ІНСТИТУТ КІБЕРНЕТИКИ
імені В.М. Глушкова НАН УКРАЇНИ

ІНСТИТУТ МАТЕМАТИКИ ТА ІНФОРМАТИКИ
імені Володимира Андрунакієвича МОЛДОВИ
ІНСТИТУТ СИСТЕМ УПРАВЛІННЯ
НАН АЗЕРБАЙДЖАНУ

Матеріали
7–ї міжнародної наукової конференції
МОДЕЛЮВАННЯ І ОПТИМІЗАЦІЯ У ТРАНСПОРТІ ТА ЛОГІСТИЦІ

присвяченої
85–річчю з дня народження
академіка НАН України Наума Зуселевича Шора
21 – 25 березня 2022 року

Kyiv–Chisinau–Baku–2022

V.M. GLUSHKOV INSTITUTE OF CYBERNETICS
OF THE NATIONAL ACADEMY OF SCIENCES OF UKRAINE
VLADIMIR ANDRUNACHIEVICI INSTITUTE OF
MATHEMATICS AND COMPUTER SCIENCE OF MOLDOVA
INSTITUTE OF CONTROL SYSTEMS
OF AZERBAIJAN NATIONAL ACADEMY OF SCIENCES

Proceedings

of the 7–th International Scientific Conference

MODELING AND OPTIMIZATION IN TRANSPORT AND LOGISTICS

Dedicated to

85th Anniversary of the Birth of

Academician of the NASU Naum Z. Shor

March 21 – 25, 2022

Kyiv–Chisinau–Baku–2022

НОВЫЕ ВАРИАНТЫ МЕТОДОВ ЭЛЛИПСОИДОВ[1]

П.И. СТЕЦЮК,
Институт кибернетики имени В.М. Глушкова
НАН Украины, Киев, Украина
stetsyukp@gmail.com
А. ФИШЕР,
Институт вычислительной математики
технического университета Дрездена, Германия
Andreas.Fischer@tu-dresden.de
О.Н. ХОМЯК,
Институт кибернетики имени В.М. Глушкова
НАН Украины, Киев, Украина
khomiak.olha@gmail.com

*Для минимизации выпуклой функции от n переменных предложены алгоритмы **em80b** и **em99b**. Алгоритм **em80b** есть метод эллипсоидов в B-форме (Хачиян, 1980), где $n \times n$-матрица B определяет замену переменных. С его помощью можно найти приближение к точке минимума с очень высокой точностью, чего нельзя сделать с помощью метода эллипсоидов в H-форме, где корректируется симметрическая матрица $H = BB^T$. Алгоритм **em99b** является B-формой метода эллипсоидов для аппроксимации множества, полученного в результате пересечения n-мерного шара и набора определяемых гиперплоскостей (Стецюк, 1999). Его предельный вариант сходится к точке минимума не более чем за n итераций.*

Ключевые слова: *выпуклая функция, метод эллипсоидов, оператор растяжения пространства, H- и B-формы методов с преобразованием пространства.*

---

[1] Работа поддержана VolkswagenFoundation (грант № 97775)

Задача. Пусть $f(x)$ – выпуклая функция, $x \in \mathbb{R}^n$, $g(x)$ – ее субградиент $g(x) \in \mathbb{R}^n$, минимальное значение $f^* = f(x^*)$. Предположим, что точка минимума x^* − единственная. Требуется найти такую точку x_ε^*, для которой $f(x_\varepsilon^*) - f^* \le \varepsilon$, где $\varepsilon > 0$ – заданное достаточно малое число.

Для нахождения точки x_ε^* рассмотрим два новых алгоритма метода эллипсоидов (**e**llipsoid **m**ethod). Первый алгоритм будем называть алгоритмом **em80b**, а второй – алгоритмом **em99b**. Здесь символ «**b**» означает, что оба алгоритма будут представлены в **B**-форме, где корректируется $n \times n$-матрица B, которая определяет замену переменных $x = By$, где $y = Ax$ – образ точки x в преобразованном пространстве переменных, $A = B^{-1}$.

