Зміст

Розділ 1

Дослідження операцій

1.1 Задачі багатокритерійної оптимізації

13.02.2014

Математична модель:

$$\max f_i(x), i \in I_1 = \{1, \dots, m\} \tag{1.1.1}$$
$$\min f_i(x), i \in I_2 = \{m+1, \dots, M\} \tag{1.1.2}$$

Введемо наступні відношення переваги та еквівалентності на множині альтернатив.
Відношення **нестрогого переваги** $x \succeq y$ виконується, якщо:

$$\begin{cases} f_i(x) \geq f_i(y), i \in I_1 \\ f_i(x) \leq f_i(y), i \in I_2 \end{cases} \tag{1.1.3}$$

Відношення **строгої переваги** $x \succ y$, якщо виконуються попередні обмеження (1.1.3) і одне виконується строго.
Відношення **еквівалентності** $x \sim y$, якщо $f_i(x) = f_i(y), i \in I = \{1, \dots, M\}$
Альтернатива x_0 називається **паретто оптимальною** або **ефективною**, якщо не існує такої альтернативи x', для якої виконується умова:

$$\begin{cases} f_i(x') \geq f_i(x_0), i \in I_1 \\ f_i(x') \leq f_i(x_0), i \in I_2 \end{cases} \tag{1.1.4}$$

Теорема 1.1.1. *Дві ефективні альтернативи x_1 та x_2 або є непорівнянними, або еквівалентні.*

Доведення. Нехай ці дві альтернативи є порівняними між собою. Так, як x_1 ефективна альтернатива, то для неї виконується умова:

$$\begin{cases} f_i(x_1) \geq f_i(x_2), i \in I_1 \\ f_i(x_1) \leq f_i(x_2), i \in I_2 \end{cases} \tag{1.1.5}$$

Так як x_2 ефективна альтернатива, то для неї виконується умова:

$$\begin{cases} f_i(x_2) \geq f_i(x_1), i \in I_1 \\ f_i(x_2) \leq f_i(x_1), i \in I_2 \end{cases} \tag{1.1.6}$$

Порівнюючи умови (1.1.5) та (1.1.6) отримуємо, що:

$$f_i(x_1) = f_i(x_2), i \in I = \{1, \ldots, M\} \tag{1.1.7}$$

□

Нехай задані два вектора критерії $f^1(x)$ та $f^2(x)$ називаються **еквівалентними** ($f^1(x) \sim f^2(x)$), якщо вони породжують однакові відношення еквівалентності та предпочтення.
Позначимо через $f_i^0 = \max f_i(x), i \in I_1$, через $f_i^0 = \min f_i(x), i \in I_2$
Розглянемо довільну альтернативу x^* і ступінь близькості до оптимальних значень оцінимо наступним чином:

$$\Delta f_i(x^*) = \begin{cases} f_i^0 - f_i(x), i \in I_1 \\ f_i(x) - f_i^0, i \in I_2 \end{cases} \tag{1.1.8}$$

Найкращою компромісною альтернативою варто рахувати таку альтернативу, для якої ступінь відхилення від оптимальних значень по всім критеріям мінімальна.
Так як цільові функції мають різну фізичну розмірність, то зручно розглядати не саму множину цільових функцій, а еквівалентну їй множину $W(x) = \{w_i(x) | i \in I\}$, де $w_i(x)$ - це монотонні перетворення, що приводять цільові функції до безрозмірного виду. Це монотонне перетворення має задовольняти наступним вимогам:

- Мати єдину точку відліку та однакову шкалу зміни значень;
- Зберігається множина ефективних альтернатив;
- Враховувати необхідність мінімізації відхилень від оптимальних значень по всім критеріям.

Монотонне перетворення визначається наступним чином:

$$w_i(x) = \begin{cases} \dfrac{f_i^0 - f_i(x)}{f_i - f_{i,min}}, i \in I_1 \\ \dfrac{f_i(x) - f_i^0}{f_{i,max} - f_i^0}, i \in I_2 \end{cases} \tag{1.1.9}$$

$$0 < w_i(x) < 1$$

Ще одна формула для монотонного перетворення:

$$w_i(x) = \begin{cases} \dfrac{f_i^0 - f_i(x)}{f_i^0}, i \in I_1 \\ \dfrac{f_i(x) - f_i^0}{f_i^0}, i \in I_2 \end{cases} \tag{1.1.10}$$

В даному випадку шкала не одинична.

Теорема 1.1.2. *Для будь-якої альтернативи існують вагові коефіцієнти* $\exists \rho_i \geq 0, \sum_{i=1}^{M} \rho_i = 1$ *і параметри* k_0 *такий, що*

$$\rho_i \cdot w_i(x) = k_0 \tag{1.1.11}$$

Теорема 1.1.3. *Якщо для двох альтернатив* x' *та* x'' *вагові коефіцієнти співпадають, то виконується умова:*

$$\begin{cases} k'_0 = \gamma k''_0 \\ w_i(x') = \gamma w_i(x'') \end{cases} \tag{1.1.12}$$

Де γ *коефіцієнт пропорційності.*

Теорема 1.1.4. *Для того, щоб альтернатива* x_0 *була ефективною, досить того, щоб вона була єдиними розв'язком системи:*

$$\rho_i w_i(x) \leq k_{0,min} \tag{1.1.13}$$

Теорема 1.1.5. *Якщо* x_0 *ефективна альтернатива, то для неї відносно зважені відхилення від оптимальних значень однакові та мінімальні.*

1.1.1 Метод обмежень

Від багатокритеріальної задачі (1.1.1)-(1.1.2) переходимо до однокритеріальної задачі

$$\min k_0 \tag{1.1.14}$$
$$\rho_i w_i(x) \leq k_0, i \in I \tag{1.1.15}$$

Використовуючи умову для монотонного перетворення (1.1.9) перетворюємо систему (1.1.15) у:

$$f_i(x) \geq f_i^0 - \frac{k_0}{\rho_i}\left(f_i^0 - f_{i,min}\right), i \in I_1 \tag{1.1.16}$$
$$f_i(x) \leq f_i^0 - \frac{k_0}{\rho_i}\left(f_{i,max} - f_i^0\right), i \in I_2 \tag{1.1.17}$$

Необхідно знайти мініальний k_0 при якому система (1.1.16)-(1.1.17) буде сумісною.

Крок 1. $k_0 = k_0(0)$

Крок 2. Зменшуємо $k_0(1) = k_0(0) - \Delta k_0$. Перевіряємо, чи сумісні системи. Якщо сумісні, то на наступний крок;

⋮

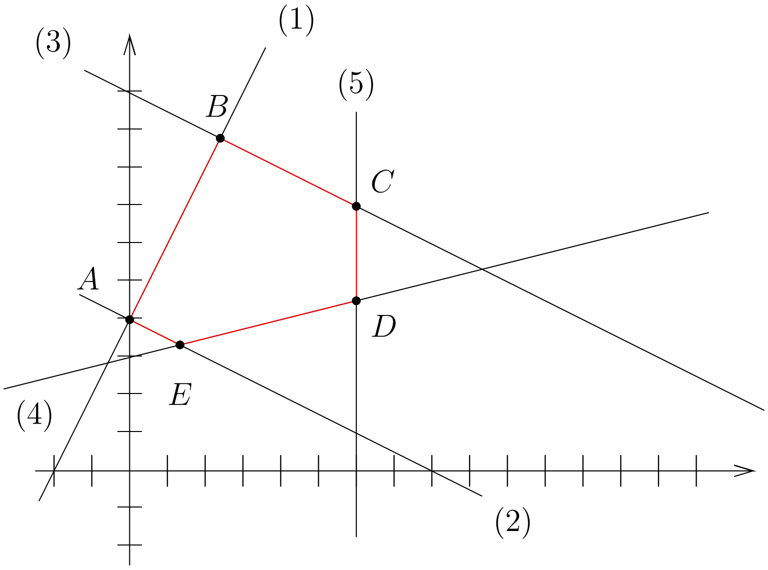
Крок n. $k_0(n-1) = k_(n-2) - \Delta k_0$

Нехай на n-тій ітерації система сумісна, а на n+1-шій вже не сумісна. Тоді нашим розв'язком буде $k_0(n-1)$.
Так як загальний метод не залежить від функціональної залежності, то для кожного виду залежності потрібно знаходити свій спосіб перевірки сумісності систем (1.1.16)-(1.1.17)

Приклад 1.1.1. Записуємо математичну модель:

$$f_1(x) = 3x_1 + 2x_2 \to \max$$
$$f_2(x) = x_1 - 3x_2 \to \min$$
$$-2x_1 + x_2 \leq 4$$
$$x_1 + 2x_2 \geq 8$$
$$x_1 + 2x_2 \leq 20$$
$$-x_1 + 4x_2 \geq 12$$
$$x_1 \leq 6$$
$$x_1, x_2 \geq 0$$
$$\rho_1 = \rho_2 = \tfrac{1}{2}$$

Знайдемо потрібні розв'язки для перетворення (1.1.9) графічним методом.



Отримали випуклий багатокутник з вершинами:
$A(0,4);B\left(2.4,\dfrac{44}{5}\right);C(6,7);D(6,4.5);E\left(\dfrac{4}{3},\dfrac{10}{3}\right)$
Будуємо вектори нормальні.
$f_1^0(C) = 32$; $f_{1,min}(A) = 8$
$f_2^0(B) = -24$; $f_{2,max}(D) = -7.5$
Отже, тепер можливо використати монотонне перетворення (1.1.9).

$$w_1(x) = \frac{32 - (3x_1 + 2x_2)}{32 - 8}$$
$$w_2(x) = \frac{x_1 - 3x_2 + 24}{-7.5 + 24}$$

Тепер ми можемо перейти до однокритеріальної задачі:

$$\min k_0 = x_3$$
$$\frac{1}{2} \cdot \frac{32 - 3x_1 - 2x_2}{32 - 8} \leq x_3$$
$$\frac{1}{2} \cdot \frac{x_1 - 3x_2 + 24}{-7.5 + 24} \leq x_3$$
$$-2x_1 + x_2 \leq 4$$
$$x_1 + 2x_2 \geq 8$$
$$x_1 + 2x_2 \leq 20$$
$$-x_1 + 4x_2 \geq 12$$
$$x_1 \leq 6$$
$$x_1, x_2 \geq 0$$
$$\rho_1 = \rho_2 = \tfrac{1}{2}$$

Зрозуміло, що кращій компромісний розв'язок буде знаходитися на прямій BC. Тобто $x_1 + 2x_2 = 20$.

А з теореми 1.1.5 можна отримати $\dfrac{32 - 3x_1 - 2x_2}{32 - 8} = \dfrac{x_1 - 3x_2 + 24}{-7.5 + 24}$

Розв'язавши дану систему можна знайти x_1, x_2.

$x_1^0 = 3.8, x_2^0 = 8.1$

1.2 Знаходження найкращої альтернативи в задачах багатокритерійного вибору

27.02.2014

Постановка задачі: Задані критерії та відношення переваги та вагові коефіцієнти важливості:

$$f_i, j = \overrightarrow{1, n} \tag{1.2.1}$$
$$R_j, j = \overrightarrow{1, n} \tag{1.2.2}$$
$$\omega_j \geq 0, j = \overrightarrow{1, n} \tag{1.2.3}$$

Необхідно знайти найкращу альтернативу. Для розв'язку цієї задачі необхідно визначити ефективний спосіб згортання векторного критерію в скалярний. Способи:

- $Q_1 = R_1 \cap R_2 \ldots \cap R_n$
- $Q_2 = \sum_{j=1}^{n} \omega_j R_j$

Отже, необхідно знайти найкращу альтернативу за обома згортками.

Крок 1. Будуємо функцію приналежності заданих відношень переваги.

$$\mu_{R_j}(x,y) = \begin{cases} 1 & , x \succ y, x \sim y \\ 0 & \end{cases} \tag{1.2.4}$$

Крок 2. Визначаємо першу згортку та будуємо функцію приналежності.

$$Q_1 = \bigcap_{j=1}^{n} R_j \tag{1.2.5}$$

$$\mu_{Q_1}(x,y) = \min\left\{\mu_{R_1}(x,y), \ldots, \mu_{R_n}(x,y)\right\} \tag{1.2.6}$$

Крок 3. Визначаємо відношення строгої переваги, тобто ступінь, з якою альтернатива x краще за альтернативу y і будуємо функцію приналежності Q_1^S.

$$\mu_{Q_1^S}(x,y) = \max\left\{0; \mu_{Q_1}(x,y) - \mu_{Q_1}(y,x)\right\} \tag{1.2.7}$$

Крок 4. Визначаємо множину альтернатив, що не домінують, тобто Q_1^n. Тобто, ступінь з якої альтернатива x не домінується ніякої іншою альтернативою.

$$\mu_{Q_1^n} = 1 - \max\left\{\mu_{Q_1^S}(y,x)\right\} \tag{1.2.8}$$

Крок 5. Визначаємо другу згортку за формулою $Q_2 = \sum_{j=1}^{n} \omega_j R_j$ і будуємо функцію приналежності:

$$\mu_{Q_2}(x,y) = \sum_{j=1}^{n} \omega_j \mu_{R_j}(x,y) \tag{1.2.9}$$

Крок 6. Визначаємо відношення строгої переваги за другою згорткою Q_2^S і будуємо функцію приналежності.

$$\mu_{Q_2^S}(x,y) = \max\left\{0; \mu_{Q_2}(x,y) - \mu_{Q_2}(y,x)\right\} \tag{1.2.10}$$

Крок 7. Визначаємо множину альтернатив, що не домінують за другою згорткою Q_2^n і будуємо функцію приналежності.

$$\mu_{Q_2^n} = 1 - \max\left\{\mu_{Q_2^S}(y,x)\right\} \tag{1.2.11}$$

Крок 8. Визначаємо множину альтернатив, що не домінують за обома згортками $Q^n(X) = Q_1^n(X) \cap Q_2^n(X)$ і будуємо функцію приналежності:

$$\mu_{Q^n}(x) = \min\left\{\mu_{Q_1^n}(x), \mu_{Q_2^n}(x)\right\} \tag{1.2.12}$$

Крок 9. Визначаємо найкращу альтернативу за обома згортками. Найкращою альтернативою є така альтернатива x_0, для якої виконується:

$$\mu_{Q^n}(x_0) = \max\left\{\mu_{Q^n}(x)\right\} \tag{1.2.13}$$

Якщо ступінь альтернативи, що не домінує є 1, то це **чітко не домінуєма альтернатива**.

