МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ

МАУП

МЕТОДИЧНІ МАТЕРІАЛИ ЩОДО ЗАБЕЗПЕЧЕННЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

з дисципліни

“МЕТОДИ ОПТИМІЗАЦІЇ”

(для спеціалістів)

Київ
ДП «Видавничий дім «Персонал»
2010

Підготовлено доктором технічних наук, професором кафедри прикладної математики та програмування *І. В. Бейком* та кандидатом фізико-математичних наук, доцентом кафедри прикладної математики та програмування *П. М. Зіньком*

Затверджено на засіданні кафедри прикладної математики та програмування (протокол № 1 від 30.08.07)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

Бейко І. В., Зінько П. М. Методичні матеріали щодо забезпечення самостійної роботи студентів з дисципліни "Методи оптимізації" (для спеціалістів). — К.: ДП «Вид. дім «Персонал», 2010. — 34 с.

Методичні матеріали містять пояснювальну записку, зміст самостійної роботи з дисципліни, а також список літератури для самостійної роботи студентів денної та заочної форм навчання, які вивчають дисципліну "Методи оптимізації".

ЗМІСТ

Відповідальний за випуск *А. Д. Вегеренко*
Редактор *С. Г. Рогузько*
Комп’ютерне верстання *А. М. Голянда*

Зам. № ВКЦ-3504

Папір офсетний.
Друк ротаційний трафаретний. Наклад 30 пр.

Міжрегіональна Академія управління персоналом (МАУП)
03039 Київ-39, вул. Фрометівська, 2, МАУП

ДП «Видавничий дім «Персонал»
03039 Київ-39, просп. Червонозоряний, 119, літ. XX

Свідоцтво про внесення до Державного реєстру
суб’єктів видавничої справи ДК № 3262 від 26.08.2008

ПОЯСНЮВАЛЬНА ЗАПИСКА

Мета навчальної дисципліни “Методи оптимізації” — допомогти студентам опанувати методи розв’язування задач оптимізації реальних систем у реальних умовах неповних даних за допомогою сучасного математично-комп’ютерного інструментарію.

Самостійну роботу студенти виконують у бібліотеці, у навчальних кабінетах, у комп’ютерних класах та вдома. Самостійна робота студентів організаційно і методично планується і спрямовується як особиста творча праця студента. При виконанні самостійної роботи студенти отримують індивідуальні консультації та допомогу викладачів та фахівців деканату. Самостійна робота студентів передбачає роботу на практичних заняттях, опрацювання конспектів лекцій і літературних джерел, роботу в бібліотеці, зокрема електронних бібліотеках Інтернету, вивчення навчального матеріалу за підручниками, навчальними посібниками, а також підготовку доповідей, рефератів, написання курсових робіт; пошукову і науково-дослідну діяльність та самотестування. Самостійна робота студентів з дисципліни “Методи оптимізації” здійснюється з використанням навчально-методичних матеріалів та математично-комп’ютерного інструментарію з метою поглиблення теоретичних знань та набуття навичок їх практичного застосування. Постійне самостійне навчання допомагає поглибити знання та набути вмінь, необхідних для висококваліфікованого фахівця. Завдання викладача на аудиторних заняттях — дати студентам основи знань, навчити їх самостійно вчитися та здобувати нові знання.

Індивідуальна робота студентів сприятиме поглибленню їх знань з дисципліни “Методи оптимізації” шляхом творчого вивчення сучасної проблематики теорії оптимізації. У процесі самостійної індивідуальної роботи використовуються тестові та розрахункові завдання. За допомогою тестів визначається рівень засвоєння студентом основних принципів та методичних положень, на які спирається сучасна теорія оптимізації і дослідження операцій. Вивчення курсу передбачає оглядові лекції, анотації до кожної теми курсу, тестові завдання, практичні завдання та завдання для індивідуальної роботи. Лекційний матеріал призначається для спрямування студентів у найраціональнішому напрямі опанування нових знань і акцентуванні уваги на найскладніших і вузлових питаннях навчальної дисципліни. Конспектування лекцій допомагає студенту зберігати важливу інформацію та аналізувати її в подальшому. При підготовці до практичних занять студент використовує конспект лекцій. Якщо у конспекті бракує матеріалу з окремих питань, або вони винесені на самостійне опрацювання, студент опрацьовує рекомендовані підручники, навчальні посібники та відповідне програмне забезпечення для ПК.

Під час виконання самостійної роботи допоможе вміння знаходити необхідний матеріал в міжнародних фондах Інтернету за URL-адресами Web-документів та Ftp-файлів з літературних джерел у географічно віддалених фондах та архівах, а також шляхом участі у мережних конференціях, де можна отримати відповіді та поради щодо розшукуваної інформації. В Інтернеті існують різні інформаційно-пошукові системи (http: // www. yahoo. com, http: // www. portal. edu. ru, http: // www. ipl. org та ін.), що грунтуються на інформації, розподіленій у тематичних розділах за ключовими словами.

ЗМІСТ САМОСТІЙНОЇ РОБОТИ з дисципліни “МЕТОДИ ОПТИМІЗАЦІЇ”

Тема 1. Градієнтні методи розв’язування задач безумовної оптимізації. Градієнтний метод мінімізації диференційованої функції. Градієнтний метод мінімізації з розтяганням простору.

Тема 2. Градієнтний метод пошуку сідлових точок.

Тема 3. Методи розв’язування задач квадратичного програмування. Метод спряжених градієнтів для мінімізації квадратичної функції при лінійних обмеженнях. Метод спряжених градієнтів для задачі квадратичного програмування з обмеженнями типу нерівностей. Модифікація методу спряжених напрямків для задач квадратичного програмування великої розмірності. Стійкі алгоритми розв’язування задач квадратичного програмування.

Тема 4. Методи лінеаризації для розв’язування задач нелінійного програмування.

Тема 5. Методи мінімаксної оптимізації. Методи відшукання стаціонарної точки мінімаксу. Метод розв’язування мінімаксної задачі з обмеженнями, заданими нерівностями.

Додаткова

8. *Базилевич В., Лук’янов В., Писаренко Н., Квіцинська Н.* Методи оптимізації і дослідження операцій: Опорний конспект лекцій. — К.: Четверта хвиля, 1997. — 248 с.
9. *Поляк Б. Т.* Введение в оптимизацию. — М.: Наука, 1983.
10. *Пшеничный Б. Н., Данилин Ю. М.* Численные методы в экстремальных задачах. — М.: Наука, 1975.
11. *Будаговська С. та ін.* Методи оптимізації і дослідження операцій та макроекономіка. — К.: Основи, 1998. — 518 с.
12. *Макконнелл К. Р., Брю С. Л.* Аналітична економія. Принципи, проблеми і політика: Пер. з англ. / Наук. ред. перекладу Т. Панчишина. — Л.: Просвіта, 1999. — Ч. 2. Методи оптимізації і дослідження операцій.
13. *Нуреев Р.* Сборник задач по микроэкономике. — М.: ИНФРА-М, 2002. — 432 с.
14. *Шндайк Р. С, Рубінфельд Д. Л.* Методи оптимізації і дослідження операцій / Пер. з англ. А. Олійника, Р. Скільського. — К.: Основи, 1996. — 646 с.
15. *Самуельсон П. А., Нордгауз В. Д.* Методи оптимізації і дослідження операцій: Пер. з англ. / За ред. С. Панчишина. — К.: Основи, 1998. — 676 с.
16. *Ястремський О., Гриценко О.* Основи мікроекономіки. — К.: Знання, 1998. — 674 с.

то керування

$$u_i\,(y(A, c)) \equiv \arg\max_{u(i)\in Di}[(y(A, c), f_i\,(u_i)) + f_i\,(u_i)]$$

є розв'язком задачі 2.

Теорема 3. Якщо

$$f_i(u_i)=B_iu_i + d_i, f_i\,(u_i) = (h_i, u_i),$$

$$D_i = \{u_i \mid \| u_i \|_{v(i)} \le \alpha_i\}$$

$$v(i) \in\{1, 2, 3\}, \| u \|_1 \equiv \max_j | (u)_j |,$$

$$\| u \|_2 \equiv\sum_j | (u)|_1, \| u \|_3 \equiv(\sum_j(u)_j^{\,2})^{1/2} |,$$

де $(u)_j$ — j-*та* компонента вектора u, то керування

$$u_i\,(y(A, c)) \equiv s\,(B_i{}^*y_i(A, c) +h_i, \alpha_i, v(i))$$

є в умовах теореми 2 розв'язком задачі 2,

де $(s(a,\alpha, 1))_i = \alpha\,\mathrm{sign}\,(a)_i$,

$(s(a,\alpha, 3) = \alpha\, a_i/\| a \|_3$,

$(s(a,\alpha, 2))_i = \alpha\,(a)\,\mathrm{sign}(a)_i$,

$\alpha^i\,(a)$ — розв'язок α^i системи

$$\alpha^i \ge 0, \sum \alpha^i= \alpha, \forall i \notin I \equiv \arg\max_i | (u)_i| \alpha^i = 0. \qquad (3)$$

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Бейко И. В., Бублик Б. Н., Зінько П. Н.* Методи і алгоритми розв'язування задач оптимізації. — К.: Вища шк., 1993. — 512 с.
2. *Ермольев Ю. М., Ляшко И. И., Михалевич В. С., Тюптя В. И.* Математические методы исследования операцій. — К.: Вища шк., 1979. — 312 с.
3. *Жук М. В., Щербина Ю. М.* Збірник задач з методів оптимізації. — Львів, ЛДУ, 1997.
4. *Шор Н. З.* Методы минимизации недифференцируемых функций и их приложения. — К.: Наук. думка, 1979.
5. *Задоя О.* Методи оптимізації і дослідження операцій. — Дніпропетровськ, 2001.
6. *Кириленко В.* Методи оптимізації і дослідження операцій: Навч. посіб. — К.: Таксон, 1998. — 334 с.
7. *Методи* оптимізації і дослідження операцій: Навч. посіб. для студ. вищ. навч. закл. / Н. О. Гончарова, А. І. Ігнатюк, Н. А. Малиш та ін. — К.: МАУП, 2005. — 304 с.

Тема 6. Методи відшукання сідлових точок. Градієнтні методи відшукання сідлових точок. Узагальнений градієнтний метод для відшукання сідлових точок.

Тема 7. Методи розв'язування стохастичних мінімаксних задач.

Тема 8. Методи розв'язуючих операторів для оптимального керування та оцінювання.