Алгоритм **em80b** – метод эллипсоидов, который предложили независимо Д.Б. Юдин и А.С. Немировский [1] и Н.З. Шор [2]. Он использует ***B***-форму метода эллипсоидов, которую для задачи линейного программирования использовал Л.Г. Хачиян в 1980 году [3]. Она построена подобно тому, как Н.З. Шор построил B-форму своего варианта метода эллипсоидов в статье [2]. Ее вычислительная реализация представлена в программе **emshor** [4, 5].

В 1981 году в статье [6] М. Грьотшель, Л. Ловас и А. Схрейвер рассматривали метод эллипсоидов в H-форме, где корректируется симметрическая матрица $H = BB^T$. Условимся этот вариант метода эллипсоидов называть алгоритмом **em81h**. На примере тестовых задач покажем, что с его помощью нельзя найти точку минимума с очень высокой точностью, что позволяет сделать алгоритм **em80b**.

Алгоритм **em99b** является B-формой метода эллипсоидов, описанных вокруг пересечения шара и набора определяемых гиперплоскостей [7], на каждой его итерации объем локализации точки x^* уменьшается в α раз, где $\alpha > 1$ – коэффициент растяжения пространства. Если $\alpha = \infty$, то количество итераций алгоритма **em99b** совпадает с количеством итераций предельного варианта метода, предложенного Н.З. Шором в 1970 году [8].

Алгоритм **em99b** использует значение f^* и число $m\left(m \ge 1\right)$, такие, что субградиент $g\left(x\right)$ удовлетворяет условию

$$\left(x - x^*\right)^T g\left(x\right) = m\left(f\left(x\right) - f^*\right),\ \forall x \in \mathbb{R}^n. \tag{1}$$

Условию (1) удовлетворяют дифференцируемые однородные (с показателем σ) выпуклые функции ($m = \sigma$), квадратичные выпуклые функции ($m = 2$), некоторые классы кусочно-линейных выпуклых функций с острым минимумом ($m = 1$).

Алгоритм **em80b**

(B-форма метода эллипсоидов, Хачиян, 1980)

Шаг 0. Выбираем точку $x_0 \in \mathbb{R}^n$ и радиус r_0 такими, чтобы $\left\|x_0 - x^*\right\| \le r_0$. Выбираем $\varepsilon > 0$, полагаем $B_0 := I_n$ и $k := 0$.

Шаг 1. Вычислим $f\left(x_k\right)$ и $g_k = g\left(x_k\right)$. Если $\left\|B_k^\top g_k\right\| \le \varepsilon / r_0$, то ОСТАНОВ: $k^* := k$, $x_\varepsilon^* := x_k$.

Шаг 2. Вычисляем $x_{k+1} := x_k - \dfrac{r_0}{n+1} B_k \xi_k$, где $\xi_k := \dfrac{B_k^\top g_k}{\left\|B_k^\top g_k\right\|}$.

Шаг 3. Обновляем $B_{k+1} := \dfrac{n}{\sqrt{n^2 - 1}}\left(B_k + \left(\sqrt{\dfrac{n-1}{n+1}} - 1\right)\left(B_k \xi_k\right)\xi_k^\top\right)$.

Шаг 4. Устанавливаем $k := k + 1$ и переходим к Шагу 1.

Комментарий. На каждой k-ой итерации алгоритма $n \times n$-матрица B_k определяет замену переменных $x = B_k y$, где $y = A_k x$ – образ точки x в преобразованном пространстве переменных, $A_k = B_k^{-1}$. На каждой итерации реализуется субградиентный спуск с постоянным шагом $h_0 = r_0 / \left(n+1\right)$ для выпуклой функции $\varphi\left(y\right) = f\left(B_k y\right)$ в преобразованном пространстве переменных.

Действительно, если обе части формулы $x_{k+1} = x_k - h_0 B_k \xi_k$ умножить слева на матрицу A_k, то получим

$$y_{k+1} = A_k x_{k+1} = A_k x_k - h_0 \xi_k = y_k - h_0 \frac{B_k^T g_f(x_k)}{\left\| B_k^T g_f(x_k) \right\|} = y_k - h_0 \frac{g_\varphi(y_k)}{\left\| g_\varphi(y_k) \right\|},$$

где $g_\varphi(y_k) = B_k^T g_f(x_k)$ есть субградиентом функции $\varphi(y) = f(B_k y)$ в точке $y_k = A_k x_k$ пространства переменных $y = A_k x$.