Приклад 1.2.1. Розглядаються наступні альтернативи при купівлі квартири:

- x_1 - купити квартиру в центрі Києва;
- x_2 - купити квартиру на околоці Києва;
- x_3 - купити квартиру десь взагалі на краю Києва.

Вибір буде відбуватися за наступними критеріями:

- Властивість квартири;
- Затрати на дорогу;
- Загазованість навколишнього середовища;
- Досуг після роботи.

Задані коефіцієнти вагомості $\omega_1 = 0.4, \omega_2 = 0.25, \omega_3 = 0.2, \omega_4 = 0.15$.

$$\begin{array}{ll} R_1: & x_2 \succ x_1, x_3 \succ x_2 \\ R_2: & x_2 \succ x_3, x_2 \succ x_1, x_1 \sim x_3 \\ R_3: & x_2 \sim x_3, x_2 \succ x_1 \\ R_4: & x_1 \succ x_2, x_2 \sim x_3 \end{array}$$

Будуємо функцію приналежності:

$\mu_{R_1}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	0	0
x_2	1	1	0
x_3	1	1	1

$\mu_{R_2}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	0	1
x_2	1	1	1
x_3	1	0	1

$\mu_{R_3}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	0	0
x_2	1	1	1
x_3	1	1	1

$\mu_{R_4}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	1	1
x_2	0	1	1
x_3	0	1	1

Визначаємо першу згортку:

$\mu_{Q_1}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1

Будуємо відношення строгої приналежності та недомінуючість:

$\mu_{Q_1^s}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	0	0	0
x_2	0	0	0
x_3	0	0	0
$\mu_{Q_1^n}(x)$	1	1	1

Визначаємо другу згортку:

$\mu_{Q_2}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	1	0.15	0.4
x_2	0.85	1	0.6
x_3	0.85	0.75	1

Будуємо відношення строгої приналежності та недомінуючість:

$\mu_{Q_2^S}(x_i, x_j)=$

	x_1	x_2	x_3
x_1	0	0	0
x_2	0.7	0	0
x_3	0.45	0.15	0
$\mu_{Q_2^n}(x)$	0.3	0.85	1
$\mu_{Q_1^n}(x)$	1	1	1
$\mu_{Q^n}(x)$	0.3	0.84	1

Отже, чітку не домінуюча альтернатива це x_3.

Розділ 2

Дискретне програмування

2.1 Дискретне програмування:початок

$$\max f(x_1, \ldots, x_n) \tag{2.1.1}$$

$$g_i(x_1, \ldots, x_n) \leq b_i, i = \overrightarrow{1, m} \tag{2.1.2}$$

$$x_j \geq 0, j = \overrightarrow{1, n} \tag{2.1.3}$$

$$x_j \in D_j \tag{2.1.4}$$

Типи задач:

- Задачі програмування в цілих числах;
- Задачі булевого програмування;
- Загальний випадок дискретності D_j - скінченна множина.

2.1.1 Особливості

При розв'язку задач програмування в цілих числах в загальному випадку неможливо використати прийом заміни задач неперервним аналогом, а далі округлення до найближчого цілого.
Точні методи розв'язку:

- Метод площин, що відсікають;
- Метод гілок та границь;
- Метод аналізу та відкидання варіантів;
- Методи динамічного програмування.

Наближені методи пошуку:

- Метод випадкового пошуку;
- Метод вектора спаду;
- Метод локальної оптимізації;
- Модифікації точних методів.

2.2 Метод площин,що відсікають

$$\max\left\{\sum_{j=1}^{n} c_j x_j\right\} \tag{2.2.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = \overrightarrow{1,m} \tag{2.2.2}$$

$$x_j \geq 0 \tag{2.2.3}$$

$$x_j \in \mathbb{Z} \tag{2.2.4}$$

Ідея методу: від дискретної задачі переходимо до неперервного аналогу. Тобто, розв'язуємо задачу (2.2.1)-(2.2.3). Якщо отриманий розв'язок цілий, то це й буде шуканий розв'язок. В іншому випадку, до системи обмеження додаємо нове обмеження, яке не задовольняє знайдений оптимальний, але не цілий план, але задовольняє будь-який цілий. Знову розв'язуємо неперервGну задачу. Знаходимо оптимальний розв'язок і перевіряємо, чи виконується умова (2.2.4). Далі аналогічно. Обмеження, що вводяться таким чином називаються **правильнім відсіканням**. Якщо правильне відсікання працює ефективно, то за скінченну кількість ітерацій ми прийдемо до оптимального розв'язку або до ознаки нерозв'язності.

2.2.1 Умова збіжності

Якщо цільова функція на множині розв'язків обмежена цілком (зверху і знизу), а множина оптимальних розв'язків є випуклим багатогранником і рядок, що спрямовує формується за симплекс-таблицею з не цілою компонентою, а рядок, що спрямовує формується по рядку симплекс-таблиці за умовою $\Theta_j = \min\left|\frac{\Delta_l}{x_{il}}\right|_{x_{il}<0}$, то алгоритм таки працює. Виведемо формулу для правильного відсікання. 06.03.2014

Нехай всі обмеження задачі задані в канонічному виді

$$Ax = B \tag{2.2.5}$$

Розмірність матриці $A = (m \times n)$. Ранг матриці A дорівнює m. Нехай $m < n$. Базис містить m векторів. Тоді m стовпчиків матриці A утворює базисну матрицю. Інші стовпчики утворюють небазисну матрицю. Таким же чином розіб'ємо вектор x на базисний та небазисний і запишемо систему (2.2.5) в наступному вигляді

$$A_b x_b + A_{nb} x_{nb} = B \tag{2.2.6}$$

Помножимо систему (2.2.6) на матрицю, обернену до базисної

$$A_b^{-1} A_b x_b + A_b^{-1} A_{nb} x_{nb} = A_b^{-1} B \tag{2.2.7}$$

І з системи (2.2.7) запишемо розв'язок системи (2.2.5)

$$x_b = A_b^{-1}B - A_b^{-1}A_{nb}x_{nb} \tag{2.2.8}$$

$$x_i = a_{i0} - \sum_{j \in I_{nb}} a_{ij}x_j \tag{2.2.9}$$

Позначимо через γ_{i0} дробову частину a_{i0}

$$a_{io} = [a_{i0}] + \gamma_{i0} \tag{2.2.10}$$

А через γ_{ij} дробову частину a_{ij}

$$a_{ij} = [a_{ij}] + \gamma_{ij} \tag{2.2.11}$$

Підставим (2.2.10)-(2.2.11) у (2.2.9)

$$x_i - [a_{i0}] + \sum_{j \in I_{nb}} [a_{ij}]\, x_j = \gamma_{i0} - \sum_{j \in I_{nb}} \gamma_{ij}x_j \tag{2.2.12}$$

Нехай x_i ціле. Позначимо $\gamma_{i0} - \sum_{j \in I_{nb}} \gamma_{ij}x_j = \xi$ - також буде цілою.

Можливі два випадки:

- $\xi \geq 1$;
- $\xi \leq 0$.

Розглянемо перший випадок ($\xi \geq 1$):

$$\gamma_{i0} - \sum_{j \in I_{nb}} \gamma_{ij}x_j \geq 1 \tag{2.2.13}$$

$$\Rightarrow \gamma_{i0} \geq 1 \sum_{j \in I_{nb}} \gamma_{ij}x_j \geq 1 \tag{2.2.14}$$

Це протиріччя з умовою, що γ_{i0} - дробова частина. Отже, можливий лише один варіант ($\xi \leq 0$):

$$\gamma_{i0} - \sum_{j \in I_{nb}} \gamma_{ij}x_j \leq 0 \tag{2.2.15}$$

$$\gamma_{i0} \leq \sum_{j \in I_{nb}} \gamma_{ij}x_j \tag{2.2.16}$$

Не цілий план цій умові не задовольняє, а цілій задовольняє. Отримали правильно формулу, запишемо її:

$$-\sum_{j \in I_{nb}} \gamma_{ij}x_j \leq -\gamma_{i0} \tag{2.2.17}$$

Приводимо обмеження (2.2.17) до канонічного виду

$$-\sum_{j \in I_{nb}} \gamma_{ij}x_j + x_{n+1} = -\gamma_{i0} \tag{2.2.18}$$

Цей рядом дописуємо у симплекс-таблицю оптимального розв'язку неперервної задачі. І далі розв'язуємо задачу двоїстим симплекс-методом. $n+1$ рядок буде спрямовуючим. Сукупність ітерацій двоїстим симплекс-методом починаючи від цього розв'язку та до нового оптимального називається **великою ітерацією**. Можливі наступні результати великої ітерації:

Ознака оптимальності: $\forall i : x_i = x_{i0} \geq 0, x_i \in \mathbb{Z}$;

Ознака формування нового відсікання: $\forall i : x_i = x_{i0} \geq 0, \exists i^* : x_{i^*} \notin \mathbb{Z}$;

Ознака неров'язності: $\exists i^* : x_{i^*} = x_{i^*0} < 0, x_{ij} \geq 0$.

Змінні $x_{n+1}, x_{n+2}, \ldots$ називається **додатковими**, вони виводяться з базису на першій ітерації. Якщо якусь додаткову змінну знову необхідно ввести в базис, то відповідний рядок та стовпчик можна викинути. Дробові частини завжди записуються зі знаком мінус.

2.3 Метод гілок та границь для задач програмування в цілих числах

Записуємо математичну модель

$$\max \sum_{j=1}^{n} c_j x_j \tag{2.3.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = \overrightarrow{1, m} \tag{2.3.2}$$

$$x_j \geq 0, j = \overrightarrow{1, n} \tag{2.3.3}$$

$$x_j \in \mathbb{Z}, j = \overrightarrow{1, n} \tag{2.3.4}$$

Перший етап

Відкидаємо умову того, що $x_j \in \mathbb{Z}$ і розв'язуємо неперервну задачу (2.3.1)-(2.3.3). Оптимальний розв'язок позначимо як x_0. Якщо даний розв'язок цілий, то воно і буде шуканим. В іншому випадку визначаємо верхню межу (оцінку) для шуканого розв'язку. Оценку будемо позначати буквою $\xi(G_0) = f(x_0)$. І йдемо на першу ітерацію.

Ітерації

Нехай проведено k-ітерацій. Є ряд підмножин $\left\{G_1^k, \ldots, G_v^k\right\}$. Для кожної підмножини пораховані оцінки $\left\{\xi\left(G_1^k\right), \ldots, \xi\left(G_v^k\right)\right\}$. $k+1$-ітерація.

Перший етап: ветвление

Для ще одного ветвления обираємо підмножину з максимальною оцінкою. Нехай це підмножина $G_i^k : \xi\left(G_i^k\right) = \max\left\{\xi\left(G_j^k\right)\right\}$. G_j^k - множина вершин, що вісять. Тобто ті підмножини, які ще не ветвлились. Розбиваємо підмножину G_i^k на дві множини, що не перетинаються: $G_i^k = G_1^{k+1} \cup G_2^{k+1}$. Для цього в плані x_i^k обираємо деяку не цілу компоненту. Нехай це $x_r = x_{r0}$. Першу множину отримуємо таким чином:

$$G_1^{k+1} = G_i^K \cap \{x_r : x_r \leq [x_{r0}]\} \tag{2.3.5}$$

Другу множину отримуємо за такою формулою:

$$G_2^{k+1} = G_i^K \cap \{x_r : x_r \geq [x_{r0}] + 1\} \tag{2.3.6}$$

Другий етап

Знаходимо оптимальні розв'язки на підмножинах і обчислюємо оцінки для підмножин.
В симплекс-таблицю оптимального розв'язку на підмножині G_i^k дописуємо обмеження

$$x_r \leq [x_{ro}] \tag{2.3.7}$$

Розв'язуємо цю задачу двоїстим симплекс-методом і оптимальний план на підмножині G_1^{k+1} позначимо x_1^{k+1}. І оцінку $\xi\left(G_1^{k+1}\right) = f\left(x_1^{k+1}\right)$.
В симплекс-таблицю оптимального розв'язку на підмножині G_i^k дописуємо обмеження:

$$x_r \geq [x_{ro}] + 1 \tag{2.3.8}$$

Розв'язуємо цю задачу двоїстим симплекс-методом і оптимальний план на підмножині G_2^{k+1} позначимо x_2^{k+1}. І оцінку $\xi\left(G_2^{k+1}\right) = f\left(x_2^{k+1}\right)$.

Третій етап: перевірка ознаки оптимальності

Якщо план x_1^{k+1} цілий і оцінка для відповідної підмножини $\xi\left(G_1^{k+1}\right) \geq \max\xi\left(G_j^k\right)$, то план x_1^{k+1} і буде оптимальним планом. У іншому випадку на наступну ітерацію.

Особливості

- Обмеження, що вводяться грають роль відсікань;
- Якщо на деякій підмножині $G_i = \emptyset$ задача нерозв'язна, то оцінка для відповідної підмножини $\xi\left(G_i\right) = +\infty$, якщо розв'язується задача мінімізації та $\xi\left(G_i\right) = -\infty$, якщо розв'язується задача максимізації;
- Якщо всі коефіцієнти цільової функції цілі, то оцінку можна визначити за такою формулою $\xi\left(G_0\right) = [f\left(x_0\right)] + 1$;
- Якщо розв'язується задача мінімізації, то для ще одного ветвления обирається підмножина з мінімальною оцінкою, а ознакою оптимальності буде те, що план є цілим та $\xi\left(G_1^{k+1}\right) \leq \min\xi\left(G_j^k\right)$.

