

## 5. Методи нульового порядку багатомірної оптимізації

Методи нульового порядку застосовуються у тих випадках, коли з якихось причин визначення градієнта цільової функції неможливе. Вони використовуються у тих випадках, коли функція  $f_0$  задана алгоритмічно, зокрема, коли для обчислення значень функції за тих чи інших значеннях аргументу потрібно провести натурний або числовий експеримент. Ці методи придатні, коли функція  $f_0$  недиференційована або її значення визначаються з похибкою, що призводить до великих неточностей при обчисленні похідних.

Методи прямого пошуку є методами, у яких використовуються лише значення функції.

Розглянемо функцію двох змінних. Її лінії постійного рівня наведені на рис. 5.1, а мінімум знаходиться в точці  $(u_1^*, u_2^*)$ . Найпростішим методом пошуку є *метод покоординатного спуску*. Із точки **A** ми робимо пошук мінімуму вздовж напрямку осі  $u_1$  і, таким чином, знаходимо точку **B**, в якій дотична до лінії постійного рівня паралельна осі  $u_1$ . Потім, роблячи пошук із точки **B** у напрямку осі  $u_2$ , отримуємо точку **C**, роблячи пошук паралельно осі  $u_1$ , отримуємо точку **D**, і т.д. Таким чином, ми приходимо до оптимальної точки. Будь-який із одномірних методів, описаних у попередній главі, може бути використаний тут для пошуку вздовж осі. Таким чином цю ідею можна застосувати для функцій  $n$  змінних.

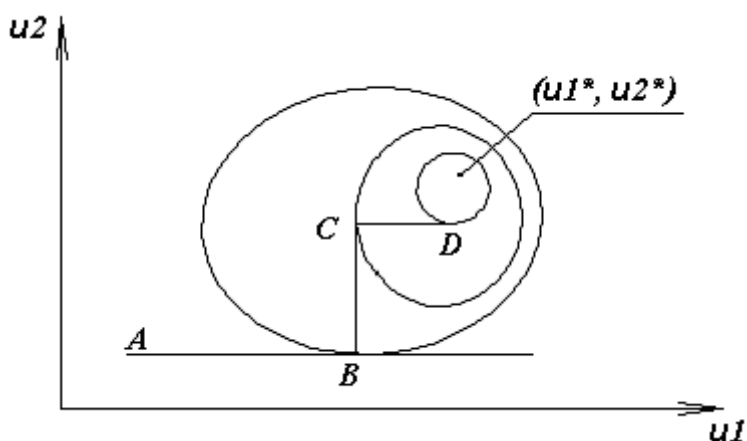


Рис. 5.1. Траєкторія руху методом по координатного спуску

Теоретично даний метод ефективний у випадку єдиного мінімуму функції. Але на практиці він виявляється занадто повільним. Тому були розроблені більш складні методи, що використовують більше інформації на підставі вже отриманих значень функції.

Було запропоновано кілька функцій, які через свої властивості є тестовими для таких методів. Нижче наведено кілька прикладів таких функцій.

Функція Розенброка:

$$f(u_1, u_2) = 100(u_2 - u_1^2)^2 + (1 - u_1)^2; \quad u^* = (1; 1). \quad (5.1)$$

Функція Пауела:

$$f(u) = (u_1 + 10u_2)^2 + 5(u_3 - u_4)^2 + (u_2 - 2u_3)^4 + 10(u_1 - u_4)^2; \quad u^* = (0; 0; 0; 0). \quad (5.2)$$

Двовимірний експонентна функція:

$$f(u_1, u_2) = \sum_a \left[ (e^{-au_1} - e^{-au_2}) - (e^{-a} - e^{-10a}) \right], \quad \text{де } a = 0,1 (0,1) 1^*; \quad u^* = (1; 10). \quad (5.3)$$

Будь-яка серйозна оптимізаційна процедура повинна ефективно вирішувати задачі (5.1), (5.2), (5.3) та інші тестові задачі.

### 5.1. Метод Гауса-Зейделя.

Розглянута вище стратегія пошуку покладена в основу методу Гауса-Зейделя. Так, для задачі  $f(\vec{u}) \rightarrow \min_{\vec{u}}$ , де  $\vec{u} \in R^n$  алгоритм пошуку можна записати таким чином:

$$u_i^{onm} = \arg \min_{u_i} f(u_1^{onm}, u_2^{onm}, \dots, u_i, \dots, u_{n-1}, u_n)$$

Основна ідея цього методу полягає в тому, що пошук екстремуму здійснюється за допомогою почергової зміни незалежних змінних вздовж однієї з координатних осей.

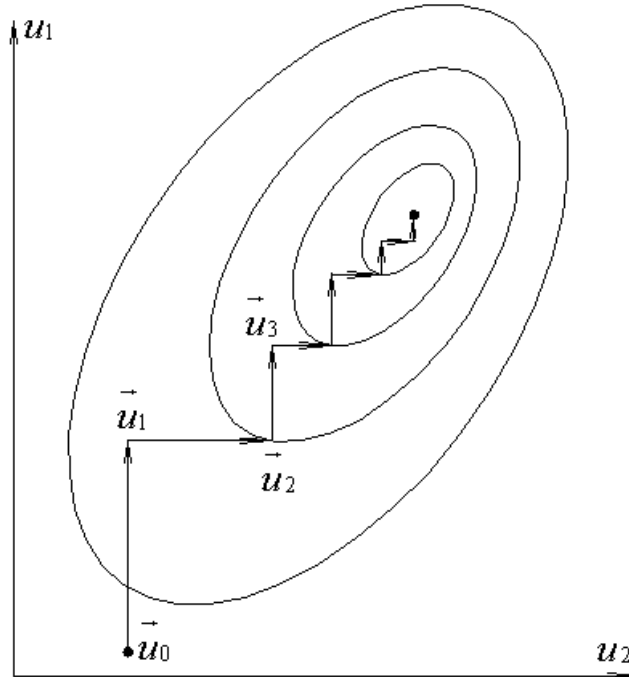


Рис. 5.2. Траєкторія руху методом Гауса-Зейделя

Порядок дій згідно алгоритму можна записати так:

1) Вибір вихідної точки  $\vec{u}_k$ ,  $k=0$  і першого напрямку пошуку  $\vec{e}_j$ ,  $j=1, \dots, m$ , наприклад,  $\vec{e}_1 = \{1, 0, 0, \dots, 0\}^T$  – одиничний вектор, що збігається у напрямку з віссю  $u_1$ .

2) Визначення  $\vec{u}_{k+1} = \vec{u}_k + h_k \vec{e}_j$ , де  $h_k$  – кроковий множник, обраний з умови мінімізації  $f_0(\vec{u})$  по напрямку  $\vec{e}_j$ .

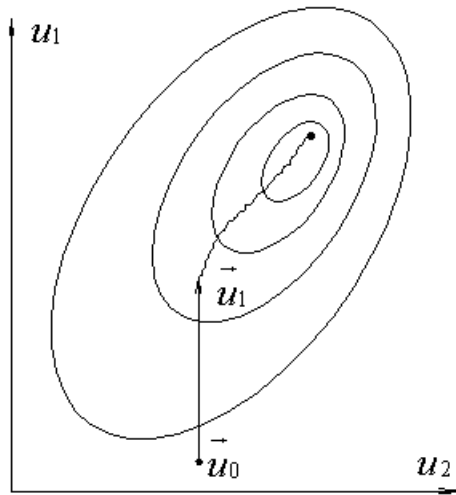
3) Якщо  $j < n$ , де  $n$  – число незалежних змінних, то присвоюється  $k=j$ ,  $j=j+1$  і виконується пункт 2.

Якщо  $j=n$ , то присвоюється  $\vec{u}_k = \vec{u}_0$ ,  $k=0$  і виконується пункт 1.

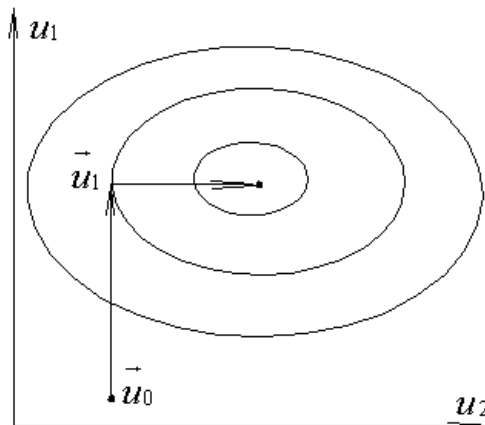
Траєкторія руху цим методом для функції двох змінних зображена на рис. 4.6.

Процес пошуку припиняється, якщо на жодному з напрямків уздовж координатних осей не спостерігається зменшення цільової функції.

Ефективність методу Гауса-Зейделя істотно залежить від спрямованості ліній рівня цільової функції. Так, для цільової функції, зображеної на рис. 5.3, застосування методу Гауса-Зейделя неефективно, а для цільової функції, зображеної на рис. 5.4, метод Гауса-Зейделя забезпечує знаходження екстремуму всього за два кроки.



*Рис. 5.3. Випадок неефективного використання метода Гауса-Зейделя*



*Рис. 5.4. Випадок ефективного використання метода Гауса-Зейделя*

## 5.2. Метод Розенброка.

Метод Розенброка заснований на тому, що замість безперервного пошуку по координатах, що відповідають напрямкам незалежних змінних, після кожного циклу координатного пошуку здійснюється поворот системи координат таким чином, що одна з координат збігається з напрямком усього кроку в попередньому циклі координатного пошуку.

Метод Розенброка по суті своїй зводиться до знаходження такої системи координат, у якій напрямки пошуку витягуються вздовж головних осей квадратичної апроксимації цільової функції.

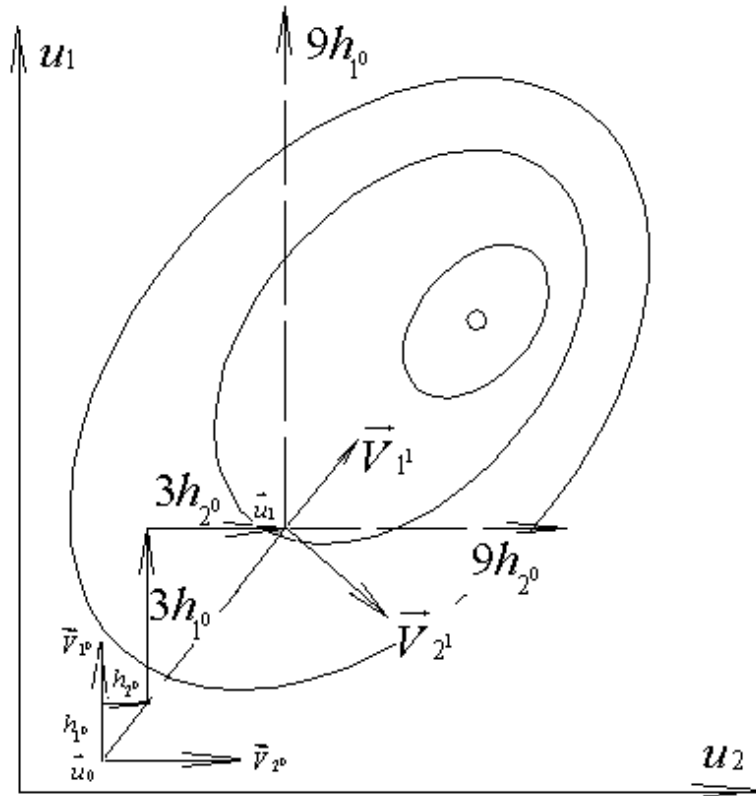


Рис. 5.5. Траєкторія руху методом Розенброка

Алгоритм методу для пошуку максимуму має вигляд (рис. 5.5):

1. Вибір вихідної точки  $\bar{u}_k$ ,  $k = 0$ .
2. Вибір ортогональних напрямків пошуку  $\bar{V}_{jk}$ ,  $j = 1, \dots, m$  і величин кроків  $h_{jk}$  по кожному з напрямків пошуку. Як ортогональні напрямки пошуку  $\bar{V}_{j0}$  вибираються напрямки, що відповідають напрямкам координатних осей  $\bar{V}_{j0} = \bar{e}_j$ ,  $j = \overline{1, m}$ .
3. Введення допоміжної величини  $S_j$ ,  $j = 1, \dots, m$ . Присвоєння  $S_j = 0$ .
4. Здійснення пошукових кроків по всіх ортогональних напрямках:

$$\bar{u} = \bar{u}_k + h_{jk} \bar{V}_{jk}.$$

- 4.1. Якщо  $j \geq m$ , то присвоєння  $j = 1$  і перехід до наступного пункту, якщо  $j < m$ , то  $j = j + 1$  і виконання пункту 4.1.

- 4.2. Присвоєння  $\bar{u}_k = \bar{u}$ ,  $S_j = S_j + h_{jk}$ ,  $j = 1, \dots, m$  і вибір нових величин кроку:

$$h_{jk} = \begin{cases} \alpha \cdot h_{jk}, & \text{якщо } f_0(\bar{u}) > f_0(\bar{u}_k), j = 1, \dots, m, \\ \beta \cdot h_{jk}, & \text{якщо } f_0(\bar{u}) < f_0(\bar{u}_k), j = 1, \dots, m. \end{cases} \quad (5.4)$$

Рекомендується приймати значення  $\alpha = 3$ ;  $\beta = -\frac{1}{2}$ .

Перехід до пункту 4.1.

Ці операції необхідно виконувати доти, поки цільова функція не перестане поліпшуватися. У результаті буде досягнута точка  $\bar{u}_{k+1}$ .

5. Визначення нових напрямків пошуку (нових ортогональних векторів)  $\bar{V}_{j,k+1}$ .