Очевидно, что обновление B-матрицы (см. Шаг 3) требует $O(n^2)$ операций. Это обусловлено использованием оператора растяжения пространства $R_\alpha(\xi): \mathbb{R}^n \to \mathbb{R}^n$, который определен как

$$R_\alpha(\xi) := I_n + (\alpha - 1)\xi\xi^\top, \qquad (2)$$

где $\xi \in \mathbb{R}^n$, $\|\xi\| = 1$ – направление растяжения, а $I_n \in \mathbb{R}^{n \times n}$ – единичная матрица.

Теорема 1. Последовательность точек $\{x_k\}_{k=0}^{k^*}$, которую генерирует алгоритм **em80b**, удовлетворяет неравенствам

$$\left\| A_k (x_k - x^*) \right\| \le r_0, \quad A_k = B_k^{-1}, \quad k = 0, 1, 2, \ldots, k^*. \qquad (3)$$

Скорость сходимости алгоритма **em80b** определяется отношением объема эллипсоида $E_k = \left\{ x : \left\| A_k (x_k - x) \right\| \le r_0 \right\}$, локализующего x^* на k-й итерации, к объему эллипсоида $E_{k-1} = \left\{ x : \left\| A_{k-1} (x_{k-1} - x) \right\| \le r_0 \right\}$, локализующего x^* на $(k-1)$-й итерации, $1 \le k \le k^*$.

Теорема 2. На каждой итерации k, где $1 \le k \le k^*$, отношение объемов эллипсоидов E_k и E_{k-1} есть величина постоянная и равная

$$q_n = \frac{vol(E_k)}{vol(E_{k-1})} = \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2 - 1}} \right)^n < \exp\left\{ -\frac{1}{2n} \right\} < 1. \qquad (4)$$

Для итерации k^* имеем $x_\varepsilon^* = x_{k^*}$, так как $f(x_{k^*}) - f^* \le \varepsilon$.

Алгоритм **em80b** можно представить в H-форме, где корректируется симметрическая матрица $H = BB^T$. H -форму метода эллипсоидов будем называть алгоритмом **em81h**. Этот вариант метода эллипсоидов использован в [6] (авторы – Grötschel M., Lovász L., Schrijver A.) для обоснования полиномиальных алгоритмов решения комбинаторных задач (GLS, 1981).

Алгоритм **em81h**

(H -форма метода эллипсоидов, GLS, 1981).

Шаг 0. Выбираем точку $x_0 \in \mathbb{R}^n$ и радиус r_0 такими, чтобы $\left\| x_0 - x^* \right\| \le r_0$. Выбираем $\varepsilon > 0$, полагаем $H_0 := I_n$ и $k := 0$.

Шаг 1. Вычислим $f\left(x_k\right)$ и $g_k = g\left(x_k\right)$. Если $\sqrt{g_k^T H_k g_k} \le \varepsilon / r_0$, то ОСТАНОВ: $k^* := k$, $x_\varepsilon^* := x_k$.

Шаг 2. Вычисляем $x_{k+1} := x_k - \dfrac{r_0}{n+1} H_k \xi_k$, где $\xi_k := \dfrac{g_k}{\sqrt{g_k^T H_k g_k}}$.

Шаг 3. Обновляем $H_{k+1} := \dfrac{n^2}{n^2 - 1}\left(H_k - \dfrac{2}{n+1} H_k \xi_k \xi_k^\top H_k \right)$.

Шаг 4. Устанавливаем $k := k + 1$ и переходим к Шагу 1.