Основні питання для самостійного опрацювання:

1. Методи градієнтного спуску з розтяганням простору.
2. Методи узагальнених градієнтів для випуклих функцій.
3. Лінійні та нелінійні задачі оптимізації з обмеженнями.
4. Методи множників Лагранжа для задач нелінійного програмування.
5. Градієнтні методи для задач нелінійного програмування.
6. Методи узагальнених градієнтів для задач випуклого програмування.

Знання, які студент повинен здобути під час самостійної роботи:

- основні постановки задач оптимізації функцій багатьох змінних;
- аналітичні методи відшукання мінімальних значень функцій багатьох змінних при обмеженнях з використанням методу множників Лагранжа;
- ітераційні методи наближеного розв'язування задач оптимізації;
- градієнтні методи оптимізації з обмеженнями;
- методи прискореної збіжності для мінімізації диференцьованих функцій багатьох змінних.

Уміння, які студент повинен набути під час самостійної роботи:

- розв'язувати задачі відшукання мінімальних значень функцій багатьох змінних аналітичними методами;
- будувати алгоритми для розв'язування задач великої розмірності;
- розв'язувати задачі оптимізації методами Ньютона;
- розв'язувати задачі оптимізації методами Монте-Карло;
- будувати алгоритми і розв'язувати задачі нелінійної оптимізації методами множників Лагранжа;
- будувати алгоритми оптимізації негладких функцій за допомогою узагальнених градієнтів;
- розв'язувати задачі оптимізації методами розтягання простору.

Індивідуальні завдання

Тип завдання: розробка алгоритмів та програм для розв'язування задач лінійного та нелінійного програмування, проведення чисельних експериментів для оптимізації параметрів алгоритмів; ознайомлення та опрацювання тематичної літератури; розробка алгоритмів та програм для розв'язування задач оптимізації функцій багатьох змінних.

Мета завдання: перевірка знань студентів, здобутих у процесі вивчення методів та алгоритмів для розв'язування задач лінійного та нелінійного програмування.

Самостійна робота: проаналізувати дані результатів, отримані в індивідуально проведених обчислювальних експериментах.

Ключові терміни: *задачі оптимізації функцій багатьох змінних, необхідні та достатні умови мінімуму, алгоритми прискореної оптимізації, задачі лінійного програмування.*

З метою стислого опису задач оптимізації та алгоритмів їх розв'язування скористаємося наступними математичними позначеннями та символами:

R — множина дійсних чисел;

$\alpha, \beta, \gamma, \delta, \varepsilon, \lambda, \mu, \nu, \rho, \sigma, \tau$ – дійсні числа;

i, j, k, l, m, n, s — цілі числа;

R^n — n-вимірний евклідовий простір, тобто множина векторів $x = (x_1, x_2, ..., x_n)$ всі компоненти яких $x_1, x_2, ..., x_n$ є дійсними числами;

$a, b, c, d, e, f, g, h, x, y, z,...$ і $a^i, b^i, a^k, x^k,...$ — вектори, тобто $a \in R^n$, $x \in R^k$,...

A, B, C, H — матриці;

a_{ij}, b_{ij}, c_{ij}, і h_{ij} — елементи i-го рядка та j-го стовпця відповідно матриць A, B, C і H;

$\Im, K, L$ — множина індексів (цілих чисел)

(x, y) — скалярний добуток векторів x та y, тобто $(x, y) = \sum_i x_i y_i$;

$\|x\|$ — евклідова норма вектора x, тобто $\|x\| = (x, x)^{1/2}$;

f_i — скалярна функція;

$f_i : X \to Y$ — функція, яка визначена на множині X і приймає значення із множини Y;

$f_i : X \to R^n$ — n-вимірна вектор-функція $f = (f_1, f_2, ..., f_n)$;

A^T — матриця, транспонована до матриці A;

A^{-1} — матриця, обернена до матриці A;

$\nabla f_i(x)$ — градієнт функції f_i в точці x, тобто

яке за даними спостережень zk, k=1,2,..., що надходять із "приладу спостережень"

$$z_k = C_k(x_k, x_{k-1}, ..., u_k, u_{k-1}, ..., u_1), \; z_k \in Z_k, \qquad (2)$$

забезпечує максимальне значення $B(x, u, z)$ для заданої функції B.

Для відшукання оптимального керування $u^* \in D$, яке на множині допустимих керувань D максимізує функцію B на траєкторіях $x=x(u)$ керованої системи (1), (2), тобто для всіх $u \in D$ виконуються нерівності

$$B(x(u), u, z(u)) \le B(x(u^*), u^*, z(u^*)),$$

можна скористатися методами розв'язуючих операторів. Методи розв'язуючих операторів ґрунтуються на розв'язках β функціонального рівняння $\beta(u, z, A(x, u, z), C(x, u)) - B(x, u, z) = 0$. У разі, як це буває в реальних задачах із залежною частиною $u^2 = U(x, u, z)$ керування $u_k \equiv (u_k^1, u_k^2)$, функціональне рівняння для розв'язуючого оператора β_{ABCD} набуває вигляду

$$\beta(u^1, U(x, u, z), z, A(x, u, z), C(x, u)) = B(x, u, z).$$

Алгоритм 1 відшукання u^* за допомогою β_{ABCD} ґрунтується на теоремі 1.

Теорема 1. Якщо $\forall\, u \in D, z \in Z, X(u, z) \ne \varnothing$ і керована система має єдиний розв'язок, то розв'язок $u(z)$ задачі параметричної оптимізації $u(z) \equiv \arg\min u \in D\, \beta_{ABCD}(u, z)$ є шуканим розв'язком задачі 1.

Задача 2. Знайти керування $u \in D$ для системи

$$A_{k,k} x_k + A_{k,k-1} x_{k-1} + ... + A_{k,1} x_1 + f_k(u) = 0$$
$$k = 1, 2, ..., N,$$

яке дає максимальне на заданій множині D значення для функції

$$B(x, u) \equiv \sum (c_i, x_i) + f_{N+1}(u)$$

при заданих матрицях $A_{k,i}$ векторах c_i і функціях f_k.

Алгоритм 2. Обчислити розв'язок $(y_N, y_{N-1}, ..., y_1) \equiv y(A, c)$ системи

$$c_k + \sum A_j^*{}_{,k} y_k = 0, \; k = N, N-1, ..., 1,$$

і далі обчислити керування $u(y(A, c)) = \arg\max_{u \in D} F(u, y(A, c))$ для функції

$$F(u, y) \equiv \sum (y_i, f_i(u)) + f_{N+1}(u).$$

Теорема 2. Якщо $y(A, c)$ — єдиний розв'язок керованої системи і $\forall\, u \in D\; X(u) \ne \varnothing$, то $B(X(u), u) = F(u, y(A, c))$ і керування $u(y(A, c))$ є шуканим розв'язком задачі 2.

Наслідок 1. Якщо $D = D_1 \times D_2 \times ... \times D_N$,

$$f_i(u) \equiv f_i(u_i), f_{N+1}(u) \equiv \sum f_i(u_i), u_i \in D_i,$$

II. Вибрати довільні дійсні числа z_i^0, $i \in [0:N_0]$, початкове значення кроку $\rho_0 > 0$, додатні константи $\alpha > 0$, $\beta > 0$ і покласти $k = 0$.

Основний цикл.

III. Знайти індекс i_k з умови $z_{i_k}^k(\omega) = \max\limits_{i\in[0:N_k]} z_i^k(\omega)$.

IV. За незалежною реалізацією ω^k випадкового параметра ω обчислити градієнт $\nabla_x(\varphi(x^k(\omega), y^{i_k}, \omega^k))$ та наступне наближення

$$x^{k+1}(\omega) = \pi_X(x^k(\omega) - \rho_k \nabla_x \varphi(x^k(\omega), y^{i_k}, \omega^k)).$$

V. Для всіх $i\in[0:N_k]$ обчислити значення $\phi(x^{k+1}(\omega), y^i, \omega^k)$ і побудувати нову сітку $Y_{N_{k+1}} = \left\{y^i \mid y^i \in Y,\ i \in \left[0:N_{k+1}\right]\right\}$, яка зберігає усі точки сітки Y_{N_k} і відстань від довільного $\overline{y} \in Y$ до сітки $Y_{N_{k+1}}$ не перевищує $\alpha\rho_{k+1}^{1+\beta}$.

VI. Обчислити $z_i^{k+1}(\omega) = z_i^k(\omega) + \sigma_{k+1}(\varphi(x^{k+1}(\omega), y^i, \omega^k) - z_i^k(\omega))$ для всіх $i\in[0:N_k]$.

VII. Покласти $i = N_k + 1$.

VIII. Знайти точку y^{j_i} сітки Y_{N_k}, яка належить кулі радіусом $\alpha\rho_{k+1}$ з центром у точці y^i і має найменший індекс $j_i \in [0:N_k]$.

IX. Покласти $z_i^{k+1}(\omega) = z_{j_i}^k(\omega)$.

X. Якщо $i < N_{k+1}$, то задати $i = i + 1$ і перейти на крок VIII.

XI. Задати $k = k + 1$ і перейти на крок III.

Теорема 2. *Якщо виконані умови теореми 1 і послідовність $\{\sigma_k\}_{k=0}^{\infty}$ задовольняє умові $\lim\limits_{s\to\infty}\sum_{j=s}^{m} \sigma_j = \infty$ при $m - s \to \infty$ (наприклад $\sigma_k = k^{-\nu}$, $\nu\in(1/2,1)$), то для ліпшицевої функції $E\varphi(x,y,\omega)$ майже при кожному ω граничні точки послідовності $\{x^k(\omega)\}_{k=0}^{\infty}$, породженої алгоритмом 2, мінімізують функцію $\max\limits_{y\in Y} E\varphi(x,y,\omega)$ на множині X.*

8. Методи розв'язуючих операторів для оптимального керування та оцінювання

Задача 1. Знайти таке керування $u \equiv (u_1, u_2, ..., u_N) \in D$ для керованої системи

$$A_k(x_k, x_{k-1}, ..., x_1, u_k, u_{k-1}, ..., u_1, z) = 0, \qquad (1)$$

$$x_k \in X_k,\ u_k \in U_k,\ k = 1, 2, ..., N,$$

$$\nabla f_i(x) = \left(\frac{\partial f_i(x)}{\partial x_1}, \dots, \frac{\partial f_i(x)}{\partial x_n}\right);$$

$\nabla f(x)$ — матриця Якобі для вектор-функції f в точці x, тобто елемент i-го рядка та j-го стовпця даної матриці дорівнює частинній похідній $\partial f_i(x)/\partial x_j$;

$\nabla^2_{xx} f_i(x)$ – матриця Гессе для функції f_i в точці x, тобто елемент k-го рядка і j-го стовпця даної матриці дорівнює $\partial^2 f_i(x)/\partial x_k \partial x_j$;