Приклад 2.3.1.

$$\max(2x_1 + 3x_2) \tag{2.3.9}$$
$$x_1 + 4x_2 \le 14 \tag{2.3.10}$$
$$2x_1 + 3x_3 \le 12 \tag{2.3.11}$$
$$x_1, x_1 \ge 0 \tag{2.3.12}$$
$$x_1, x_2 \in \mathbb{Z} \tag{2.3.13}$$

Приводимо до канонічного виду та позбавляємося від умови цілих чисел.

$$\max(2x_1 + 3x_2) \tag{2.3.14}$$
$$x_1 + 4x_2 + x_3 = 14 \tag{2.3.15}$$
$$2x_1 + 3x_3 + x_4 = 12 \tag{2.3.16}$$
$$x_1, x_1 \ge 0 \tag{2.3.17}$$

Далі розв'язуємо звичайним симплекс методом. Оскільки отриманий результат не цілий, переходимо до гілляння. Отримуємо таблицю:

			2	3	0	0	
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
3	x_2	$\frac{16}{5}$	0	1	$\frac{2}{5}$	$-\frac{1}{5}$	0
2	x_1	$\frac{6}{5}$	1	0	$-\frac{3}{5}$	$\frac{4}{5}$	0
	Δ	12	0	0	0	1	
$\leftarrow$	x_5	$-\frac{1}{5}$	0	0	$-\frac{2}{5}$	$-\frac{4}{5}$	1
					$\uparrow$		

Будуємо нову симплекс-таблицю:

			2	3	0	0		
	B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6
3	x_2	3	0	1	0	-1	1	0
2	x_1	$\frac{3}{2}$	1	0	0	2	$-\frac{3}{2}$	0
0	x_3	$\frac{1}{2}$	0	0	1	2	$-\frac{5}{2}$	0
$\leftarrow$	x_6	$-\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$ $\uparrow$	1

І ще одну нову симплекс-таблицю:

			2	3	0	0	
	B_x	A_0	A_1	A_2	A_3	A_4	A_5
3	x_2	2	0	1	$\frac{2}{5}$	$-\frac{1}{5}$	0
2	x_1	3	1	0	$-\frac{3}{5}$	$\frac{4}{5}$	0
	x_3	3					

Отримали розв'язок $f(x) = 12$

20.03.2014

Приклад 2.3.2. Задача комвіяжера на шість міст.
Необхідно виконати процедуру приведення. Для цього в кожному рядку шукаємо мінімальний (позначимо його h_i) і віднімаємо від усіх елементів рядку.

$$c = \begin{array}{c|c|c|c|c|c|c|c} & 0 & 1 & 2 & 3 & 4 & 5 & h_i \\ \hline 0 & x & 5 & 1 & 10 & 6 & 8 & 1 \\ \hline 1 & 2 & x & 3 & 8 & 5 & 10 & 2 \\ \hline 2 & 4 & 8 & x & 6 & 4 & 9 & 4 \\ \hline 3 & 6 & 9 & 5 & x & 3 & 7 & 3 \\ \hline 4 & 8 & 7 & 7 & 4 & x & 6 & 4 \\ \hline 5 & 10 & 4 & 9 & 3 & 2 & x & 2 \\ \hline \end{array} \tag{2.3.18}$$

Перетворуюємо стовбички, для цього в кожному шукаємо мінімальний (позначаємо його H_j) і потім віднімаємо від усіх елементів стовпчику.

$$C' = \begin{array}{cccccc} & x & 4 & 0 & 9 & 5 & 7 \\ & 0 & x & 1 & 6 & 3 & 8 \\ & 0 & 4 & x & 2 & 0 & 5 \\ & 3 & 6 & 2 & x & 0 & 4 \\ & 4 & 3 & 3 & 0 & x & 2 \\ & 8 & 2 & 7 & 1 & 0 & x \\ \hline H_j & 0 & 2 & 0 & 0 & 0 & 2 \end{array} \tag{2.3.19}$$

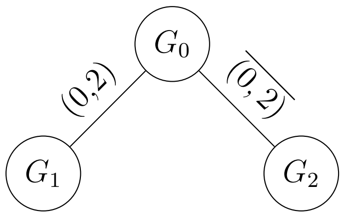
Далі шукаємо оцінки α_r - це наступний найменший після мінімального у рядку, а β_m - наступний найменший після мінімального у стовпчику.

$$C_0 = \left(\begin{array}{c|ccccccc|} & 0 & 1 & 2 & 3 & 4 & 5 & \alpha_i \\ \hline 0 & x & 2 & 0 & 9 & 5 & 5 & 2 \\ 1 & 0 & x & 1 & 6 & 3 & 6 & 1 \\ 2 & 0 & 2 & x & 2 & 0 & 3 & 0 \\ 3 & 3 & 4 & 2 & x & 0 & 2 & 2 \\ 4 & 4 & 1 & 4 & 0 & x & 0 & 0 \\ 5 & 8 & 0 & 7 & 1 & 0 & x & 0 \\ \hline \beta_j & 0 & 1 & 1 & 1 & 0 & 2 & \end{array}\right) \tag{2.3.20}$$

Отримали оцінку: $G_0(\xi) = \sum_i h_i + \sum_j H_j = 20$. Далі, обчислюємо оцінки для нульових елементів за формулою

$$c_{ij}: \quad \Theta(c_{ij}) = \alpha_i + \beta_j \tag{2.3.21}$$

І розбиваємо по тому елементу, де $\max \Theta$. Це елемент (0,2), $\Theta_{0,2} = 3$.
Розбиваємо множину на підмножини:



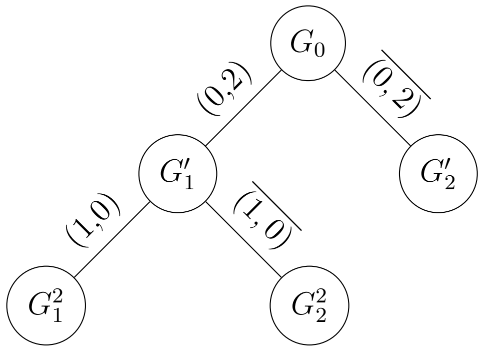
Отримали оцінку для другої підмножини $G_2(\xi) = G_0(\xi) + \Theta_{0,2} = 23$. Отже, має йти на G_1.
Викреслимо другий стовпчик та нульовий рядок (як у переході), забороняємо перехід (2,0) та отримаємо таку матрицю:

$$C_1' = \begin{array}{c|ccccc|c|c|} & 0 & 1 & 3 & 4 & 5 & h_i & \alpha_r \\ \hline 1 & 0 & x & 0 & 3 & 0 & 0 & 3 \\ 2 & x & 2 & 2 & 0 & 3 & 0 & 2 \\ 3 & 3 & 3 & x & 0 & 2 & 0 & 2 \\ 4 & 4 & 1 & 0 & x & 0 & 0 & 0 \\ 5 & 8 & 0 & 1 & 0 & x & 0 & 0 \\ \hline H_j & 0 & 0 & 0 & 0 & 0 \\ \hline \beta_m & 3 & 1 & 1 & 0 & 2 \end{array} \tag{2.3.22}$$

Знайдемо оцінку $G_1(\xi)$ за формулою:

$$G_1^2(\xi) = G_0(\xi) + \sum_i h_i + \sum_j H_j = 20 \tag{2.3.23}$$

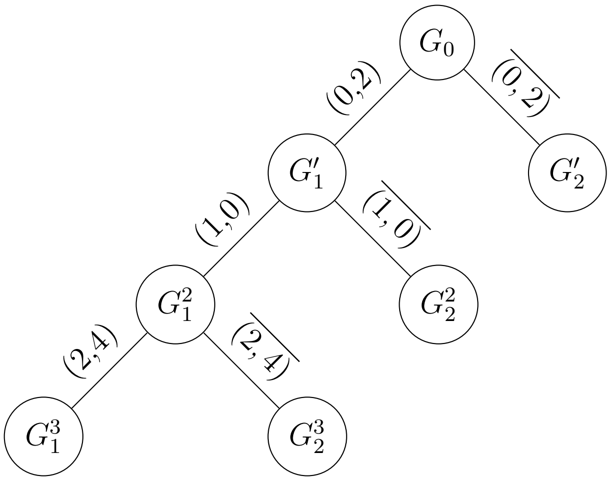
Розіб'ємо підмножину за таким самим алгоритмом. $\Theta_{1,0} = 6$:



Отримали оцінку : $G_2^2(\xi) = G_1(\xi) + \Theta_{1,0} = 26$.
Знову викреслюємо стовпчик і рядок за переходом (1,0). Також забороняємо перехід (2,1), оскільки він є транзитивним для $(2,0) \to (0,1)$.

$$C_1^2 = \begin{array}{c|cccc|c|c|} & 1 & 3 & 4 & 5 & h_i & \alpha_r \\ \hline 2 & x & 2 & 0 & 3 & 0 & 2 \\ 3 & 4 & x & 0 & 2 & 0 & 2 \\ 4 & 1 & 0 & x & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & x & 0 & 0 \\ \hline H_j & 0 & 0 & 0 & 0 \\ \hline \beta_m & 1 & 1 & 0 & 2 \end{array} \tag{2.3.24}$$

Отже, $G_1^2(\xi) = G_1(\xi) + \sum_i h_j + \sum_j H_j = 20$.
Знайдемо нове розбиття. $\Theta_{2,4} = \Theta_{3,4} = \Theta_{4,5} = 2$. Немає різниці, яку обирати, тому оберемо (2,4).



Отримали оцінку $G_2^3\left(\xi\right) = G_1^2\left(\xi\right) + \Theta_{2,4} = 22$
Знову викреслюємо рядок і стовпчик за переходом (2,4) та забороняємо (4,1) як транзитивний від $(4,2) \to (2,1)$.

$$C_1^3 = \begin{array}{c|ccc|c} & 1 & 3 & 5 & h_i \\ \hline 3 & 4 & x & 2 & 2 \\ 4 & x & 0 & 0 & 0 \\ 5 & 0 & 1 & x & 0 \end{array} \tag{2.3.25}$$

Віднімаємо двійку та отримуємо:

$$C_1^3 = \begin{array}{c|ccc|c|c|} & 1 & 3 & 5 & h_i & \alpha_i \\ \hline 3 & 2 & x & 0 & 2 & 2 \\ 4 & x & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & x & 0 & 1 \\ \hline H_J & 0 & 0 & 0 \\ \hline \beta_j & 2 & 1 & 0 \end{array} \tag{2.3.26}$$

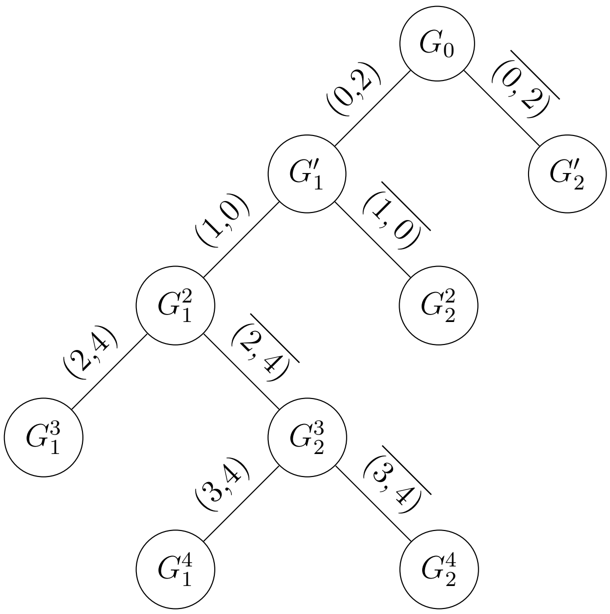
Отже, $G_1^3\left(\xi\right) = G_1^2\left(\xi\right) + \sum\limits_i h_i + \sum\limits_j H_j = 20 + 2 = 22$
Оскільки оцінки у обох варіантах однакові, оберемо G_2^3 та розберемося з тим, що відбувається, коли ми заперечуємо перехід.
Розглянемо гілку G_2^3 та побудуємо для неї матрицю. Для цього просто забороняємо перехід $(2,4)$.:

$$C_2^3 = \begin{array}{c|cccc|c} & 1 & 3 & 4 & 5 & h_i \\ \hline 2 & x & 2 & x & 3 & 2 \\ 3 & 4 & x & 0 & 2 & 0 \\ 4 & 1 & 0 & x & 0 & 0 \\ 5 & 0 & 1 & 0 & x & 0 \\ \hline \end{array} \tag{2.3.27}$$

Після віднімання:

$$C_2^3 = \begin{array}{c|cccc|c|c} & 1 & 3 & 4 & 5 & h_i & \alpha_r \\ \hline 2 & x & 0 & x & 1 & 2 & 1 \\ 3 & 4 & x & 0 & 2 & 0 & 2 \\ 4 & 1 & 0 & x & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & x & 0 & 0 \\ \hline H_j & 0 & 0 & 0 & 0 & & \\ \hline \beta_m & 1 & 0 & 0 & 1 & & \end{array} \tag{2.3.28}$$

