

П.І. СТЕЦЬОК

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна,
e-mail: *stetsyukp@gmail.com*.

А. ФІШЕР

Технічний університет Дрездена, Інститут обчислювальної математики,
Дрезден, Німеччина, e-mail: *Andreas.Fischer@tu-dresden.de*.

О.М. ХОМ'ЯК

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна,
e-mail: *khomiak.olha@gmail.com*.

УНІФІКОВАНЕ ПРЕДСТАВЛЕННЯ КЛАСИЧНОГО МЕТОДУ ЕЛІПСОЇДІВ¹

Анотація. Розглянуто параметричну версію методу еліпсоїдів з масштабуванням простору за скалярним параметром $\lambda > 0$. Для певних значень параметра λ вона зводиться до відомих варіантів методу еліпсоїдів Шора, Хачіяна, Немировського та Юдіна. Обґрунтовано властивості двох алгоритмічних реалізацій параметризованого методу еліпсоїдів. Перший алгоритм базується на оновленні несиметричної матриці B , а другий оновлює симетричну матрицю $H = BB^T$. Описано застосування алгоритмів для розв'язання задач опуклого програмування та для знаходження сідлової точки опукло-увігнутої функції.

Ключові слова: метод еліпсоїдів, розтягнення простору, опукле програмування, сідлова точка.

ВСТУП

Віктор Михайлович Глушков у VIII главі «Математические методы оптимизации» своєї книги «Основы безбумажной информатики» [1] приділив значну увагу розвиненню теорії оптимізації, зокрема, і методу еліпсоїдів. Він зазначав, що метод еліпсоїдів є окремим випадком розробленого в Інституті кібернетики методу узагальненого градієнтного спуску з розтягненням простору в напрямку узагальненого градієнта, а також, що завдяки завершеності теорії збіжності метод еліпсоїдів використано для побудови та обґрунтування поліноміального алгоритму розв'язання задачі лінійного програмування з цілими коефіцієнтами.

Класичний метод еліпсоїдів запропонували в 1976 р. Д.Б. Юдін та А.С. Немировський [2]. Вони отримали цей метод зі схеми послідовних відсікань та назвали його модифікованим методом центрованих перерізів. Незалежно від цього метод еліпсоїдів був відкритий і Н.З. Шором у 1977 р. [3]. У цій праці метод представлено як окремий випадок субградієнтних методів з розтягненням простору, які були запропоновані Н.З. Шором наприкінці шістдесятих років [4, 5]. Зауважимо, що в 1970 р. Н.З. Шор був за крок до відкриття класичного методу еліпсоїдів. У цьому легко переконатися, якщо порівняти доведення теореми про збіжність методу еліпсоїдів, наведеної в [3], з доведенням теореми 2 у [5].

На основі субградієнтних методів з розтягненням простору в [6] запропоновано параметричну версію методу еліпсоїдів з масштабуванням простору за скалярним параметром $\lambda > 0$. Для певних значень параметра λ отримані відомі варіанти методу еліпсоїдів, а саме Н.З. Шора [3], А.С. Немировського та Д.Б. Юдіна (див. [7, с. 76]), Л.Г. Хачіяна (див. [8, лема 4]). Параметризований метод еліпсоїдів дає змогу не тільки об'єднати зазначені варіанти методів еліпсоїдів, а також запропонувати нові його варіанти [9].

У цій статті наведено властивості двох алгоритмічних реалізацій параметризованого методу еліпсоїдів. Перший алгоритм базується на оновленні неси-

¹Роботу підтримано грантом Volkswagen Foundation № 97775 та грантом Національного фонду досліджень України № 2021.01/0136

метричної матриці B , як у методі еліпсоїдів, запропонованому Н.З. Шором, а другий оновлює симетричну матрицю $H = BB^T$, як запропоновано В.А. Скоковим [10] для субградієнтних методів з розтягненням простору. Крім того, описано, як метод еліпсоїдів можна застосувати до задачі безумовної мінімізації опуклої функції, до розв'язання задач опуклого програмування з обмеженнями та для знаходження сідлової точки опукло-увігнутої функції.

Матеріал викладено наступним чином: у розд. 1 описано B -форму параметризованого методу еліпсоїдів та його властивості, у розд. 2 дано детальне доведення теореми 1 із розд. 1, у розд. 3 описано H -форму параметризованого методу еліпсоїдів, а в розд. 4 продемонстровано, як метод еліпсоїдів можна застосувати до задач опуклого програмування та до проблеми сідлової точки опукло-увігнутої функції.

1. B -ФОРМА ПАРАМЕТРИЗОВАНОГО МЕТОДУ ЕЛІПСОЇДІВ

Розглянемо таку задачу. Нехай задано відображення $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Припустимо, що існує така точка $x^* \in \mathbb{R}^n$, що $g(x)^T (x - x^*) \geq 0$ для всіх $x \in \mathbb{R}^n$ і $g(x) \neq 0$ для всіх $x \neq x^*$. Не обмежуючи загальності, будемо вважати, що x^* — єдина точка. Для знаходження наближення до точки x^* можна використовувати метод еліпсоїдів та його параметризовані версії.

Алгоритм 1. B -форма параметризованого методу еліпсоїдів.