$\hat{\nabla} f_i(x)$ – узагальнений градієнт функції f_i в точці x;

$\tilde{\nabla} f_i(x)$ – квазіградієнт функції f_i в точці x;

$G \underline{\underline{\Delta}} Q$ – G дорівнює Q за визначенням (G еквівалентно Q);

$G \supset Q$ — G містить Q;

$G \subset Q$ — G міститься в Q;

$G \bigcup Q$ – об'єднання G і Q;

$G \bigcap Q$ – перетин G і Q;

$G \times Q$ – декартовий добуток G і Q;

$\{x | G\}$ – множина всіх x, для яких виконується твердження G;

$x \neq y$ — x не дорівнює y;

$x \notin Q$ — x належить Q;

$x \notin Q$ — x не належить Q;

$\varnothing$ — порожня множина;

I — одинична матриця;

(α, β) – відкритий інтервал $\{x | \alpha < x < \beta\}$;

$[\alpha, \beta]$ — замкнутий інтервал $\{x | \alpha \le x \le \beta\}$;

$\max\limits_{i\in\Im}$ — максимум по всім i, які належать множині $\Im$;

R^n_+ — множина $\{x | x \ge 0,\ x \in R^n\}$;

$\operatorname{diam} Y$ — діаметр множини Y;

$\operatorname{int} X$ — внутрішність множини X;

$\operatorname{tr} A$ — слід матриці A;

$\operatorname{rang} A$ — ранг матриці A;

$\operatorname{Ent}(t)$ — ціла частина числа t;

$\mathrm{R}(A)$ — образ матриці A;

$[1:n]$ — множина цілих чисел від 1 до n включно;

$\operatorname{co} X$ – опукла оболонка множини X;

$\det A$ — визначник матриці A;

$O(\alpha)$ - величина порядку α;

$o(\alpha)$ - величина нескінченно мала порівняно з α;

$(b\vdots A)$ – матриця, яка утворена стовпцем b і стовпцями матриці A;

$\overset{\text{м.н.}}{=}$ – дорівнює майже напевно;

$[t]_+$ — $[t]_+ = \max\{0, t\}$;

$(\beta_{ij})_{i=1,\ldots,m}^{j=1,\ldots,n}$ – матриця розмірності $m \times n$, (i, j)-им елементом якої є β_{ij};

$\{x, y, ..., z\}$ — множина, яка складається із елементів $x, y, ..., z$;

$\max\{t_1, t_2, ..., t_n\}$ — максимальний елемент із множини $\{t_1, t_2, ..., t_n\}$;

P {подія S} — ймовірність події S;

$\forall\, x \in X$ — для всіх елементів x із множини X;

$\exists\, x \in X$ — існує такий елемент x в множині X;

$\arg\min\limits_{x\in X} f_i(x)$ – той елемент x^* із множини X, на якому досягається найменше значення $f_i(x^*)$ функції f_i на множині X, тобто

$$\forall x \in X \quad f_i(\arg\min_{x\in X} f_i(x)) \le f_i(x);$$

$\operatorname{Arg}\min\limits_{x\in X} f_i(x)$ – множина всіх елементів $\arg\min\limits_{x\in X} f_i(x)$;

$\arg\max\limits_{x\in X} f_i(x)$ – елемент x^{**} із множини X, на якому досягається найбільше значення $f_i(x^{**})$ функції f_i на множині X, тобто

$$\forall x \in X \quad f_i(\arg\max_{x\in X} f_i(x)) \ge f_i(x);$$

X^* — множина розв'язків задачі оптимізації;

$E\xi$ – математичне сподівання випадкової величини ξ;

$E(\xi / x)$ – умовне математичне сподівання при заданому x;

$D\xi$ – дисперсія випадкової величини ξ;

Ω — простір елементарних подій ω.

1. Градієнтні методи розв'язування задач безумовної оптимізації

1.1. Градієнтні методи мінімізації диференційованих функцій

Задача 1. Знайти значення x^*, на якому диференційована функція $f: R^n \to R^1$ досягає найменшого значення $f(x^*)$.

Відомо, що розв'язком задачі 1 є вектор x^*, який задовольняє нерівності $f(x^*) \le f(x)$ для всіх значень x. Із твердження наступної теореми випливає, що розв'язок x^* є серед розв'язків системи алгебраїчних рівнянь $f'(x) = 0$.

Алгоритм 1

Початок.

I. Вибрати довільні початкове наближення $x^0 \in X$, натуральне число $N = N_s$, дійсні числа $\boldsymbol{z}_i^0$, $i \in [0:N]$ і покласти $k = 0$.

II. Побудувати сітку $Y_N = \left\{ y^i \mid y^i \in Y,\ i \in [0:N] \right\}$.

Основний цикл.

III. Знайти індекс i_k із умови $\boldsymbol{z}_{i_k}^k = \max\limits_{i\in[0:N]} \boldsymbol{z}_i^k$.

III. На незалежній реалізації ω^k випадкового параметра ω обчислити стохастичний градієнт $\nabla_x \phi(x^k(\omega), y^{i_k}, \omega^k)$ та наступне наближення

$$x^{k+1}(\omega) = \pi_X(x^k(\omega) - \rho_k \nabla_x \varphi(x^k(\omega), y^{i_k}, \omega^k)).$$

IV. Для всіх $i \in [0:N]$ обчислити величини $\varphi(x^{k+1}(\omega), y^i, \omega^k)$,

$$z_i^{k+1}(\omega) = z_i^k(\omega) + \sigma_k(\varphi(x^{k+1}(\omega), y^i, \omega^k) - z_i^k(\omega)),$$

покласти $k = k + 1$ і перейти на крок III.

Теорема 1. *Якщо опукла вниз по x функція $E\varphi(x,y,\omega)$ та її ліпшицевий градієнт задовольняють умовам*

$$E\left|\varphi(x,y,\omega)\right|^2 < \infty,\ \ E\left\|\nabla_x \varphi(x,y,\omega)\right\|^2 < \infty,\ \ y \in Y,\ x \in X,$$

то майже при кожному ω граничні точки $\overline{x}^{N_s}$ послідовності $\left\{x^k(\omega)\right\}_{k=0}^{\infty}$, породженої алгоритмом 1 за кроковими множниками ρ_k і σ_k, які задовольняють умовам

$$\sum_{k=0}^{\infty} \sigma_k^2 < \infty,\ \ \sum_{k=0}^{\infty} \rho_k = \infty,\ \ \rho_k / \sigma_k \to 0 \text{ при } k \to \infty;$$

$$\rho_{k+1} / \rho_k \to 1 \text{ при } k \to \infty,$$

мінімізують $\max\limits_{i\in[0:N]} E\varphi(x, y^i, \omega)$ на множині X.

За допомогою алгоритму 1 розв'язуємо задачу на ущільнювальній послідовності сіток $\left\{\boldsymbol{Y}_{N_s}\right\}_{s=0}^{\infty}$

Алгоритм 2 (модифікація алгоритму 1)

Початок.

I. Вибрати початкове наближення $x^0 \in R^n$, натуральне число і N_0 сітку

$$Y_{N_0} = \left\{ y^i \mid y^i \in Y,\ i \in [0:N_0] \right\}.$$

$$\varphi(x,\ y^*) \le \varphi(x^*,\ y^*) \le \varphi(x^*,\ y),\quad \forall x \in X,\ y \in Y,$$

для увігнуто-опуклої функції $\varphi(x, y)$ на замкнутих опуклих множинах X, Y в евклідових просторах R^n і R^m, для яких множини сідлових точок $X^* \times Y^*$,

$$X^* = \{x \mid \varphi(x,\ y^*) = \max_{x' \in X} \varphi(x',\ y^*),\quad x \in X\},$$
$$Y^* = \{y \mid \varphi(x^*,\ y) = \min_{y' \in Y} \varphi(x^*,\ y'),\quad y \in Y\},$$

є не порожніми і обмеженими.

Алгоритм 3

Початок.

I. Вибрати довільне початкове наближення $(x^0,\ y^0) \in X \times Y$ і покласти $k = 0$.

Основний цикл.

II. Обчислити узагальнені градієнти $g_x(x^k, y^k)$ та $g_y(x^k, y^k)$ для функції $\varphi(x,\ y)$ за аргументами x та y в точці (x^k, y^k).

III. За допомогою операторів проектування π_X, π_Y обчислити наступні наближення

$$x^{k+1} = \pi_X(x^k + \rho_k g_x \varphi(x^k,\ y^k)),\ y^{k+1} = \pi_Y(y^k - \rho_k g_y(x^k,\ y^k)).$$

IV. Покласти $k = k + 1$ та перейти на крок II.

Теорема 3. *Якщо на допустимій множині $X \times Y$ узагальнені градієнти обмежені і крокові множники ρ_k задовольняють умовам*

$$\rho_k > 0,\quad k = 0, 1, \ldots;\quad \lim_{k\to\infty} \sum_{i=0}^{k} \rho_i = \infty,$$

то $\min\limits_{(x^*,\ y^*) \in X^* \times Y^*} \|(x^k,\ y^k) - (x^*,\ y^*)\| \to 0$ при $k \to \infty$.

7. Методи розв'язування стохастичних мінімаксних задач

Задача. Знайти розв'язок $\arg\min\limits_{x\in X}\max\limits_{y\in Y} E\varphi(x,y,\omega)$ задачі стохастичної мінімаксної оптимізації для функції $\varphi: X \times Y \times \Omega \to R^1$ із неперервним математичним сподіванням $f(x,y) \underline{\underline{\Delta}} E\varphi(x,y,\omega)$ та градієнтом $\nabla_x \varphi(x,y,\omega)$ на опуклій, замкненій та обмеженій множині $X \subset R^n$ і на замкненій та обмеженій множині $Y \subset R^m$.

Теорема. Якщо $x^* = a$, то градієнт $f'(x)$ диференційованої функції f у точці $x = a$ дорівнює нулю, тобто $f'(a) = 0$.

Справедливість твердження даної теореми випливає із того, що диференцьовна функція при всіх z задовольняє асимптотичному рівнянню

$$f(a+z) = f(a) + (f'(a), z) + o(\|z\|).$$

Із цього рівняння при $z = -\varepsilon f'(a)$ маємо рівняння

$$f(a+ \varepsilon f'(a)) = f(a) - \varepsilon(f'(a), f'(a)) + o(\|\varepsilon f'(a)\|)$$

і для значення $\varepsilon>0$, яке задовольняє нерівності $o(\|\varepsilon f'(a)\|) < \varepsilon(f'(a), f'(a))/2$, отримаємо нерівність

$$f(a+ \varepsilon f'(a)) < f(a) - \varepsilon(f'(a), f'(a))/2 < f(a),$$

яка суперечить оптимальності a. Звідси випливає, що x^* є серед розв'язків рівняння $f'(x)=0$.