Знаходимо перехід. $\Theta_{3,4} = \Theta_{5,4} = 2$

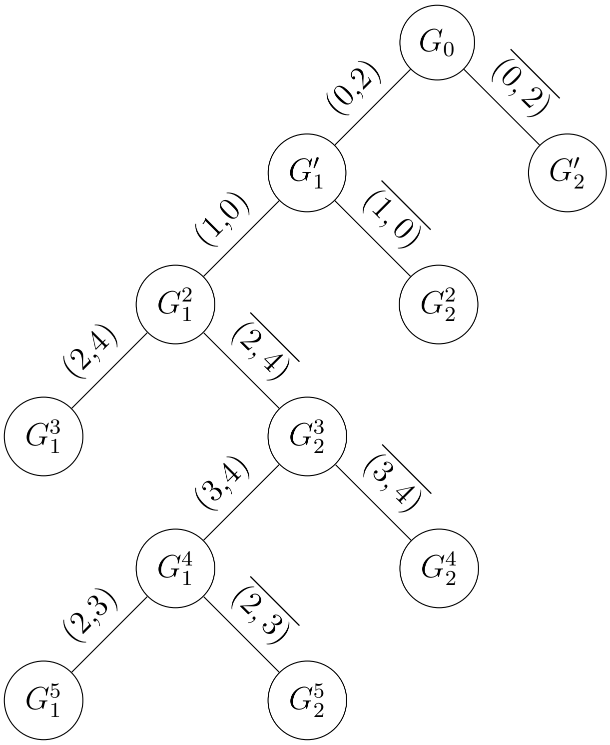


Розглянемо перехід G_1^4 та побудуємо для нього матрицю:

$$C_1^4 = \begin{array}{c|ccc|c|c} & 1 & 3 & 5 & h_i & \alpha_r \\ \hline 2 & x & 0 & 1 & 0 & 1 \\ 4 & 1 & x & 0 & 0 & 1 \\ 5 & 0 & 1 & x & 0 & 1 \\ \hline H_j & 0 & 0 & 0 & & \\ \hline \beta_m & 1 & 1 & 1 & & \end{array} \tag{2.3.29}$$

Оцінка вийшла $G_1^4(\xi) = G_2^3(\xi) + \sum_i h_i + \sum_j H_j = 22$

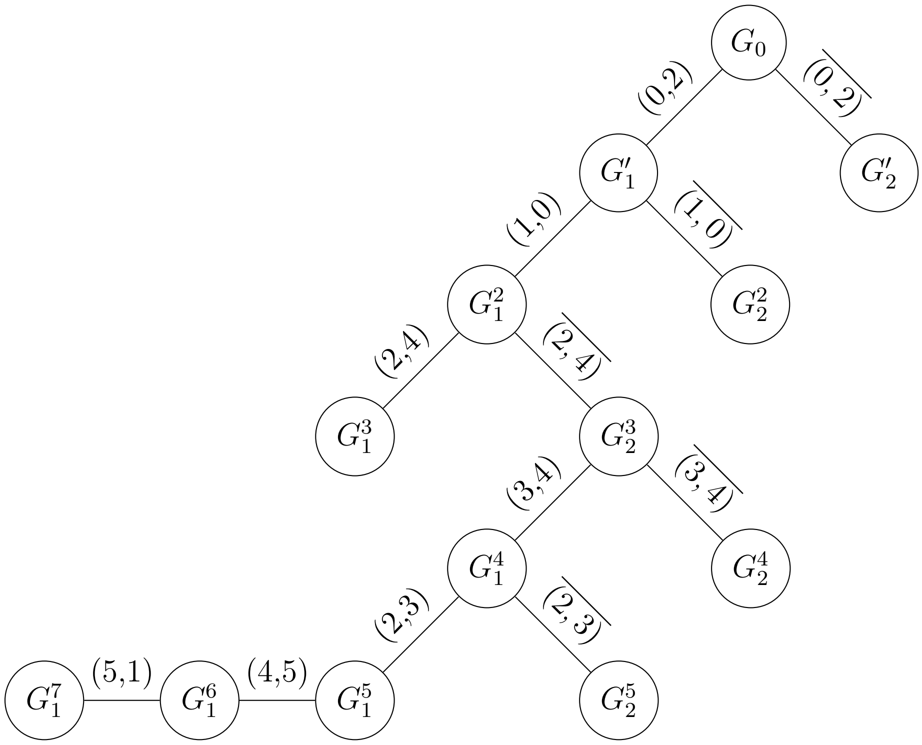
Знайдемо перехід. $\Theta_{2,3} = \Theta_{4,5} = \Theta_{5,1} = 2$



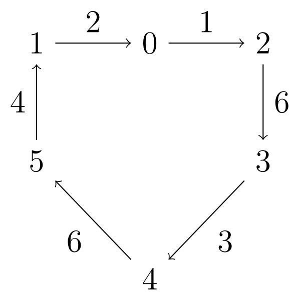
Отримали оцінку $G_2^5(\xi) = G_1^4 + \Theta_{2,3} = 24$.
Розглянемо матрицю для G_1^5

$$C_1^5 = \begin{array}{c|cc|c} & 1 & 5 & h_i \\ \hline 4 & x & 0 & 0 \\ 5 & 0 & x & 0 \\ \hline H_j & 0 & 0 & \end{array} \tag{2.3.30}$$

Отримали оцінку $G_1^5(\xi) = G_1^4(\xi) + \sum\limits_i h_i + \sum\limits_j H_j = 22$. Отже, ми знайшли оптимальний шлях, оскільки далі є лише ті переходи, які потрібні для завершення шляху:



Намалюємо графік руху:



2.4 Метод послідовного аналізу та відсів варіантів

Ідея методу: розв'язується задача дискретної оптимізації і на етапі побудови дерева варіантів відкидаються недопустимі варіанти або неефективні з точки зору цільової функції. Для цього будуються дві процедури відсіву ω_1 - відсів за обмеженнями, а ω_2 - відсів по значенням цільової функції.

$$\min f \tag{2.4.1}$$

$$g_p(x) \le g_p^*, \quad i = \overrightarrow{1,q} \tag{2.4.2}$$

$$g_p(x) \ge g_p^*, \quad i = \overrightarrow{q+1,Q} \tag{2.4.3}$$

$$x \in \mathbb{X} \tag{2.4.4}$$

Нехай x^p забезпечує оптимум при фіксованому $x_j = x_{j_{k_j}}$.

2.4.1 Процедура ω_1

$$g_p\left(x^p|x_j = x_{j_{k_j}}\right) > g_p^*, \quad i = \overrightarrow{1, q} \tag{2.4.5}$$

$$g_p\left(x^p|x_j = x_{j_{k_j}}\right) \geq g_p^*, \quad i = \overrightarrow{q+1, Q} \tag{2.4.6}$$

Використовуючи процедури відсіву (2.4.5) та (2.4.6) виконуємо процедуру ω_1 для всіх обмежень та для всіх змінних. Використовуємо процедуру ω_1 до того часу, поки відсів не закічиться. Множину варіантів, що залишилася позначимо як $\mathbb{X}(l)$.

Твердження 2.4.1. *Якщо при двох послідовних використань процедури ω_1 множина точок оптимуму не змінюється, використання процедури ω_1 закінчується*

Твердження 2.4.2. *Допустима множина при використанні процедури ω_1 не зменшується.*

При використанні процедури ω_1 до початкової множини можливі наступні випадки:

- $\mathbb{X}(l) = \emptyset$: задача нерозв'язна.
- $\mathbb{X}(l) \neq \emptyset$ в цьому випадку йдемо на процедуру ω_2

2.4.2 Процедура ω_2

Будуємо допоміжну задачу.

$$\min f(x) \tag{2.4.7}$$

$$f(x) \leq f_1^* \tag{2.4.8}$$

$$x \in \mathbb{X}(l) \tag{2.4.9}$$

Де, f_1^* - це поріг відсіву, який визначається так:

$$f_1^* = \frac{1}{2}\left(\min_{x\in\mathbb{X}(l)} f(x) + \max_{x\in\mathbb{X}(l)} f(x)\right) \tag{2.4.10}$$

Нехай x^p забезпечує оптимум функції $f(x)$ при фіксованому $x_j = x_{j_{k_j}}$. Записуємо умову оцінки

$$f\left(x^p|x_j = x_{j_{k_j}}\right) > f_1^* \tag{2.4.11}$$

Тобто $x_{j_{k_j}}$ відсівається, якщо виконується умова (2.4.11). Виконуємо процедуру ω_2 для всіх змінних. Якщо відсів стався, переходимо на процедуру ω_1. В іншому випадку необхідно ввести новий поріг:

$$f_2^* = \frac{1}{2}\left(f_1^* + \min_{x\in\mathbb{X}(l)} f(x)\right) \tag{2.4.12}$$

Виконуємо процедуру ω_2 з новим порогом і, якщо відсів відбувся, йдемо на процедуру ω_1. Якщо відсіву знову не відбулося, то збільшуємо поріг.
Якщо після використання процедури ω_1 множина варіантів, що залишилася, порожня, необхідно перейти на процедуру ω_2 і розширити множину варіантів. Для цього вводимо новий поріг:

$$f_3^* = \frac{1}{2}\left(f_2^* + \max_{x\in\mathbb{X}(l)} f(x)\right) \tag{2.4.13}$$

Використовуємо процедури ω_1 та ω_2 багатократно до того часу, поки не буде знайдемо оптимальний розв'язок. При чому, на k-тому кроку відсів відбувається за умовою:

$$f\left(x^p | x_j = x_{j_{k_j}}\right) > f_k^* \tag{2.4.14}$$

Де f_k^* - поріг відсіву

$$f_k^* = \begin{cases} \frac{1}{2}\left(f_{k-1}^* + \max\limits_{x\in\mathbb{X}(l)} f(x)\right), & \mathbb{X}(l)\uparrow \\ \frac{1}{2}\left(f_{k-1}^* + \min\limits_{x\in\mathbb{X}(l)} f(x)\right), & \mathbb{X}(l)\downarrow \\ \frac{1}{2}\left(f_{k-1}^* + f_{k-2}^*\right), & \mathbb{X}(l)\uparrow after \mathbb{X}(l)\downarrow\ or\ reverse \end{cases} \tag{2.4.15}$$

2.5 Метод послідовного відсіву варіантів цілого програмування

Запишемо модель:

$$\max \sum_{j=1}^{n} c_j x_j \tag{2.5.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = \overrightarrow{1, m} \tag{2.5.2}$$

$$x_j \ge 0, \qquad j = \overrightarrow{1, n} \tag{2.5.3}$$

$$x_j \in \mathbb{Z} \tag{2.5.4}$$

Для кожної змінної визначаємо найменше і найнижче значення. Умови (2.5.3) та (2.5.4) у загальному виді:

$$0 \le d_j^{(1)} \le x_j \le d_j^{(2)} \tag{2.5.5}$$

$d_j^{(1)} = 0$ - нижня границя, а верхня границя визначається як:

$$d_j^{(2)} = \min\left\{\left[\frac{b_i}{a_{ij}}\right]_{a_{ij}>0}\right\}, a_{ik} \ge 0 \tag{2.5.6}$$

Розділ 3

Нелінійне програмування

03.04.2014

$$\max f(x_1, \dots, x_n) \tag{3.0.1}$$

$$g_i(x_1, \dots, x_n) \geq 0, i = \overrightarrow{1, m} \tag{3.0.2}$$

І одна з цих функцій нелінійна. Тобто, або критерій, або хоча б одне з обмежень.

Теорема 3.0.1. *Якщо $f(x)$ - неперервна функція, що задана на допустимій множині $R(x)$. То функція $f(x)$ на цій множині досягає свого максимуму (мінімуму) в одній або кількох точках, що належать одній з наступних множин:*

- *Множина стаціонарних точок - S_1;*
- *Множина точок границі - S_2;*

Нехай $f(x)$ задана на множині $R(x)$, функція $f(x)$ називається **випуклою вгору функцією**, якщо

$$\forall x_1, x_1 \in R(x); \exists 0 < k < 1 : f\left(kx_1 + (1-k)x_2\right) \geq kf(x_1) + (1-k)f(x_2) \tag{3.0.3}$$

Нехай $f(x)$ задана на множині $R(x)$, функція $f(x)$ називається **випуклою вниз функцією**, якщо

$$x_1, x_1 \in R(x); 0 < k < 1 : f\left(kx_1 + (1-k)x_2\right) \leq kf(x_1) + (1-k)f(x_2) \tag{3.0.4}$$

В стаціонарній точці x_0 функція досягає свого максимуму (мінімум), якщо в околиці цієї точки функція випукла вниз (випукла вверх).
Нехай $f(x)$ неперервна функція і має не менше як другу похідну. Позначимо $f_{ij}(x_0) = \left.\dfrac{\partial^2 f(x)}{\partial x_i \partial x_j}\right|_{x=x_0}$
функція строго випукла вниз, якщо:

$$\Delta_1 = f_{11} < 0; \Delta_2 = \begin{vmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} > 0 \end{vmatrix}$$

Функція строго випукла вверх, якщо:

$$\Delta_1 = f_{11} > 0; \Delta_2 = \begin{vmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} > 0 \end{vmatrix}$$

3.1 Метод множників Лагранджа

Метод множників Лагранджа допомагає знайти мінімум або максимум функції при обмежені рівності:

$$\min f(x) \tag{3.1.1}$$

$$h_i(x) = 0, i = \overrightarrow{1, m} \tag{3.1.2}$$

Вводимо $\lambda_1, \ldots, \lambda_m$. Кількість змінних співпадає з кількістю обмежень та ці множники є **множниками Лагрнаджа**. Вони знаконевизначенні.

$$L(X, \Lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) \tag{3.1.3}$$

x_0 є розв'язком задачі (3.1.1)-(3.1.2) якщо $\exists \Lambda_0$:

$$\frac{\partial L(x_0, \Lambda_0)}{\partial x_j} = 0 \tag{3.1.4}$$

$$\frac{\partial L(x_0, \Lambda_0)}{\partial \lambda_i} = 0 \tag{3.1.5}$$

Умова (3.1.5) отримується з обмежень. Залишилося довести умову (3.1.4). Доведемо для випадку трьох змінних та двух обмежень:

$$\min f(x_1, x_2, x_3) \tag{3.1.6}$$

$$h_1(x_1, x_2, x_3) = 0 \tag{3.1.7}$$

$$h_2(x_1, x_2, x_3) = 0 \tag{3.1.8}$$

Нехай x_1 точка мінімуму. І в цій точці градієнти обмежень лінійнонезалежні. Нехай ми виразили $x_2 = u(x_1)$ та $x_3 = v(x_1)$. Знайдемо частинні похідні в цій точці

$$\begin{cases} \dfrac{\mathrm{d}\, f}{\mathrm{d}\, x_1} = \dfrac{\partial f}{\partial x_1} + \dfrac{\partial f}{\partial x_2} \cdot \dfrac{\mathrm{d}\, u}{\mathrm{d}\, x_1} + \dfrac{\partial f}{\partial x_3} \cdot \dfrac{\mathrm{d}\, v}{\mathrm{d}\, x_1} = 0 \\ \dfrac{\mathrm{d}\, h_1}{\mathrm{d}\, x_1} = \dfrac{\partial h_1}{\partial x_1} + \dfrac{\partial h_1}{\partial x_2} \cdot \dfrac{\mathrm{d}\, u}{\mathrm{d}\, x_1} + \dfrac{\partial h_1}{\partial x_3} \cdot \dfrac{\mathrm{d}\, v}{\mathrm{d}\, x_1} = 0 \\ \dfrac{\mathrm{d}\, h_1}{\mathrm{d}\, x_1} = \dfrac{\partial h_2}{\partial x_1} + \dfrac{\partial h_2}{\partial x_2} \cdot \dfrac{\mathrm{d}\, u}{\mathrm{d}\, x_1} + \dfrac{\partial h_2}{\partial x_3} \cdot \dfrac{\mathrm{d}\, v}{\mathrm{d}\, x_1} = 0 \end{cases} \tag{3.1.9}$$

