. *Скоков В.А.* Замечание к методам минимизации, использующим операцию растяжения пространства // Кибернетика. 1974. № 4.
7. *Щепакин М.Б.* О методе ортогонального спуска // Кибернетика. 1987. № 1.
8. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. М., 1985.
9. *Lemarechal C.* Numerical Experiments in Nonsmooth Optimization // Ed. *Nurminski E.A.* // Progress in Nondifferentiable Optimization. Laxenburg (Austria), 1982.
10. *Hock W., Schittkowski K.* Test Examples for Nonlinear Programming Codes // Lecture Notes in Economics and mathematical systems. 1981. N 187.
11. *Поляк Б.Т.* Введение в оптимизацию. М., 1983.
12. *Ким К.В., Малков У.Х., Скоков В.А., Черкасский Б.В.* Пакет анализа оптимизационных экономических моделей // Пакеты прикладных программ. Методы оптимизации. М., 1984.

Поступила в редакцию
20 I 1993