С помощью алгоритма **em80b** можно найти точку x_ε^* с очень высокой точностью $\varepsilon \approx 10^{-15} \div 10^{-10}$, чего нельзя сделать с помощью алгоритма **em81h**. Это подтверждают вычислительные эксперименты для двух выпуклых кусочно-линейных функций

$$f_1(x) = \sum_{i=1}^{n} i\left|x_i - 1\right| \quad \text{и} \quad f_2(x) = \sum_{i=1}^{n} 10^{i-1}\left|x_i - 1\right|,$$

для которых $f_1^* = f_1(x^*) = f_2^* = f_2(x^*) = 0$, $x^* = \left(1,1,\ldots,1\right)^T$. Функция $f_2\left(x\right)$ является овражной, степень вытянутости ее поверхностей уровня для заданного n равна 10^{n-1} – отношение максимального коэфициента при $\left|x_n - 1\right|$ к минимальному при $\left|x_1 - 1\right|$.

Вычислительные эксперименты проводились на персональном компьютере с процессором AMD Ryzen 5 4500U 2.38 GHz, 16 GB в системе Windows 10 с помощью GNU Octave версии 6.2.0. Результаты при различных значениях ε представлены в таблицах 1 и 2. Здесь k^* – количество итераций, затраченное на поиск точки x_ε^*, $f_1\left(x_\varepsilon^*\right)$ и $f_2\left(x_\varepsilon^*\right)$ – значения функций $f_1(x)$ и $f_2(x)$ на итерации k^*, $\left\|B_k^*\right\|$ – евклидова норма матрицы B на итерации k^*.

ТАБЛИЦА 1. Сравнение алгоритмов **em80b** и **em81h** для $n = 5$

	Алгоритм **em80b**			Алгоритм **em81h**		
ε	k^*	$f_1\left(x_\varepsilon^*\right)$	$\left\|x_\varepsilon^* - x^*\right\|$	k^*	$f_1\left(x_\varepsilon^*\right)$	$\left\|x_\varepsilon^* - x^*\right\|$
1.0e-01	268	5.0e-03	1.3e-03	268	5.0e-03	1.3e-03
1.0e-02	422	1.3e-04	4.7e-05	422	1.2e-04	4.8e-05
1.0e-03	**534**	**1.3e-04**	**4.8e-05**	**470**	**9.2e-04**	**1.5e-04**
1.0e-04	**633**	**1.6e-05**	**3.4e-06**	**470**	**9.2e-04**	**1.5e-04**
1.0e-05	**713**	**2.1e-07**	**4.7e-08**	**470**	**9.2e-04**	**1.5e-04**
ε	k^*	$f_2\left(x_\varepsilon^*\right)$	$\left\|x_\varepsilon^* - x^*\right\|$	k^*	$f_2\left(x_\varepsilon^*\right)$	$\left\|x_\varepsilon^* - x^*\right\|$
1.0e+01	251	2.1e-01	5.0e-02	251	2.1e-01	5.0e-02
1.0e+00	375	2.4e-02	4.9e-03	362	1.6e-01	3.9e-02
1.0e-01	**488**	**1.4e-02**	**6.0e-03**	**406**	**3.1e-01**	**6.4e-02**
1.0e-02	**583**	**1.7e-03**	**1.5e-04**	**406**	**3.1e-01**	**6.4e-02**
1.0e-03	**718**	**1.5e-05**	**3.8e-06**	**406**	**3.1e-01**	**6.4e-02**

Из табл. 1 видно, что алгоритм **em81h** не способен решить задачу при малых значениях $\varepsilon \approx 10^{-5} \div 10^{-2}$ (выделено жирным).

В табл. 2 показано, что *B*-форма метода эллипсоидов является устойчивой для нахождения приближения к точке минимума выпуклой функции с очень высокой точностью.

ТАБЛИЦА 2. Сравнение алгоритмов **em80b** и **emshor** для *n=25*

	Алгоритм **em80b**			Алгоритм **emshor**		
ε	k^*	$f_1\left(x_\varepsilon^*\right)$	$\left\|B_k^*\right\|$	k^*	$f_1\left(x_\varepsilon^*\right)$	$\left\|B_k^*\right\|$

10^{-3}	16215	3.5e-05	3.1e-05	16215	3.5e-05	7.2e-11
10^{-4}	19060	8.8e-08	3.1e-06	19060	8.8e-08	7.3e-13
10^{-5}	21887	3.8e-07	3.4e-07	22033	3.8e-07	6.5e-15
10^{-7}	27701	3.8e-09	3.2e-09	27731	3.9e-09	7.0e-19
10^{-10}	36387	1.5e-12	3.0e-12	36453	1.7e-12	6.0e-25
10^{-13}	41845	4.4e-16	1.9e-14	42000	2.9e-15	3.8e-29
10^{-14}	41912	0.0e+00	1.8e-14	42106	0.0e+00	3.3e-29
10^{-15}	41912	0.0e+00	1.8e-14	42106	0.0e+00	3.3e-29