Перепишемо систему (3.1.9) у такому вигляді:

$$\begin{vmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} & \dfrac{\partial f}{\partial x_3} \\ \dfrac{\partial h_1}{\partial x_1} & \dfrac{\partial h_1}{\partial x_2} & \dfrac{\partial h_1}{\partial x_3} \\ \dfrac{\partial h_2}{\partial x_1} & \dfrac{\partial h_2}{\partial x_2} & \dfrac{\partial h_2}{\partial x_3} \end{vmatrix} \cdot \begin{vmatrix} 1 \\ \dfrac{\mathrm{d}\, u}{\mathrm{d}\, x_1} \\ \dfrac{\mathrm{d}\, v}{\mathrm{d}\, x_1} \end{vmatrix} = 0 \tag{3.1.10}$$

Оскільки вектор не рівний нулю, то рядки матриці лінійнозалежні, тобто: $a, b, c \neq 0$:

$$a\nabla f(x^*) + b\nabla h_1(x^*) + c\nabla h_2(x^*) = 0 \tag{3.1.11}$$

$a \neq 0$:

$$a = 0 \Rightarrow b\nabla h_1(x^*) + c\nabla h_2(x^*) = 0 \tag{3.1.12}$$

А це протиріччя з тим, що їх градієнти лінійнонезалежні.

$$\nabla f(x^*) + \frac{b}{a}\nabla h_1(x^*) + \frac{c}{a}\nabla h_2(x^*) = 0 \tag{3.1.13}$$

Перепозначивши константи ми доведемо умову (3.1.4)

$$\nabla f(x^*) + \lambda_1\nabla h_1(x^*) + \lambda_2\nabla h_2(x^*) = 0 \tag{3.1.14}$$

$$\frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} + \lambda_1\frac{\partial h_1}{\partial x_1} + \lambda_2\frac{\partial h_2}{\partial x_1} = 0$$
$$\frac{\partial L}{\partial x_2} = \frac{\partial f}{\partial x_2} + \lambda_1\frac{\partial h_1}{\partial x_2} + \lambda_2\frac{\partial h_2}{\partial x_2} = 0$$
$$\frac{\partial L}{\partial x_3} = \frac{\partial f}{\partial x_3} + \lambda_1\frac{\partial h_1}{\partial x_3} + \lambda_2\frac{\partial h_2}{\partial x_3} = 0$$

3.2 Теорема Куно-Таккера

Нехай задана наступна задача нелійнійного програмування

$$\min f_(x) \tag{3.2.1}$$

$$g_i(x) \leq 0, i = \overrightarrow{1, m} \tag{3.2.2}$$

І нехай x^* - точка мінімуму функції $f(x)$. І в цій точці градієнти обмежень лінійнонезалежні. Тоді існує такий вектор $\exists\Lambda^* \geq 0$, що виконуються наступні умови:

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) = 0 \tag{3.2.3}$$

$$\sum_{i=1}^{m} \lambda_i^* g_i(x^*) = 0 \tag{3.2.4}$$

Умова (3.2.4) це умова доповнюючою нежорсткості.

Доведення. Позначимо через $I = \{i : g_i(x) = 0\}$. Розкладемо $f(x^* + tz)$ в ряд Тейлора, z - вектор в околиці x^*, t - малий скаляр.

$$f(x^* + tz) = f(x^*) + t\nabla f(x^*)z^{\mathrm{T}} + \Theta \tag{3.2.5}$$

Аналогічно для активних обмежень:

$$g_i(x^* + tz) = g_i(x^*) + t\bigtriangledown g_i(x^*)^{\mathrm{T}} z + \Theta \tag{3.2.6}$$

Помітно, що система

$$\begin{cases} \bigtriangledown f(x^*)^{\mathrm{T}} z < 0 \\ \bigtriangledown g(x^*)^{\mathrm{T}} z \leq 0 \end{cases} \tag{3.2.7}$$

не виконується завдяки умові на точки x^*. □

Теорема 3.2.1 (Лема Фарташа)**.** *Для будь-якої матриці A виконується одна з умов:*

$$AX < 0 \tag{3.2.8}$$

$$\Lambda^{\mathrm{T}} A = 0 \tag{3.2.9}$$

Одночасно вони виконуватися не можуть.

Продовжуємо доведення теореми. Розглянемо таку матрицю:

$$A = \begin{vmatrix} \bigtriangledown f(x^*) \\ \bigtriangledown g_i(x^*), i \in I \end{vmatrix} \tag{3.2.10}$$

Тоді, використовуючи лему Фарташа $\exists \Lambda^* = \{\lambda_0^*, \lambda_i^*\} \geq 0$ така, що

$$\lambda_0^* \bigtriangledown f(x^*) + \sum_{i \in I} \lambda_i^* \bigtriangledown g(x^*) = 0 \tag{3.2.11}$$

Нехай $\lambda_j = 0, j \notin I$ і за таких припущень можна записати:

$$\sum_{i=1}^{m} \lambda_i^* g_i(x^*) = 0 \tag{3.2.12}$$

Тоді можемо записати:

$$\lambda_0^* \bigtriangledown f(x^*) + \sum_{i=1}^{m} \lambda_i^* \bigtriangledown g_i(x^*) = 0 \tag{3.2.13}$$

$\lambda_0^* \neq 0$, оскільки, якщо $\lambda_0^* = 0$, то

$$\sum_{i=1}^{m} \lambda_i^* \bigtriangledown g_i(x^*) = 0 \tag{3.2.14}$$

А я умова протирічить тому, що градієнти лінійнонезалежні.
Поділимо систему (3.2.13) на λ_0^*

$$\bigtriangledown f(x^*) + \sum_{i=1}^{m} \frac{\lambda_i^*}{\lambda_0^*} \bigtriangledown g_i(x^*) = 0 \tag{3.2.15}$$

□

3.2.1 Використання теореми Куно-Таккера

Нехай задані наступна задача нелінійного програмування

$$\min f(x) \tag{3.2.16}$$

$$g_i(x) \leq 0, i = \overrightarrow{1, m} \tag{3.2.17}$$

$$x \geq 0 \tag{3.2.18}$$

Всі функції випукли догори функції.
Зробимо заміну $x_j = -h_j(x)$. Отримали ще таке обмеження:

$$h_j(x) \leq 0, j = \overrightarrow{1, n} \tag{3.2.19}$$

Запишемо для задачі (3.2.16),(3.2.17),(3.2.19) функцію Лагранджа

$$L(X, \Lambda, U) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) + \sum_{j=1}^{n} u_j h_j(x) \tag{3.2.20}$$

Використаємо теорему Куно-Таккера і записуємо умову оптимальності:

$$\nabla L(X, \Lambda, U) = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{n} u_j \nabla h_j(x^*) = 0 \tag{3.2.21}$$

$$\sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) = 0 \tag{3.2.22}$$

$$\sum_{j=1}^{n} u_j \nabla h_j(x^*) = 0 \tag{3.2.23}$$

$$\lambda_i^* g_i(x^*) = 0 \tag{3.2.24}$$

$$u_j x_j^* = 0 \tag{3.2.25}$$

Перепишемо умову (3.2.21) в наступному вигляді:

$$\frac{\partial L(X, \Lambda, U)}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(x^*)}{\partial x_j} - u_j = 0 \tag{3.2.26}$$

Перепишемо це ще раз, як бачимо, один аргумент зник:

$$\frac{\partial L(X, \Lambda)}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(x^*)}{\partial x_j} = u_j \geq 0 \tag{3.2.27}$$

Також, запишемо обмеження в такій формі:

$$\frac{\partial L(X, \Lambda)}{\partial \lambda_i} = g_i(x) \leq 0 \tag{3.2.28}$$

Також, з отриманного та формул (3.2.24) та (3.2.25) отримуємо:

$$\frac{\partial L(X, \Lambda)}{\partial x_j} \cdot x_j^* = 0 \tag{3.2.29}$$

$$\frac{\partial L(X, \Lambda)}{\partial \lambda_i} \cdot \lambda_i^* = 0 \tag{3.2.30}$$

x^* оптимальний розв'язок задачі (3.2.16)-(3.2.18) тоді і тільки тоді, коли $\exists \Lambda^* \geq 0$, що виконуються умови (3.2.27),(3.2.28),(3.2.29),(3.2.30), де (3.2.29),(3.2.30) - умови додаткової нежорсткості. Нехай задана наступна задача нелінійного програмування:

$$\max f(x) \tag{3.2.31}$$
$$g_i(x) \geq 0, i = \overrightarrow{1, m} \tag{3.2.32}$$
$$x \geq 0 \tag{3.2.33}$$

Всі функції є випуклими донизу.

Теорема 3.2.2. *x^* оптимальний розв'язок задачі (3.2.31)-(3.2.33) тоді і тільки тоді, коли $\exists \Lambda^* \geq 0$ такий, що виконуються:*

$$\frac{\partial L(X^*, \Lambda^*)}{\partial x_j} \leq 0 \tag{3.2.34}$$
$$\frac{\partial L(X^*, \Lambda^*)}{\partial x_j} x_j^* = 0 \tag{3.2.35}$$
$$\frac{\partial L(X^*, \Lambda^*)}{\partial \lambda_i} \geq 0 \tag{3.2.36}$$
$$\frac{\partial L(X^*, \Lambda^*)}{\partial \lambda_i} \lambda_i^* = 0 \tag{3.2.37}$$

Умови (3.2.35) та (3.2.37) - це умови додаткової нежорсткості.

Нехай задана наступна задача квадратичного програмування:

$$\max B^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} C x \tag{3.2.38}$$
$$Ax \leq A_0 \tag{3.2.39}$$
$$x \geq 0 \tag{3.2.40}$$

Функція (3.2.38) випукла донизу

Теорема 3.2.3. *x^* оптимальний розв'язок задачі (3.2.38)-(3.2.40), коли $\exists m$-вимірні $\Lambda \geq 0, W \geq 0$ і такий n-мірний $V \geq 0$,що виконуються наступні умови:*

$$B + Cx + A^{\mathrm{T}} \Lambda + V = 0 \tag{3.2.41}$$
$$A_0 - Ax - W = 0 \tag{3.2.42}$$
$$X^{\mathrm{T}} V = 0 \tag{3.2.43}$$
$$\Lambda^{\mathrm{T}} W = 0 \tag{3.2.44}$$

Умови (3.2.43),(3.2.44) - це умови додаткової нежосткості.

Доведення. Запишемо функцію Лагранджа для задачі (3.2.38)-(3.2.40):

$$L(X, \Lambda) = B^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} C x + \Lambda^{\mathrm{T}} (A_0 - Ax) \tag{3.2.45}$$

□

3.3 Метод можливих напрямів

17.04.2014

3.3.1 Метод можливих напрямів з лінійними обмеженням

$$\min f(x) \tag{3.3.1}$$

$$A \cdot X \leq B \tag{3.3.2}$$

$$H \cdot X = U \tag{3.3.3}$$

Нехай x_1 допустима точка задачі (3.3.1)-(3.3.3) для якої:

$$\begin{cases} A_1 x_1 = B_1 \\ A_2 x_1 < B_2 \end{cases} \tag{3.3.4}$$

s - можливий напрямок в точці x_1, якщо виконується умова:

$$A_1 S \leq 0 \tag{3.3.5}$$

$$HS = 0 \tag{3.3.6}$$

Вектор s можливий напрямок спуску в точці x_1, якщо виконуються умови(3.3.5),(3.3.6) і при цьому:

$$\triangledown f^{\mathrm{T}}(x_1)\, s < 0 \tag{3.3.7}$$

Для того, щоб знайти такий вектор, необхідно мінімізувати $\min \triangledown f^{\mathrm{T}}(x_1)$. Така умова називається **умовою нормування**. В результаті, отримуємо наступну задачу:

$$\triangledown f^{\mathrm{T}}(X_1)\, s < 0 \tag{3.3.8}$$

$$A_1 S \leq 0 \tag{3.3.9}$$

$$HS = 0 \tag{3.3.10}$$

$$-1 \leq S_j \leq 1 \tag{3.3.11}$$

Оптимальний розв'язок даної задачі позначимо як s_1. Якщо значення цільової функції дорівнює нулю, то поточна точка оптимальний розв'язок задачі. Тобто, точка Куно-Такера. В іншому випадку все сумно.

3.3.2 Алгоритм

Початковий етап Нехай задана задача (3.3.1)-(3.3.3). Шукаємо допустиму точку x_1 і йдемо на першу ітерацію. Нехай проведено $k-1$ ітерація, знайдена точка x_k, для якої

$$\begin{cases} A_1 x_k = B_1 \\ A_2 x_k < B_2 \end{cases}$$

Перший етап Складаємо та розв'язуємо задачу

$$\min \nabla f^{\mathrm{T}}(x_k)s \tag{3.3.12}$$
$$A_1 S \leq 0 \tag{3.3.13}$$
$$HS = 0 \tag{3.3.14}$$
$$-1 \leq s_j \leq 1 \tag{3.3.15}$$

І знайшли оптимальний розв'язок s_k. Якщо значення функції менше нуля, то s_k можливий напрямок спуску і ми йдемо на другий етап. А в іншому випадку x_k оптимальний розв'язок задачі.