- 5.1. Введення додаткових векторів  $\bar{W}_j$ :

$$\bar{W}_j = \sum_{j=1}^n S_j \bar{V}_{jk}, j = 1, \dots, m. \quad (5.5)$$

- 5.2. Обчислення одиничних векторів:

$$\vec{V}_{1,k+1} = \frac{\vec{W}_1}{\|\vec{W}_1\|}$$

$$\vec{V}_{j,k+1} = \frac{\vec{W}_j - \sum_{i=1}^{j-1} (\vec{W}_j^T \vec{V}_{i,k+1}) \vec{V}_{i,k+1}}{\left\| \vec{W}_j - \sum_{i=1}^{j-1} (\vec{W}_j^T \vec{V}_{i,k+1}) \vec{V}_{i,k+1} \right\|}, \quad j = 2, 3, \dots, m. \quad (5.6)$$

5.3. Присвоєння:

$$\vec{V}_{j,k} = \vec{V}_{j,k+1}, \quad j = 1, \dots, m$$

$$S_j = 0, \quad j = 1, \dots, m \quad (5.7)$$

і перехід до пункту 4.

Процес пошуку переривається, коли величина  $\vec{W}_1$  після численних послідовних наближень залишається малою, а напрямок пошуку залишається незмінним. Очевидно, що  $\vec{W}_1$  є єдиним вектором від  $\vec{u}_k$  до  $\vec{u}_{k+1}$ .

На рис. 5.6 показана траєкторія пошуку при мінімізації функції Розенброка методом Розенброка (числа позначають кількість обчислень цільової функції).

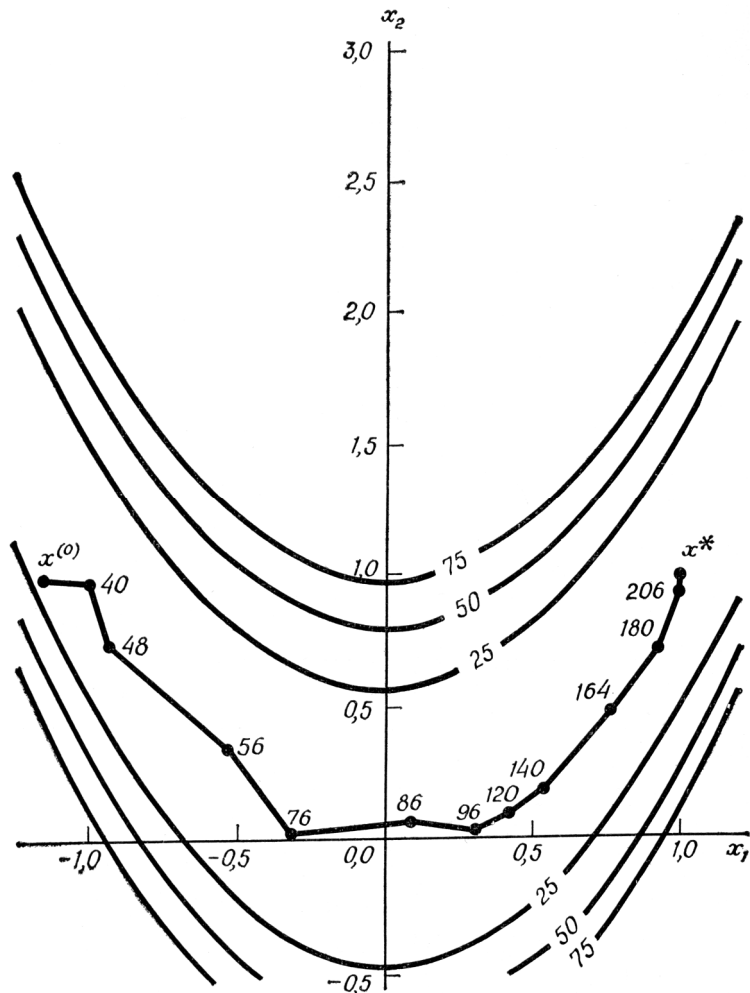


Рис. 5.6. Траєкторія пошуку при мінімізації функції Розенброка методом Розенброка

#### Приклад 5.1. Ілюстрація роботи алгоритму Розенброка

Мінімізувати  $f(x) = 4(x_1 - 5)^2 + (x_2 - 6)^2$ . Початкова точка  $x^{(0)} = \begin{bmatrix} 8 \\ 9 \end{bmatrix}$ .

Розв'язок. У точці  $x^{(0)}$  значення функції  $f(x^{(0)}) = 45,000$ ;  $h_1 = h_2 = 0,10$ . Напрями початкового пошуку співпадають з координатними вісями  $x_1$  та  $x_2$ :  $\vec{V}_1^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ,  $\vec{V}_2^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ .

Спочатку обрахуємо  $f(x)$  у точці

$$x = \begin{bmatrix} 8,00 + 0,10 \cdot 1 \\ 9,00 + 0,10 \cdot 0 \end{bmatrix} = \begin{bmatrix} 8,10 \\ 9,00 \end{bmatrix}.$$

Тут  $f(x) = 47,44$ , тобто має місце невдача. Потім обрахуємо  $f(x)$  у точці

$$x = \begin{bmatrix} 8,00 + 0,10 \cdot 0 \\ 9,00 + 0,10 \cdot 1 \end{bmatrix} = \begin{bmatrix} 8,00 \\ 9,10 \end{bmatrix}.$$

Тут  $f(x) = 45,61$  и знову має місце невдача. Таким чином, на наступному циклі і  $h_1$ , і  $h_2$  мають бути помножені на  $\beta = -\frac{1}{2}$ , або  $h_3 = h_4 = -0,50 \cdot 0,10 = -0,05$ . Крок здійснюється з останнього вдалого значення  $x$ , тобто з точки  $x = \begin{bmatrix} 8 \\ 9 \end{bmatrix}$ .

Спочатку обрахуємо значення  $f(x)$  у точці

$$x = \begin{bmatrix} 8,00 - 0,05 \cdot 1 \\ 9,00 - 0,05 \cdot 0 \end{bmatrix} = \begin{bmatrix} 7,95 \\ 9,00 \end{bmatrix}.$$

Тут  $f(x) = 43,81$ , що свідчить про успіх. Потім обрахуємо  $f(x)$  у точці

$$x = \begin{bmatrix} 7,95 - 0,05 \cdot 0 \\ 9,00 - 0,05 \cdot 1 \end{bmatrix} = \begin{bmatrix} 7,95 \\ 8,95 \end{bmatrix},$$

де  $f(x) = 43,125$ , що знову свідчить про успіх. На наступному циклі  $h$  помножується на 3, тобто  $h_5 = h_6 = 3 \cdot (-0,05) = -0,15$ .

Нижче наведені декілька послідовних циклів першого етапу.

Номер пошуку	$h$	$x_1$	$x_2$	$f(x)$	Успіх ( $S$ ) або невдача ( $F$ )
5	-0,15	7,80	8,95	40,06	$S$
6	-0,15	7,80	8,80	39,20	$S$
7	-0,45	7,35	8,80	29,93	$S$
8	-0,45	7,35	8,35	27,61	$S$
9	-1,35	6,00	8,35	9,522	$S$
10	-1,35	6,00	7,00	5,000	$S$
12	-4,05	1,95	7,00	32,21	$F$
13	-4,05	6,00	2,95	13,30	$F$

Тепер вже у кожному координатному напрямі за успіхом відбулася невдача, і, таким чином, закінчився нульовий етап пошуку. Потім обраховуються нові напрями пошуку так, щоб  $\vec{V}_1^{(1)}$  був напрямлений вздовж вектора, що йде з точки  $x^{(0)} = \begin{bmatrix} 8 \\ 9 \end{bmatrix}$  у  $x^{(1)} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}$ , при чому остання точка відповідає найкращому значенню  $f(x)$ , одержаному на нульовому етапі;  $\vec{V}_2^{(1)}$  ортонормований до  $\vec{V}_1^{(1)}$ . Вектори  $\vec{W}_1^{(0)}$  та  $\vec{W}_2^{(0)}$  вираховуються за формулою (5.5), а  $\vec{V}_1^{(1)}$  та  $\vec{V}_2^{(1)}$  - за формулою (5.6).

$$\vec{W}_1^{(0)} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, \vec{W}_2^{(0)} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}, \vec{V}_1^{(1)} = \frac{\begin{bmatrix} -2 \\ -2 \end{bmatrix}}{\sqrt{(-2)^2 + (-2)^2}} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T,$$

$$\vec{V}_2^{(1)} = \frac{\begin{bmatrix} 0 & -2 \end{bmatrix}^T - \begin{bmatrix} 0 & -2 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T}{\left\| \begin{bmatrix} 0 & -2 \end{bmatrix}^T - \begin{bmatrix} 0 & -2 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^T \right\|} = \frac{\begin{bmatrix} 1 & -1 \end{bmatrix}^T}{\sqrt{1^2 + (-1)^2}} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

У першому циклі на першому етапі пошуку ( $k=1$ ) прийmemo  $h_1 = h_2 = -4,05 \cdot (-0,5) = 2,025$ . Спочатку обрахуємо  $f(x)$  у напрямку пошуку  $\vec{V}_1^{(1)}$  у точці

$$x = \begin{bmatrix} 6,000 + (2,025) \cdot (-0,706) \\ 6,000 + (2,025) \cdot (-0,706) \end{bmatrix} = \begin{bmatrix} 4,568 \\ 5,568 \end{bmatrix}.$$

Тут  $f(x) = 0,9327$ , що свідчить про успіх. Потім обрахуємо  $f(x)$  у напрямку пошуку  $\vec{V}_2^{(1)}$  у точці

$$x = \begin{bmatrix} 4,568 + (2,025) \cdot 0,706 \\ 5,568 + (2,025) \cdot (-0,706) \end{bmatrix} = \begin{bmatrix} 6,000 \\ 4,136 \end{bmatrix}.$$

Тут  $f(x) = 7,474$ , що свідчить про невдачу. Решта даних для першого етапу наведена нижче:

Номер пошуку	$h$	$x_1$	$x_2$	$f(x)$	Успіх ( $S$ ) або невдача ( $F$ )
3	6,075	0,272	1,272	11,175	$F$
4	-1,125	3,852	6,284	5,351	$F$
5	-3,038	6,716	7,716	14,722	$F$
6	0,506	4,926	5,210	0,646	$S$
7	1,518	3,852	4,136	8,743	$F$
8	1,518	6,000	4,136	7,473	$F$

Оскільки рух у напрямку 1 призводить до успіху в перших двох випадках, а у напрямку 2 – до невдачі, то у пошуках 3 та 4 беремо  $h_3 = 3 \cdot 2,025 = 6,075$  і  $h_4 = -0,5 \cdot 2,025 = -1,0125$  відповідно. Після невдачі у пошуках 3 та 4  $h_3 = 6,075$  та  $h_4 = -1,0125$  помножуються на  $-0,5$ , що дає відповідно  $h_5$  та  $h_6$ . Після пошуку 6 мав місце успіх по обом напрямам, а після пошуку 8 відбулися дві послідовні невдачі, тому у відповідності до даного алгоритму на цьому кроці етап 1 завершується. На цьому етапі завершимо розгляд даного прикладу. Проте наведена вище процедура була продовжена, поки критерій збіжності не був задоволений (що потребувало 111 обчислень цільової функції, а відносне відхилення її значення від правильного значення, рівного 0, становило  $5,5 \cdot 10^{-11}$ , відносне відхилення  $x_1$  становило  $3,0 \cdot 10^{-6}$ , а відхилення  $x_2$  становило  $4,4 \cdot 10^{-6}$ ).

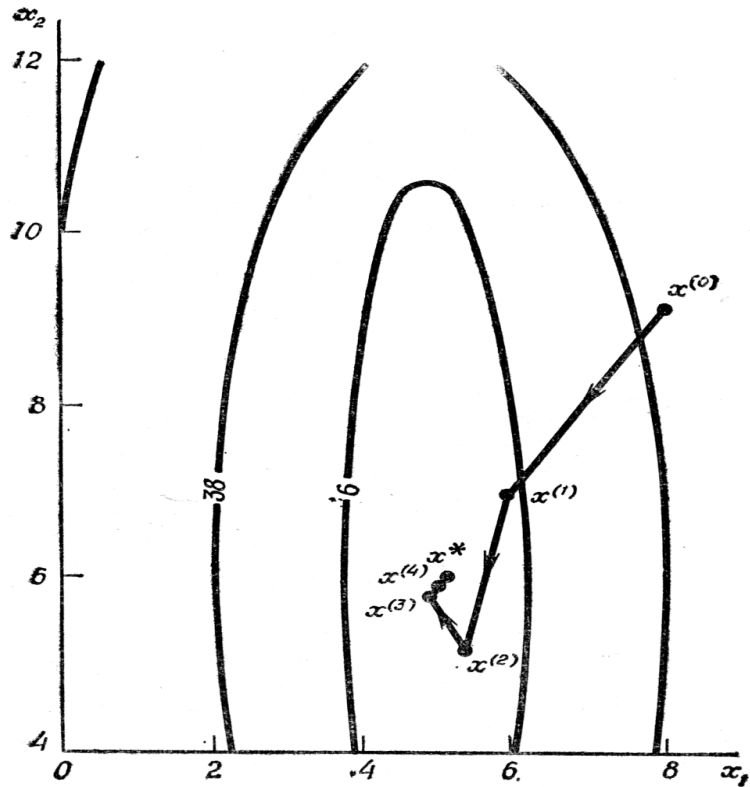


Рис. 5.7. Траекторія основної частини мінімізації на перших чотирьох етапах

На рис. 5.7 показана траекторія основної частини мінімізації на перших чотирьох етапах (34 обчислення функції); після проведення цих етапів були отримані  $x^{(4)} = [5,036 \ 5,938]^T$  та  $f(x^{(4)}) = 9,46 \cdot 10^{-3}$ . Інші 77 обчислень функції знадобились для збільшення точності  $x$  та  $f(x)$ .