У наступному алгоритмі 1 реалізовано градієнтний метод відшукання оптимального розв'язку за ітераційною формулою

$$x^{k+1} = x^k - \lambda_k f'(x^k),\ k=1, 2, \ldots,$$

з використанням спеціального вибору кроків λ_k.

Алгоритм 1

Початок.

I. Вибрати початкове наближення $x^0 \in R^n$, значення параметрів $\lambda_0 > 0$, $\varepsilon > 0$, $s > 0$ і покласти $k = 0$.

Основний цикл.

II. Обчислити градієнт $f'(x^k)$ у точці x^k.

III. Якщо $\|f'(x^k)\| \le \varepsilon$, то покласти $x^* = x^k$ і завершити алгоритм, інакше перейти до кроку IV.

IV. Обчислити наступне наближення до оптимального розв'язку

$$x^k+1 = x^k - \lambda_k f'(x^k).$$

V. Якщо $f(x^{k+1}) > f(x^k) - s\ (\lambda_k)$, то покласти $\lambda_k = \lambda_k/2$ і повернутися до кроку IV.

VI. Покласти $k = k + 1$ і перейти на крок II.

Теорема. *Якщо x^* є точкою локального мінімуму майже диференційованої функції $f_0(x)$, то обчислена за алгоритмом 1 послідовність $\{x^k\}_{s=1}^{\infty}$ забезпечить виконання нерівності $\|f'(x^k)\| \le \varepsilon$ за скінченну кількість ітераційних кроків k.*

1.2. Градієнтний метод мінімізації з розтяганням простору

Задача. Знайти оптимальне значення $x^* = \arg\min\limits_{x\in R^n} f_0(x)$ для заданої майже диференційованої функції $f_0: R^n \to R^1$.

Для обчислення оптимального значення x^* за допомогою градієнтного методу з розтяганням простору будують і використовують оператор розтягання простору. Оператор розтягання простору R^n у напрямку $\xi \in R^n (\|\xi\| = 1)$ з коефіцієнтом $\alpha \geq 0$ позначається $G_\alpha(\xi)$ і обчислюється за формулою

$$G_\alpha(\xi)x = \alpha\gamma_\xi(x)\xi + d_\xi(x) = x + (\alpha - 1)(x, \xi)\xi$$

з попереднім представленням вектора x у формі

$$x = \gamma_\xi(x)\xi + d_\xi(x);$$

$$(\xi, d_\xi(x)) = 0;$$

$$\gamma_\xi(x) = (x, \xi),\ d_\xi(x) = x - (x, \xi)\xi.$$

Наступний ітераційний алгоритм 1 побудований на обчисленні наближення x^{k+1} за ітераційною формулою

$$x^{k+1} = x^k - B_k\rho_k\xi^k,$$

де ξ^k – нормований псевдоградієнт функції $\varphi_k(y) \underline{\underline{\Delta}} f_0(B_k y)$, B_k — оператор, обернений до оператора A_k, $A_k = G_{\alpha_k}(\xi^{k-1})A_{k-1}$), ρ_k – кроковий множник. Оператори B_{k+1} також обчислюються за ітераційними формулами:

$$B_{k+1} = B_k G_{\beta_{k+1}}\xi^k; \quad B_0 = I, \quad \beta_{k+1} \underline{\underline{\Delta}} 1/\alpha_{k+1}.$$

Алгоритм 1

Початок.

I. Вибрати початкове наближення $x^0 \in R^n$ і матрицю $B_0 = I$ (I — одинична $n \times n$-матриця); покласти $k = 0$.

Основний цикл.

II. Обчислити псевдоградієнт $g(x^k)$ функції f_0 у точці x^k.

III. Якщо $g(x^k) = 0$, то покласти $x^* = x^k$ і завершити алгоритм, інакше перейти до кроку IV.

IV. Обчислити і нормувати псевдоградієнт $\tilde{\boldsymbol{g}}(\boldsymbol{y}^k)$ функції $\phi_k(y) = f_0(B_k y)$ у точці $\boldsymbol{y}^k = \boldsymbol{B}_k^{-1}\boldsymbol{x}^k$:

$$\tilde{\boldsymbol{g}}(\boldsymbol{y}^k) = \boldsymbol{B}_k^T\ \boldsymbol{g}(\boldsymbol{x}^k),\ \xi^k = \tilde{\boldsymbol{g}}(\boldsymbol{y}^k) / \|\ \tilde{\boldsymbol{g}}(\boldsymbol{y}^k)\ \|.$$

V. Обчислити наступне наближення до оптимального розв'язку:

$$x^{k+1} = x^k - \rho_k B_k \xi^k.$$

VI. Обчислити коефіцієнти α_{k+1}, $\beta_{k+1} = 1/a_{k+1}$ та оператор B_{k+1} (обернений до оператора A_{k+1}) за формулою

$$B_{k+1} = B_k G_{\beta_{k+1}}(\xi^k).$$

VII. Покласти $k = k + 1$ і перейти на крок II.

Теорема 1. *Для довільного числа $\delta > 0$ існує таке число $\rho(\delta) > 0$, що для початкового наближення $(x^0,\ y^0) \in Z(\delta)$,*

$$Z(\delta) \underline{\underline{\Delta}} \{(x,\ y) \mid \min_{(x^*,\ y^*) \in X^* \times Y^*} \|(x,\ y) - (x^*,\ y^*)\| \leq \delta, (x,\ y) \in X \times Y\},$$

і для послідовності $\{\rho_k\}_{k=0}^\infty$, $\rho_k > 0, \rho_k \to 0, \sum_{k=0}^\infty \rho_k = \infty, \rho_k \leq \rho(\delta)$, алгоритм 1 генерує збіжну до множини сідлових точок $X^* \times Y^*$ послідовність $\{(x^k,\ y^k)\}_{k=0}^\infty$.

Алгоритм 2 (алгоритм із постійним кроком)

Початок.

I. Вибрати довільне початкове наближення $(x^0,\ y^0) \in X \times Y$, вибрати крок $\rho > 0$ і покласти $k = 0$.

Основний цикл.

II. Обчислити градієнти

$\nabla_x \varphi(x^k,\ y^k)$, $\nabla_y \varphi(x^k,\ y^k)$ і наступні наближення

$$x^{k+1} = \pi_X(x^k + \rho\nabla_x \varphi(x^k,\ y^k)), y^{k+1} = \pi_Y(y^k - \rho\nabla_y \varphi(x^k,\ y^k))$$

за допомогою операторів проектування π_X і π_Y відповідно на множини X і Y.

III. Покласти $k = k + 1$ і перейти до кроку II.

Теорема 2. *Якщо внутрішня у множині $X \times Y$ точка (x^*, y^*) є єдиною сідловою точкою функції $\varphi(x,\ y)$ з іншими похідними і не існує числа λ та ненульового вектора $(u,\ v) \in R^n \times R^m$, які задовольняють умовам*

$$\nabla_{xx}^2 \varphi(x^*,\ y^*)u = 0, \quad \nabla_{yy}^2 \varphi(x^*,\ y^*)v = 0;$$

$$\nabla_{yx}^2 \varphi(x^*,\ y^*)u = \lambda v, \quad \left(\nabla_{yx}^2 \varphi(x^*,\ y^*)\right)^T v = \lambda u,$$

то для довільного числа $\delta > 0$ існує постійний крок $\rho = \rho(\delta) > 0$, з яким алгоритм 2 генерує із початкового наближення $(x^0,\ y^0) \in Z(\delta)$ послідовність $\{(x^k,\ y^k)\}_{k=0}^\infty$, яка зі швидкістю геометричної прогресії збігається до сідлової точки (x^*, y^*).

6.2. Узагальнений градієнтний метод для відшукання сідлових точок

Задача 2. Знайти сідлову точку $(x^*, y^*) \in X \times Y$, яка задовольняє нерівностям

Теорема 3. *Гранична точка* $x^*_{\varepsilon\mu}$ *послідовності* $\{\boldsymbol{x}^{\boldsymbol{k}}\}_{\boldsymbol{k}=0}^{\infty}$, *породженої алгоритмом 3, задовольняє умові* $0 \in \tilde{L}_{\varepsilon\mu}(x^*_{\varepsilon\mu})$ *і є* (ε, μ) *- квазістаціонарною точкою функції* $\max\limits_{i \in \Im} \varphi_i(x)$ *на множині X.*

6. Методи відшукання сідлових точок

6.1. Градієнтні методи відшукання сідлових точок

Задача 1. Знайти точку $(x^*, y^*) \in X \times Y$, яка задовольняє нерівностям сідлової точки

$$\varphi(x, y^*) \le \varphi(x^*, y^*) \le \varphi(x^*, y), \quad \forall x \in X,\ y \in Y,$$

для неперервно диференційованої опуклої уверх по x і опуклої униз по y на множині $X \times Y$ функції $\varphi(x, y)$.

Вважаємо, що функція $\varphi(x, y)$ має непорожню обмежену множину сідлових точок $X^* \times Y^*$ і для довільного $x \in (X \setminus X^*)$ виконується строга нерівність $\varphi(x, y^*) < \varphi(x^*, y^*)$. Цим умовам задовольняють, зокрема, функції Лагранжа для задачі опуклого програмування:

$$\arg\max_{x \in X_1} f_0(x), \quad X_1 = \{x \mid f_i(x) \ge 0, \quad i = 1, \ldots, m, \quad x \in R^n\},$$

із диференційованими (та опуклими) функціями $f_j,\ j = 0, 1, \ldots, m$, якщо функція f_0 строго опукла і виконується умова Слейтера.

Алгоритм 1

Початок.

I. Вибрати початкове наближення $(x^0, y^0) \in X \times Y$ і покласти $k = 0$.

Основний цикл.

II. Вибрати крокові множники ρ_k, які задовольняють умовам теореми 1, обчислити градієнти $\nabla_x \varphi(x^k, y^k)$ та $\nabla_y \varphi(x^k, y^k)$ для функції $\varphi(x, y)$ у точці (x^k, y^k) і обчислити наступні наближення:

$$x^{k+1} = \pi_X(x^k + \rho_k \nabla_x \varphi(x^k, y^k)), \quad y^{k+1} = \pi_Y(y^k - \rho_k \nabla_y \varphi(x^k, y^k)),$$

де π_Q – оператор проектування на множину Q.