Крок 0. Вибираємо точку $x_0 \in \mathbb{R}^n$, несингулярну матрицю $B_0 \in \mathbb{R}^{n \times n}$ і радіус $r_0 > 0$ такими, щоб $\|B_0^{-1}(x_0 - x^*)\| \leq r_0$. Крім того, вибираємо $\lambda > 0$ і вважаємо, що $k := 0$.

Крок 1. Обчислюємо $g(x_k)$. Якщо $g(x_k) = 0$, то ЗУПИНКА: $k^* := k$, $x^* := x_k$.

Крок 2. Обчислюємо $x_{k+1} := x_k - h_k B_k \xi_k$, де $\xi_k := \frac{B_k^T g(x_k)}{\|B_k^T g(x_k)\|}$, $h_k = \frac{1}{n+1} r_k$.

Крок 3. Оновлюємо $B_{k+1} := \lambda \left(B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \right)$ і $r_k := \frac{1}{\lambda \sqrt{\frac{n-1}{n+1}}} r_k$.

Крок 4. Установлюємо $k := k + 1$ і переходимо до кроку 1.

На кожній ітерації алгоритму оновлюється матриця B_k , яка пов'язана із заміною змінних $x = B_k y$, де $y = A_k x$ — відображення точки x у перетвореному просторі змінних, $A_k = B_k^{-1}$. Очевидно, що оновлення B -матриці (див. крок 3) потребує $O(n^2)$ операцій. Це зумовлено використанням параметра $\lambda > 0$ та оператора розтягнення простору $R_\alpha(\xi): \mathbb{R}^n \rightarrow \mathbb{R}^n$, який визначається як

$$R_\alpha(\xi) := I_n + (\alpha - 1) \xi \xi^T, \quad (1)$$

де $\xi \in \mathbb{R}^n$, $\|\xi\| = 1$ — напрямок розтягнення, а $I_n \in \mathbb{R}^{n \times n}$ — одинична матриця. Властивості оператора розтягнення простору детально вивчені в [11]. Вважаючи, що $\beta := 1/\alpha$, а оператор «оберненого» розтягнення позначений $R_\alpha^{-1}(\xi)$, отримуємо $R_\alpha^{-1}(\xi) = R_\beta(\xi)$ і $B_{k+1} = \lambda B_k R_\beta(\xi_k)$. Останній вираз показує, яку роль відіграє оператор розтягнення простору для оновлення B -матриці. Для оновлення A -матриць маємо співвідношення

$$A_{k+1} = \frac{1}{\lambda} R_\alpha(\xi_k) A_k, \quad (2)$$

яке впливає з ланцюга рівностей

$$A_{k+1} = B_{k+1}^{-1} (\lambda B_k R_\beta (\xi_k))^{-1} = \frac{1}{\lambda} R_\beta^{-1} (\xi_k) B_k^{-1} = \frac{1}{\lambda} R_{1/\beta} (\xi_k) A_k = \frac{1}{\lambda} R_\alpha (\xi_k) A_k.$$

В алгоритмі 1 використовується $\alpha = \sqrt{\frac{n+1}{n-1}}$.

Теорема 1. Послідовність точок $\{x_k\}$, яку генерує алгоритм 1, задовольняє нерівність

$$\|A_k(x_k - x^*)\| = \|B_k^{-1}(x_k - x^*)\| \leq r_k, \quad k \in \mathbb{N}. \quad (3)$$

Детальне доведення теореми 1 наведено у розд. 2.

Множина точок x , яка задовольняє нерівність $\|B_k^{-1}(x_k - x)\| \leq r_k$, є еліпсоїдом $\mathcal{E}_k := \{x \mid \|B_k^{-1}(x_k - x)\| \leq r_k\}$, що містить точку x^* . Еліпсоїд $\mathcal{E}_k$ має об'єм

$$\text{vol}(\mathcal{E}_k) = \frac{v_0 r_k^n}{\det A_k}, \quad (4)$$

де v_0 — об'єм одиничної n -вимірної кулі, $\det A_k$ — визначник матриці A_k .

Отже, швидкість збіжності алгоритму 1 визначатиметься відношенням об'єму еліпсоїда $\mathcal{E}_{k+1}$, що локалізує точку x^* на $(k+1)$ -й ітерації, до об'єму еліпсоїда $\mathcal{E}_k$, що локалізує x^* на k -й ітерації.

Теорема 2. Нехай x_k і x_{k+1} згенеровані алгоритмом 1. Тоді відношення об'ємів еліпсоїдів $\mathcal{E}_{k+1}$ і $\mathcal{E}_k$ не залежить від k і дорівнює

$$q_n := \frac{\text{vol}(\mathcal{E}_{k+1})}{\text{vol}(\mathcal{E}_k)} = \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2+1}} \right)^n < \exp\left\{-\frac{1}{2n}\right\} < 1. \quad (5)$$

Крім того, $x^* \in \mathcal{E}_k$ для всіх $k \in \mathbb{N}$.