Другий етап Складаємо та розв'язуємо наступну задачу

$$\min f(x_k + \lambda s_k) \tag{3.3.16}$$
$$0 \leq \lambda \leq \lambda_{\max} \tag{3.3.17}$$

Де права частина у обмеження, це максимальне значення λ при якому точка $x_k + \lambda s_k$ залишається допустимою. Далі покладемо $x_{k+1} = x_k + \lambda_k s_k$ і йдемо на наступну ітерацію.

Приклад 3.3.1.

$$\min 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \tag{3.3.18}$$
$$x_1 + x_2 \leq 2 \tag{3.3.19}$$
$$x_1 + 5x_2 \leq 5 \tag{3.3.20}$$
$$-x_1 \leq 0 \tag{3.3.21}$$
$$-x_2 \leq 0 \tag{3.3.22}$$
$$\nabla f(x) = \begin{pmatrix} 4x_1 - 2x_2 + 4 \\ 4x_2 - 2x_1 - 6 \end{pmatrix} \tag{3.3.23}$$
$$\min -4s_1 - 6s_2 \tag{3.3.24}$$
$$-s_1 \leq 0 \tag{3.3.25}$$
$$-s_2 \leq 0 \tag{3.3.26}$$
$$-1 \leq s_1, s_2 \leq 1 \tag{3.3.27}$$

$s_1 = s_2 = 1; F = -10$
Ідемо на другий етап:

$$\min 2\left(\lambda s_1\right)^2 + 2\left(\lambda s_2\right)^2 - 2\lambda^2 s_1 s_2 - 4\lambda s_1 - 6\lambda s_2 \tag{3.3.28}$$

Спростимо до нормального виду

$$\min 2\lambda^2 - 10\lambda \tag{3.3.29}$$
$$4\lambda - 5 = 0 \tag{3.3.30}$$
$$\lambda = 2.5 \tag{3.3.31}$$

Також є обмеження:

$$\begin{cases} 2\lambda \leq 2 \\ 6\lambda \leq 5 \end{cases} \Rightarrow \lambda = \frac{5}{6} \tag{3.3.32}$$

Отже, $\lambda = \dfrac{5}{6}$

3.3.3 Метод можливих напрямів для задач з нелінійними обмеженнями

$$\min f(x) \tag{3.3.33}$$
$$g_i(x) \leq 0, i = \overrightarrow{1, m} \tag{3.3.34}$$

Нехай x допустима точка задачі (3.3.33)-(3.3.34). Через множину $I = \{i : g_i(x) = 0\}$ - множину індексів активних обмежень. Нехай, якщо $f(x)$ і $g_i(x), i \in I$ - диференційовані, а $g_i(x), i \notin I$ - неперервні. Вектор s можливий напрям спуску в точці x, якщо

$$\nabla f^{\mathrm{T}}(x)s < 0 \tag{3.3.35}$$
$$\nabla g_i{}^{\mathrm{T}}(x)s < 0, i \in I \tag{3.3.36}$$

Для того, щоб знайти цей вектор, потрібно $\min\left\{\max\left\{\nabla f^{\mathrm{T}}(x)s, \nabla g_i{}^{\mathrm{T}}(x)s\right\}\right\}$. Позначимо $\max\left\{\nabla f^{\mathrm{T}}(x)s, \nabla g_i{}^{\mathrm{T}}(x)s\right\} = z$. В результаті отримуємо наступну задачу

$$\min z \tag{3.3.37}$$
$$\nabla f^{\mathrm{T}}(x)s - z \leq 0 \tag{3.3.38}$$
$$\nabla g_i{}^{\mathrm{T}}(x)s - z \leq 0, i \in I \tag{3.3.39}$$
$$-1 \leq s_j \leq 1 \tag{3.3.40}$$

Позначимо оптимальний розв'язок данної задачі як s_0, z_0. Якщо $z_0 = 0$, то поточна точка є оптимальним розв'язком.

Алгоритм

Початковий етап Знаходимо допустиму точку і йдемо на першу ітерацію. Нехай проведена $k-1$ ітерація я знайдена точка x_k. Позначимо через $I = \{i : g_i(x_k) = 0\}$ - множину індексів активних обмежень.

Перший етап Складаємо та розв'язуємо наступну задачу:

$$\min z \tag{3.3.41}$$
$$\nabla f^{\mathrm{T}}(x)s - z \leq 0 \tag{3.3.42}$$
$$\nabla g_i{}^{\mathrm{T}}(x)s - z \leq 0, i \in I \tag{3.3.43}$$
$$-1 \leq s_j \leq 1 \tag{3.3.44}$$

Оптимальний розв'язок позначили s_k, z_k. Якщо $z_k = 0$, то x_k оптимальний розв'язок задачі. Інакше йдемо на другий етап

Другий етап Складаємо та розв'язуємо наступну задачу

$$\min f(x_k + \lambda s_k) \tag{3.3.45}$$
$$0 \leq \lambda \leq \lambda_{\max} \tag{3.3.46}$$

Де права частина у обмеження, це максимальне значення λ при якому точка $x_k + \lambda s_k$ залишається допустимою. Далі покладемо $x_{k+1} = x_k + \lambda_k s_k$ і йдемо на наступну ітерацію.

Приклад 3.3.2.

$$\min{(x_1 - 2x_2)}^2 + 5\,{(x_1 - 8)}^2 \tag{3.3.47}$$

$$x_1^2 - 5x_2 \leq 10 \tag{3.3.48}$$

$$x_1 + 2x_2 \leq 8 \tag{3.3.49}$$

$$\nabla f(x) = \begin{pmatrix} 2(x_1 - 2x_2) + 10\,(x_1 - 8) \\ -4\,(x\_2x_2) \end{pmatrix} \tag{3.3.50}$$

$$\nabla f(x') = \begin{pmatrix} -80 \\ 0 \end{pmatrix} \tag{3.3.51}$$

Складемо задачу:

$$\min z \tag{3.3.52}$$

$$-80s_1 - z \leq 0 \tag{3.3.53}$$

$$-1 \leq s_1 \leq 1 \tag{3.3.54}$$

$s_1 = 1, z = -80$

$$\min\left\{{(\lambda s_1)}^2 + 5\,{(\lambda s_1 - 8)}^2\right\} \tag{3.3.55}$$

$$2\lambda + 10\,(\lambda - 8) = 0 \tag{3.3.56}$$

$$\lambda = \frac{20}{3} \tag{3.3.57}$$

Розглянемо обмеження:

$$\begin{cases} \lambda^2 \leq 10 \\ \lambda \leq 8 \end{cases} \Rightarrow \lambda = \sqrt{10} \tag{3.3.58}$$

Запишемо нову точку:

$$x = \begin{pmatrix} \sqrt{10} \\ 0 \end{pmatrix} \tag{3.3.59}$$

Розділ 4

Геометричне програмування

4.1 Прості задачі геометричного програмування

Прості задачі - задачі без обмежень.

$$g(t) = \sum_{i=1}^{n} U_i(t) = \sum_{i=1}^{n} c_i t_1^{a_{i1}} \cdot \ldots \cdot t_m^{a_{im}} \tag{4.1.1}$$

Для розв'язку таких задач використаємо нерівності о середньому геометричному та середньому арифметичному.

$$\frac{1}{n}\sum_{i=1}^{n} U_i \leq \prod_{i=1}^{n} U_i^{1/n} \tag{4.1.2}$$

$$\frac{1}{n}\sum_{i=1}^{n} U_i\delta_i \geq \prod_{i=1}^{n} U_i^{\delta_i} \tag{4.1.3}$$

$$\delta_i \geq 0 \tag{4.1.4}$$

$$\sum_{i=1}^{n} \delta_i = 1 \tag{4.1.5}$$

$$U_i\delta_i = u_i \tag{4.1.6}$$

$$u_i(t) = c_i t_1^{a_{i1}} \cdot \ldots \cdot t_m^{a_{im}} \tag{4.1.7}$$

$$\sum_{i=1}^{n} u_i(t) \geq \prod_{i=1}^{n} \left(\frac{u_i}{\delta_i}\right)^{\delta_i} = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right) t_1^{D_1} \ldots t_m^{D_m} \tag{4.1.8}$$

$$D_j = \sum_{i=1}^{n} a_{ij}\delta_i \tag{4.1.9}$$

$$\tag{4.1.10}$$

Підберемо вагові коефіцієнти таким чином, щоб:

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0 \tag{4.1.11}$$

І підставимо:

$$g(t) \le V(\delta) \tag{4.1.12}$$

δ_i - це змінні, для яких виконуються умови нормалізації, ортогоналізації та невід'ємності, тобто умови (4.1.4),(4.1.5),(4.1.11)

Теорема 4.1.1. *Нехай t^* точка мінімума функції (4.1.1), тоді існує δ^* для якої виконуються умови (4.1.4),(4.1.5),(4.1.11) і в цій точці значення двоїстої функції співпадає зі значенням прямої функції $V(\delta^*) = g(t^*)$*

Доведення.

$$\frac{\partial g(t^*)}{\partial t_j} = 0 \tag{4.1.13}$$

$$\frac{\partial g(t^*)}{\partial t_j} \cdot t_j^* = \sum_{i=1}^{n} \frac{\partial u_i(t^*)}{\partial t_j} \cdot t_j^* = \sum_{i=1}^{n} u_i(t^*) a_{ij} = 0 \tag{4.1.14}$$

Так як $g(t^*) \neq 0$:

$$\sum_{i=1}^{n} \frac{u_(t^*)}{g(t^*)} a_{ij} = 0 \tag{4.1.15}$$

$$\frac{u_(t^*)}{g(t^*)} = \delta_i^* \tag{4.1.16}$$

$$\sum_{i=1}^{n} \delta_i^* a_{ij} = 0 \tag{4.1.17}$$

Отже, умова ортогональності виконується
Перевіримо умову нормалізації:

$$\sum_{i=1}^{n} \delta_i^* = \sum_{i=1}^{n} \frac{u_(t^*)}{g(t^*)} = 1 \tag{4.1.18}$$

$$g(t^*) = \prod_{i=1}^{n} g(t^*)^{\delta_i^*} = \prod_{i=1}^{n} \left(\frac{u_i(t^*)}{\delta_i^*} \right)^{\delta_i^*} = v(\delta^*) \tag{4.1.19}$$

□

4.2 Загальний випадок з обмеженнями

24.04.2014

$$g_o(t) = \sum_{i \in I_0} c_i t_1^{a_{i1}} \cdot \ldots \cdot t_m^{a_{im}} \tag{4.2.1}$$

$$g_k(t) = \sum_{i \in I_k} c_i t_1^{a_{i1}} \cdot \ldots \cdot t_m^{a_{im}} \le 1, k = \overrightarrow{1,p} \tag{4.2.2}$$

$$I_0 = \{1, n_0\} \tag{4.2.3}$$

$$I_k = \{m_k, \ldots, n_k\}, k = \overrightarrow{1,p} \tag{4.2.4}$$

До задачі (4.2.1)-(4.2.2) запишемо двоїсту задачу:

$$\max v(\delta) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \prod_{k=1}^{p} \Lambda_k^{\Lambda_k(\delta)}(\delta) \tag{4.2.5}$$

$$\sum_{i \in I_0} \delta_i = 1 \tag{4.2.6}$$

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0 \tag{4.2.7}$$

$$\delta_i \geq 0 \tag{4.2.8}$$

4.2.1 Зв'язок прямої та двоїстої задачі

1. Кожному позіному відповідає двоїста змінна і навпаки: $t_m^{a_{im}} \leftrightarrow \delta_i$
2. Для двоїстої змінної, що відповідає позіномам цільової функції прямої задачі виконується умова нормалізації.
3. Кожному обмеженню відповідає: $g_k(t) \leq 1 \leftrightarrow \Lambda_k(\delta) = \sum\limits_{i \in I_k} \delta_i$
4. Кількість умов ортогональності визначається кількістю змінних прямої задачі.

Задача називається **сумісною**, якщо існує хоча б одна точка, що задовольняє її обмеження.

Теорема 4.2.1. *Нехай пряма задача сумісна. Тобто існує точка t' для якої виконується $g_k(t') \leq 1, k = \overrightarrow{1,p}$.*

1. *Тоді двоїста задача також сумісна, і існує точка δ^*, для якої виконуються умови (4.2.6)-(4.2.8). І в цій точці $v(\delta^*) = \max v(\delta)$*
2. $g(t^*) = v(\delta^*)$
3. *Якщо δ^* оптимальний розв'язок двоїстої задачі, то існує оптимальний розв'язок прямої задачі з компонентами, що визначаються із наступної системи.*

$$c_i t_1^{*,a_{i1}} \cdot t_m^{*,a_{im}} = \begin{cases} \delta_i^* v(\delta^*), i \in I_0 \\ \dfrac{\delta_i^*}{\Lambda_k(\delta^*)}, i \in I_k \end{cases} \tag{4.2.9}$$

4.3 Алгоритм

Перший крок Записуємо двоїсту задачу

Другий крок Визначаємо складність задачі як

$$d = n - m - 1 \tag{4.3.1}$$

Де, n - кількість позіномів, а m - кількість змінних у прямій задачі.
Якщо $d = 0$, то розв'язок системи єдине і йдемо на сьомий крок.
В іншому випадку, йдемо на третій.

Третій крок Записуємо загальний розв'язок системи (15)-(16). Записуємо систему (23).

Четвертий шаг Записуємо максимізуюче рівняння, записуємо систему (39). Кількість таких рівнянь співпадає зі ступеню складності задачі. Де $\lambda_{k,j}$ визначається з системи (29), $\lambda_{k,r}$ із системи (26). Численне значення для k_j визначається з системи (33).