### 5.3. Метод пошуку по симплексу.

Перші спроби розв'язку оптимізаційних задач без обмежень на основі прямого пошуку пов'язані з використанням одномірних методів оптимізації. Як правило, при реалізації таких методів допустима область визначення показника якості функціонування системи (цільової функції) замінюється дискретною множиною (гратами) точок простору керованих змінних, а потім використовуються різні стратегії зменшення області, що містить розв'язок задачі. Часто ця процедура виявляється еквівалентною рівномірному пошуку у вузлах ґрат і, отже, непридатною для розв'язку задач із числом змінних, що перевищує 2. Більш корисна ідея полягає у виборі базової точки й оцінюванні значень цільової функції в точках, що оточують базову. Наприклад, при розв'язку задачі із двома змінними можна скористатися квадратним зразком, зображеним на рис. 4.12. Потім «найкраща» з п'яти досліджуваних точок вибирається в якості наступної базової точки, навколо якої будується аналогічний зразок. Якщо жодна з кутових точок не має переваги перед базовою, то розміри зразка варто зменшити, після чого продовжити пошук.

Одна із стратегій пошуку, що викликає особливий інтерес покладена в основу *методу пошуку по симплексу*, запропонованого Спендлі, Хекстом і Хімсвортом. Процедура симплексного пошуку Спендлі, Хекста і Хімсворта базується на тому, що експериментальним зразком, що містить найменшу кількість точок, є регулярний симплекс. Регулярний симплекс в  $N$ -мірному просторі є багатогранником, утвореним  $N+1$  рівновіддаленими одна від одної точками-вершинами. Наприклад, у випадку двох змінних симплексом є рівносторонній трикутник; у тривимірному просторі симплекс – тетраедр. В алгоритмі симплексного пошуку використовується важлива властивість симплексів, відповідно до якої новий симплекс можна побудувати на будь-якій грані початкового симплекса шляхом переносу обраної вершини на належну відстань уздовж прямої,



проведеної через центр ваги інших вершин початкового симплекса. Отримана в такий спосіб точка є вершиною нового симплекса, а обрана при побудові вершина початкового симплексу відкидається. Неважко побачити, що при переході до нового симплексу потрібно одне обчислення значення цільової функції. Рис. 4.13 ілюструє процес побудови нового симплексу на площині.

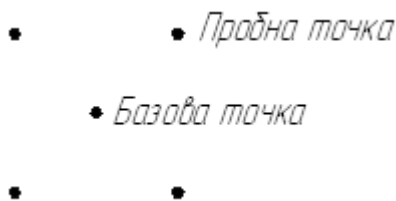


Рис. 5.8. Квадратний зразок (частковий випадок кубічного зразку)

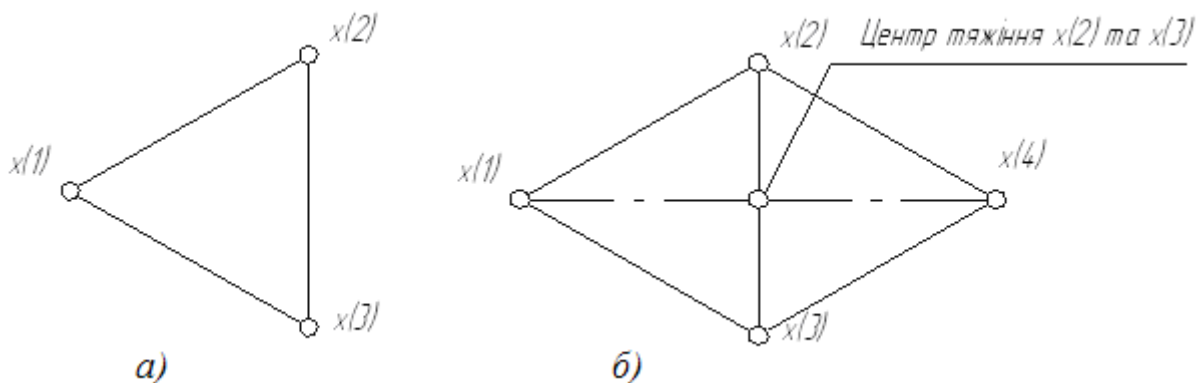


Рис. 5.9. Побудова нового симплексу: а) - початковий симплекс; б) - новий симплекс

Робота алгоритму симплексного пошуку починається з побудови регулярного симплекса у просторі незалежних змінних й оцінювання значень цільової функції у кожній з вершин симплексу. При цьому визначається вершина, якій відповідає найбільше значення цільової функції. Потім знайдена вершина проектується через центр ваги інших вершин симплексу в нову точку, що використовується як вершина нового симплексу. Якщо функція спадає досить плавно, ітерації тривають доти, поки або не буде накрыта точка мінімуму, або не почнеться циклічний рух по двох або більше симплексах. У таких ситуаціях можна скористатися наступними трьома правилами.

*Правило 1. «Накриття» точки мінімуму*

Якщо вершина, якій відповідає найбільше значення цільової функції, побудована на попередній ітерації, то замість неї береться вершина, якій відповідає наступне по величині значення цільової функції.

*Правило 2. Циклічний рух*

Якщо деяка вершина симплексу не виключається протягом більш ніж  $M$  ітерацій, то необхідно зменшити розміри симплексу за допомогою коефіцієнта редукції і побудувати новий симплекс, обравши в якості базової точку, якій відповідає мінімальне значення цільової функції. Спендлі, Хекст і Хімсворт запропонували обчислювати  $M$  по формулі:

$$M=1.65N+0.05N^2,$$

де  $N$  - розмірність задачі, а  $M$  округляється до найближчого цілого числа. Для застосування даного правила потрібно встановити величину коефіцієнту редукції.

*Правило 3. Критерій закінчення пошуку*

Пошук завершується, коли або розміри симплекса, або різниці між значеннями функції у вершинах стають досить малими. Щоб можна було застосовувати ці правила, необхідно задати величину параметру закінчення пошуку.

Реалізація досліджуваного алгоритму заснована на обчисленнях двох типів: (1) побудові регулярного симплексу при заданих базовій точці й масштабному множнику і (2) розрахунку координат відбитої точки. Побудова симплексу є досить простою

процедурою, тому що з елементарної геометрії відомо, що при заданих початковій (базовій) точці  $x(0)$  і масштабному множнику  $\alpha$  координати інших  $N$  вершин симплекса в  $N$ -мірному просторі обчислюються за формулою:

$$x(i) = \begin{cases} x_j(0) + \delta_1, & \text{якщо } j \neq i, \\ x_j(0) + \delta_2, & \text{якщо } j = i, \end{cases} \quad (5.8)$$

для  $i$  та  $j=1,2,3,\dots,N$ .

Прирости  $\delta_1$  і  $\delta_2$ , що залежать лише від  $N$  і обраного масштабного множника  $\alpha$ , визначаються по формулах:

$$\delta_1 = \left[ \frac{(N+1)^{1/2} + N - 1}{N\sqrt{2}} \right] \alpha, \quad (5.9)$$

$$\delta_2 = \left[ \frac{(N+1)^{1/2} + N - 1}{N\sqrt{2}} \right] \alpha, \quad (5.10)$$

Помітимо, що величина масштабного множника  $\alpha$  вибирається дослідником, виходячи з характеристик розв'язуваної задачі. При  $\alpha=1$  ребра регулярного симплекса мають одиничну довжину.

Обчислення другого типу, пов'язані з відбиттям щодо центра ваги, також представляють собою нескладну процедуру. Нехай  $x(j)$  - точка, що підлягає відбиттю. Центр ваги інших  $N$  точок розташованих у точці:

$$x_c = \frac{1}{N} \sum_{\substack{i=0 \\ i \neq j}}^N x(i). \quad (5.11)$$

Всі точки прямої, що проходить через  $x(j)$  і  $x_c$ , задаються формулою:

$$x = x(j) + \lambda(x_c - x(j)) \quad (5.12)$$

При  $\lambda=0$  отримуємо вихідну точку  $x(j)$ , тоді як значення  $\lambda=1$  відповідає центру ваги  $x_c$ . Для того щоб побудований симплекс мав властивість регулярності, відбиття повинне бути симетричним. Отже, нова вершина знаходиться при  $\lambda=2$ . Таким чином:

$$x(j)_{\text{нов}} = 2x_c - x(j)_{\text{попереднє}} \quad (5.13)$$

Проілюструємо обчислювальну схему методу наступним прикладом.

### **Приклад 5.2. Обчислення відповідно до методу пошуку по симплексу**

Мінімізувати  $f(x) = (1-x_1)^2 + (2-x_2)^2$ .

Розв'язок. Для побудови вихідного симплексу потрібно задати початкову точку і масштабний множник. Нехай  $x^{(0)} = [0,0]^T$  і  $\alpha=2$ . Тоді:

$$\delta_1 = \left[ \frac{\sqrt{3}+1}{2\sqrt{2}} \right] \alpha = 1.9318$$

$$\delta_2 = \left[ \frac{\sqrt{3}-1}{2\sqrt{2}} \right] \alpha = 0.5176$$

Використовуючи ці два параметри, обчислимо координати двох інших вершин симплексу:

$$x^{(1)} = [0+0,5176; 0+1,9318]^T = [0,5176; 1,9318]^T,$$

$$x^{(2)} = [0+1,9318; 0+0,5176]^T = [1,9318; 0,5176]^T,$$

яким відповідають значення цільової функції  $f(x^{(1)}) = 0,2374$  і  $f(x^{(2)}) = 3,0658$ . Тому що  $f(x^{(0)})=5$ , необхідно відбити точку  $x^{(0)}$  щодо центра ваги двох інших вершин симплексу:

$$x_c = \frac{1}{2} \sum_{i=1}^2 x^{(i)} = \frac{1}{2} (x^{(1)} + x^{(2)}).$$

Використовуючи формулу (4.22), отримуємо:

$$x^{(3)} = x^{(1)} + x^{(2)} - x^{(0)},$$

$$x^T = [2,4494; 2,4494]^T.$$

В отриманій точці  $f(x^{(3)}) = 2,3027$ , тобто спостерігається зменшення цільової функції. Новий симплекс утворений точками  $x^{(1)}$ ,  $x^{(2)}$  і  $x^{(3)}$ . Відповідно до алгоритму варто відбити точку  $x^{(2)}$ , якій відповідає найбільше значення цільової функції, щодо центра ваги точок  $x^{(1)}$  і  $x^{(3)}$ . Ітерації тривають доти, поки не знадобиться застосування правил 1, 2 і 3, які були сформульовані вище.

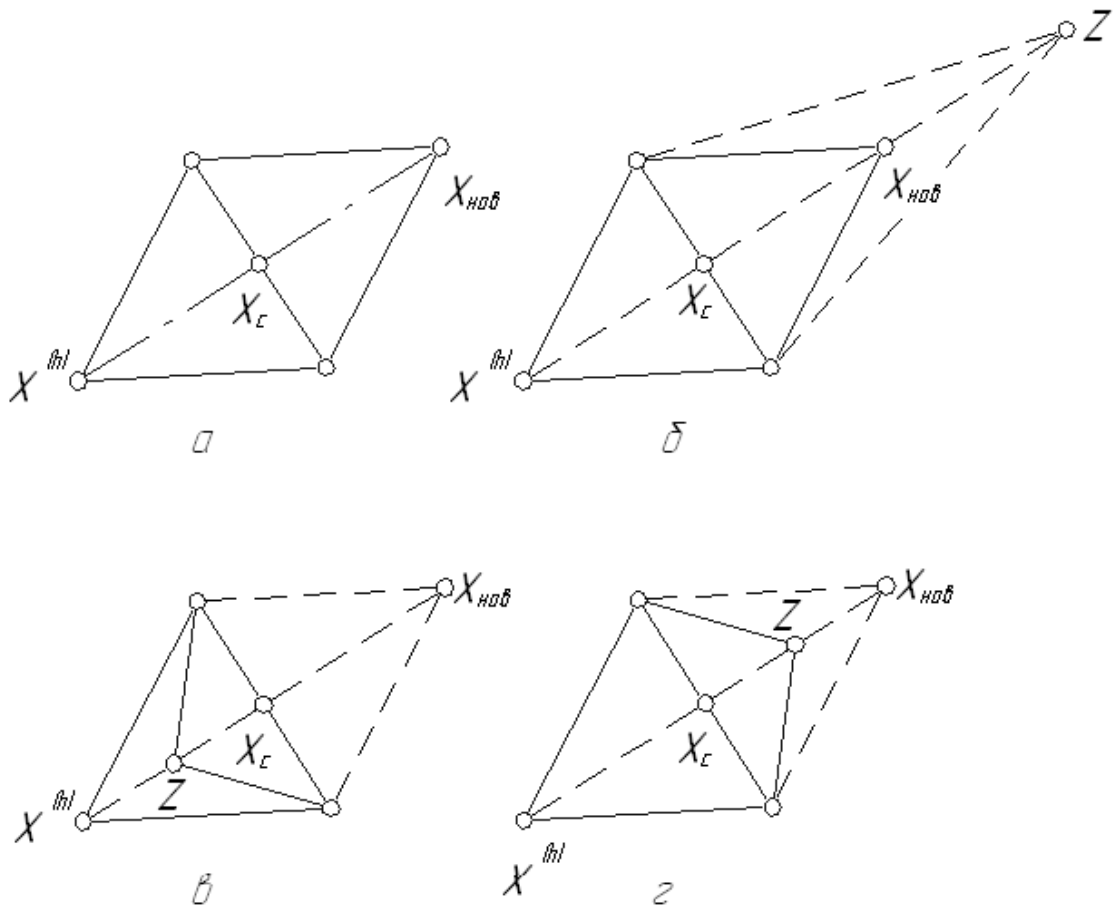
Розглянутий вище алгоритм має кілька очевидних переваг:

1. Розрахунки й логічна структура методу відрізняються порівняльною простотою, і, отже, відповідна програма для ЕОМ виявляється короткою.
2. Рівень вимог до об'єму пам'яті ЕОМ невисокий, масив має розмірність  $(N+1, N+2)$ .
3. Використовується порівняно невелике число задалегідь установлених параметрів: масштабний множник  $\alpha$ , коефіцієнт зменшення множника  $\alpha$  (якщо застосовується правило 2) і параметри закінчення пошуку.
4. Алгоритм виявляється ефективним навіть у тих випадках, коли помилка обчислення значень цільової функції велика, оскільки при його реалізації оперують найбільшими значеннями функції у вершинах, а не найменшими.