Доведення. Згідно з (2) маємо $A_{k+1} = \frac{1}{\lambda} R_\alpha (\xi_k) A_k$ і для визначників матриць A_{k+1} та A_k справедливо $\det A_{k+1} = \left(\frac{1}{\lambda}\right)^n \det R_\alpha (\xi_k) \det A_k$. Враховуючи (4) і те, що $\det R_\alpha (\xi) = \alpha$, знаходимо для (5) коефіцієнт зменшення об'єму

$$\begin{aligned} q_n &= \frac{\text{vol}(\mathcal{E}_{k+1})}{\text{vol}(\mathcal{E}_k)} = \frac{v_0 r_{k+1}^n \det A_k}{v_0 r_k^n \det A_{k+1}} = \left(\frac{r_{k+1}}{r_k} \right)^n \frac{\det A_k}{\left(\frac{1}{\lambda}\right)^n \det R_\alpha (\xi_k) \det A_k} = \\ &= \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2-1}} \right)^n. \end{aligned}$$

Доведення нерівності в (5) (див. [12], стор. 77–78) базується на використанні співвідношення

$$\sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2-1}} \right)^n = \left(\left(\frac{n}{n+1} \right)^{n+1} \left(\frac{n}{n+1} \right)^{n-1} \right)^{1/2}.$$

Теорема 2 доведена.

Згідно з теоремою 2 маємо $q_n < \exp\left\{-\frac{1}{2n}\right\} < 1$, що означає, що для великих n коефіцієнт зменшення об'єму апроксимується асимптотичною формулою $q_n \approx 1 - \frac{1}{2n}$. Отже, для того щоб у 10 разів зменшити об'єм еліпсоїда, який локалізує точку x^* , потрібно зробити K ітерацій ($K = -\frac{\ln 10}{\ln q_n} \approx (2 \ln 10)n \approx 4.6n$).

Враховуючи, що теореми 1 та 2 справедливі для будь-якого параметра $\lambda \in (0, \infty)$, із алгоритму 1 можна отримати як відомі, так і нові варіанти методів еліпсоїдів, адаптувавши цей λ . Так, наприклад, для методу еліпсоїдів Шора [3] достатньо вибрати $\lambda = 1$, за яким формули для кроку 3 будуть мати вигляд

$$B_{k+1} := B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \quad \text{і} \quad r_{k+1} := \frac{n}{\sqrt{n^2 - 1}} r_k, \quad (6)$$

які є найекономнішими за кількістю арифметичних операцій. Для методу еліпсоїдів Хачіяна [8] достатньо вибрати $\lambda = \frac{n}{\sqrt{n^2 - 1}}$, при цьому формули для кроку 3 будуть мати вигляд

$$B_{k+1} := \frac{n}{\sqrt{n^2 - 1}} \left(B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \right) \quad \text{і} \quad r_{k+1} := r_k, \quad (7)$$

які гарантують постійну величину кроку в перетвореному просторі на кожній ітерації алгоритму 1. Варіанту методу еліпсоїдів Немировського та Юдіна [7] відповідає $\lambda = \left(\frac{n+1}{n-1} \right)^{1/2n}$, за яким для кроку 3 маємо такі формули:

$$B_{k+1} := \left(\frac{n+1}{n-1} \right)^{1/2n} \left(B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \right) \quad \text{і} \quad r_{k+1} := \left(\frac{n-1}{n+1} \right)^{1/2n} \frac{n}{\sqrt{n^2 - 1}} r_k.$$

У [9] наведено варіант методу еліпсоїдів, для якого $\lambda = \left(\frac{n}{\sqrt{n^2 - 1}} \right)^{3/2}$ та формули для кроку 3 мають вигляд

$$B_{k+1} := \left(\frac{n}{\sqrt{n^2 - 1}} \right)^{3/2} \left(B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \right) \quad \text{і} \quad r_{k+1} := \left(\frac{n}{\sqrt{n^2 - 1}} \right)^{1/2} r_k.$$

Усі варіанти (з різними λ) теоретично еквівалентні, хоча невеликі відмінності можуть спостерігатися через накопичення помилок обчислень.

2. ДОВЕДЕННЯ ТЕОРЕМИ 1 (АЛГОРИТМ 1)

Доведення проведемо за аналогією з доведенням, яке для методу еліпсоїдів було зроблено Н.З. Шором [3]. Для доведення будемо використовувати співвідношення

$$R_\alpha^T(\xi) R_\alpha(\xi) = R_{\alpha^2}(\xi), \quad (8)$$

яке випливає з властивостей (1) — оператора розтягнення простору (див. [11], стор. 68–69).

Доведення теореми 1 проведемо методом індукції по k . Для $k=0$ нерівність (3) переходить у $\|B_0^{-1}(x_0 - x^*)\| \leq r_0$ та виконується за припущенням. Припустимо, що нерівність (3) виконується для $k = \bar{k}$. Доведемо її справедливості для $k = \bar{k} + 1$.

Враховуючи співвідношення (8) і те, що з (2) випливає $A_{k+1} = \frac{1}{\lambda} R_{\alpha}(\xi_{\bar{k}}) A_{\bar{k}}$, маємо ланцюжок рівностей:

$$\begin{aligned} ||A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)||^2 &= \langle A_{\bar{k}+1}(x_{\bar{k}+1} - x^*), A_{\bar{k}+1}(x_{\bar{k}+1} - x^*) \rangle = \\ &= \langle \frac{1}{\lambda} R_{\alpha}(\xi_{\bar{k}}) A_{\bar{k}}(x_{\bar{k}+1} - x^*), \frac{1}{\lambda} R_{\alpha}(\xi_{\bar{k}}) A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle = \\ &= \frac{1}{\lambda^2} \langle A_{\bar{k}}(x_{\bar{k}+1} - x^*), R_{\alpha}^T(\xi_{\bar{k}}) R_{\alpha}(\xi_{\bar{k}}) A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle = \\ &= \frac{1}{\lambda^2} \langle A_{\bar{k}}(x_{\bar{k}+1} - x^*), R_{\alpha^2}(\xi_{\bar{k}}) A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle = \\ &= \frac{1}{\lambda^2} \langle A_{\bar{k}}(x_{\bar{k}+1} - x^*), A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle + \frac{1}{\lambda^2} (\alpha^2 - 1) \langle \xi_{\bar{k}}, A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle^2 = \\ &= \frac{1}{\lambda^2} ||A_{\bar{k}}(x_{\bar{k}+1} - x^*)||^2 + (\alpha^2 - 1) \frac{1}{\lambda^2} \langle \xi_{\bar{k}}, A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle^2, \end{aligned}$$