III. Якщо умова завершення алгоритму не виконується, то покласти $k = k + 1$ і перейти на крок II.

Теорема. *Якщо x^* є точкою локального мінімуму майже диференційованої функції $f_0(x)$, то обчислена за алгоритмом 1 послідовність* $\{x^k\}_{s=1}^{\infty}$ *задовольняє нерівностям*

$$\left\| x^k - x^* \right\| \le \sigma, \quad \sigma > 0,\ k = 1, 2, \ldots;$$

$$1 + \delta \le \alpha_k \le \bar{\alpha}, \qquad \delta > 0,$$

і існує така підпослідовність $\{\boldsymbol{x}^{k_s}\}_{\boldsymbol{s}=1}^{\infty}$, *що норма вектора* $\tilde{\mathbf{g}}(\boldsymbol{y}_s^k)$ *зменшується, при деякому значенні константи* $\gamma > 0$, *із швидкістю*

$$\| \ \tilde{\mathbf{g}}(\boldsymbol{y}_s^k) \ \| \ \le \gamma \Big(\prod_{j=1}^{k_s} \alpha_j\Big)^{-\frac{1}{n}}, \ s = 1, 2, \ldots$$

Алгоритм 1 можна використати для обчислення розв'язку системи нелінійних рівнянь $f_i(x) = 0,\ i = 1, 2, \ldots n$, як розв'язку задачі оптимізації

$$\min_{\boldsymbol{x}} \max_{1 \le \boldsymbol{i} \le \boldsymbol{n}} |f_i(x)|.$$

2. Градієнтний метод пошуку сідлових точок

Задача. Знайти сідлову точку (x^*, y^*),

$$\varphi(x^*, y^*) = \min_{x \in R^n} \max_{y \in R^m} \varphi(x, y) = \max_{y \in R^m} \min_{x \in R^n} \varphi(x, y),$$

де $\varphi : R^n \times R^m \to R^1$ – задана функція, що задовольняє вимогам: для кожного фіксованого вектора $y \in R^m$ функція $\varphi(x, y)$ опукла і неперервно диференційована по x; для кожного фіксованого вектора $x \in R^n$ функція $\varphi(x, y)$ увігнута по y; множина сідлових точок $X^* \times Y^*$ непорожня.

На k-й ітерації алгоритму рух до наступного наближення (x^{k+1}, y^{k+1}) по змінній x здійснюється у напрямку антиградієнта функції $\varphi(x, y^k)$ по аргументу x, а по змінній y здійснюється у напрямку узагальненого градієнта функції $\varphi(x^k, y)$ по аргументу y.

Алгоритм 1

Початок.

I. Вибрати довільне початкове наближення $(x^0, y^0) \in R^n \times R^m$, вибрати константу $\alpha < \infty$, яка задовольняє умовам теореми 1, і покласти $k = 0$.

Основний цикл.

II. Обчислити у точці (x^k, y^k) градієнт $\nabla_x \varphi(x^k, y^k)$ функції $\varphi(x, y)$ по аргументу x у точці (x^k, y^k).

III. Обчислити в точці (x^k, y^k) узагальнений градієнт $\varphi_y(x^k, y^k)$ функції $\varphi(x, y)$ по аргументу y і обчислити наступні наближення:

$$x^{k+1} = \begin{cases} x^k - \rho_k \varphi_x(x^k, y^k), & \text{якщо } (x^k, y^k) \in S_x \times S_y, \\ x^0, & \text{якщо } (x^k, y^k) \notin S_x \times S_y; \end{cases}$$

$$y^{k+1} = \begin{cases} y^k - \rho_k' \varphi_y(x^k, y^k), & \text{якщо } (x^k, y^k) \in S_x \times S_y, \\ y^0, & \text{якщо } (x^k, y^k) \notin S_x \times S_y, \end{cases}$$

де $S_x \underline{\underline{\Delta}} \{x \mid \|x\| \le \alpha, x \in R^n\}$, $S_y \underline{\underline{\Delta}} \{y \mid \|y\| \le \alpha, y \in R^m\}$.

IV. Покласти $k = k + 1$ і перейти на крок II.

Теорема. *Якщо крокові множники* ρ_k *і* ρ_k' *задовольняють умовам*

$$\rho_k \to +0, \quad \rho_k' \to +0, \quad \rho_k' / \rho_k \to 0 \text{ при } k \to \infty;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k' = \infty,$$

і початкове наближення (x^0, y^0) *належить множині* $S_x \times S_y$ *при такому значенні* $\alpha < \infty$*, що:*

при $x \notin S_x$ *і для всіх* $y \in S_y$ *виконується нерівність*

$$(\varphi_x(x, y), x) > 0;$$

при $y \notin S_y$ *і для всіх* $x \in S_x$ *виконуються нерівності*

$$\max_{\{\hat{\nabla}_y \varphi(x,y)\}} (\varphi_y(x, y), y) < 0;$$

$$\|\varphi_y(x, y)\| \le \beta_1 < \infty, \quad \forall x \in S_x, \quad y \in S_y,$$

то всі граничні точки послідовності $\{y^k\}$ *належать множині* Y^*.

3. Методи розв'язування задач квадратичного програмування

3.1. Метод спряжених градієнтів для мінімізації квадратичної функції при лінійних обмеженнях

Задача. Знайти оптимальне значення

$$x^* = \arg\min_{x \in X} \left[\frac{1}{2}(\boldsymbol{Cx}, \boldsymbol{x}) + (\boldsymbol{d}, \boldsymbol{x}) \right]$$

на допустимій множині $X = \{x \mid (a^i, x) - b_i = 0, \quad i \in \Im\}$ із скінченною множиною індексів $\Im$, лінійно-незалежними векторами a_i та додатньо визначеною матрицею C.

VIII. Покласти $k = k + 1$ і перейти на крок II.

Теорема 2. *Якщо для початкового наближення* x^0 *множина*

$$X^0 \underline{\underline{\Delta}} \left\{ x \left| \max_{i \in \Im} \varphi_i(x) \le \max_{i \in \Im} \varphi_i(x^0), \ x \in X \right. \right\}$$

обмежена, то будь-яка гранична точка нескінченної послідовності $\{x^k\}_{k=0}^{\infty}$*, породженої алгоритмом 2, є стаціонарною точкою функції* $\max_{i \in \Im} \varphi_i(x)$ *на множині X.*

Алгоритм 3

Початок. *I.* Вибрати початкове наближення $x^0 \in X$, константи $\varepsilon > 0$, $\mu > 0$ і покласти $k = 0$.

Основний цикл.

II. За множинами індексів

$$K_\varepsilon(x^k) = \{i \mid \max_{i \in \Im} \varphi_i(x^k) - \varphi_i(x^k) \le \varepsilon, \ i \in \Im\};$$

$$Q_\varepsilon(x^k) = \{j \mid -\mu \le f_j(x^k) \le 0, \ j \in \Im_1\}$$

обчислити множину векторів $\tilde{H}_{\varepsilon\mu}(x^k) = H_\varepsilon(x^k) \cup H_\mu'(x^k)$,

де $H_\varepsilon(x^k) = \{\nabla \varphi_i(x^k), \quad i \in K_\varepsilon(x^k)\}, H_\mu'(x^k) = \{\nabla f_j(x^k), \quad j \in Q_\varepsilon(x^k)\}$.

III. За допомогою алгоритму мінімізації квадратичної функції з обмеженнями знайти найближчу до початку координат точку $z_{\varepsilon\mu}^k$ опуклої оболонки $\tilde{L}_{\varepsilon\mu}(x^k)$ множини $\tilde{H}_{\varepsilon\mu}(x^k)$. Якщо $z_{\varepsilon\mu}^k$ =0, то покласти $x_{\varepsilon\mu}^* = x^k$ і завершити алгоритм 3. В іншому разі обчислити нормований напрямок

$$h_{\varepsilon\mu}(x^k) = - \ z_{\varepsilon\mu}^k / \left\| z_{\varepsilon\mu}^k \right\|$$

(ε, μ) - найшвидшого спуску функції $\max_{i \in \Im} \varphi_i(x)$ в точці x^k.

IV. Обчислити наступне наближення $x^{k+1} = x^k + \rho_k h_{\varepsilon\mu}(x^k)$ за кроком ρ_k, який задовольняє умовам

$$\max_{i \in \Im} \varphi_i(x^k + \rho_k h_{\varepsilon\mu}(x^k)) = \min_{\rho \ge 0} \max_{i \in \Im} \varphi_i(x^k + \rho h_{\varepsilon\mu}(x^k));$$

$$x^k + \rho_k h_{\varepsilon\mu}(x^k) \in X.$$

V. Покласти $k = k + 1$ і перейти на крок II.

$$H_{00}\left(x^k\right) = H_0\left(x^k\right) \cup H_0'\left(x^k\right),$$

$$H_0\left(x^k\right) = \left\{\nabla\varphi_i\left(x^k\right),\ i \in K_0\left(x^k\right)\right\};$$

$$H_0'\left(x^k\right) = \left\{\nabla f_j\left(x^k\right),\ j \in Q_0\left(x^k\right)\right\}.$$

Якщо опукла оболонка $\tilde{L}_{00}\left(x^k\right)$ цих векторів містить нуль-вектор, то покласти $x^* = x^k$ і завершити алгоритм 2.

III. Покласти $s = 0$.

IV. Покласти $\varepsilon = \varepsilon_s$, $\mu = \mu_s$, $\alpha = \alpha_s$.

V. На множинах індексів

$$K_\varepsilon\left(x^k\right) = \left\{i \middle| \max_{i\in\Im}\varphi_i\left(x^k\right) - \varphi_i\left(x^k\right) \le \varepsilon,\ i \in \Im\right\},$$

$$Q_\varepsilon\left(x^k\right) = \left\{j \mid -\mu \le f_j\left(x^k\right) \le 0,\ j \in \Im_1\right\}$$

обчислити множину векторів

$$\tilde{H}_{\varepsilon\mu} = H_\varepsilon\left(x^k\right) \cup H_\mu'\left(x^k\right),$$

де $H_\varepsilon\left(x^k\right) = \left\{\nabla\varphi_i\left(x^k\right),\ i \in K_\varepsilon\left(x^k\right)\right\}$,

$$H_\mu'\left(x^k\right) = \left\{\nabla f_j\left(x^k\right),\ j \in Q_\varepsilon\left(x^k\right)\right\},$$

і обчислити найближчу до початку координат точку $z_{\varepsilon\mu}$ опуклої оболонки $\tilde{L}_{\varepsilon\mu}\left(x^k\right)$ множини $\tilde{H}_{\varepsilon\mu}\left(x^k\right)$. Якщо не виконується нерівність $\left\|z_{\varepsilon\mu}\right\| \ge \alpha$, то покласти $\varepsilon_{s+1} = \varepsilon_s/2$, $\mu_{s+1} = \mu_s/2$, $\alpha_{s+1} = \alpha_s/2$, $\boldsymbol{s} = \boldsymbol{s}+1$ і перейти до кроку IV.