П'ятий крок Розв'язуємо систему (39) і із цієї системи визначаємо r_j^*.

Шостий крок Підставляємо r_j^* в систему (23) і знаходимо оптимальний розв'язок двоїстої задачі.

Сьомий крок Розв'язуємо систему та з цієї системи визначаємо оптимальний розв'язок двоїстої задачі δ_i^*.

Восьмий крок Підставляємо δ_i^* в цільову функцію двоїстої задачі і визначаємо значення цільової функції $v(\delta^*) = g(t^*)$

Дев'ятий крок Знаходимо оптимальний розв'язок прямої задачі за системою (4.3.17)

Приклад 4.3.1.

$$\min g_i(t) = 40t_1t_2 + 20t_2t_3 \tag{4.3.2}$$

$$g_i(t) = \frac{1}{5}t_1^{-1}t_2^{-\frac{1}{2}} + \frac{3}{5}t_2^{-1}t_3^{-\frac{2}{3}} \leq 1 \tag{4.3.3}$$

Кількість позиномів 4. Запишемо двоїсту задачу:

$$\max v(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1}\left(\frac{20}{\delta_2}\right)^{\delta_2}\left(\frac{\frac{1}{5}}{\delta_3}\right)^{\delta_3}\left(\frac{\frac{3}{5}}{\delta_4}\right)^{\delta_4} \cdot (\delta_3+\delta_4)^{\delta_3+\delta_4} \tag{4.3.4}$$

$$\delta_1 + \delta_2 = 1 \tag{4.3.5}$$

$$\delta_1 - \delta_3 = 0 \tag{4.3.6}$$

$$\delta_1 + \delta_2 - \frac{1}{2}\delta_3 - \delta_4 = 0 \tag{4.3.7}$$

$$\delta_2 - \delta_4 = 0 \tag{4.3.8}$$

Порахуємо складність: $d = 4 - 3 - 1 = 0$

$$\delta_1^* = \delta_2^* = \delta_3^* = \frac{1}{2} \tag{4.3.9}$$

$$\delta_4^* = \frac{3}{4} \tag{4.3.10}$$

Знайдемо значення цільової функції $v(\delta^*) = 40 = g(t^*)$
Складемо систему для знаходження розв'язку прямої задачі:

$$40t_1t_2 = \delta_1^* \cdot v(\delta^*) = 20 \tag{4.3.11}$$

$$20t_2t_3 = \delta_2^* \cdot v(\delta^*) = 20 \tag{4.3.12}$$

$$\frac{1}{5}t_1^{-1}t_2^{-\frac{1}{2}} = \frac{\delta_3^*}{\delta_3^* + \delta_4^*} \tag{4.3.13}$$

$$\frac{3}{5}t_2^{-1}t_3^{-\frac{2}{3}} = \frac{\delta_4^*}{\delta_3^* + \delta_4^*} \tag{4.3.14}$$

$$\tag{4.3.15}$$

Отримали розв'язок: $t_1^* = \frac{1}{2}, t_2^* = t_3^* = 1$

$$\max v(\delta) = \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \prod_{k=1}^{p} \Lambda_k^{\Lambda_k(\delta)}(\delta) \tag{4.3.16}$$

$$\sum_{i \in I_0} \delta_i = 1 \tag{4.3.17}$$

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0 \tag{4.3.18}$$

$$\delta_i \geq 0 \tag{4.3.19}$$

Якщо $d > 0$, то загальний розв'язок системи (4.3.17)-(4.3.19) записується у виді:

$$\delta_i(r) = b_i^{(0)} \sum_{j=1}^{d} r_j b_j^{(j)} \tag{4.3.20}$$

Де $b_i^{(0)}$ вектор нормалізації записується так:

$$b^{(0)} = \begin{cases} \sum\limits_{i \in I_0} = 1 \\ \sum\limits_{i=1}^{n} a_{ij}\delta_i = 0 \end{cases} \tag{4.3.21}$$

І вектори нев'язки:

$$b^{(j)} = \begin{cases} \sum\limits_{i \in I_0} \delta_i' = 0 \\ \sum\limits_{i=1}^{n} a_{ij}\delta_i' = 0 \end{cases} \tag{4.3.22}$$

$$\Lambda_k(r) = \sum_{i \in I_k} \delta_i(r) \tag{4.3.23}$$

Підставимо (4.3.20) в (4.3.23)

$$\Lambda_k(r) = \sum_{i \in I_k} b_i^{(0)} + \sum_{j=1}^{d} r_j \sum_{i \in I_k} b_i^{(j)} \tag{4.3.24}$$

Введемо такі позначення:

$$\sum_{i \in I_k} b_i^{(0)} = \Lambda_k^{(0)} \tag{4.3.25}$$

$$\sum_{i \in I_k} b_i^{(j)} = \Lambda_k^{(j)} \tag{4.3.26}$$

Підставимо (4.3.26),(4.3.25) в (4.3.24)

$$\Lambda_k = \Lambda_k^{(0)} + \sum_{j=1}^{d} r_j \Lambda_k^{(j)} \tag{4.3.27}$$

Підставимо (4.3.27) у цільову функцію:

$$v(r) = \prod_{i=1}^{n} c_i^{\,b_i^{(0)} + \sum\limits_{j=1}^{d} r_j b_j^{(j)}} \prod_{i=1}^{n} \delta_i^{-\delta_i(r)}(r) \prod_{k=1}^{p} \Lambda_k^{\Lambda_k(r)}(r) \tag{4.3.28}$$

$$\prod_{i=1}^{n} c_i^{b_i^{(0)}} = k_0 \tag{4.3.29}$$

$$\prod_{i=1}^{n} c_i^{b_i^{(j)}} = k_j \tag{4.3.30}$$

Підставимо ці позначення у вираз (4.3.28)

$$v(r) = k_0 \prod_{j=1}^{d} k_j^{r_j} \prod_{i=1}^{n} \delta_i^{-\delta_i(r)}(r) \prod_{k=1}^{p} \Lambda_k^{\Lambda_k(r)}(r) \tag{4.3.31}$$

Так як функція $\ln v(r)$ вогнута та монотонно зростає, то функції $\ln v(r)$ та $v(r)$ мають однакову множину масимізуючих точок. Переходимо до логарифму

$$\ln v(r) = \ln k_0 + \sum_{j=1}^{d} r_j \ln k_j - \sum_{i=1}^{n} \delta_i(r) \ln \delta_i(r) \sum_{k=1}^{p} \Lambda_k(r) \ln \Lambda_k(r) \tag{4.3.32}$$

$$\frac{\partial \ln v(r)}{\partial r_j} = \ln k_j - \sum_{j=1}^{d} b_i^{(j)} \ln \delta_i(r) - \sum_{i=1}^{n} \frac{\partial \delta_i(r)}{\partial r_j} + \sum_{k=1}^{p} \Lambda_k^{(j)} \ln \Lambda_k(r) + \sum_{k=1}^{p} \frac{\partial \Lambda_k(r)}{\partial r_j} = 0 \tag{4.3.33}$$

Так як:

$$\sum_{i=1}^{n} \frac{\partial \delta_i(r)}{\partial r_j} = \sum_{i \in I_k} \sum_{k=1}^{p} \frac{\partial \delta_i(r)}{\partial r_j} = \sum_{k=1}^{p} \frac{\partial \Lambda_k(r)}{\partial r_j} \tag{4.3.34}$$

$$\ln k_j = \sum_{j=1}^{d} b_j^{(j)} \ln \delta_i(r) - \sum_{k=1}^{p} \Lambda_k^{(j)} \ln \Lambda_k(r) \tag{4.3.35}$$

Отже, отримали:

$$k_j = \prod_{i=1}^{n} \delta_i^{b_i}(r) \prod_{k=1}^{p} \Lambda_k^{-\Lambda_k^{(j)}}(r) \tag{4.3.36}$$

Розділ 5

Динамічне програмування

5.1 Динамічне програмування

15.05.2014

$$\max \sum_{j=1}^{n} f_j(x_j) \tag{5.1.1}$$

$$\sum_{j=1}^{n} c_{ij} x_j \leq b \tag{5.1.2}$$

$$x_j \geq 0 \in \mathbb{Z} \tag{5.1.3}$$

Позначимо через

$$z^* = \max \left\{ f_n(x_n) + \max \left\{ \sum_{j=1}^{n-1} f_i(x_i) \right\} \right\} \tag{5.1.4}$$

Позначимо

$$\max \sum_{j=1}^{n-1} f_j(x_j) \tag{5.1.5}$$

За умови

$$\sum_{j=1}^{n-1} a_j x_j \leq b - a_n x_n \tag{5.1.6}$$

Як $\Lambda_{n-1}(\xi)$

$$\Lambda_{n-1}(\xi) = \max \left\{ f_{n-1}(x_{n-1}) + \max \left\{ \sum_{j=1}^{n-2} f_i(x_i) \right\} \right\} \tag{5.1.7}$$

Позначимо

$$\max \sum_{j=1}^{n-2} f_j(x_j) \tag{5.1.8}$$

За умови

$$\sum_{j=1}^{n-2} a_j x_j \leq \xi - a_{n-1} x_{n-1} \tag{5.1.9}$$

Як $\Lambda_{n-2}(\xi_1), \xi_1 = \xi - a_{n-1}x_{n-1}$ Підставляємо позначення в систему (5.1.7) та отримуємо

$$\Lambda_{n-1}(\xi) = \max\{f_{n-1}(x_{n-1}) + \Lambda_{n-2}(\xi - a_{n-1}b_{n-1})\} \tag{5.1.10}$$

5.1.1 Основне рекуренте співвідношення Белмана

$$\Lambda_k(\xi) = \max_{x_k \le [A_k]}\{f_k(x_k) + \Lambda_{k-1}(\xi - a_k x_k)\} \tag{5.1.11}$$

5.2 Основні властивості задачі

1. Задача має допускати інтерпретацію n-крокового процесу прийняття рішень;
2. На кожному кроці має бути задана деяка множина параметр ξ і ця множина від кроку до кроку не змінюється;
3. Розв'язок, знайдений на k-тому кроці не має впливати на розв'язки попередніх кроків;
4. Оптимальна стратегія k-того кроку залежить лише від поточного стану ξ.

Даний підхід дозволяє замінити n-вимірну оптимізацію n-кроковим процесом прийняття рішень.
Існує дві схеми розв'язку.

- Розв'язок в прямому напрямі. Дана схема використовується, якщо заданий кінцевий стан системи;
- Розв'язок в зворотньому напрямі. Ця схема використовується, якщо задані початкові умови процеси.

В задачі (5.1.1)-(5.1.3) заданий кінцевий стан системи і цю задачу розв'язуємо у прямому напрямі.

Перший крок

$$\Lambda_1(\xi) = \max_{x \le [\xi/a]} f_1(x_1), \xi = \overline{0, b} \tag{5.2.1}$$

Для кожного ξ визначаємо Λ_1. На кожному кроці будується таблиця

ξ	x_1	Λ
0		
$\vdots$		
b		

Деякий крок Використовуючи співвідношення (5.1.11) обчислюємо $\Lambda_2, \ldots$

Останній крок

$$\xi = b: \ \Lambda_n(b) = \max\left\{f_n(x_n) + \Lambda_{n-1}\left(b - a_n x_n\right)\right\} \tag{5.2.2}$$

З цієї системи визначаємо x_n^0. А далі покладемо $\xi^0 = b - a_n x_n^0$. І так далі визначаємо $x_{n-1}^0, \ldots, x_1^0$.

5.3 Задача про розподіл трудових ресурсів

Постановка задачі:
Планується робота фірми на n періодів.
m_j - оптимальна кількість робітників в j-тий місяць.
x_j - фактична кількість робітників в j-тий місяць.
Відхилення від оптимальної кількості приводить до витрати за такою функцією:

$$g_j(x_j - m_j) \tag{5.3.1}$$

Можна наймати або звільняти співробітників на початку кожного періоду. Затрати на це записуються як

$$f_j(x_j - x_{j-1}) \tag{5.3.2}$$

Необхідно визначити таку кількість співробітників, при якій мінімізуються сумарні затрати.

$$\min\left(\sum_{j=1}^{n} f_j(x_j - x_{j-1}) + \sum_{j=1}^{n} g_j\left(x_j - m_j\right)\right) \tag{5.3.3}$$

5.3.1 Перший випадок:початкова умова

Задана початкова умова x_0. Отже, розв'язуємо задачу у зворотньому порядку

Перший крок Визначаємо Λ_n:

$$\Lambda_n\left(\xi\right) = \min\left(f_n\left(x_n - \xi\right) + g_n\left(x_n - m_n\right)\right), \xi = x_{n-1}, \xi = \overline{0, m_{max}} \tag{5.3.4}$$

Наступні кроки

$$\Lambda_k\left(\xi\right) = \min\left(f_k\left(x_k - \xi\right) + g_k\left(x_k - m_k\right) + \Lambda_{k+1}\left(x_k\right)\right) \xi = x_{k-1}, \xi = \overline{0, m_{max}} \tag{5.3.5}$$

За цим співвідношенням визначаємо усі $\Lambda_{n-1}, \ldots, \Lambda_2$.