Перераховані фактори характеризують метод пошуку по симплексу як досить корисний при проведенні обчислень у реальному часі.

Алгоритм має також ряд істотних недоліків.

1. Не виключене виникнення труднощів, пов'язаних із масштабуванням, оскільки всі координати вершин симплексу залежать від того самого масштабного множника  $\alpha$ . Щоб обійти такі труднощі, у практичних задачах треба промасштабувати всі змінні для того, щоб їх значення були приблизно однаковими по величині.
2. Алгоритм працює занадто повільно, тому що отримана на попередніх ітераціях інформація не використовується для прискорення пошуку.
3. Не існує простого способу розширення симплексу, що не вимагає перерахунку значень цільової функції у всіх точках зразка. Таким чином, якщо з якої-небудь причини цільова функція зменшується (наприклад, якщо зустрічається область із вузьким «яром» або «хребтом»), то пошук повинен тривати із зменшеною величиною кроку.



**Рис. 5.10.** Розтяг і стиск симплексу  
*a* – нормальне відбиття, *б* – розтяг, *в* – стиск, *г* – стиск.

Модифікована процедура пошуку по симплексу, розроблена Нелдером і Мідом, частково усуває деякі з перерахованих недоліків. Незабавно помітити, що хоча формула для визначення вершин регулярного симплекса виявляється досить зручною при побудові вихідного зразка, однак вагомим підстав для збереження властивості регулярності симплексу в процесі пошуку немає. Отже, при відбитті симплекса існує можливість як його розтягування, так і стискання. При розрахунках по методу Нелдера і Міда використовуються вершини симплексу  $x^{(h)}$  (якій відповідає найбільше значення цільової функції  $f^{(h)}$ ),  $g^{(h)}$  (якій відповідає наступне по величині значення цільової функції  $f^{(g)}$ ) і  $x^{(i)}$  (якій відповідає найменше значення цільової функції  $f^{(i)}$ ). Нагадаємо, що відбиття вершини симплексу здійснюється по прямій:

$$x = x^{(h)} + \lambda(x_c - x^{(h)}), \quad (5.14)$$

$$x = x^{(h)} + (1 + \theta)(x_c - x^{(h)}). \quad (5.15)$$

При  $\theta = 1$  має місце так назване нормальне відбиття симплексу, оскільки точка  $x_{нов}$  розташовується на відстані  $\|x_c - x^{(i)}\|$  від точки  $x_c$ . Якщо  $-1 \leq \theta < 1$ , то спостерігається стисле відбиття, або *стиск* симплексу, тоді як вибір  $\theta > 1$  забезпечує розтягнуте відбиття, або *розтягування* симплексу. На рис. 5.8 наведені можливі варіанти відбиття. Три значення параметра  $\theta$ , використовувані при нормальному відбитті, стиску і розтягуванні, позначаються через  $\alpha$ ,  $\beta$ , і  $\gamma$  відповідно. Реалізація методу починається з побудови вихідного симплексу і визначення точок  $x^{(h)}$ ,  $x^{(g)}$ ,  $x^{(i)}$  і  $x_c$ . Після нормального відбиття здійснюється перевірка значень цільової функції за критерієм закінчення пошуку в точках відбитого симплексу. Якщо пошук не закінчений, то за допомогою тестів, наведених на рис. 5.8, вибирається одна з операцій – нормальне відбиття, розтягування або стиск. Ітерації тривають, поки зміни значень цільової функції у вершинах симплексу не стануть

незначними. Як задовільні значення параметрів  $\alpha$ ,  $\beta$ , і  $\gamma$  Нелдер і Мід рекомендують використовувати  $\alpha = 1$ ,  $\beta = 0,5$ , і  $\gamma = 2$ .

Результати окремих чисельних експериментів показують, що метод Нелдера - Міда має достатню ефективність і високу надійність в умовах наявності випадкових збурень або помилок при визначенні значень цільової функції. В 1969 р. Бокс і Дрейпер стверджували, що цей метод є «найбільш ефективним із всіх відомих методів послідовної оптимізації». В 1972 р. Паркінсон і Хатчінсон досліджували вплив вибору параметрів  $\alpha$ ,  $\beta$ ,  $\gamma$  і способу побудови вихідного симплексу на ефективність пошуку. Вони встановили, що орієнтація вихідного симплексу на відміну від його форми є істотним чинником, що впливає на процедуру пошуку, і запропонували використати значення параметрів  $(\alpha; \beta; \gamma) = (2; 0,25; 2,5)$ . Такий вибір параметрів дозволив забезпечити гарну роботу алгоритму при повторенні послідовних розтягувань симплексу.

#### 5.4. Метод Хука-Дживса.

Пошук згідно цього методу складається із послідовності *кроків пошуку для дослідження функції навколо базисної точки*, з якої у випадку успіху робиться *пошук за зразком*.

Опис цієї процедури наведено нижче:

- A. Вибрати початкову базисну точку  $b_1$  і крок довжиною  $h_j$  для кожної змінної  $x_j$ ,  $j=1,2,\dots,n$ . Довжина кроку може бути обрана однаковою для всіх змінних.
- B. Досліджувати  $f(X)$  навколо базисної точки  $b_1$  з метою отримання даних про локальне поведіння функції  $f(X)$ . Ці дані будуть використовуватися для знаходження потрібного напрямку пошуку за зразком, за допомогою якого можна сподіватися досягти більшого спадання функції. *Пошук, що досліджує, навколо базисної точки  $b_1$  виконується таким чином:*
  - 1) Обчислити значення функції  $f(b_1)$  у базисній точці  $b_1$ .
  - 2) Кожна змінна по черзі змінюється додаванням довжини кроку. Таким чином, ми обчислюємо значення функції  $f(b_1+h_1e_1)$ , де  $e_1$  – одиничний вектор у напрямку осі  $x_1$ . Якщо це призводить до зменшення значення функції, то  $b_1$  замінюється на  $b_1+h_1e_1$ . У протилежному випадку обчислюємо значення функції  $f(b_1-h_1e_1)$ , і якщо її значення зменшується, то  $b_1$  замінюємо на  $(b_1-h_1e_1)$ . Якщо жоден із виконаних кроків не призводить до зменшення значення функції, то точка  $b_1$  залишається незмінною і розглядаються зміни в напрямку осі  $x_2$ , тобто знаходиться значення функції  $f(b_1+h_2e_2)$  і т.д. Коли будуть розглянуті всі  $n$  змінних, ми будемо мати нову базисну точку  $b_2$ .
  - 3) Якщо  $b_2=b_1$ , тобто зменшення функції не було досягнуто, то дослідження повторюється навколо тієї ж базисної точки  $b_1$ , але із зменшеною довжиною кроку. На практиці задовільним є зменшення кроку (кроків) у десять разів від початкової довжини.
  - 4) Якщо  $b_2 \neq b_1$ , то виконується *пошук за зразком*.
- C. При пошуку за зразком використовується інформація, отримана в процесі дослідження, і мінімізація функції здійснюється пошуком у напрямку, заданому зразком. Ця процедура виконується в такий спосіб:
  - 1) Розумно рухатися із базисної точки  $b_2$  у напрямку  $(b_2-b_1)$ , оскільки пошук у цьому напрямку вже привів до зменшення значення функції. Тому обчислимо функцію в точці зразку:  $P_1 = b_1 + 2(b_2 - b_1)$ .  
У загальному випадку:  $P_i = b_i + 2(b_{i+1} - b_i)$ .
  - 2) Потім дослідження варто продовжувати навколо точки  $P_1$  ( $P_i$ ).
  - 3) Якщо найменше значення на кроці C.2) менше значення в базисній точці  $b_2$  (у загальному випадку  $b_{i+1}$ ), то отримують нову базисну точку  $b_3$  (у загальному випадку  $b_{i+2}$ ), після чого варто повторити крок C.1). У протилежному випадку не робити пошук за зразком із точки  $b_2$  ( $b_{i+1}$ ), а продовжити дослідження в точці  $b_2$  ( $b_{i+1}$ ).

- 4) Завершити цей процес, коли довжина кроку (довжини кроків) буде зменшена до заданого малого значення.