який запишемо у вигляді співвідношення

$$\lambda^2 ||A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)||^2 = ||A_{\bar{k}}(x_{\bar{k}+1} - x^*)||^2 + (\alpha^2 - 1) \langle \xi_{\bar{k}}, A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle^2. \quad (9)$$

Далі, розшифруємо обидва доданки у правій частині (9), для чого використаємо співвідношення

$$A_{\bar{k}}(x_{\bar{k}+1} - x^*) = A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}}, \quad (10)$$

яке з урахуванням того, що $A_{\bar{k}} = B_{\bar{k}}^{-1}$ і чергова точка в алгоритмі 1 обчислюється за формулою з кроку 2, випливає з ланцюжка рівностей

$$\begin{aligned} A_{\bar{k}}(x_{\bar{k}+1} - x^*) &= A_{\bar{k}}(x_{\bar{k}} - h_{\bar{k}} B_{\bar{k}} \xi_{\bar{k}} - x^*) = A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} A_{\bar{k}} B_{\bar{k}} \xi_{\bar{k}} = \\ &= A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}}. \end{aligned}$$

Перший доданок у правій частині (9) можна записати у вигляді рівності

$$||A_{\bar{k}}(x_{\bar{k}+1} - x^*)||^2 = ||A_{\bar{k}}(x_{\bar{k}} - x^*)||^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2, \quad (11)$$

яка з урахуванням (9) і того, що $||\xi_{\bar{k}}||=1$, випливає з ланцюжка рівностей

$$\begin{aligned} ||A_{\bar{k}}(x_{\bar{k}+1} - x^*)||^2 &= ||A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}}||^2 = \\ &= \langle A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}}, A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}} \rangle = \\ &= \langle A_{\bar{k}}(x_{\bar{k}} - x^*), A_{\bar{k}}(x_{\bar{k}} - x^*) \rangle - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2 \langle \xi_{\bar{k}}, \xi_{\bar{k}} \rangle = \\ &= ||A_{\bar{k}}(x_{\bar{k}} - x^*)||^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2 ||\xi_{\bar{k}}||^2 = \\ &= ||A_{\bar{k}}(x_{\bar{k}} - x^*)||^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2. \end{aligned}$$

Враховуючи співвідношення (10), для квадрата скалярного добутку в другому доданку правої частини (9) маємо такий ланцюжок рівностей:

$$\begin{aligned} \langle A_{\bar{k}}(x_{\bar{k}+1} - x^*), \xi_{\bar{k}} \rangle^2 &= \langle A_{\bar{k}}(x_{\bar{k}} - x^*) - h_{\bar{k}} \xi_{\bar{k}}, \xi_{\bar{k}} \rangle^2 = \\ &= (\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle - h_{\bar{k}} \langle \xi_{\bar{k}}, \xi_{\bar{k}} \rangle)^2 = (\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle - h_{\bar{k}} ||\xi_{\bar{k}}||^2)^2 = \\ &= (\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle - h_{\bar{k}})^2 = \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2. \end{aligned}$$

Отже, квадрат зазначеного скалярного добутку можна записати у вигляді

$$\langle A_{\bar{k}}(x_{\bar{k}+1} - x^*), \xi_{\bar{k}} \rangle^2 = \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2. \quad (12)$$

Підставляючи (11) і (12) у (9) маємо

$$\begin{aligned} \lambda^2 \|A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)\|^2 &= \|A_{\bar{k}}(x_{\bar{k}+1} - x^*)\|^2 + (\alpha^2 - 1) \langle \xi_{\bar{k}}, A_{\bar{k}}(x_{\bar{k}+1} - x^*) \rangle^2 = \\ &= \|A_{\bar{k}}(x_{\bar{k}} - x^*)\|^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2 + \\ &+ (\alpha^2 - 1) (\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle^2 - 2h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + h_{\bar{k}}^2) = \\ &= \|A_{\bar{k}}(x_{\bar{k}} - x^*)\|^2 - 2\alpha^2 h_{\bar{k}} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle + \\ &+ (\alpha^2 - 1) \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle^2 + \alpha^2 h_{\bar{k}}^2 = \|A_{\bar{k}}(x_{\bar{k}} - x^*)\|^2 - \\ &- \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle (2\alpha^2 h_{\bar{k}} - (\alpha^2 - 1) \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle) + \alpha^2 h_{\bar{k}}^2, \end{aligned}$$

звідки з урахуванням $h_k = \frac{1}{n+1} r_k$ і $\alpha = \sqrt{\frac{n+1}{n-1}}$ маємо

$$\begin{aligned} \lambda^2 \|A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)\|^2 &= \|A_{\bar{k}}(x_{\bar{k}} - x^*)\|^2 - \\ &- \frac{2}{n-1} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle (r_{\bar{k}} - \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle) + \frac{1}{n^2 - 1} r_{\bar{k}}^2. \end{aligned} \quad (13)$$