VI. Обчислити вектор $h_{\varepsilon\mu}\left(x^k\right)$ — напрям (ε,μ) - квазінайшвидшого спуску функції $\max\limits_{i\in\Im}\varphi_i\left(x\right)$ в точці x^k за формулою

$$h_{(\varepsilon,\mu)}\left(x^k\right) = -\left(1/\left\|z_{\varepsilon\mu}\right\|\right)z_{\varepsilon\mu}.$$

VII. Обчислити ρ_k з умови

$$\max_{i\in\Im}\varphi_i\left(x^k + \rho_k h_{\varepsilon\mu}\left(x^k\right)\right) = \min_{\rho\ge 0}\max_{i\in\Im}\varphi_i\left(x^k + \rho h_{\varepsilon\mu}\left(x^k\right)\right);$$

і обчислити наступне значення $x^k + \rho_k h_{\varepsilon\mu}\left(x^k\right) \in X$.

Алгоритм 1

Початок.

I. Вибрати довільне початкове наближення $x^0 \in X$.

II. Обчислити матрицю H за формулою $H = A^T(AA^T)^{-1}A$, де А — матриця, рядками якої є n-вимірні вектори-рядки $(a^i)^{\mathrm{T}}$, $i \in \Im$.

III. Обчислити градієнт функції $f_0(x) \underline{\underline{\Delta}} \frac{1}{2}(Cx,x)+(d,x)$ в точці $x = x^0$ за формулою $\nabla f_0(x^0) = Cx^0 + d$.

IV. Обчислити вектор $h^1 = -(I - H)\nabla f_0(x^0)$, де I — одинична матриця. Якщо $h^1 = 0$, то завершити обчислення.

V. Обчислити $\rho_1 = -(\nabla f_0(x^0), h_1)/(h_1, Ch^1)$, $x^1 = x^0 + \rho_1 h^1$.

VI. Покласти $k = 1$, $g^0 = h^1$.

Основний цикл.

VII. Обчислити n-вимірний вектор $g^k = (I - H)\nabla f_0(x^k)$, де $\nabla f_0(x^k) = Cx^k + d$.

Якщо $g^k = 0$, то завершити обчислення.

VIII. Обчислити n-вимірний вектор $h^{k+1} = -g^k + \dfrac{\left\|g^k\right\|^2}{\left\|g^{k-1}\right\|^2}h^k$.

IX. Обчислити

$\rho_{k+1} = -(\nabla f_0(x^k),\ h^{k+1})/(h^{k+1},\ Ch^{k+1})$, $x^{k+1} = x^k + \rho_{k+1}h^{k+1}$, покласти $k = k + 1$ і перейти на крок VII.

Теорема. *Якщо функція $f_0(x) \underline{\underline{\Delta}} \frac{1}{2}(Cx,x)+(d,x)$ має на множині X скінченний мінімум, то послідовність $\{x^k\}$, породжена алгоритмом 1, збігається до точки мінімуму x^* задачі 1 за скінченну кількість ітерацій $k \le n$.*

3.2. Метод спряжених градієнтів для задачі квадратичного програмування з обмеженнями, заданими нерівностями

Задача. Для квадратичної функції $f_0(x) \underline{\underline{\Delta}} \frac{1}{2}(Cx,x)+(d,x)$ з додатньо визначеною матрицею C і заданим вектором d знайти $\arg\min\limits_{x\in X}\left[\frac{1}{2}(\boldsymbol{Cx},\boldsymbol{x})+(\boldsymbol{d},\boldsymbol{x})\right]$, де $X \underline{\underline{\Delta}} \{x \mid x_j \ge 0,\ i \in \Im^+\}$, $\Im^+$ — задана підмножина множини $\{1, 2, ..., n\}$.

Алгоритм 2

Початок.

I. Обрати довільне початкове наближення $x^0 \in X$.

II. Знайти підмножину $\Im(x^0)$, де $\Im(x) \underline{\underline{\Delta}} \{i \mid x_i = 0,\ i \in \Im^+\}$.

III. Обчислити градієнт $\nabla f_0(x^0) \underline{\underline{\Delta}} \left(\frac{\partial f_0(x^0)}{\partial x_1}, \ldots, \frac{\partial f_0(x^0)}{\partial x_n} \right)^T$.

IV. Якщо $\frac{\partial f_0(x^0)}{\partial x_i} = 0$ для всіх i, що не належать підмножині $\Im(x^0)$, то перейти на крок V; інакше перейти на крок XVI.

V. Якщо $\frac{\partial f_0(x^0)}{\partial x_i} = 0$ для всіх i із підмножини $\Im(x^0)$, то покласти $x^* = x^0$ і завершити алгоритм 2; інакше покласти

$\Im'(x^0) = \left\{ i \mid \frac{\partial f_0(x^0)}{\partial x_i} \geq 0, \quad i \in \Im(x^0) \right\}$ і перейти на крок VI.

VI. Покласти $h^1 = -\nabla f_0(x^0)$.

VII. Покласти $k = 0$ і перейти на крок X.

Основний цикл.

VIII. Обчислити $\nabla f_0(x^k) \underline{\underline{\Delta}} \left(\frac{\partial f_0(x^k)}{\partial x_1}, \ldots, \frac{\partial f_0(x^k)}{\partial x_n} \right)^T$. Якщо $\forall i \notin \Im'(x^0)$ $\frac{\partial f_0(x^k)}{\partial x_i} = 0$, то покласти $x^0 = x^k$ і перейти на крок II; інакше перейти на крок IX.

IX. Обчислити вектор $h^{k+1} = -\nabla f_0(x^k) + \frac{\left\| \nabla f_0(x^k) \right\|^2}{\left\| \nabla f_0(x^{k-1}) \right\|^2} h^k$.

X. Обчислити $\rho_{k+1} = -(\nabla f_0(x^k), h^{k+1}) / (h^{k+1}, Ch^{k+1})$.

XI. Визначити $\overline{\Im}(x^0)-$ множину всіх $i \notin \Im'(x^0)$, для яких $h_i^{k+1} < 0$.

XII. Обчислити $\overline{\rho}_{k+1} = \min\limits_{i \in \overline{\Im}(x^0)} (-x_i^k / h_i^{k+1})$.

XIII. Якщо $\rho_{k+1} < \overline{\rho}_{k+1}$, то покласти

5.2. Метод розв'язування мінімаксної задачі з обмеженнями, заданими нерівностями

Задача 2. Знайти $\arg \min\limits_{x \in X} \max\limits_{i \in \Im} \varphi_i(x)$ на множині

$X \underline{\underline{\Delta}} \left\{ x \middle| f_j(x) \leq 0, j \in \Im_1,\ x \in R^n \right\}$. Розглядається випадок випуклих неперервно диференційованих функцій f_j, $j \in \Im_1$, для яких виконується умова Слейтера $\inf\limits_{x \in R^n} \max\limits_{i \in \Im_1} f_i(x) < 0$, та неперервно диференційованих функцій $\varphi_i : R^n \to R^1,\ i \in \Im$.

Точку $x^* \in X$ називається стаціонарною точкою функції $\max\limits_{i \in \Im} \varphi_i(x)$ на множині X, якщо многогранник $\tilde{L}(x^*)$:

$$\tilde{L}(x^*) = \operatorname{co} H^0(x^*),\ H^0(x^*) = H(x^*) \bigcup H_1(x^*);$$

$$H(x^*) = \left\{ \nabla \varphi_i(x^*),\ i \in K_0(x^*) \right\};$$

$$H_1(x^*) = \left\{ \nabla f_j(x^*),\ j \in Q_0(x^*) \right\};$$

$$K_0(x^*) = \left\{ i \middle| \max_{i \in \Im} \varphi_i(x^*) = \varphi_i(x^*),\ i \in \Im \right\};$$

$$Q_0(x^*) = \left\{ j \mid f_j(x^*) = 0,\ j \in \Im_1 \right\})$$

містить початок координат.

Якщо функція $\max\limits_{i \in \Im} \varphi_i(x)$ є опуклою вниз на множині X, то стаціонарна точка x^* є точкою глобального мінімуму.

Алгоритм 2

Початок.

I. Вибрати довільне початкове наближення $x^0 \in X$, додатні константи $\varepsilon_0 > 0$, $\mu_0 > 0$, $\alpha_0 > 0$ і покласти $k = 0$.

Основний цикл.

II. На множинах індексів

$$K_0(x^k) = \left\{ i \mid \varphi_i(x^k) = \max_{i \in \Im} \varphi_i(x^k),\ i \in \Im \right\};$$

$$Q_0(x^k) = \left\{ j \mid f_j(x^k) = 0,\ j \in \Im_1 \right\}$$

обчислити множину векторів

обчислити $\psi(x^k)=\min\limits_{x\in X^k}\max\limits_{i\in K_0(x^k)}\left(\nabla\varphi_i(x^k),\ x-x^k\right)$. Якщо $\psi(x^k)=0$, то покласти $x^*=x^k$ і завершити алгоритм 1.

III. Покласти $j=0$.

IV. Покласти $\varepsilon=\varepsilon_j$.

На множині $K_\varepsilon(x^k)=\left\{i\mid \max\limits_{i\in\Im}\varphi_i(x^k)-\varphi_i(x^k)\le\varepsilon,\ \ i\in\Im\right\}$ обчислити $\psi_\varepsilon(x^k)=\min\limits_{x\in X^k}\max\limits_{i\in K_\varepsilon(x^k)}\left(\nabla\varphi_i(x^k),\ x-x^k\right)$.

Якщо не виконується нерівність $\psi_\varepsilon\left(x^k\right)\le-\alpha_0\varepsilon/\varepsilon_0$, то покласти $\varepsilon_{j+1}=\varepsilon_j/2,\ \ j=j+1$ і перейти на крок IV.

V. Обчислити вектор y^k, який задовольняє умову

$$\psi_\varepsilon\left(x^k\right)=\max_{i\in K_\varepsilon\left(x^k\right)}\left(\nabla\varphi_i\left(x^k\right),y^k-x^k\right).$$

VI. Обчислити крок ρ_k із умови

$$\max_{i\in\Im}\varphi_i\left(x^k+\rho_k\left(y^k-x^k\right)\right)=\min_{\rho\in[0,1]}\max_{i\in\Im}\varphi_i\left(x^k+\rho_k\left(y^k-x^k\right)\right).$$

VII. Обчислити наступне наближення

$$x^{k+1}=x^k+\rho_k\left(y^k-x^k\right).$$

VIII. Покласти $k=k+1$ і перейти на крок IV.

Теорема 1. Якщо для початкового значення x^0 множина

$$X\left(x^0\right)=\left\{x\left|\max_{i\in\Im}\varphi_i\left(x\right)\le\max_{i\in\Im}\varphi_i\left(x^0\right),\ \ x\in X\right.\right\}$$

обмежена, то гранична точка x^* послідовності $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$, породженої алгоритмом 1, є стаціонарною точкою функції $\max\limits_{i\in\Im}\varphi_i\left(x\right)$ на множині X.