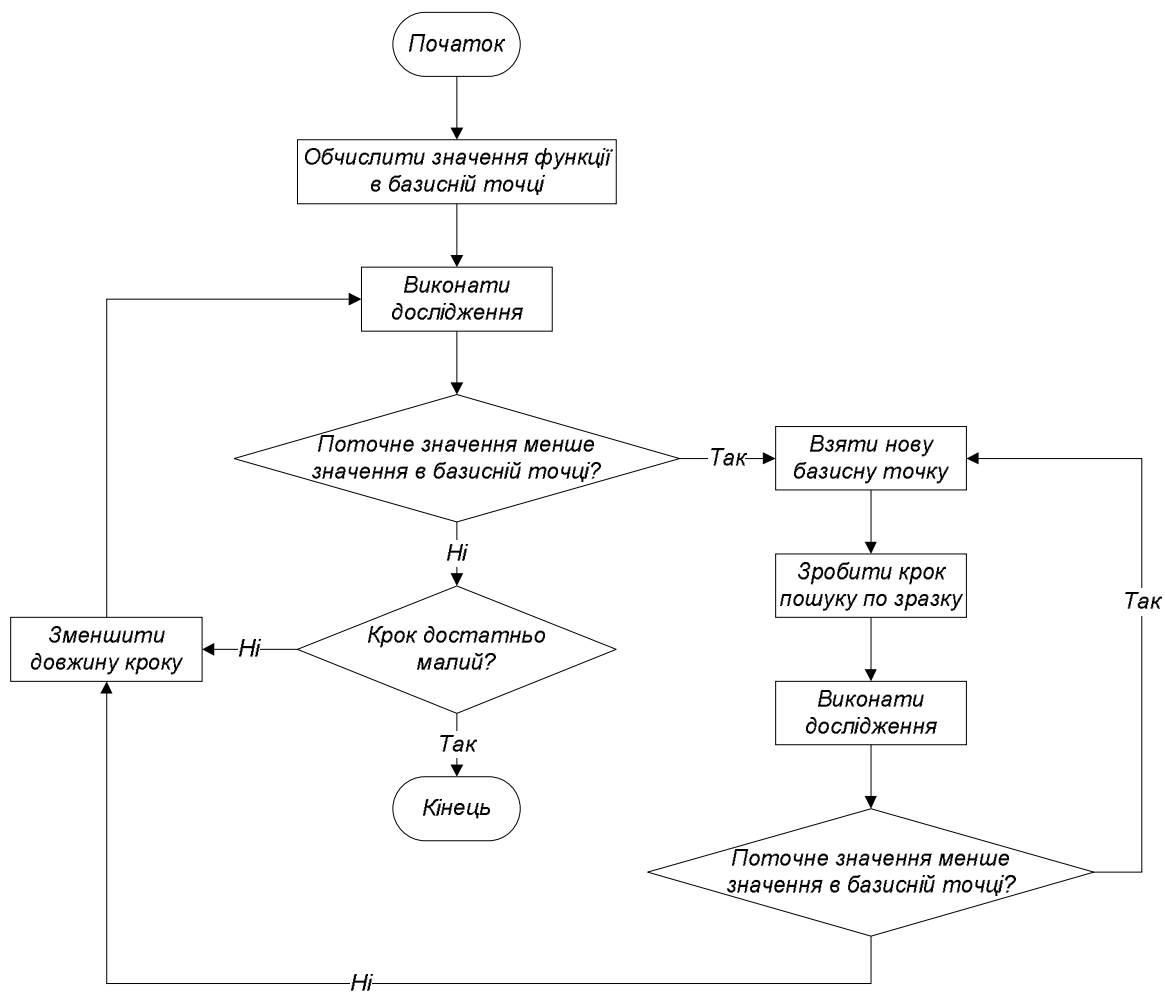


Рис. 5.11. Інформаційна схема метода Хука-Джівса

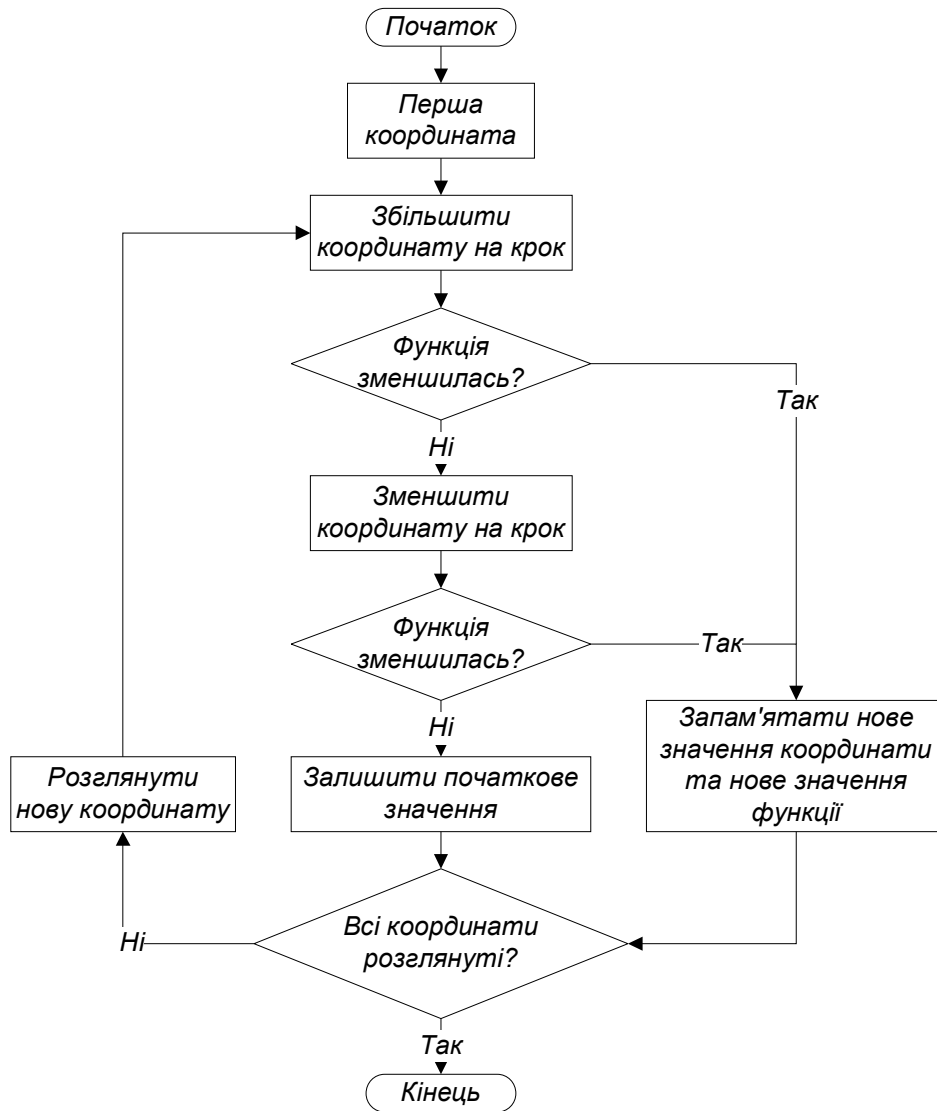


Рис. 5.12. Інформаційна схема досліджув чого пошуку

**Приклад 5.3. Розв'язок задачі оптимального параметричного синтезу регулятора методом Хука-Дживса.**

Однією із важливих задач оптимізації в теорії управління є задача оптимального параметричного синтезу регулятора. Цю задачу можна розв'язати методами моделювання і оптимізації для регуляторів різноманітної структури, а також для різноманітних об'єктів керування – лінійних, із запізненням, нелінійних та ін.

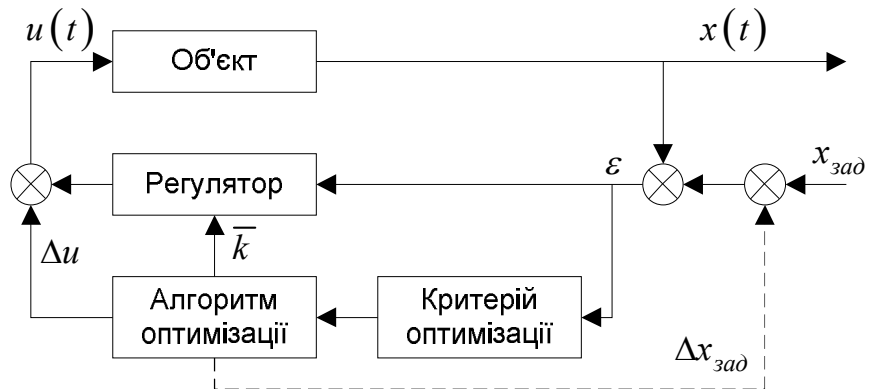


Рис. 5.13. Структурна схема методу моделювання та оптимізації одноконтурної САР

Структурна схема методу моделювання та оптимізації одноконтурної САР (системи автоматичного регулювання) зображена на рис. 5.13.

Моделюючи процес, знаходимо сигнал розбалансу  $e(t)$  і використовуємо його для знаходження інтегрального критерію оптимізації. Математична постановка задачі оптимального параметричного синтезу АСР по інтегральному критерію в загальному вигляді може бути записана так:

$$I = \int_{t_H}^{t_K} f_0(\bar{x}, \bar{u}, \bar{g}, \bar{e}), \rightarrow \min_{k \in V_k}$$

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, \bar{b}) + W(t)$$

$$\bar{y} = \bar{\varphi}(x, \bar{c})$$

$$\bar{u} = \bar{\psi}(\bar{x}, \bar{k})$$

$$\bar{u} \in V_u, \bar{x} \in V_x, k \in V_k$$

$\bar{b}, \bar{c}$  – вектори параметрів об'єкту і вимірювального приладу;

$\bar{k}$  – вектор параметрів регулятора

$V_k, V_u, V_x$  – відкриті і закриті частини;

$W(t)$  – детерміноване збурення.

Для рішення цієї задачі використовують методи нульового порядку. Розглянемо задачу ОПС пропорційно-інтегрального регулятора для об'єкта першого порядку із запізненням.

#### Змістовна постановка задачі

Необхідно визначити параметри динамічної настройки пропорційно-інтегрального регулятора для відпрацювання заданого сигналу при наявності обмежень на керування.

#### Математична постановка задачі

Передаточна функція об'єкту: 
$$W_{об} = \frac{Y(p)}{U(p)} = \frac{2 \cdot e^{-30 \cdot p}}{100 \cdot p + 1}$$

Інтегральний критерій має вигляд: 
$$I = \int_{t_H}^{t_K} e^2 \cdot t^\alpha \cdot dt \rightarrow \min_{K_p, T_u}$$

Передаточна функція регулятора: 
$$W_p = K_p \left( 1 + \frac{1}{T_u \cdot p} \right), \bar{k} = \begin{bmatrix} K_p \\ T_u \end{bmatrix}$$

Обмеження на керування: 
$$u_{\min} \leq u \leq u_{\max}$$

Процедуру оптимізації виконати з використанням алгоритму Хука-Дживса.

Для знаходження значень  $K_p$  і  $T_u$  методом Хука-Дживса подається збурення  $\Delta x_{зад}$ . Сигнал розбалансу поступає на блок, що обчислює критерій оптимізації та передає одержану величину на блок, що реалізує алгоритм оптимізації. Останній коректує параметри настройки регулятора,  $\bar{k}$ . Потім подається нове збурення і дії повторюються. Оптимізація продовжується доти, доки зміна значення критерію оптимальності не стає меншою за задану точність, тобто  $|I^{(k)} - I^{(k-1)}| < \varepsilon$ , що свідчить про досягнення екстремуму.

Процес пошуку оптимальних настройок регулятора методом Хука-Дживса можна реалізувати програмно. На рис. 5.14 наведено розрахунок настройок регулятора з використанням програми „Параметрический синтез одноконтурной АСР.v.1.1”, розробленої Бунке А.С. (інтерфейс програми на російській мові). Початкові значення параметрів настройки регулятора одержані інженерною методикою:  $K_p = 1.17$ ,  $T_u = 70 \text{ с}$ . Кінцеві значення після восьми кроків оптимізації становлять:  $K_p = 0,87$ ,  $T_u = 105 \text{ с}$ .



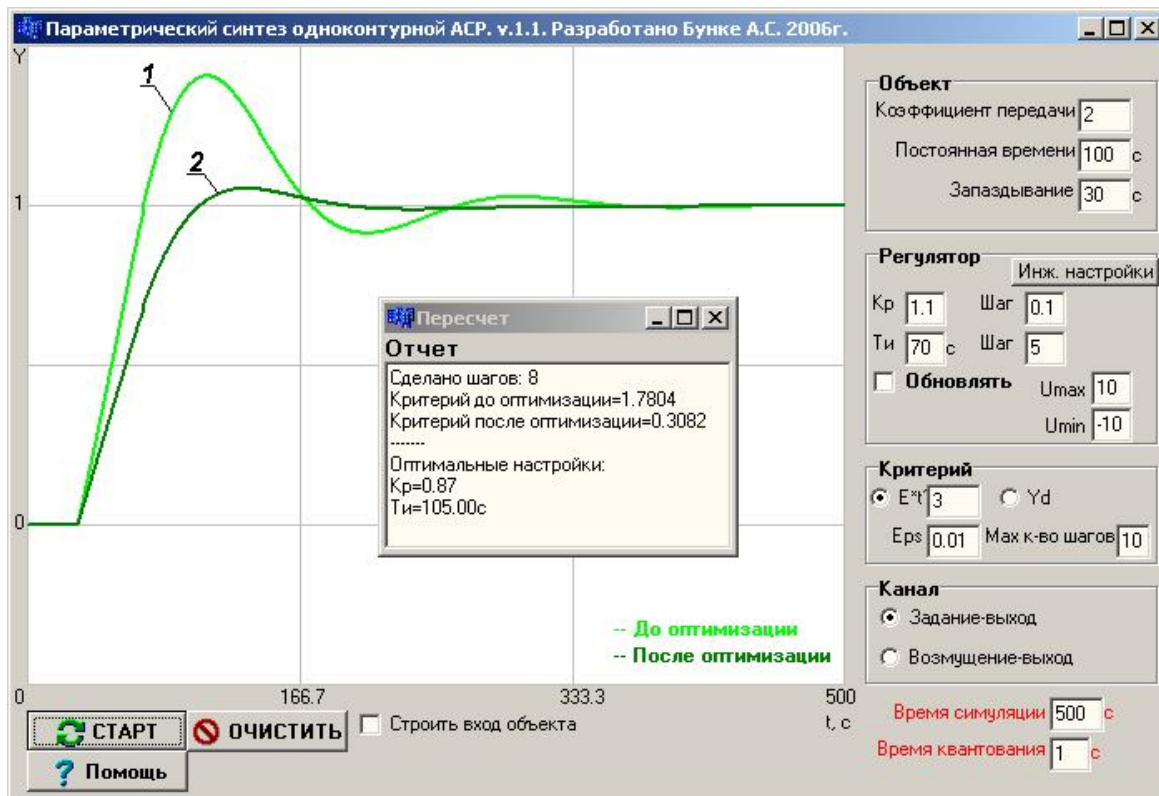


Рис. 5.14. Розрахунок оптимальних параметрів регулятора методом моделювання і оптимізації  
 1 – перехідний процес до оптимізації; 2 – перехідний процес після восьми кроків оптимізації

### Приклад 5.3. Розв’язок задачі оптимального параметричного синтезу регуляторів каскадної САУ.

Розглянемо задачу синтезу стабілізуючого та коректуючого ПІ-регуляторів каскадної САУ методом Хука-Дживса. Система має вигляд, зображений на рис. 5.15.

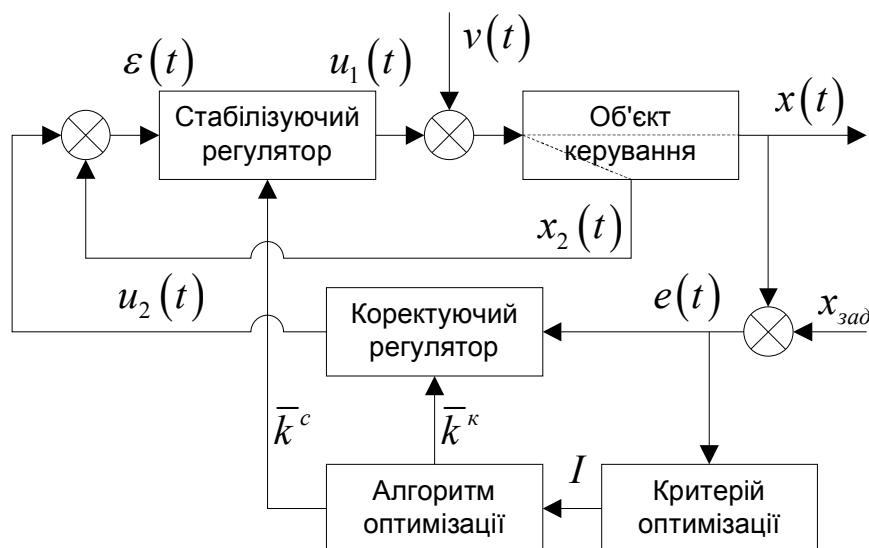


Рис. 5.15. Структурна схема методу моделювання та оптимізації каскадної САУ

#### Математична постановка задачі

Передаточна функція об'єкту:  $W_{об}^{инерц.} = \frac{1 \cdot e^{-15 \cdot p}}{150 \cdot p + 1}$ ,  $W_{об}^{опер.} = \frac{1 \cdot e^{-0.5 \cdot p}}{5 \cdot p + 1}$

Інтегральний критерій має вигляд:  $I = \int_{t_n}^{t_k} e^2 \cdot t^\alpha \cdot dt \rightarrow \min_{K_p, T_i}$

$$\text{Передаточна функція регулятора: } W_p^k = K_p^k \left( 1 + \frac{1}{T_u^k \cdot p} \right), \quad \bar{k}^k = \begin{bmatrix} K_p^k \\ T_u^k \end{bmatrix},$$

$$W_p^c = K_p^c \left( 1 + \frac{1}{T_u^c \cdot p} \right), \quad \bar{k}^c = \begin{bmatrix} K_p^c \\ T_u^c \end{bmatrix}$$

На рис. 5.16 наведено розрахунок настройок регулятора з використанням програми „Параметрический синтез ПИ для Каскадной системы”, розробленої Бунке А.С. (інтерфейс програми на російській мові). Початкові значення параметрів настройки регуляторів одержані інженерною методикою:  $K_p^k = 7$ ,  $T_u^k = 105 \text{ c}$ ;  $K_p^c = 7$ ,  $T_u^c = 3.5 \text{ c}$ . Кінцеві значення після 30 кроків пошуку становлять:  $K_p^k = 5.2$ ,  $T_u^k = 150 \text{ c}$ ;  $K_p^c = 8.5$ ,  $T_u^c = 0.5 \text{ c}$ .