Далі, для оцінювання знака добутку

$$\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle (r_{\bar{k}} - \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle),$$

що входить до правої частини співвідношення (13), оцінимо знаки обох його співмножників. Перший співмножник буде невід'ємним. Оскільки маємо $\langle x - x^*, g(x) \rangle \geq 0$ для всіх $x \in \mathbb{R}^n$, його легко оцінити так:

$$\begin{aligned} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle &= \left\langle A_{\bar{k}}(x_{\bar{k}} - x^*), \frac{B_{\bar{k}}^T g(x_{\bar{k}})}{\|B_{\bar{k}}^T g(x_{\bar{k}})\|} \right\rangle = \\ &= \frac{1}{\|B_{\bar{k}}^T g(x_{\bar{k}})\|} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), B_{\bar{k}}^T g(x_{\bar{k}}) \rangle = \\ &= \frac{1}{\|B_{\bar{k}}^T g(x_{\bar{k}})\|} \langle B_{\bar{k}} A_{\bar{k}}(x_{\bar{k}} - x^*), g(x_{\bar{k}}) \rangle = \frac{1}{\|B_{\bar{k}}^T g(x_{\bar{k}})\|} \langle x_{\bar{k}} - x^*, g(x_{\bar{k}}) \rangle \geq 0. \end{aligned}$$

З урахуванням того, що

$$0 \leq \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle \leq \|A_{\bar{k}}(x_{\bar{k}} - x^*)\| \leq r_{\bar{k}},$$

другий співмножник оцінюється так:

$$r_{\bar{k}} - \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle \geq 0.$$

З невід'ємності обох співмножників випливає, що

$$\frac{2}{n-1} \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle (r_{\bar{k}} - \langle A_{\bar{k}}(x_{\bar{k}} - x^*), \xi_{\bar{k}} \rangle) \geq 0. \quad (14)$$

Далі, врахувавши (14) і те, що $\|A_{\bar{k}}(x_{\bar{k}} - x^*)\| \leq r_{\bar{k}}$, співвідношення (13) запишемо у вигляді

$$\begin{aligned} \lambda^2 \|A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)\|^2 &\leq \|A_{\bar{k}}(x_{\bar{k}} - x^*)\|^2 + \frac{1}{n^2 - 1} r_{\bar{k}}^2 \leq \\ &\leq r_{\bar{k}}^2 + \frac{1}{n^2 - 1} r_{\bar{k}}^2 = \left(\frac{n^2}{n^2 - 1} \right) r_{\bar{k}}^2, \end{aligned}$$

звідки маємо нерівність

$$\|A_{\bar{k}+1}(x_{\bar{k}+1} - x^*)\|^2 = \|B_{\bar{k}+1}(x_{\bar{k}+1} - x^*)\|^2 \leq \frac{1}{\lambda^2} \left(\frac{n}{\sqrt{n^2 - 1}} \right)^2 r_{\bar{k}}^2 = r_{\bar{k}+1}^2,$$

з якої випливає справедливість нерівності (3) для $k = \bar{k} + 1$.

Теорема 1 доведена.

3. H-ФОРМА ПАРАМЕТРИЗОВАНОГО МЕТОДУ ЕЛІПСОЇДІВ

Наведену раніше B -форму параметризованого методу еліпсоїдів (алгоритм 1) можна записати в H -формі за допомогою додатно визначених симетричних матриць $H_k = B_k B_k^T$. При цьому спрощується формула для оновлення матриці $H_{k+1} = B_{k+1} B_{k+1}^T$ з урахуванням того, що для $\beta = \frac{1}{\alpha} = \sqrt{\frac{n-1}{n+1}}$ спрощується вираз $\beta^2 - 1 = \frac{n-1}{n+1} - 1 = -\frac{2}{n+1}$.

Розглянемо H -форму, яка відповідає алгоритму 1.

Алгоритм 2. H -форма параметризованого методу еліпсоїдів.

Крок 0. Вибираємо точку $x_0 \in \mathbb{R}^n$, додатно визначену симетричну матрицю $H_0 \in \mathbb{R}^{n \times n}$ і радіус $r_0 > 0$ такими, щоб $(x_0 - x^*)^T H_0^{-1} (x_0 - x^*) \leq r_0^2$. Крім того, вибираємо $\lambda > 0$ і вважаємо, що $k := 0$.

Крок 1. Обчислюємо $g(x)$. Якщо $g(x) = 0$, то ЗУПИНКА: $k^* := k$, $x^* := x_k$.

Крок 2. Обчислюємо $x_{k+1} := x_k - h_k \frac{H_k g(x_k)}{\sqrt{g(x_k)^T H_k g(x_k)}}$, де $h_k := \frac{1}{n+1} r_k$.

Крок 3. Оновлюємо $H_{k+1} := \lambda^2 \left(H_k - \frac{2}{n+1} \frac{H_k g(x_k) g(x_k)^T H_k}{g(x_k)^T H_k g(x_k)} \right)$ і $r_{k+1} := \frac{1}{\lambda} \frac{n}{\sqrt{n^2 - 1}} r_k$.

Крок 4. Установлюємо $k := k + 1$ і переходимо до кроку 1.