Якщо множина X^k є строго опуклою і початок координат не належить опуклій оболонці $L_\varepsilon\left(x^k\right)$, яка натягнута на вектори $\nabla\varphi_i\left(x^k\right)$, $i\in K_\varepsilon\left(x^k\right)$, то задача обчислення $\psi_\varepsilon\left(x^k\right)$ зводиться до задачі максимізації на $L_\varepsilon\left(x^k\right)$ неперервно диференційованої функції $\theta\left(z\right)\triangleq\min\limits_{v\in X^k}\left(z,v-x^k\right)$ з градієнтом $\nabla\theta\left(z\right)=v\left(z\right)-x^k$, де $v\left(z\right)\in X^k$ задовольняє умові $\theta\left(z\right)=\left(z,v\left(z\right)-x^k\right)$.

$$x_i^{k+1}=x_i^k+\rho_{k+1}h_i^{k+1},\ \ i\notin\Im'(x^0);$$

$$x_i^{k+1}=x_i^k=0,\ \ i\in\Im'(x^0),$$

і перейти на крок XIV; інакше покласти

$$x_i^{k+1}=x_i^k+\bar\rho_{k+1}h_i^{k+1},\ \ i\notin\Im'(x^0);$$

$$x_i^{k+1}=x_i^k=0,\ \ i\in\Im'(x^0),$$

і перейти на крок XV.

XIV. Покласти $k=k+1$ і перейти на крок VIII.

XV. Покласти $x^0=x^{k+1}$ і перейти на крок II.

XVI. Покласти $\Im'(x^0)=\Im(x^0)$ і перейти на крок VI.

3.3. Модифікація методу спряжених напрямків для задач квадратичного програмування великої розмірності

Задача. Знайти мінімальне значення функції

$$f_0(x)=\left(\frac{1}{2}(x,Cx)+(d,x)\right),$$

при обмеженнях $(a^j,x)=b_j,\ \ j=1,\dots,m;\ \ \upsilon_i\le x_i\le\omega_i,\ \ i=1,\dots,n,$ де $C-n\times n$-матриця $(C\ge0)$; $d-n$-вимірний вектор; $a^j=(a_1^j,\dots,a_n^j);\ \ b_j,\ \ j=1,\dots,m;\ \upsilon_i,\ \ \omega_i,\ \ i=1,\dots,n-$ дійсні числа.

Наведений нижче алгоритм 3 орієнтований на розв’язування задач з великою кількістю змінних x_i, $i=1,\dots,n$, і меншою кількістю обмежень m. На кроках XII—XIV мінімізуємо (не більше ніж за $n-m-s$ ітерацій, s — кількість елементів у множині $\Im^+\bigcup\Im^-$.) квадратичну функцію $f_0(x)$ на підпросторі, заданому рівняннями:

$$x_i^k=\omega_i,\ \ i\in\Im^+;\ \ x_i^k=\upsilon_i,\ \ i\in\Im^-,(a^j,x)=b_j,\ \ j=1,\dots,m.$$

Алгоритм 3

Початок.

I. Вибрати довільне початкове наближення x^0, яке задовольняє нерівностям $\upsilon_i\le x_i\le\omega_i,\ \ i=1,\dots,n$.

II. Сформувати $m\times n$-матрицю A із рядками $a^j,\ \ j=1,\dots,m.$

III. Покласти $k=0$.

Основний цикл.

IV. Обчислити множину індексів $\Im=\Im^+\bigcup\Im^-$, де $\Im^{+}=\{i \mid x_i^k \ge \omega_i,\ \ i=1,\dots,n\}$, $\Im^{-}=\{i \mid x_i^k \le \upsilon_i,\ \ i=1,\dots,n\}$.

V. Обчислити градієнт $\nabla f_0(x^k)=Cx^k+d$.

VI. Сформувати матрицю $\tilde{\boldsymbol{B}}$, яка утворюється із матриці B заміною всіх рядків з номерами $j\in \Im^{+}\bigcup \Im^{-}$ нульовими рядками, і сформувати матриці B_{+} (B_{-}), рядками яких є рядки матриці B з номерами $j\in \Im^{+}$ ($j\in \Im^{-}$).

VII. Для довільного n-вимірного вектора-стовпчика g визначити оператора:

$$Q_{\Im}g=\overline{g}-\tilde{A}^{T}(\tilde{A}\tilde{A}^{T})^{-1}\tilde{A}\tilde{g};\ G_{\Im}g=\begin{pmatrix}G_{\Im}^{+}g\\ G_{\Im}^{-}g\end{pmatrix},$$

де $G_{\Im}^{+}g=(A_{+})^{T}(\tilde{A}\tilde{A}^{T})^{-1}\tilde{A}\tilde{g}-g_{+};\ \ G_{\Im}^{-}g=g_{-}-(A_{-})^{T}(\tilde{A}\tilde{A}^{T})^{-1}\tilde{A}\tilde{g}.$

VIII. Обчислити вектори $Q_{\Im}\nabla f_0(x^k),\ G_{\Im}\nabla f_0(x^k)$ і покласти $q=G_{\Im}\nabla f_0(x^k)$.

IX. Якщо $Q_{\Im}\nabla f_0(x^k)\neq 0,$ то перейти на крок XII; інакше перейти на крок X.

X. Якщо для всіх $i = 1, ..., n$ виконується нерівність $q_i \ge 0,$ то завершити алгоритм 3 (у цьому разі x^k є наближеним розв'язком задачі).

XI. Оновити множини $\Im^{+}$ і $\Im^{-}$ видаленням із них одного з номерів i, для яких $q_i < 0$ (зі зміною множин $\Im^{+}$ і $\Im^{-}$ змінюються також і оператори $Q_{\Im},\ G_{\Im}$).

XII. Покласти $s=0$, $y^0=x^k,\ \ h^0=0.$

XIII. Обчислити

$$h^{s+1}=-Q_{\Im}\nabla f_0(y^s)+\left(\left\|Q_{\Im}\nabla f_0(y^s)\right\|^2/\left\|Q_{\Im}\nabla f_0(y^{s-1})\right\|^2\right)h,$$

$$\rho_{s+1}=-(\nabla f_0(y^s),h^{s+1})/(h^{s+1},Ch^{s+1}).$$

Якщо $\rho_{s+1}=0,$ то покласти $x^{k+1} = y^s$ і перейти на крок XV; інакше перейти на крок XIV.

XIV. Обчислити $\overline{\rho}_{s+1}=\min\left\{\min\limits_{h_i^{s+1}>0}\dfrac{\omega_i-y_i^k}{h_i^{s+1}},\min\limits_{h_i^{s+1}>0}\dfrac{y_i^k-\upsilon_i}{h_i^{s+1}}\right\}$. Якщо $\rho_{s+1}<\overline{\rho}_{s+1}$, то покласти $y^{s+1}=y^s+\rho_{s+1}h^{s+1},\ s=s+1,$ і перейти на крок XIII; інакше покласти $x^{k+1}=y^s+\overline{\rho}_{s+1}h^{s+1}$.

то покласти $i = i + 1$ і перейти на крок VII.

IX. Обчислити наступне наближення $x^{k+1}=x^k+\rho_k h(x)$, покласти $k = k + 1$ і перейти на крок II.

Теорема 1. *Якщо задача квадратичного програмування*

$$\arg\min_{\boldsymbol{h}}\left((\nabla \boldsymbol{f}_0(\boldsymbol{x}),\boldsymbol{h})+\frac{1}{2}(\boldsymbol{h},\boldsymbol{h})\right),\ (\nabla f_j(x),h)+f_j(x)\le 0,\quad j\in\Im_{\delta}(x),$$

має розв'язок h^* при будь-якому x із обмеженої множини

$$X_{\alpha}=\{x\big|\,f_0(x)+\alpha\max_{j\in\Im}f_j(x)\le f_0(x^0)+\delta\max_{j\in\Im}f_j(x^0),\ \ x\in\Re^n\},$$

існують множники Лагранжа $\lambda_j(x),\ \ j\in\Im_{\delta}(x)$, $\sum\limits_{j\in\Im_{\delta}(x)}\lambda_j(x)\le\alpha$ і для деякого індекса $j \in \Im$ $f_j(x) = 0$, то утворена алгоритмом 1 послідовність $\{\boldsymbol{x}^k\}_{k=0}^{\infty}$ має граничну точку x^*, яка задовольняє необхідним умовам мінімуму функції f_0 на множині X.

Якщо у точці x^* виконуються достатні умови локального мінімуму і кількість індексів у множині $\Im_0(x^*)$ дорівнює n, то нескінченна послідовність $\{x^k\}_{\kappa=0}^{\infty}$, утворена алгоритмом 1′ при $\rho=1$, збігається в деякому околі точки x^* із квадратичною швидкістю $\| x^{k+1}-x^{*}\|\le\beta_2\| x^{k}-x^{*}\|^2,\quad \beta_2<\infty$.

5. Методи мінімаксної оптимізації

5.1. Методи відшукання стаціонарної точки мінімаксу

Задача 1. Знайти $\arg\min\limits_{x\in X}\max\limits_{i\in\Im}\varphi_i(x)$ для заданих неперервно диференційованих функцій $\varphi_i : X\to R^1$, заданої допустимої опуклої і замкненої множини $X \subset R^n$ і заданої множини індексів $i \in \Im$.

Алгоритм 1

Початок.

I. Вибрати довільне початкове наближення $x^0 \in X$, константи $\varepsilon_0>0$, $\alpha_0>0$ і покласти $k = 0$.

Основний цикл.