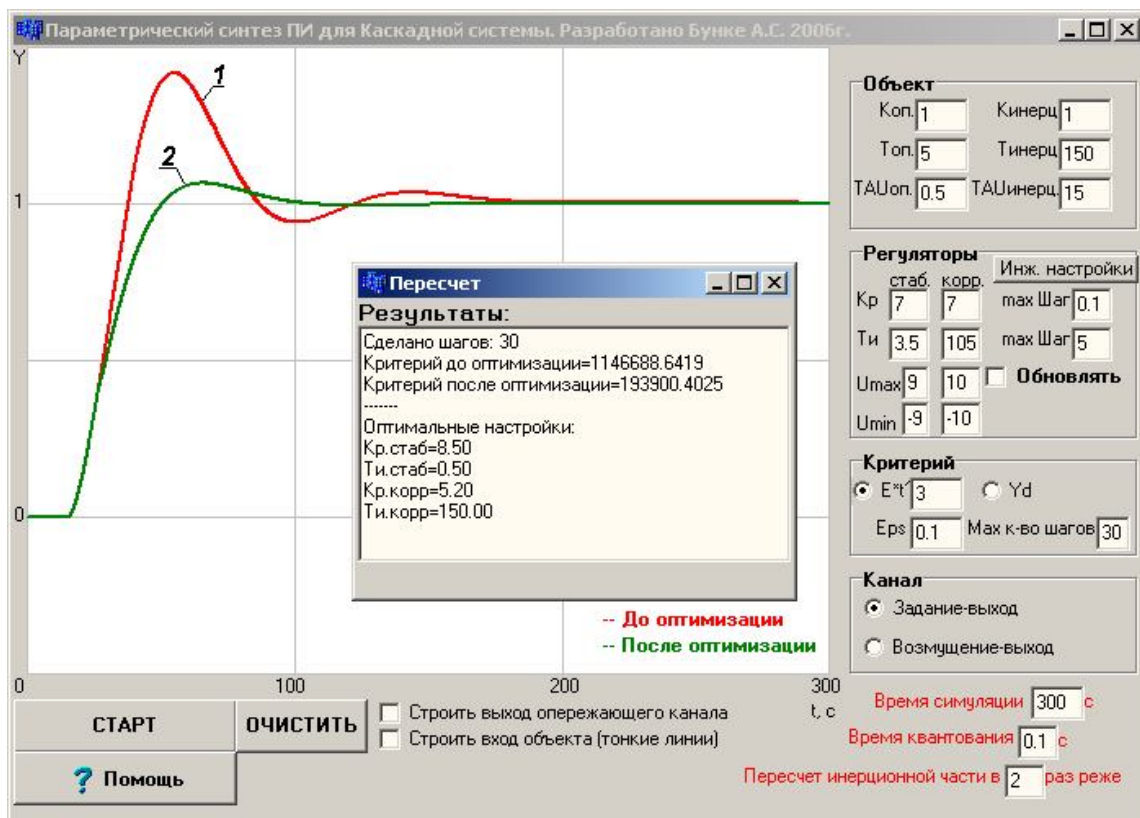


Рис. 5.16. Розрахунок каскадної АСР методом моделювання і оптимізації.

1 – перехідний процес до оптимізації; 2 – перехідний процес після 30 кроків оптимізації.

### 5.5. Методи випадкового пошуку

Всі розглянуті дотепер методи відносились до регулярних методів пошуку екстремуму, тобто до методів, у яких напрямок і величина робочого кроку вибираються не випадковим чином. Це означає, що при тих самих умовах задачі оптимізації, визначених алгоритмом, щоразу буде обрано один і той же напрямок пошуку та величина робочого кроку.

Існують так звані методи випадкового пошуку, що відрізняються від регулярних (детермінованих) методів оптимізації навмисним введенням елементів випадковості. Завдяки цьому, у тих самих умовах можуть бути обрані різні напрямки і величини робочого кроку в процесі пошуку екстремуму.

Хоча методи випадкового пошуку можуть бути формально віднесені до методів нульового порядку, але наявність елементів випадковості (випадкового напрямку й випадкової точки) дозволяє виділити їх у самостійну групу.

### 5.5.1. Послідовність випадкових чисел

Елементи випадковості в методах випадкового пошуку реалізуються за допомогою випадкових чисел, рівномірно розподілених на деякому числовому відрізку.

Найбільш простим способом отримання випадкових чисел є вибірка їх зі спеціальних таблиць. Однак введення більших масивів випадкових чисел в ЕОМ приводить до зайвого завантаження пам'яті обчислювальної машини.

Тому найбільше поширення знайшли програмні способи одержання послідовностей випадкових чисел, засновані на використанні певних алгоритмів. Необхідно відзначити, що за допомогою цих алгоритмів можуть бути знайдені лише послідовності псевдовипадкових чисел, які насправді не є випадковими, а можуть розглядатися такими лише в тому випадку, якщо об'єм вибірки випадкових чисел не занадто великий.

Звичайно для програмних способів отримання випадкових чисел використовуються наступні алгоритми, засновані на застосуванні рекурентних співвідношень.

#### Алгоритм I

$$Z_{i+1} = K \cdot Z_i \bmod M$$

Це означає, що кожне наступне випадкове число  $Z_{i+1}$  дорівнює остачі від ділення на  $M$  добутку числа  $Z_i$  на постійний множник  $K$ . Рекомендується вибирати  $K = 5^{17}$ ,  $M = 2^{42}$ ,  $Z_0 = 1$ .

#### Алгоритм II

$$Z_{i+1} = (K_1 \cdot Z_i + K_2) \bmod K_3$$

Це означає, що кожне наступне випадкове число  $Z_{i+1}$  дорівнює остачі від ділення величини  $(K_1 \cdot Z_i + K_2)$  на  $K_3$ . Рекомендується вибирати  $K_1 = 13077$ ,  $K_2 = 6925$ ,  $K_3 = 32767$ ,  $Z_0 = 13$ . Для знаходження випадкового напрямку  $\vec{a}$  за допомогою послідовності випадкових чисел  $Z_i$ ,  $i = 1, \dots, m$  необхідно пронормувати кожне з  $m$  випадкових чисел:

$$a_i = \frac{Z_i}{\sqrt{\sum_{i=1}^m Z_i^2}}, \quad i = 1, \dots, m. \quad (5.16)$$

Звідси випливає, що:

$$\sum_{i=1}^m a_i^2 = 1, \quad (5.17)$$

тому випадковий вектор  $\vec{a}$  має фактично довжину, рівну одиниці, і характеризує випадковий напрямок в  $m$ -мірному просторі.

Координати випадкової точки в  $m$ -мірному просторі  $u_i$ ,  $i = 1, \dots, m$  також можуть бути знайдені за допомогою сукупності випадкових чисел  $Z_i$ ,  $i = 1, \dots, m$  рівномірно розподіленої на якому-небудь інтервалі.

Нехай випадкові числа  $Z_i$  рівномірно розподілені в інтервалі  $[-c, +c]$ , а  $m$ -мірний допустимий простір для  $u_i$  задається умовами:

$$\alpha \leq u_i \leq \beta, \quad i = 1, \dots, m, \quad (5.18)$$

тоді:

$$u_i = \frac{z_i(\beta - \alpha) + c \cdot (\beta + \alpha)}{2 \cdot c}, \quad i = 1, \dots, m. \quad (5.19)$$

Нехай випадкові числа  $z_i$  рівномірно розподілені в інтервалі  $[\alpha, \beta]$ , а  $m$ -мірний допустимий простір для  $u_i$  задається умовами:  $-c \leq u_i \leq c$ ,  $i = 1, \dots, m$ .

Тоді:

$$u_i = \frac{(2 \cdot z_i - \alpha - \beta) \cdot c}{\beta - \alpha} \quad i = 1, \dots, m \quad (5.20)$$

Наведені співвідношення (5.19) і (5.20) придатні для перетворення рівномірно розподілених випадкових чисел з одного інтервалу в будь-який інший.

### 5.5.2. Сліпий пошук

Метод заснований на визначенні величини цільової функції в ряді випадкових точок  $\vec{u}_k$ , обраних у просторі незалежних змінних.

Спочатку випадковим чином вибирається початкова точка  $\vec{u}_1$ , у якій обчислюється значення цільової функції  $f_0(\vec{u}_1)$ . Далі вибирається  $\vec{u}_2$  й розраховується  $f_0(\vec{u}_2)$ . Якщо  $f_0(\vec{u}_2)$  більше (при пошуку максимуму)  $f_0(\vec{u}_1)$ , то значення  $\vec{u}_2$  й  $f_0(\vec{u}_2)$  запам'ятовуються. Потім вибирається точка  $\vec{u}_3$ , обчислюється  $f_0(\vec{u}_3)$ , порівнюється з  $f_0(\vec{u}_2)$  і т. д.

При досить великій кількості досліджених точок можна досягти високого ступеню точності у визначенні екстремуму.

Для зменшення числа досліджуваних точок метод сліпого пошуку можна модифікувати одним з наступних способів.

По-перше, заздалегідь обмежується число досліджуваних випадкових точок, після чого краща точка приймається в якості початкової для будь-якого іншого, у тому числі й детермінованого, алгоритму.

По-друге, простір розв'язків звужується симетрично щодо кращої із заздалегідь обмеженого числа досліджуваної точки, і надалі сліпий пошук триває в цьому значно меншому просторі.

### 5.5.3. Метод випадкових напрямків

Алгоритм цього методу полягає в тому, що із точки  $\vec{u}_k$ , у якій відоме значення  $f_0(\vec{u}_k)$  здійснюється крок у випадковому напрямку  $\vec{a}_k$ . Величина кроку визначається величиною крокового множника  $h_k$ .

У результаті знаходиться нова точка:

$$\vec{u}_{k+1} = \vec{u}_k + \vec{h}_k \cdot \vec{a}_k \quad (5.21)$$

Якщо при пошуку максимуму  $f_0(\vec{u}_{k+1}) > f_0(\vec{u}_k)$ , то робиться новий крок із точки  $\vec{u}_{k+1}$  у випадковому напрямку  $\vec{a}_{k+1}$  і т.д.

Якщо ж  $f_0(\vec{u}_{k+1}) < f_0(\vec{u}_k)$ , то вибирається наступний випадковий напрямок  $\vec{a}_k$  і робиться повторний крок із точки  $\vec{u}_k$ .

Пошук закінчується, якщо після виконання серії кроків більшого значення цільової функції знайти не вдається.

Величина крокового множника  $h_k$  може бути постійною, або змінною і вибиратися як за допомогою регулярних алгоритмів, так і за допомогою послідовностей випадкових чисел.

Існує модифікація методу випадкових напрямків – метод випадкових напрямків із зворотним кроком. Помітною її особливістю є те, що при невдалому кроці  $h_k \cdot \vec{a}_k$  із точки  $\vec{u}_k$  відразу здійснюється крок у зворотному напрямку –  $h_k \cdot \vec{a}_k$ , тобто реалізується алгоритм:

$$\vec{u}_{k+1} = \vec{u}_k + \vec{h}_k \cdot \vec{a}_k \cdot \text{sign} \left[ f_0(\vec{u}_{k+1}) - f_0(\vec{u}_{k+1}) \right] \quad (5.22)$$

Алгоритм (5.22) при достатньому віддаленні від екстремуму виявляється досить ефективним. Інша модифікація методу випадкових напрямків називається методом випадкових напрямків з лінійним прогнозом. Особливість цієї модифікації полягає в тому,

що після невдалого кроку із точки  $\bar{u}_k$  робиться крок у протилежному напрямку, як у методі зі зворотним кроком, але значення цільової функції в точці  $\bar{u}_{k+1} = \bar{u}_k - h_k \cdot \bar{a}_k$  не визначається, а перераховується в припущенні про лінійність цільової функції за формулою:

$$f_0(\bar{u}_{k+1}) = 2 \cdot f_0(\bar{u}_k) - f_0(\bar{u}_k + h_k \cdot \bar{a}_k) \quad (5.23)$$

Ще одна модифікація методу випадкових напрямків називається *методом найкращої проби*. У цій модифікації генерують деяке число  $m$  випадкових векторів  $\{\bar{a}_k^S\} = \{\bar{a}_k^{-1}, \bar{a}_k^{-2}, \dots, \bar{a}_k^{-m}\}$ ,  $S=1, \dots, m$ .

Потім здійснюється  $m$  пробних кроків за алгоритмом

$$\bar{u}_{k+1}^S = \bar{u}_k + h_k \cdot \bar{a}_k^S, \quad S = 1, \dots, m, \quad (5.24)$$

і як робочий крок вибирається вектор  $\bar{a}_k^S$ , що приводить до найбільшої зміни цільової функції. Точка  $\bar{u}_{k+1}^S$  приймається в якості початкової, і процес пошуку повторюється спочатку. Очевидно, що при  $m \rightarrow \infty$  процес пошуку аналогічний градієнтному методу.

## 5.6. Методи оптимізації при наявності обмежень

Крім методу множників Лагранжа, призначеного для розв'язку задач оптимізації з обмеженнями типу рівностей, існує два основних підходи до розв'язку задач умовної оптимізації.

Перший полягає в тому, що задача на екстремум при наявності обмежень зводиться до задачі безумовної оптимізації за допомогою спеціальних, так званих *штрафних функцій*. Методи, що використовують такі функції, називають методами штрафних функцій.

Другий підхід полягає в тому, що рух до екстремуму здійснюється по послідовності припустимих точок, тобто точок, що задовольняють обмеженням задачі з монотонно спадаючими (зростаючими) значеннями цільової функції. Цей підхід ілюструє метод проекції градієнта.