Вычислительные эксперименты показывают, что при $n = 10 \div 25$ алгоритмы **em80b** и **emshor** можно успешно применять для нахождения точки x_ε^* при очень малых $\varepsilon \approx 10^{-10} \div 10^{-15}$. Так, для $n = 25$ время решения задачи из табл. 2 обоими методами при максимальной точности составило менее 4 секунд. Из таблицы видим, что требуется $4.6n^2$ итераций, чтобы в десять раз улучшить отклонение найденного рекордного значения функции $f(x)$ от ее оптимального значения f^* (результат согласуется с теоремой 2).

Следует заметить, что на современных компьютерах с помощью методов эллипсоидов в *B*-форме можно решать практические задачи оптимизации при количестве переменных $n = 50 \div 100$, если вычисление значения функции и ее субградиента есть малозатратной по времени процедурой. Так, например, при $\varepsilon = 10^{-15}$ вычислительные затраты «на алгоритмические операции» программы **em80b** при минимизации «неовражной» функции $f_1(x_{itn})$ для 50 переменных составили 9.95 сек. (162034 итераций), для 75 переменных – 25.69 сек. (358623 итераций), для 100 переменных – 107.79 сек. (630570 итераций).

Ниже для нахождения точки x_ε^* опишем алгоритм **em99b**, который является *B*-формой метода эллипсоидов, описанных вокруг пересечения шара и набора гиперплоскостей. Алгоритм использует знание значения f^*; заданное число $m\left(m\geq 1\right)$ – такое, что субградиент $g\left(x\right)$ удовлетворяет условию (1); $\alpha>1$ – коэффициент растяжения пространства (допускается $\alpha=\infty$).

Алгоритм **em99b** ($\alpha>1$)

(*B*-форма метода эллипсоидов с известными m и f^*)

Шаг 0. Выбираем точку $x_0\in\mathbb{R}^n$ и радиус r_0 такими, чтобы $\left\|x_0-x^*\right\|\leq r_0$. Выбираем $\varepsilon>0$, полагаем $B_0:=I_n$ и $k:=0$.

Шаг 1. Вычислим $f_k=f\left(x_k\right)$ и $g_k=g\left(x_k\right)$. Если $f_k-f^*\leq\varepsilon$, то ОСТАНОВ: $k^*:=k$, $x_\varepsilon^*:=x_k$.

Шаг 2. Вычисляем $x_{k+1}:=x_k-\dfrac{m(f_k-f^*)}{\left\|B_k^\top g_k\right\|}B_k\xi_k$, где $\xi_k:=\dfrac{B_k^\top g_k}{\left\|B_k^\top g_k\right\|}$.

Шаг 3. Обновляем матрицу $B_{k+1}=B_k+\left(\dfrac{1}{\alpha}-1\right)\left(B_k\xi_k\right)\xi_k^T$.

Шаг 4. Обновляем радиус $r_{k+1}:=\sqrt{r_k^2-\dfrac{m^2(f_k-f^*)^2}{\left\|B_k^\top g_k\right\|^2}}$.

Шаг 5. Устанавливаем $k:=k+1$ и переходим к Шагу 1.

На каждой итерации алгоритма **em99b** $n\times n$ -матрица B_k связана с заменой переменных $x=B_ky$, $A_k=B_k^{-1}$. Ее обновление (Шаг 3) требует $O(n^2)$ операций, что обусловлено использованием оператора растяжения пространства. Величина шага в направлении нормированного субградиента в преобразованном пространстве переменных связана с шагом Поляка (Агмона-Моцкина-Шенберга)

$$y_{k+1}=A_kx_{k+1}=y_k-\frac{m(f_k-f^*)}{\left\|B_k^\top g_k\right\|}\frac{B_k^T g_f(x_k)}{\left\|B_k^T g_f(x_k)\right\|}=y_k-\frac{m(f_k-f^*)}{\left\|B_k^\top g_k\right\|}\frac{g_\varphi(y_k)}{\left\|g_\varphi(y_k)\right\|},$$

где $g_\varphi(y_k)=B_k^T g_f(x_k)$ есть субградиентом функции $\varphi\left(y\right)=f\left(B_ky\right)$ в точке $y_k=A_kx_k$ пространства переменных $y=A_kx$.