II. На множині $X^k=\left\{x\left\|x-x^k\right\|_1\le 1,\ x\in X\right\},\ \left\|y\right\|_1=\max\limits_{i\in[1:n]}\left|y_i\right|$ і на множині індексів $K_0(x^k)=\left\{i \mid \varphi_i(x^k)=\max\limits_{i\in\Im}\varphi_i(x^k),\ i\in\Im\right\}$

за лінійних обмежень

$$\hat{f}_j(x)\underline{\underline{\Delta}} f_j(x^k)+(\nabla f_j(x^k), x-x^k)\le \varepsilon_k, \quad j\in \Im_{\delta_k}.$$

Квадратичний доданок $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^k\|^2$ гарантує існування розв'язку $z(\varepsilon,\delta_k)$ допоміжної задачі на непорожній допустимій множині. Покращене $(k+1)$ -ше наближення x^{k+1} обчислюється за формулою $x^{k+1}=x^k+\rho_k(z(\varepsilon,\delta_k)-x^k)$. Різні способи визначення довжини кроку ρ_k і констант ε,δ_k визначають різні варіанти методів лінеаризації.

Задача.

Знайти оптимальне значення $\arg\min\limits_{x\in X} f_0(x)$ для функції $f_0:\Re^n\to\Re^1$ і для допустимої множини $X=\{x\,|\,f_j(x)\le 0,\ j\in\Im,\ x\in\Re^n\}$ з ліпшицевими градієнтами функцій $f_j:\Re^n\to\Re^1$.

Алгоритм 1

Початок.

I. Вибрати довільне початкове значення $x^0\in\Re^n$, покласти $k=0$, вибрати константу $0<\varepsilon<1$ і вибрати константи $\alpha>0$, $\delta>0$ за умовами теореми 1.

Основний цикл.

II. Покласти $x=x^k$.

III. Знайти множину індексів:

$$\Im_\delta(x)=\{j\,|\,f_j(x)\ge\max_{j\in\Im} f_j(x)-\delta, \quad j\in\Im\}.$$

IV. Знайти розв'язок $h=h(x)$ задачі квадратичного програмування:

$$\arg\min_{h}\left((\nabla \boldsymbol{f}_0(\boldsymbol{x}),\boldsymbol{h})+\frac{1}{2}\|\boldsymbol{h}\|^2\right),$$

$$\left(\nabla f_j(x),h\right)+f_j(x)\le 0, \quad j\in\Im_\delta(x).$$

V. Якщо $h(x)=0$, то покласти $x^*=x$ і завершити алгоритм.

VI. Покласти $i=0$.

VII. Покласти $\rho_k=\left(\frac{1}{2}\right)^i$.

VIII. Якщо не виконується нерівність

$$f_0(x+\rho_k h(x))+\alpha\max_{j\in\Im}\{0, f_j(x+\rho_k h(x))\}\le$$

$$\le f_0(x)+\alpha\max_{j\in\Im}\{0, f_j(x)\}-\rho_k\varepsilon\|h(x)\|^2,$$

XV. Покласти $k=k+1$ і перейти на крок IV.

Теорема. *Існують константи c_1, $c_2>0$, для яких побудована алгоритмом 3 послідовність y^s задовольняє нерівностям:*

$$y_i^s-\omega_i\le c_1u; \quad \upsilon_i-y_i^s\le c_1u;$$

$$\left\|Ay^s-b\right\|\le c_2u(1+\lambda^s)\left\|A\right\|+\left\|Ay^0-b\right\|,$$

де $u=2^{1-\tau}$ *— одинична похибка округлення* (τ *- кількість розрядів у мантисі*); λ *— константа, що задовольняє нерівностям*

$$\frac{1}{4}(\mathrm{cond}\,\tilde{C}+(\mathrm{cond}\,\tilde{C})^{-1}-2)\le\lambda\le\frac{1}{2}(\mathrm{cond}\,\tilde{C}+(\mathrm{cond}\,\tilde{C})^{-1}-2),$$

$$\tilde{C}\underline{\underline{\Delta}}\left\|\tilde{C}\right\|\cdot\left\|\tilde{C}^{-1}\right\|.$$

3.4. Стійкий алгоритм для розв'язування задач квадратичного програмування

Задача. Знайти розв'язок $x^*=\arg\min\limits_{x}\frac{1}{2}\left\|Gx-c\right\|^2$ при обмеженнях $Ax=b$, $x\ge 0$.

У наведеному далі алгоритмі 4 за скінченну кількість ітерацій обчислюється оптимальний розв'язок або встановлюється відсутність допустимих розв'язків.

Алгоритм 4

Початок.

I. Знайти таку множину індексів $\Im\subseteq\{1,2,\ldots,n\}$, щоб матриця $\begin{pmatrix}\boldsymbol{B}_\Im\\ \boldsymbol{Q}_\Im\end{pmatrix}$ ($B_\Im-$ підматриця матриці A, складена зі стовпців a^j, $j\in\Im$, матриці A; $Q_\Im-$ підматриця матриці G, складена із стовпців g^j, $j\in\Im$, матриці G) мала повний ранг за стовпцями, і обчислити розв'язок (y,u) системи лінійних рівнянь $B_\Im y=b$; $B_\Im^T u+Q_\Im^T Q_\Im y=Q_\Im^T c$.

Основний цикл.

II. Якщо $y>0$ то перейти на крок III; інакше перейти на крок VII.

III. Якщо для всіх $j\in\{1,\ldots,n\}\setminus\Im$ виконується нерівність

$$(a^j,u)+(g^j,Q_\Im y-c)\ge 0,$$

то обчислити оптимальний розв'язок $x^* = (x_1^*, \dots, x_n^*)$ за формулами:

$$x_j^* = \begin{cases} y_j, & \text{якщо} \quad j \in \Im, \\ 0, & \text{якщо} \quad j \notin \Im, \end{cases} \quad j = 1, \dots, n$$

і завершити алгоритм 4.

IV. Знайти індекс $k \in \{1, \dots, n\} \setminus \overline{\Im}$, для якого виконується нерівність $(a^k, u) + (g^k, Q_\Im y - c) < 0$, і покласти $\Im = \overline{\Im} \cup \{k\}$.

V. Якщо $i \in \overline{\Im}$, то покласти $\overline{y}_i = y_i$. Для інших $i \in \Im$ покласти $\overline{y}_k = 0$ і перейти на крок IX.

VI. Обчислити максимальне значення $\overline{\lambda} \in [0,1]$, при якому $\overline{\lambda} y + (1 - \overline{\lambda}) \overline{y} \geq 0$, і покласти $z_j = \overline{\lambda} y_j + (1 - \overline{\lambda}) \overline{y}_j, \quad j \in \Im$.

VII. Покласти $\overline{\Im} = \Im$.

VIII. Знайти підмножину $\tilde{\Im}$, що складається з тих індексів $k \in \overline{\Im}$, для яких $z_k = 0$, покласти $\Im = \overline{\Im} \setminus \tilde{\Im}$ і для всіх $i \in \Im$ покласти $\overline{y}_j = z_j$.

IX. За множиною $\Im$ скласти підматрицю $B_\Im$ матриці *A* і підматрицю $Q_\Im$ матриці *G*, обчислити розв'язок (y, u) системи лінійних рівнянь

$$B_\Im y = b;$$
$$B_\Im u + Q_\Im^T Q_\Im y = Q_\Im^T c$$

і перейти на крок II.

Теорема. *Якщо існує множина індексів* $\Im \subseteq \{1, \dots, n\}$, *для якої матриця* $\begin{pmatrix} \boldsymbol{B}_\Im \\ \boldsymbol{Q}_\Im \end{pmatrix}$ *має повний ранг за стовпцями і матриці* $B_\Im$ *мають повний ранг за рядками, то за скінченну кількість ітерацій алгоритм 4 приводить до оптимального розв'язку x*.*

Для обчислення матриці $\begin{pmatrix} \boldsymbol{B}_\Im \\ \boldsymbol{Q}_\Im \end{pmatrix}$ на кроці I рекомендується спочатку обчислити $(\tilde{t}^*, \tilde{x}^*) = \arg\min_{(\tilde{t}, \tilde{x})} \left\| (-b \vdots A) \begin{pmatrix} \tilde{t} \\ \tilde{x} \end{pmatrix} \right\|^2$ при обмеженнях $\tilde{t} = 1, \ (\tilde{t}, \tilde{x}) \geq 0$. Якщо виконується рівність $\left\| (-b \vdots A) \begin{pmatrix} \tilde{t}^* \\ \tilde{x}^* \end{pmatrix} \right\|^2 > 0$, то задача не має розв'язку; якщо ж $\left\| (-b \vdots A) \begin{pmatrix} \tilde{t}^* \\ \tilde{x}^* \end{pmatrix} \right\|^2 = 0$, то $\tilde{x}^*$ задовольняє умовам $\tilde{x}^* \geq 0$ і $A\tilde{x}^* = b$. Якщо $\tilde{x}^* \neq 0$, то множина стовпців a^j, для яких $\tilde{x}_j^* > 0$, лінійно-незалежна і разом із відповідними стовпцями g^j утворює матрицю $\begin{pmatrix} \boldsymbol{B}_\Im \\ \boldsymbol{Q}_\Im \end{pmatrix}$.

Якщо матриця $\begin{pmatrix} \boldsymbol{B}_\Im \\ \boldsymbol{Q}_\Im \end{pmatrix}$ має розмірність $(m + l) \times r$, то y і u визначаються за допомогою розв'язування трикутних лінійних систем

$$S(b - u) = h; \quad Vy = d;$$
$$d = W_Q^T c + Zh; \quad h = S^{-T} b - Z^T W_Q^T c; \quad W_B^T = ZS,$$

де $Z -$ $r \times m$-матриця з ортогональними стовпцями, $S -$ верхня трикутна невироджена $m \times m$-матриця, $\begin{pmatrix} B_\Im \\ Q_\Im \end{pmatrix} = WV$, $W - (m + l) \times r$-матриця з ортогональними стовпцями, $V -$ верхня трикутна невироджена $r \times r$-матриця, $W = \begin{pmatrix} W_B \\ W_Q \end{pmatrix}$, W_B має m рядків, а W_Q має l рядків.

4. Методи лінеаризації для розв'язування задач нелінійного програмування

Методи лінеаризації використовуються для відшукання розв'язку x^* задачі мінімізації нелінійної функції $f_0(x)$ за обмежень, заданих нелінійними рівняннями $f_j(x) = 0, \ j = 1, \dots, m$, та нелінійними нерівностями $f_j(x) \leq 0, \ m + l$.

У методах лінеаризації розв'язок x^* знаходиться як границя послідовності, $x^0, x^1, \dots, x^k, x^{k+1}, \dots$, де $(k + 1)$-ше наближення x^{k+1} обчислюється як розв'язок допоміжної задачі мінімізації квадратичної функції

$$\hat{\boldsymbol{f}}_0(\boldsymbol{x}) \triangleq (\nabla \boldsymbol{f}_0(\boldsymbol{x}^k), \boldsymbol{x}) + \frac{1}{2} \| \boldsymbol{x} - \boldsymbol{x}^k \|^2$$