### 5.6.1. Методи штрафних функцій

Ці методи можуть використовуватися як при лінійних, так і при нелінійних обмеженнях типу рівностей і нерівностей.

Зміст методів полягає в тому, що вихідна цільова функція  $f_0(\bar{u})$  розширюється за рахунок введення додаткових коефіцієнтів і функцій, що враховують вплив обмежень таким чином, що при порушенні обмежень різко зростає (спадає) значення розширеної цільової функції  $I_p(\bar{u})$ . Тим самим рішення задачі умовної оптимізації  $f_0(\bar{u})$  зводиться до розв'язку задачі безумовної оптимізації  $I_p(\bar{u})$ .

Математична постановка задачі умовної оптимізації:

$$f_0(\bar{u}) \rightarrow \min_{\bar{u} \in V_u} V_u = \left\{ \bar{u} \in \mathbb{R}^n \mid g_j(\bar{u}) \geq 0, j=\overline{1, m}; h_i(\bar{u}) = 0, i=\overline{m+1, s} \right\} \quad (5.25)$$

де  $f_0(\bar{u})$  - цільова функція;

$V_u$  - область, на якій може приймати значення вектор  $\bar{u}$ ;

$g_j(\bar{u}) \geq 0, j=\overline{1, m}$  - умови типу нерівності;

$h_i(\bar{u}) = 0, i=\overline{m+1, s}$  - умови типу рівності.

Припускається, що для вектора  $\bar{u}^*$ , який є рішенням цієї задачі, відоме деяке початкове наближення  $\bar{u}^{(0)}$ , можливо неприпустиме, тобто таке, що не задовольняє умовам типів рівності та нерівності задачі. За допомогою алгоритмів, які розглядаються

нижче, у просторі  $R^n$  будується кінцева послідовність точок  $\bar{u}^{(i)}$ ,  $i=\overline{1, N}$ , яка починається із заданої точки  $\bar{u}^{(0)}$  та завершується точкою  $\bar{u}^{(N)}$ , яка дає найкраще наближення до  $\bar{u}^*$  серед усіх точок побудованої послідовності. У якості  $\bar{u}^{(i)}$ ,  $i=\overline{1, N}$  беруться стаціонарні точки так званої штрафної функції – цільової функції допоміжної задачі безумовної мінімізації. За допомогою штрафної функції вихідна задача умовної мінімізації перетворюється у послідовність задач безумовної мінімізації. Конкретні методи, побудовані на вказаній загальній схемі, визначаються видом штрафної функції, а також правилами, за якими проводиться перерахунок штрафних параметрів по закінченню чергового циклу безумовної мінімізації. Штрафна функція, яка дозволяє обмежитися рішенням лише однієї задачі безумовної мінімізації, називається точною.

Методи штрафних функцій мають просту реалізацію. Важлива їх якість також у тому, що вони дають досліднику багатий матеріал безумовної оптимізації.

Штрафна функція визначається виразом

$$F(\bar{u}) = f(\bar{u}) + P(r, g(\bar{u}), h(\bar{u})) \quad (5.26)$$

де  $r$  - набір штрафних параметрів.

Функція  $P(r, g(\bar{u}), h(\bar{u}))$  називається штрафом.

Розрізняють два методи штрафних функцій – метод внутрішньої точки (бар'єрних точок) та метод зовнішньої точки.

#### **Метод внутрішньої точки (бар'єрних точок)**

Розглянемо графічну інтерпретацію задачі на прикладі.

Приклад:  $f(u) = u \rightarrow \min_u, u - 2 \leq 0$ .

Графічно задачу зображено на рис. 5.17.

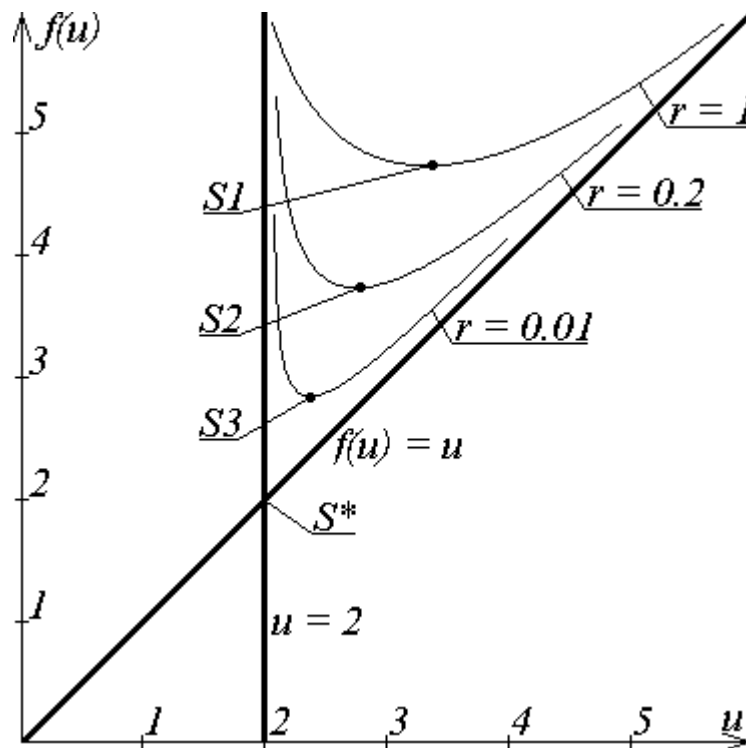


Рис. 5.17. Графічне зображення задачі умовної оптимізації

Складемо штрафну функцію:  $F(u) = u + \frac{r}{u-2}$ . Лінії рівня для  $r = 1; 0,2; 0,01$  зображені на рис. 5.17.

Легко бачити, що послідовність точок  $S_1, S_2, S_3$  прямує до точки  $S^*$  - мінімуму функції за наявності обмежень.

Таким чином, задача з обмеженнями перетворюється у послідовність підзадач безумовної оптимізації.

При розв'язуванні задачі необхідно обрати початкове значення штрафного параметра  $r$  та змінювати його після кожної під задачі безумовної оптимізації так, щоб забезпечити збіжність послідовності стаціонарних точок  $\bar{u}^{(k)}$  до оптимального значення.

Недоліки метода:

1. Необхідність пошуку допустимої точки, щоб використати її у якості початкової.
2. Штрафна функція  $F(\bar{u})$  має чітко виражену «яружну» структуру при малих  $r_k$ , що ускладнює рішення задач безумовної оптимізації  $F_k(\bar{u})$ .

3. Метод призначений для рішення задач нелінійного програмування з обмеженнями типу нерівності.

Метод передбачає побудову штрафів таким чином, щоб при наближенні вектора  $\bar{u} \in V_u$  до границі області  $V_u$  величина  $P(\bar{u}, r)$  необмежено зростає.

Траєкторія пошуку мінімуму, якщо пошук почався з внутрішньої точки, повністю буде лежати у  $V_u$ . Звідси і назва метода. Границя області  $V_u$  є своєрідним бар'єром, який неможна перейти у процесі пошуку. Звідси друга назва метода.

Для задачі

$$f(\bar{u}) \rightarrow \min_{\bar{u}}, g_i(\bar{u}) \geq 0, i=\overline{1,m}$$

у якості штрафів можуть бути побудовані наступні функції:

$$P(\bar{u}, r_k) = r_k \sum_{i=1}^m \frac{1}{g_i(\bar{u})} \tag{5.27}$$

$$P(\bar{u}, r_k) = -r_k \sum_{i=1}^m \ln g_i(\bar{u}) \tag{5.28}$$

Якщо рішення лежить на границі або біля неї, то для підвищення точності коефіцієнт  $r$  необхідно зменшувати.

Для задачі

$$f(\bar{u}) \rightarrow \min_{\bar{u}}, g_i(\bar{u}) \leq 0, i=\overline{1,m}$$

у якості штрафу може бути використано

$$P(\bar{u}, r_k) = \frac{1}{r_k} \sum_{i=1}^m (g_i(\bar{u}))^2, \tag{5.29}$$

де  $g_i(\bar{u}) = \max\{0, g_i(\bar{u})\}, i=\overline{1,m}$ .

**Метод зовнішньої точки**

Метод передбачає побудову штрафів таким чином, щоб значення перетвореної цільової функції  $F(\bar{u}, r)$  у допустимій області точно або наближено дорівнювали початковій цільовій функції  $f(\bar{u})$ , а зовні допустимої області – значно перевищували значення  $f(\bar{u})$ .

Для задачі

$$f(\bar{u}) \rightarrow \min_{\bar{u} \in V_u}, V_u = \{\bar{u} \in R^m \mid h_j(\bar{u}) = 0, j=\overline{1,m}\}$$

можуть бути використані наступні штрафи:

$$P(\bar{u}, r_k) = \frac{1}{r_k} \sum_{j=1}^m h_j(\bar{u}); \tag{5.30}$$

$$P(\bar{u}, r_k) = \frac{1}{r_k} \sum_{j=1}^m |h_j(\bar{u})|^p, \tag{5.31}$$

де  $P$  – довільне фіксоване число.

### *Алгоритм оптимізації методом штрафних функцій:*

1. Задання вихідних величин.
2. Побудувати  $F(\bar{u}, r_k)$ .
3. Знайти  $\bar{u}^{(k)} = \arg \min_{\bar{u}} F(\bar{u}, r_k)$  при фіксованому  $r_k$  з використанням метода безумовної оптимізації.
4. Перевірити умову

$$\left| F(\bar{u}^{(k+1)}, r_k) - F(\bar{u}^{(k)}, r_k) \right| \leq \varepsilon_3,$$

а також, можливо

$$\begin{aligned} \|\bar{u}^{k+1} - \bar{u}^k\| &< \varepsilon_4, \\ |f(\bar{u}^{k+1}) - f(\bar{u}^k)| &< \varepsilon_5, \\ |r_{k+1} - r_k| &< \varepsilon_6. \end{aligned}$$

Умови виконуються: Кінець.

Умови не виконуються: Перехід до наступного кроку.

5.  $r_{k+1} = r_k + \Delta r_k$ . Перехід до кроку 2.

### *Комбіновані алгоритми методу штрафних функцій*

Перетворена цільова функція може бути представлена у наступному вигляді:

$$F(\bar{u}) = f(\bar{u}) - r_k \sum_{j=1}^{m_1} \ln g_j(\bar{u}) + \frac{1}{r_k} \sum_{i=1}^{m_2} h_i^2(\bar{u}) \quad (5.32)$$

$$F(\bar{u}) = f(\bar{u}) - r_k \sum_{j=1}^{m_1} \frac{1}{g_j(\bar{u})} + \frac{1}{r_k} \sum_{i=1}^{m_2} h_i^2(\bar{u}) \quad (5.33)$$

#### **5.6.2. Метод проекції градієнта**

Цей метод придатний для розв'язку задач оптимізації при наявності лінійних обмежень типу рівностей і нерівностей.

Необхідно відзначити, що нелінійні обмеження можуть бути апроксимовані в околиці розглянутої точки за допомогою заміни нелінійних функцій членами першого порядку у відповідних розкладах у ряд Тейлора. Завдяки такій лінійній апроксимації, метод проекції градієнта може бути використаний і у випадку нелінійних обмежень.

Ідея методу полягає в тому, що робиться крок по проекції градієнта  $\nabla f_0(\bar{u}_k)$  на множину, що задається лінійними (або лінеарізованими) активними обмеженнями. Під *активними обмеженнями*  $g_j(\bar{u})$  розуміються всі обмеження типу рівностей й обмеження типу нерівностей, які в точці  $\bar{u}_k$  приймають значення рівні нулю і є лінійно незалежними. По суті, цей метод являє собою градієнтний метод у сполученні з ортогональним проектуванням антиградієнту на допустиму множину.

Алгоритм методу має вигляд:

$$\bar{u}_{k+1} = \bar{u}_k - \bar{\omega}_k, \quad (5.34)$$

$$\bar{\omega}_k = h_k \cdot P_g^k \cdot \nabla f_0(\bar{u}_k), \quad (5.35)$$

$$P_g^k = I - A_g^T (A_g \cdot A_g^T)^{-1} \cdot A_g \quad (5.36)$$

де  $h_k$  – кроковий множник, що визначає довжину кроку в напрямку  $P_g^k \cdot \nabla f_0(\bar{u}_k)$ .

Величина його вибирається зазвичай з умов, що  $f_0(\bar{u})$  повинна зменшуватися (збільшуватися) уздовж обраного напрямку, і що траєкторія руху повинна залишатися у середині допустимої області.

$A_g$  – матриця (розміром  $m \times l$ ) частинних похідних активних лінійно незалежних обмежень, строки якої являються множиною нормалей до поверхонь обмежень:



$$A_g = \begin{pmatrix} \frac{\partial g_1(\vec{u}_k)}{\partial u_1} & \dots & \frac{\partial g_1(\vec{u}_k)}{\partial u_m} \\ \dots & \dots & \dots \\ \frac{\partial g_l(\vec{u}_k)}{\partial u_1} & \dots & \frac{\partial g_l(\vec{u}_k)}{\partial u_m} \end{pmatrix}, \quad (5.37)$$

$P_g^k$  – матриця проектування градієнту  $\nabla f_0(\vec{u}_k)$  на перетин активних лінійно незалежних обмежень, що задовольняють умові:  $g_j(\vec{u}) = \vec{a}_j \cdot \vec{u} = 0$ ,  $j = 1, \dots, l$ ;

Дію алгоритму (5.34) ÷ (5.36) необхідно продовжувати доти, доки не буде виконана умова  $P_g^k \cdot \nabla f_0(\vec{u}_k) = 0$ , за якої точка  $\vec{u}_k$  задовольняє необхідним умовам Лагранжа (на практиці перевіряють виконання умови  $\|P_g^k \cdot \nabla f_0(\vec{u}_k)\| \leq \varepsilon$ ). Після цього обчислюється  $l$ -мірний вектор:

$$\vec{\xi}_k = (A_g \cdot A_g^T)^{-1} \cdot A_g \cdot \vec{u}_k \quad (5.38)$$

Якщо кожна складова  $\xi_j$ ,  $j = 1, \dots, l$  цього вектору більше або дорівнює нулю, то це означає, що  $\vec{u}_k$  є оптимальною точкою. Якщо деякі  $\xi_j < 0$ , то із різноманіття обмежень необхідно виключити відповідні  $j$ -і обмеження і повторити алгоритм (5.34) ÷ (5.36). Останні умови базуються на тому, що якщо  $P_g^k \cdot \nabla f_0(\vec{u}_k) = 0$ , то можна записати  $\nabla f_0(\vec{u}_k) = \sum_{j=1}^l \xi_j \cdot \vec{a}_j$ , де  $\xi_j$  – скалярні коефіцієнти, що складають вектор  $\vec{\xi}_k$ . В силу цього, якщо  $\xi_j \geq 0$  для всіх  $j = 1, \dots, l$ , то задовольняється умова теореми Куна-Такера і, отже,  $\vec{u}_k$  є оптимальною точкою.