Теорема 3. Последовательность точек $\{x_k\}_{k=0}^{k^*}$, которую генерирует алгоритм **em99b**, удовлетворяет неравенствам

$$\left\| A_k (x_k - x^*) \right\| \le r_k, \quad A_k = B_k^{-1}, \quad k = 0,1,2,\ldots,k^*. \qquad (5)$$

Скорость сходимости алгоритма **em99b** определяется отношением объема эллипсоида $E_k = \left\{ x : \left\| A_k (x_k - x) \right\| \le r_k \right\}$, локализующего точку x^* на k-й итерации, к объему эллипсоида $E_{k-1} = \left\{ x : \left\| A_{k-1} (x_{k-1} - x) \right\| \le r_{k-1} \right\}$, локализующего x^* на $\left(k-1\right)$-й итерации, $1 \le k \le k^*$.

Теорема 4. На каждой итерации k, где $1 \le k \le k^*$, отношение объемов эллипсоидов E_k и E_{k-1} есть величина

$$q_k(\alpha) = \frac{vol(E_k)}{vol(E_{k-1})} = \frac{1}{\alpha}\left(\frac{r_k}{r_{k-1}} \right)^n < \frac{1}{\alpha} < 1. \qquad (6)$$

Для итерации k^* имеем $x_\varepsilon^* = x_{k^*}$, учитывая, что $f(x_{k^*}) - f^* \le \varepsilon$, если $\left\| B_k^T g\left(x_{k^*}\right) \right\| \le \varepsilon / r_k$. Если $\alpha = \infty$, то $k^* \le n$.

В табл. 3 приведены результаты работы алгоритма **em99b** по нахождению точки x_ε^* при $\varepsilon = 10^{-6}$ для функций $f_1\left(x\right)$ и $f_2\left(x\right)$ при $n = 100 \div 500$ и $n = 3 \div 8$. Здесь $k^*\left(\alpha\right)$ – количество итераций при $\alpha \in \left\{2,10,100,10^6,10^{12}\right\}$, r_k^* – радиус шара на итерации k^*.

ТАБЛИЦА 3. Итерации **em99b** для $f_1\left(x\right)$ и $f_2\left(x\right)$ при $\varepsilon = 10^{-6}$

$f_2\left(x\right), f_2^* = 0,\ m = 1, x_0 = (0,\ldots,0)^T,\ r_0 = 3,$						
n	$f_2(x_0)$	$k^*(2)$	$k^*(10)$	$k^*(100)$	$k^*(10^{12})$	r_k^*
3	1.11e+02	21	10	6	3	2.449e+00
5	1.11e+04	53	21	13	5	2.000e+00
8	1.11e+07	128	50	28	8	1.000e+00
$f_1\left(x\right), f_1^* = 0,\ m = 1, x_0 = (0,\ldots,0)^T,\ r_0 = 25,$						

n	$f_1(x_0)$	$k^*(2)$	$k^*(10)$	$k^*(100)$	$k^*(10^6)$	r_k^*
100	5.05e+03	1028	447	238	100	2.291e+01
200	2.01e+04	2255	929	497	200	2.062e+01
500	1.250e+05	6303	2558	1273	500	1.118e+01

Из табл. 3 видно, что количество итераций алгоритма **em99b** уменьшается по мере увеличения коэффициента растяжения пространства α, а при очень больших значениях α количество итераций равно количеству переменных n (см. теорему 4, «если $\alpha = \infty$, то $k^* \le n$»). При этом для «сильно» овражной функции $f_2(x)$ достаточно выбрать коэффициент $\alpha = 10^{12}$, а для «слабо» овражной функции $f_1(x)$ достаточно выбрать $\alpha = 10^6$.