Послідовність $\{x_k\}$, згенерована алгоритмом 2, (теоретично) ідентична відповідній послідовності, згенерованій алгоритмом 1, за умови, що на кроці 0 останнього однакові значення для x_0 , r_0 вибираються як для алгоритму 2, а B_0 вибирається так, що $H_0 = B_0 B_0^T$. Щоб побачити, що H -форма справді створює ту саму послідовність $\{x_k\}$, можна індуктивно показати простими обчисленнями, що

$$x_{k+1} = x_k + h_k B_k \xi_k = x_k - h_k \frac{H_k g(x_k)}{\sqrt{g(x_k)^T H_k g(x_k)}},$$

$$B_{k+1}B_{k+1}^T = \lambda^2 \left(H_k - \frac{2}{n+1} \frac{H_k g(x_k) g(x_k)^T H_k}{g(x_k)^T H_k g(x_k)} \right)$$

виконується для всіх $k \in \mathbb{N}$ для алгоритму 1, що означає, що

$$H_{k+1} = B_{k+1}B_{k+1}^T$$

справедливе для $k \in \mathbb{N}$. Відповідно до цього еліпсоїд $\mathcal{E}_k$ має еквівалентні представлення за допомогою матриць B_k і H_k , а саме

$$\mathcal{E}_k = \{x \mid \|B_k^{-1}(x_k - x)\| \leq r_k\} = \{x \mid \|(x_k - x)H_k^{-1}(x_k - x)\| \leq r_k^2\}.$$

Отже, теореми 1 і 2 справедливі і для алгоритму 2.

З одного боку, для обчислення послідовності $\{x_k\}$ за допомогою алгоритму 2 потрібно вдвічі менше операцій, ніж для алгоритму 1. Використання оперативної пам'яті показує приблизно таке саме співвідношення, якщо матриці B_k є несиметричними. З іншого боку, H -форма обчислювально менш стійка, оскільки матриці H_k можуть стати несиметричними та не бути додатно визначеними. Продемонструємо це на прикладі з [9].

Для $\lambda = 1$, $N = 2$, $H_0 := I_2$, $g(x_{2k}) := (1, -1)^T$ і $g(x_{2k+1}) := (2, 1)^T$ для $k \in \mathbb{N}$ формула для оновлення H_k на кроці 3 алгоритму 2 дає

$$H_{50} = \begin{pmatrix} 8.6163e-13 & 9.5855e-14 \\ 9.5914e-14 & 1.6273e-12 \end{pmatrix}, \quad H_{70} = \begin{pmatrix} 9.9927e-18 & -1.8545e-17 \\ 4.0653e-17 & -3.5467e-17 \end{pmatrix},$$

де H_{50} є несиметричною, а H_{70} не є ані симетричною, ані додатно визначеною.

Для алгоритму 1 (B -форма параметризованого методу еліпсоїдів) таких проблем бути не може. У тому самому прикладі матриці B_{50} і B_{70} , обчислені в алгоритмі 1, такі, що

$$B_{50}B_{50}^T = \begin{pmatrix} 8.6162e-13 & 9.5889e-14 \\ 9.5889e-14 & 1.6273e-12 \end{pmatrix}, \quad B_{70}B_{70}^T = \begin{pmatrix} 1.4592e-17 & 1.6239e-18 \\ 1.6239e-18 & 2.7559e-17 \end{pmatrix},$$

тобто $H_k = B_k B_k^T$ залишаються симетричними та додатно визначеними.

Із алгоритму 2 можна отримати H -форми для відомих варіантів методів еліпсоїдів. Так, наприклад, для методу еліпсоїдів Шора аналоги формул (6) для кроку 3 в алгоритмі 2 будуть мати вигляд

$$H_{k+1} := H_k - \frac{2}{n+1} \frac{H_k g(x_k) g(x_k)^T H_k}{g(x_k)^T H_k g(x_k)} \quad \text{і} \quad r_{k+1} := \frac{1}{\lambda} \frac{n}{\sqrt{n^2 - 1}} r_k, \quad (15)$$

а для методу еліпсоїдів Хачіяна аналоги формул (7) для кроку 3 в алгоритмі 2 будуть мати вигляд

$$B_{k+1} := \frac{n^2}{n^2 - 1} \left(H_k - \frac{2}{n+1} \frac{H_k g(x_k) g(x_k)^T H_k}{g(x_k)^T H_k g(x_k)} \right) \quad \text{і} \quad r_{k+1} := r_k. \quad (16)$$

Зауважимо, що для опису методу еліпсоїдів формули (16) використовуються набагато частіше, ніж формули (15).

4. ПРИКЛАДИ ЗАДАЧ ДЛЯ ЗАСТОСУВАННЯ МЕТОДУ ЕЛІПСОЇДІВ

Розглянемо три задачі із [13], для розв'язання яких можна використовувати параметризований метод еліпсоїдів.