Метод проекції градієнту вважається досить ефективним. Його графічна інтерпретація наведена на рис. 5.18.

### Основний алгоритм методу проекції градієнту:

Вирахувати  $P_g^k$  вважаючи, що всу вектори  $\vec{a}_j$  лінійно незалежні, задана допустима точка  $u^{(t)}$  та допустима похибка зходимості  $\varepsilon > 0$ .

- 1) Вирахувати  $s^{(k)} = -P_g^k \cdot \nabla f_0(\vec{u}_k)$ .
- 2) Якщо  $\|s^{(k)}\| \leq \varepsilon$ , то обчислити  $\vec{\xi}_k$ , використовуючи формулу (5.38), та завершити обчислення. В іншому випадку продовжити обчислення.
- 3) Визначити максимальну довжину кроку:

$$h_{\max} = \min \left\{ \max \left( 0, \frac{b_j - a_j^T u^{(k)}}{a_j^T s^{(k)}} \text{ або } \infty, \text{ якщо } a_j^T s^{(k)} = 0 \right); j = 1, \dots, K \right\}.$$

- 4) Вирішити задачу одномірного пошуку: мінімізувати  $f(u^{(k)} + h_k \cdot s^{(k)})$ ,  $0 \leq h \leq h_{\max}$ .
- 5) Прийняти  $u^{(k+1)} = u^{(k)} + h_k s^{(k)}$  та перейти до п. 1.

Для обчислення  $P_g^k$  необхідно, щоб векторі коефіцієнтів  $\vec{a}_j$  в обмеженнях були лінійно незалежними. Оскільки залежні лінійні обмеження не впливають на рішення і їх можна виключити із розгляду, ця вимога не є суттєвою.

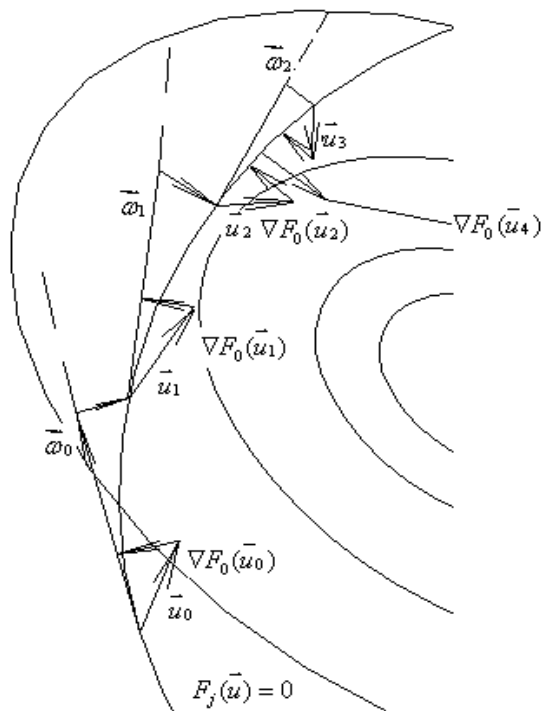


Рис. 5.18. Оптимізація методом проєкцій градієнта

**Приклад 5.4. Розв'язок задачі мінімізації методом проєкції градієнту.**

Мінімізувати  $f(x) = (x_1 - 1)^2 + (x_2 - 2)^2$

При обмеженнях  $g_1(x) = x_1 - 2x_2 \geq -2$

$g_2(x) = -x_1 - x_2 \geq -4$

$g_3(x) = x_1 \geq 0, g_4(x) = x_2 \geq 0,$

де в якості допустимої початкової точки використовується точка  $x^{(0)} = (0, 0)$ .

Розв'язок.

В точці  $x^{(0)}$  обмеження  $g_1 \geq 0$  та  $g_2 \geq 0$  неактивні, а умови невід'ємності  $g_3 \geq 0$  та  $g_4 \geq 0$  обертаються у рівності. Таким чином,

$$a_3 = (1; 0)^T, a_4 = (0; 1)^T \text{ та } A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Маємо  $AA^T = I, (AA^T)^{-1} = I$  та  $P = (I - I) = 0$ .

**Крок 1.**  $s^{(0)} = -P \cdot \nabla f = -0 \begin{pmatrix} -2 \\ -4 \end{pmatrix} = 0$ . Таким чином, необхідно перевірити відповідні

обмеження та множники

$$\xi^{(0)} = (AA^T)^{-1} A \nabla f = I \begin{pmatrix} -2 \\ -4 \end{pmatrix} = \begin{pmatrix} -2 \\ -4 \end{pmatrix}.$$

Оскільки  $\xi_2^{(0)}$  - найменший від'ємний множник, друге активне обмеження  $g_4(x) \geq 0$  виключається з множини активних обмежень. При цьому активна множина скорочується до єдиного обмеження  $g_3(x) \geq 0$ . Таким чином,

$$A = (1; 0) \text{ та } AA^T = (1; 0) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1.$$

Отже,

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1; 0) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Крок 1.

$$s^{(0)} = -P \cdot \nabla f = - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -2 \\ -4 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}.$$

**Крок 2.** Так як  $\|s^{(0)}\| = 4 > 0$ , то продовжуємо, починаючи з кроку 3.

Крок 3.

$$h_{\max} = \min \left\{ \begin{array}{l} \max \left( 0, \frac{-2-0}{-8} \right) \text{ для } g_1 \\ \max \left( 0, \frac{-4-0}{-4} \right) \text{ для } g_2 \\ \max(0, \infty) \text{ для } g_3, \text{ оскільки } (1, 0) \begin{pmatrix} 0 \\ 4 \end{pmatrix} = 0 \\ \max \left( 0, \frac{-0-0}{4} \right) \text{ для } g_4 \end{array} \right\} = \frac{1}{4}.$$

**Крок 4.** Шукаємо мінімум вздовж прямої

$$x = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + h \begin{pmatrix} 0 \\ 4 \end{pmatrix} \text{ при } 0 \leq h \leq \frac{1}{4}.$$

Функція  $f(h) = (-1)^2 + (4h - 2)^2$  монотонно спадає по  $h$ , доки значення  $h$  не стане рівним  $\frac{1}{2}$ . Отже,

$$h^* = h_{\max} = \frac{1}{4}.$$

**Крок 5.**  $x^{(1)} = (0, 1)^T$ .

В цій новій точці знову перевіряємо обмеження на активність. Очевидно, що обмеження  $g_1 \geq 0$  та  $g_3 \geq 0$  активні. Таким чином,

$$A = \begin{pmatrix} 1 & -2 \\ 1 & 0 \end{pmatrix}.$$

Легко перевірити, що  $P = 0$ ,  $s = -P \cdot \nabla f = 0$ . Вектор множників  $\xi^{(1)}$  дорівнює  $(1; -3)^T$ . Так як  $\xi_2^{(1)} < 0$ , друге обмеження  $g_3 \geq 0$  необхідно виключити з множини активних обмежень. Тепер активна множина обмежень складається лише з  $g_1 \geq 0$ .

Перераховуючи  $P$  при  $A = (1, -2)$ , одержуємо

$$P = \begin{pmatrix} 0,8 & 0,4 \\ 0,4 & 0,2 \end{pmatrix}.$$

Крок 1.

$$s^{(1)} = -P \cdot \nabla f = - \begin{pmatrix} 0,8 & 0,4 \\ 0,4 & 0,2 \end{pmatrix} \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{pmatrix} 2,4 \\ 1,2 \end{pmatrix}.$$

**Крок 2.**  $\|s^{(1)}\| > 0$ ; тому продовжуємо обчислення.

**Крок 3.** Довжина кроку  $h_{\max} = \frac{5}{6}$  визначається обмеженням  $g_2 \geq 0$ .

**Крок 4.** Далі шукаємо мінімум функції

$$f(h) = (2,4h - 1)^2 + (1 + 1,2h - 2)^2.$$

Мінімум досягається в  $h^* = \frac{1}{2} < h_{\max}$ .

**Крок 5.** Новою точкою є  $x^{(2)} = (1, 2; 1, 6)^T$ . В цій точці активне лише перше обмеження. Множина активних обмежень лишається без змін, тому немає необхідності перераховувати  $P$ . В точці  $x^{(2)}$  маємо

$$s^{(1)} = -P \cdot \nabla f = -\begin{pmatrix} 0,8 & 0,4 \\ 0,4 & 0,2 \end{pmatrix} \begin{pmatrix} 0,4 \\ -0,8 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$
$$\xi^{(2)} = (AA^T)^{-1} A \nabla f = \frac{1}{5} (1; -2) \begin{pmatrix} 0,4 \\ -0,8 \end{pmatrix} = 0,4 > 0.$$

Таким чином, точка  $x^{(2)}$  задовольняє умовам Куна-Такера.

## 5.7. Контрольні завдання

1. Знайти методом Гауса-Зейделя мінімум функції  $f(x) = (x_1 - x_2)^2 + \frac{(x_1 + x_2 - 10)^2}{9}$ .

Початкова точка  $x^0 = (0; 1)$ .

2. Надаючи параметрам  $\alpha$  і  $\beta$  в методі Розенброка різні значення, визначте найбільш ефективні з них для випадка квадратичної функції

$$f(x) = 10x_1^2 + 0,1x_2^2.$$

Повторіть те саме для

$$f(x) = 0,1x_1^2 + 10x_2^2.$$

3. Використайте пошук Розенброка для знаходження мінімуму цільової функції

$$f(x) = 3x_1^2 + x_2^2$$

при  $\alpha = 3$  і  $\beta = \frac{1}{2}$ , доки подальше покращення не стане неможливим.

4. Задана функція має мінімуми в точках  $(0; 1)$  та  $(1; 0)$  та сідлову точку  $(4^{-1/3}; 4^{-1/3})$ . Починаючи з точки  $x = (3; 3)$ , побудуйте траєкторії мінімізації (вказуючи

значення цільової функції  $f(x) = (x_1^2 + x_2^2 - 1)^2 + (x_1 + x_2 - 1)^2$  в залежності від числа ітерацій) для методів:

- а) Хука-Дживса
- б) пошук по симплексу
- в) Розенброка

Що станеться, якщо почати з точки  $(4^{-1/3}; 4^{-1/3})$ ?

Мінімізуйте задану функцію методами випадкового пошуку.

5. Знайдіть мінімум та максимум функції

$$f(x) = 20 + 0,3x_1 - 4x_2 + 0,3x_1^2 + 0,3x_2^2 + 0,4x_1x_2$$

одним з методів пошуку. Починайте з точок:

- а)  $x^0 = (0,25; 2,5)$ ,
- б)  $x^0 = (2,5; 2,5)$ ,
- в)  $x^0 = (-0,25; -2,5)$ .

6. Мінімізувати функцію

$$f(x) = (x_1 - 4)^2 + (x_2 - 4)^2$$

методом штрафних функцій при обмеженні  $h(x) = x_1 + x_2 - 5 = 0$ .

7. Мінімізувати функцію

$$f(x) = (x_1 - 4)^2 + (x_2 - 4)^2$$

методом штрафних функцій при обмеженні  $g(x) = 5 - x_1 - x_2 \geq 0$ .

8. Дана задача:

Мінімізувати функцію  $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

при обмеженнях:

$$g_1(x) = x_1 + 1 \geq 0;$$

$$g_2(x) = 1 - x_2 \geq 0;$$

$$g_3(x) = 4x_2 - x_1 - 1 \geq 0;$$

$$g_4(x) = 1 - 0,5x_1 - x_2 \geq 0.$$

Запишіть для цієї задачі штрафні функції, використовуючи:

1) логарифмічний штраф;

2) штраф, що задається оберненою функцією. Дослідіть кожну з функцій в точках  $x^{(0)} = (-1; 1)$  та  $x^{(0)} = (-0,5; 0,5)$ , які складності при цьому виникають?

9. Мінімізувати методом проекції градієнта функцію

$$f(x) = x_1^2 + x_2^2$$

при обмеженні  $h_1(x) = (x_1 - 1)^3 - x_2^2 = 0$ .

10. Мінімізувати методом проекції градієнта функцію

$$f(x) = x_1^2 + 2x_2^2$$

при обмеженнях:

$$g_1(x) = x_1^2 + x_2^2 \leq 0;$$

$$h_2(x) = 2x_1 - 2x_2 - 1 = 0.$$

11. Мінімізувати методом проекції градієнта функцію

$$f(x) = 4x_1 - x_2^2 - 12$$

при обмеженнях:

$$h_1(x) = 25 - x_1^2 - x_2^2 = 0;$$

$$g_2(x) = 10x_1 - x_1^2 + 10x_2 - x_2^2 - 34 \geq 0;$$

$$g_3(x) = x_1 \geq 0;$$

$$g_4(x) = x_2 \geq 0.$$