В последнем столбце табл. 3 отражено как уменьшается r_k^* (радиус шара локализации образа точки x^* в преобразованном пространстве переменных на заключительной итерации) по отношению к начальному радиусу r_0 для шара локализации точки x^*. Следует отметить, что при одном и том же ε величина r_k^* не зависит от того, какое значение коэффициента α используется.

Если $\alpha = \infty$, то количество итераций алгоритма **em99b** совпадает с количеством итераций «предельного варианта» для метода, который Н.З. Шор предложил для выпуклых функций с так называемыми постоянными роста M и m [8, 9].

Теорема 5 (Шор, 1970). Пусть $f(x)$ – выпуклая функция, $x \in \mathbb{R}^n$, в окрестности $S_r = \left\{x : \left\|x - x^*\right\| \le r\right\}$ субградиент $g_f(x)$ удовлетворяет двустороннему неравенству

$$m\Big(f(x) - f\left(x^*\right)\Big) \le \left(g_f(x),\ x - x^*\right) \le M\Big(f(x) - f\left(x^*\right)\Big), \quad (7)$$

где $M \ge m$ – положительные константы. Если стартовая точка $x_0 \in S_r$, $B_0 = I_n$ – единичная $n \times n$-матрица, то метод

$$x_{k+1} := x_k - \frac{2Mm}{M+m} \frac{f(x_k) - f(x^*)}{\left\| B_k^T g_f(x_k) \right\|} B_k \xi_k, \quad B_{k+1} := B_{k+1} R_{\beta_k}(\xi_k), \quad (8)$$

$$\xi_k := \frac{B_k^{\bullet} g_f(x_k)}{\left\| B_k^{\bullet} g_f(x_k) \right\|}, \ \beta_k := \frac{1}{\alpha_k} < 1, \ \alpha_k = \frac{M+m}{M-m}, \quad k = 0,1,2,\ldots \quad (9)$$

генерирует последовательность $\{x_k\}_{k=0}^{\infty}$, которая удовлетворяет неравенству

$$\left\| A_k \left(x_k - x^* \right) \right\| \le r, \ A_k = B_k^{-1}, \ k = 0,1,2,... \quad (10)$$

Для выпуклых функций с константами M и m теорема 5 определяет вариант субградиентного метода с растяжением пространства в направлении субградиента, который сходится со скоростью геометрической прогрессии по отклонению наилучшего достигнутого значения $f(x)$ от оптимального $f^* = f(x^*)$. Это обеспечивает неравенство (10), в соответствии с которым объем эллипсоида, в котором локализуется точка x^*, убывает со скоростью геометрической прогрессии со знаменателем $\dfrac{M-m}{M+m}$.

Действительно, из неравенства (10) следует локализация x^* в эллипсоиде $E_k = \left\{ x : \left\| A_k \left(x_k - x \right) \right\| \le r \right\}$ с центром в точке x_k. Отношение объемов эллипсоидов E_{k+1} и E_k задается следующим равенством:

$$\frac{vol(E_{k+1})}{vol(E_k)} = \beta_k = \frac{M-m}{M+m}. \quad (11)$$

В каждом из преобразованных пространств переменных эллипсоид, локализующий образ точки x^*, есть шаром радиуса r, и радиус не меняется при переходе в очередное преобразованное пространство переменных. Заметим, что в алгоритме **em99b** это не так, радиусы аналогичных шаров монотонно уменьшаются.

Если $M = m$, то из (11) следует «предельный вариант» метода (8)–(9), для которого коэффициент растяжения $\alpha = \infty$, или же $\beta_k = \beta = 0$ для всех $k = 0,1,2,\ldots$. Для квадратичной положительно-определенной функции можно выбирать $M = m = 2$. Для кусочно-линейной функции, надграфик которой представляет собой конус с вершиной в точке $\left(x^*, f^*\right)$, можно выбирать $M = m = 1$. Этим случаям соответствуют алгоритмы, которые сходятся за число шагов, не превышающее n. Решение невырожденной системы n линейных уравнений с n неизвестными $\left(a_i, x\right) + b_i = 0, \quad i = 1,\ldots,n$, можно заменить нахождением минимума $f\left(x\right) = \max\limits_{1 \le i \le n} \left|\left(a_i, x\right) + b_i\right|$. Если взять $f^* = 0$, $\beta_k = 0$ и применить метод (8), (9), то получим алгоритм, соответствующий известной конечной процедуре решения линейных алгебраических систем – методу ортогонализации градиентов.