Задача 1. Мінімізація опуклої функції. Метод еліпсоїдів можна використовувати для знаходження x^* — точки мінімуму опуклої функції $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Мінімальне значення f позначається $f^* := f(x^*)$. Для простоти припустимо, що x^* є єдиним мінімізатором f . Щоб застосувати алгоритм 1 до щойно описаної проблеми мінімізації, розглянемо відображення $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$, яке використовується в розд. 1 та буде спеціалізовано на $g_f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, де $g_f(x)$ є субградієнтом f від x , що означає

$$\langle x - x^*, g_f(x) \rangle \geq f(x) - f(x^*) = f(x) - f^* \geq 0 \text{ для всіх } x \in \mathbb{R}^n. \quad (17)$$

Крім того, порівняно з алгоритмом 1 на кроці 1 можна використовувати більш відповідний критерій завершення, який гарантує, що алгоритм не зупиняється раніше, ніж

$$f(x_k) \leq f^* + \varepsilon \quad (18)$$

виконується для деякого попередньо визначеного $\varepsilon > 0$. Цей критерій виводиться так. Для будь-яких $(x_k, B_k, r_k) \in \mathbb{R}^n \times \mathbb{R}^{n \times n} \times (0, \infty)$, згенерованих алгоритмом 1, отримуємо

$$\begin{aligned} f(x_k) - f^* &\leq \langle x_k - x^*, g_f(x_k) \rangle = \langle B_k^{-1}(x_k - x^*), B_k^T g_f(x_k) \rangle \leq \\ &\leq \|B_k^{-1}(x_k - x^*)\| \|B_k^T g_f(x_k)\| \leq r_k \|B_k^T g_f(x_k)\|, \end{aligned}$$

де перша нерівність випливає з (17), а остання є наслідком $x^* \in \mathcal{E}_k$ згідно з теоремою 1. Отже, якщо $r_k \|B_k^T g_f(x_k)\| \leq \varepsilon$, то (18) виконується.

Задача 2. Задача опуклого програмування. Розглянемо задачу з обмеженнями

$$\min_{x^* \in \mathbb{R}^n} f_0(x) \quad (19)$$

за обмежень

$$f_i(x) \leq 0, \quad i=1, 2, \dots, m, \quad (20)$$

де для $i=0, 1, \dots, m$ функції $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ є опуклими і $g_i: \mathbb{R}^n \rightarrow \mathbb{R}^n$ позначає відображення так, що $g_i(x)$ є субградієнтом f в x . Припустимо, що задача (19), (20) має єдиний розв'язок x^* і для неї виконується умова Слейтера. Крім того, нехай відображення $g_i: \mathbb{R}^n \rightarrow \mathbb{R}^n$ визначається як

$$g(x) = g_{i(x)}(x),$$

де відображення $x \rightarrow i(x)$ задовольняє умови

$$\begin{aligned} i(x) &= 0, & \text{якщо } f_1(x) \leq 0, \dots, f_m(x) \leq 0, \\ i(x) &\in \{i \mid f_i(x) > 0\}, & \text{якщо } \max\{f_1(x), \dots, f_m(x)\} > 0. \end{aligned} \quad (21)$$

Далі, можна побачити, що $\langle x - x^*, g(x) \rangle \geq 0$ справедливо для всіх $x \in \mathbb{R}^n$. Для цього спочатку розглянемо випадок, коли x задовольняє $\max\{f_1(x), \dots, f_m(x)\} \leq 0$. Відповідно до (21) маємо $g(x) = g_0(x)$. Крім того, враховуючи $f_0(x) \geq f_0(x^*)$ і опуклість f , маємо

$$\langle x - x^*, g(x) \rangle = \langle x - x^*, g_0(x) \rangle \geq f_0(x) - f_0(x^*) \geq 0.$$

Якщо інакше, $\max\{f_1(x), \dots, f_m(x)\} > 0$, то з (21) випливає $f_j(x) > 0$ для $j := i(x)$. Враховуючи опуклість f_j і те, що $f_j(x) \leq 0$, отримуємо

$$\langle x - x^*, g(x) \rangle = \langle x - x^*, g_j(x) \rangle \geq f_j(x) - f_j(x^*) \geq 0.$$

Отже, $\langle x - x^*, g(x) \rangle \geq 0$ справедливо для всіх $x \in \mathbb{R}^n$.

Задача 3. Сідлові точки опукло-увігнутих функцій. Нехай $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ є опукло-увігнутою функцією, тобто $f(\cdot, y): \mathbb{R}^n \rightarrow \mathbb{R}$ є опуклою для будь-якого $y \in \mathbb{R}^m$ і $f(x, \cdot): \mathbb{R}^m \rightarrow \mathbb{R}$ є увігнутою для будь-якого $x \in \mathbb{R}^n$. Пара $z^* := (x^*, y^*) \in \mathbb{R}^n \times \mathbb{R}^m$ називається сідловою точкою f , якщо

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*) \quad \text{для всіх } (x, y) \in \mathbb{R}^n \times \mathbb{R}^m.$$

Крім того, з $z := (x, y)$ множина $G_x(z) \in \mathbb{R}^n$ містить всі субградієнти $f(\cdot, y)$ відносно x , тоді як $G_y(z) \in \mathbb{R}^m$ містить всі суперградієнти $f(x, \cdot)$ відносно y . Отже, можна визначити відображення $g: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \times \mathbb{R}^m$ за допомогою

$$g(z) := \begin{pmatrix} g_x(z) \\ -g_y(z) \end{pmatrix} \quad \text{із } (g_x(z), g_y(z)) \in G_x(z) \times G_y(z).$$

За наведеним визначенням сідлової точки та за опуклістю функцій $f(\cdot, y)$ і $-f(x, \cdot)$ для будь-яких фіксованих y та x відповідно, випливає, що

$$\begin{aligned} 0 \leq f(x, y^*) - f(x^*, y) &= f(x, y^*) - f(x, y) + f(x, y) - f(x^*, y) \leq \\ &\leq -\langle g_y(z), y - y^* \rangle + \langle g_x(z), x - x^* \rangle = \langle g(z), z - z^* \rangle. \end{aligned}$$

Отже, $\langle g(z), z - z^* \rangle \geq 0$ виконується для всіх $z \in \mathbb{R}^n \times \mathbb{R}^m$, тому параметричний метод еліпсоїдів може бути використаний для апроксимації сідлової точки z^* .