Выводы. Предложены два новых варианта методов эллипсоидов – алгоритмы **em80b** и **em99b** для минимизации выпуклой функции от n переменных. Они представлены в *B*-форме, где корректируется $n \times n$ -матрица *B*, которая определяет замену переменных. Показано, что с их помощью можно найти приближение к точке минимума выпуклой функции с очень высокой точностью, чего нельзя сделать с помощью этих же методов в *H*-форме, где корректируется симметрическая матрица $H = BB^T$. Методы в *H*-форме можно использовать для задач малой размерности, такие как задачи плоского местоположения. Даже на задачах «небольшого» размера они страдают численной неустойчивостью и низкой производительностью на практике.

Алгоритм **em80b** является одной из малоизвестных форм классического метода эллипсоидов, где для того, чтобы на порядок улучшить отклонение найденного рекордного значения функции $f\left(x\right)$ от ее оптимального значения f^* нужно сделать $4.6n^2$ итераций. Алгоритм **em80b** можно успешно применять для минимизации выпуклой функции, когда $n = 10 \div 25$. Это подтверждено результатами вычислительных экспериментов для оптимизации «сильно» и «слабо» овражных негладких функций.

Алгоритм **em99b** является В-формой метода эллипсоидов для аппроксимации множества, полученного в результате пересечения n-мерного шара и набора гиперплоскостей. На каждой его итерации объем локализации точки минимума уменьшается в α раз, где $\alpha > 1$ – коэффициент растяжения пространства. Предельный вариант алгоритма **em99b** (соответствует $\alpha = \infty$) сходится к точке минимума не более, чем за n итераций.

Развитие идей метода эллипсоидов, построение ускоренных модификаций методов эллипсоидов и исследование их связи с *r*-алгоритмами Шора-Журбенко и другими методами минимизации негладких функций можно найти в книге [10]. В ней значительное внимание уделяется специальным видам одноранговых линейных операторов и обоснованию на их основе субградиентных методов с преобразованием пространства, для которых скорость сходимости не хуже, чем в эффективных реализациях *r*-алгоритмов.

Авторы выражают благодарность Журбенко Н.Г. за полезные замечания, которые позволили улучшить изложение материала.

Литература

1. Юдин Д.Б., Немировский А.С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач. Экономика и математические методы. 1976. Вып. 2. С. 357–369.
2. Шор Н.З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования. Кибернетика. 1977. № 1. С. 94–95.
3. Хачиян Л.Г. Полиномиальные алгоритмы в линейном программировании. Ж. вычисл. матем. и матем. физ. 1980. Т. 20. № 1. С. 51–68.
4. Измаилов А.Ф., Стецюк П.И., Фишер А. Алгоритм emshor и его octave реализация. Компьютерная математика. 2019. № 1. С. 132–142.

5. Stetsyuk P., Fischer A., Khomyak O. The Generalized Ellipsoid Method and Its Implementation. In: Jacimovic M., Khachay M., Malkova V., Posypkin M. (eds) *Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science*. 2020. Vol 1145. Springer, Cham, pp. 355–370.

6. Grötschel M., Lovász L., Schrijver A. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica. 1981. Vol 1. No 2. pp. 169–197.

7. Стецюк П.И. К методам эллипсоидов. Теория оптимальных решений. Киев: Институт кибернетики им. В.М.Глушкова НАН Украины, 1999. С. 27–33.

8. Шор Н.З. Использование операций растяжения пространства в задачах минимизации выпуклых функций. Кибернетика. 1970. № 1. С. 6–12.

9. Шор Н. З. О скорости сходимости обобщенного градиентного спуска с растяжением пространства // Кибернетика. 1970. № 2. С. 80—85.

10. Стецюк П.И. Методы эллипсоидов и *r*-алгоритмы. Кишинэу. Эврика. 2014. 488 с.