Метод еліпсоїдів також має низку інших застосувань, наприклад, для спеціальних класів варіаційних нерівностей [14], для розв'язання координуючих негладких задач, які виникають під час застосування схем декомпозиції для блочних задач великої розмірності [15] тощо. При цьому достатньо визначити правило побудови гіперплощин, які локалізують шукану точку, та умову зупинки ітераційного процесу в кроці 1 алгоритмів 1 та 2.

ВИСНОВКИ

У статті проведено детальний аналіз параметричної версії класичного методу еліпсоїдів з масштабуванням простору за скалярним параметром $\lambda > 0$. Для певних значень параметра λ ця версія забезпечує відомі варіанти методу еліпсоїдів, а саме варіанти Шора, Хачіяна, Немировського та Юдіна. Досліджено властивості двох алгоритмічних реалізацій параметризованого методу еліпсоїдів. Перший алгоритм базується на оновленні несиметричної матриці B , а другий оновлює симетричну матрицю $H = BB^T$. Описано застосування алгоритмів для розв'язання задач опуклого програмування та для знаходження сідлових точок опукло-увігнутих функцій.

Таку саму параметризацію можна застосувати для узагальненого методу еліпсоїдів [16, 17], де для локалізації шуканої точки використовується наближений до оптимального еліпсоїд, який забезпечує асимптотичну швидкість збіжності узагальненого методу еліпсоїдів такою ж, як і в класичному методі еліпсоїдів.

СПИСОК ЛІТЕРАТУРИ

1. Глушков В.М. Основы безбумажной информатики. Изд. 2-е. Москва: Наука, 1987. 552 с.
2. 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.

3. Шор Н.З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования. *Кибернетика*. 1977. № 1. С. 94–95.
4. Шор Н.З., Билецкий В.И. Метод растяжения пространства для ускорения сходимости в задачах овражного типа. *Труды семинара Науч. совета АН УССР по кибернетике «Теория оптимальных решений»*. Киев. 1969. № 2. С. 3–18.
5. Шор Н.З. Использование операции растяжения пространства в задачах минимизации выпуклых функций. *Кибернетика*. 1970. № 1. С. 6–12.
6. Stetsyuk P.I., Fischer A., Khomiak O.M. Ellipsoid methods with space scaling. *Proc. of the 8th International Conference on Control and Optimization with Industrial Applications (COIA 2022)* (24–26 Aug., 2022, Baku, Azerbaijan). 2022. P. 402–404.
7. Nemirovsky A.S., Yudin D.B. Problem complexity and method efficiency in optimization. New York: John Wiley, 1983.
8. Khachiyan L.G. Polynomial algorithms in linear programming, USSR Comput. Math. Math. Phys. 1980. Vol. 20, N 1. P. 53–72.
9. Fischer A., Khomyak O., Stetsyuk P. The ellipsoid method and computational aspects. *Commun. Optim. Theory*. 2023. Vol. 21. P. 1–14.
10. Скоков В.А. Замечание к методам минимизации, использующим операцию растяжения пространства. *Кибернетика*. 1974. № 4. С. 115–117.
11. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. Киев: Наук. думка, 1979. 200 с.
12. Grötschel M., Lovasz L., Schrijver A. Geometric algorithms and combinatorial optimization. Berlin: Springer-Verlag, 1988. 362 p.
13. Шор Н.З. Новые направления в развитии методов негладкой оптимизации. *Кибернетика*. 1977. № 6. С. 87–91.
14. Семенов В.В. Варіаційні нерівності: теорія та алгоритми. Київ: ВПЦ «Київський університет», 2021. 167 с.
15. Shor N.Z. Nondifferentiable optimization and polynomial problems. Amsterdam: Kluwer, 1998.
16. Стецюк П.И., Фесюк А.В., Хомяк О.Н. Обобщенный метод эллипсоидов. *Кибернетика и системный анализ*. 2018. Т. 54, № 4. С. 70–80.
17. 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*. Cham: Springer, 2020. Vol. 1145. P. 355–370.

P.I. Stetsyuk, A. Fischer, O.M. Khomiak

UNIFIED PRESENTATION OF THE CLASSICAL ELLIPSOID METHOD

Abstract. A parametric version of the ellipsoid method with space scaling by a scalar parameter $\lambda > 0$ is considered. For certain values of the parameter λ , it is reduced to the available versions of the ellipsoid method, namely, the Shor, Khachiyan, Nemirovsky, and Yudin ones. The properties of two algorithmic implementations of the parameterized ellipsoid method are proved. The first algorithm is based on updating the asymmetric matrix B , and the second one updates the symmetric matrix $H = BB^T$. The applications of the algorithms for solving convex programming problems and for finding the saddle point of a convex-concave function are described.

Keywords: ellipsoid method, dilation of space, convex programming, saddle point.

Надійшла до редакції 23.06.2023