Останній крок На цьому кроці $\xi = x_0$

$$\Lambda_1\left(x_0\right) = \min\left\{f_1\left(x_1 - x_0\right) + g_1\left(x_1 - m_1\right) + \Lambda_2\left(x_1\right)\right\} \tag{5.3.6}$$

За формулою (5.3.6) визначаємо x_1^o. А далі визначаємо $x^o = x_1^o$ і із таблиці попередніх кроків визначаємо $x_2^o, \ldots, x_n^o$

5.3.2 Другий випадок: кінцева умова

Розв'язуємо задачу у прямому напрямі

Перший крок

$$\Lambda_1(\xi) = \min\{f_1(x_1 - x_0) + g_1 x_1 - m_1 + f_2(\xi - x_1) + g_2(\xi - m_2)\}$$

$$\xi = x_2, \quad \xi = \overline{0, m_{\max}} \quad (5.3.7)$$

Наступні кроки

$$\Lambda_k(\xi) = \min\{f_{k+1}(\xi - x_k) + g_{k+1}\xi - m_{k+1} + \Lambda_{k-1}(x_k)\}$$

$$\xi = x_{k+1}, \quad \xi = \overline{0, m_{\max}} \quad (5.3.8)$$

За цим співвідношенням визначаємо $\Lambda_2, \ldots, \Lambda_{n-1}$

Останній крок На останньому кроці $\xi = m_{n+1}$

$$\Lambda_n(m_{n+1}) = \min\{f_{n+1}(m_{n+1} - x_n) + \Lambda_{n-1}(x_n)\} \quad (5.3.9)$$

З цього визначаємо x_n^o, покладемо $\xi^o = \xi_n^o$ і визначаємо далі $x_{n-1}^o, \ldots, x_1^o$

5.4 Задача керування запасами

Основні компоненти:

- Система постачання. Може бути централізованою та децентралізованою;
- Попит на предмети постачання". Стаціонарний або не станціонарний. Детермінований або випадковий;
- Способи поповнення запасів: моментальна доставка, затримка на фіксований інтервал часу, затримка на випадковий інтервал часу;
- Функції затрат: затрати на поставку, затрати на зберігання і витрати внаслідок можливого дефіциту;
- Обмеження: на максимальну вартість поставки, на максимальний об'єм запасів і на ймовірність дефіциту

Постановка задачі: планується робота системи постачання на n періодів
x_j - запас, що створюється в j-тий період або замовлення j-того періоду
y_j - залишок від запасів $j-1$-шого періоду
d_j - попит в j-тому періоді
$c_j(x_j)$ - затрати на поставку
$s_j(x_j + y_j - d_j)$ - затрати на зберігання

Необхідно визначити запас, що створюється в кожному періоді або замовлення кожного періоду, за якого мінімізуються сумарні витрати.

$$\min \sum_{j=1}^{n} c_j\left(x_j\right) + \sum_{j=1}^{n} s_j\left(x_j + y_j - d_j\right) = \sum_{j=1}^{n} f_j\left(x_j, y_{j+1}\right) \tag{5.4.1}$$

$$x_j + y_j - d_j = y_{j+1}, \quad j = \overrightarrow{1,n} \tag{5.4.2}$$

Майже у всіх випадках буде $y_1 = y_{n+1} = 0$, а в інших випадках вони будуть відомі. Отже, задачу можна розʼязувати у обох напрямках. Розвʼяжемо задачу у прямому напрямі:

$$\Lambda_k\left(\xi\right) = \min\left\{f_k\left(x_k, \xi\right) + \Lambda_{k-1}\left(\xi + d_k - x_k\right)\right\}, \quad \xi = y_{k+1} \tag{5.4.3}$$

За цим співвідношенням визначаємо $\Lambda_1, \dots, \Lambda_{n-1}$.
На останньому кроці $\xi = y_{n+1} = 0$

$$\Lambda_n\left(y_{n+1}\right) = \min\left\{f_n\left(x_n, y_{n+1} + \Lambda_{n-1}\left(y_{n+1} + d_n - x_n\right)\right)\right\} \tag{5.4.4}$$

За цим співвідношенням визначаємо x_n^o і покладемо $\xi^o = y_{n+1} + d_n - x_n^o$ і визначаємо за ним $x_{n-1}^o, \dots, x_1^o$.
Зробимо наступні припущення

- $c_j(x_j)$ - вогнута
- $s_j(y_{j+1})$ - лінійна функція

В силу вогнутості мінімум функції $c_j(x_j)$ досягається в одній з крайніх точно допустимої множини розвʼязків, яка визначається обмеженням (5.4.2). Так як таких обмежень n, то оптимальний розвʼязок містить не більше ніж n невідомих, що не дорівнюють нулю.
Тобто

$$x_j^o \cdot y_j^o \Rightarrow \begin{cases} x_j^o = 0; y_j^o > 0 \\ x_j^O, y_j^0 = 0 \end{cases} \tag{5.4.5}$$

З цього видно, що замовлення не буде, якщо в початку j-того періоду мався деякий запас. Тобто, $x_j^0 \in \{0, d_j, d_j + d_{j+1}, \dots\}$. Тобто замовлення дорівнює сумі попиту за деяке число періодів.
Отже, у співвідношенні (5.4.3) оптимум досягається або за $x_k = 0$, або за $x_k = \xi + d_k$.
Звідси отримуємо

$$\Lambda_k\left(\xi\right) = \min \begin{matrix} f_k\left(0, \xi\right) + \Lambda_{k-1}\left(\xi + d_k\right) \\ f_k\left(\xi + d_k, \xi\right) + \Lambda_{k-1}\left(0\right) \end{matrix} \tag{5.4.6}$$

$$\Lambda_{k-1}\left(\xi + d_k\right) = \min \begin{matrix} f_{k-1}\left(0, \xi + d_k\right) + \Lambda_{k-2}\left(\xi + d_k + d_{k-1}\right) \\ f_k\left(\xi + d_k + d_{k-1}, \xi + d_k\right) + \Lambda_{k-2}\left(0\right) \end{matrix} \tag{5.4.7}$$

Отже, основне рекуренте співвідношення матиме такий вигляд:

$$\Lambda_k\left(\xi\right) = \min_{i \le k} g_k(i) \tag{5.4.8}$$

Де $g_k(i)$ - витрати за k періодів, за умови, що останній раз замовлення було у i-тому періоді.

$$g_k(i) = \Lambda_{i-1}(\xi) + f_i\left(\xi + \sum_{u=i}^{k} d_u, \xi + \sum_{u=i+1}^{k} d_u\right) + \sum_{j=i+1}^{k} f_j\left(0; \xi + \sum_{u=j+1}^{k} d_u\right) \quad (5.4.9)$$

І останнє, щоб все взагалі стало просто:

$$c_j(x_j) = A_j\delta_j(x_j), \quad \delta_j(x_j) = \begin{cases} 1, x_j > 0 \\ 0, x_j = 0 \end{cases} \quad (5.4.10)$$

В результаті отримуємо наступне рекуренте співвідношення

$$\Lambda_k(0) = \min_{i \le k} g_k(i) \quad (5.4.11)$$

$$g_k(i) = \Lambda_{i-1}(0) + A_i + \sum_{j=1}^{k=1} s_j \sum_{u=j+1}^{k} d_u \quad (5.4.12)$$

5.5 Математичні задачі динамічного програмування

22.05.2014

$$\max \sum_{j=1}^{n} f_j(x_j) \quad (5.5.1)$$

$$\sum_{j=1}^{n} a_{1i}x_j \le b_1 \quad (5.5.2)$$

$$\sum_{j=1}^{n} a_{2i}x_j \le b_2 \quad (5.5.3)$$

$$x_j \ge 0 \in \mathbb{Z} \quad (5.5.4)$$

Так як два обмеження, необхідно ввести два параметра $\xi_1 = \overline{0, b_1}; \xi_2 = \overline{0, b_2}$. Задані початкові умови, отже розв'язуємо у прямому напрямку. Запишемо рекурентну формулу:

$$\Lambda_k(\xi_1, \xi_2) = \max_{x_k \le \min\left\{\left[\frac{\xi_1}{a_{1k}}\right]; \left[\frac{\xi_2}{a_{2k}}\right]\right\}} \{f_k(x_k)\Lambda_{k-1}(\xi_1 - a_{1k}x_k; \xi_2 - a_{2k}x_k)\} \quad (5.5.5)$$

Використовуючи це співвідношення визначаємо $\Lambda_1, \ldots, \Lambda_{n-1}$. На останньому кроці $\xi_1 = b_1, \xi_2 = b_2$

$$\Lambda_n(b_1, b_2) = \max_{x_k \le \min\left\{\left[\frac{b_1}{a_{1k}}\right]; \left[\frac{b_2}{a_{2k}}\right]\right\}} \{f_n(x_n)\Lambda_{n-1}(b_1 - a_{1n}x_n; b_2 - a_{2n}x_n)\} \quad (5.5.6)$$

З цього співвідношення знаходимо x_n^o. Далі, покладемо $\xi_1^o = b_1 - a_{1n}x_n^o; \xi_2^o = b_2 - a_{2n}x_n^o$ і за таблицями останніх кроків визначаємо інші оптимальні.
Кількість операцій експоненціально зростає відносно кількості значень. Таке явище Белман назвав **прокляття розмірності**.

5.5.1 Метод множників Лагранджа для зниження розмірності задачі

Нехай задана наступна задача динамічного програмування

$$\max \sum_{j=1}^{n} f_j(x_j) \tag{5.5.7}$$

$$\sum_{j=1}^{n} a_{1i}x_j = b_1 \tag{5.5.8}$$

$$\sum_{j=1}^{n} a_{2i}x_j = b_2 \tag{5.5.9}$$

$$x_j \geq 0 \in \mathbb{Z} \tag{5.5.10}$$

Введемо друге обмеження (обмеження (5.5.9)) в цільову функцію:

$$\max \left(\sum_{j=1}^{n} f_j(x_j) - \lambda \sum_{j=1}^{n} a_{2j}x_j \right) \tag{5.5.11}$$

$$\sum_{j=1}^{n} a_{1j}x_j = b_1 \tag{5.5.12}$$

Так як у нас лише одне обмеження, то потрібно ввести лише один параметр $\xi = \overline{0, b_1}$. Запишемо рекурентне співвідношення.

$$\Lambda_k(\xi) = \max \left\{ f_k(x_k) - \lambda a_{2k}x_k + \Lambda_{k-1}\left(\xi - a_{1k}x_k\right) \right\} \tag{5.5.13}$$

Використовуючи рекурентне співвідношення визначаємо $x_j^o(\lambda)$. Так як λ - невідома, задаємо початкове значення, визначаємо x_j^o і перевіряємо, чи виконується обмеження (5.5.9). Якщо воно виконується, то даний розв'язок і буде шуканим. В іншому випадку λ необхідно скоригувати, при цьому, якщо $\sum\limits_{j=1}^{n} a_{2j}x_j(\lambda) > b_2$, то λ потрібно збільшити. В іншому випадку - зменшуємо.

5.6 Прямі методи одновимірного пошуку

Запишемо математичну модель:

$$\min f(x) \tag{5.6.1}$$

$$a \leq x \leq b \tag{5.6.2}$$

$f(x)$ - нелінійна функція, $[a, b]$ - інтервал невизначеності.
Нехай $[\lambda, \mu] \subset [a, b]$. Якщо $f(\lambda) > f(\mu)$, то новий інтервал невизначеності $[\lambda, b]$.
В іншому випадку, новий інтервал невизначеності $[a, \mu]$.

5.6.1 Метод Фібоначчі

Початковий етап Числа Фібоначчі

$$F_0 = F_1 = 1 \quad F_{n+1} = F_n + F_{n-1}$$

Нехай $[a_1, b_1]$ - початковий інтервал невизначеності. Задаємося константою розрізнюваності $\varepsilon > 0$ та кінцевою довжиною інтервалу невизначеності $l > 0$. Визначаємо кількість ітерацій

$$n : \quad F_n > \frac{b_1 - a_1}{l} \tag{5.6.3}$$

Визначаємо μ_1, λ_1 за такими формулами:

$$\lambda_1 = a_1 + \frac{F_{n-2}}{F_n}(b_1 - a_1) \tag{5.6.4}$$

$$\mu_1 = a_1 + \frac{F_{n-1}}{F_n}(b_1 - a_1) \tag{5.6.5}$$

Визначили $f(\lambda_1)$ та $f(\mu_1)$ і йдемо на першу ітерацію.

k-та ітерація та перший крок Порівнюємо $f(\lambda_k)$ та $f(\mu_k)$. Якщо $f(\lambda_k) > f(\mu_k)$ йдемо на другий крок, інакше на третій.

Другий крок $a_{k+1} = \lambda_{k_1}; b_{k+1} = b_{k_1}; \lambda_{k+1} = \mu_k$, де

$$\mu_{k+1} = a_{k+1} + \frac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1}) \tag{5.6.6}$$

Якщо $k = n - 2$ на четвертий крок, інакше порахували значення функцій і йдемо на наступну ітерацію.

Третій крок $a_{k+1} = a_k; b_{k+1} = \mu_k; \mu_{k+1} = \lambda_k$

$$\lambda_{k+1} = a_{k+1} + \frac{F_{n-k-2}}{F_{n-k}}(b_{k+1} - a_{k+1}) \tag{5.6.7}$$

Якщо $k = n - 2$ на четвертий крок, інакше порахували значення функцій і йдемо на наступну ітерацію.

Четвертий крок Покладемо $\lambda_n = \lambda_{n-1}, \mu_n = \lambda_n + \varepsilon$. Якщо $f(\lambda_n) > f(\mu_n)$, то $a_n = \lambda_n, b_n = b_{n-1}$. В іншому випадку $a_n = a_{n-1}; b_n = \mu_n$.
Оптимальний розв'язок лежить в інтервалі $[a_n, b_n]$.

5.6.2 Дихотомічний пошук

Початковий етап Нехай $[a_1, b_1]$ - інтервал невизначеності, Задаємося константою розрізнюваності $\varepsilon > 0$ та кінцевою довжиною інтервалу невизначеності $l > 0$.
Визначаємо λ_1, μ_1

$$\mu_1 = \frac{b_1 + a_1}{2} + \varepsilon \qquad \lambda_1 = \frac{b_1 + a_1}{2} - \varepsilon \tag{5.6.8}$$

Йдемо на першу ітерацію.

k-та ітерація і перший крок Якщо $b_k - a_k \leq l$, то оптимальний розв'язок лежить в інтервалі $[a_k, b_k]$. І іншому випадку порівнюємо $f(\lambda_k)$ та $f(\mu_k)$. Якщо $f(\lambda_k) > f(\mu_k)$, то другий крок. Інакше - третій.

Другий крок $a_{k+1} = \lambda_k; b_{k+1} = b_k; \lambda_{k+1} = \mu_k$

$$\mu_{k+1} = \frac{b_{k+1} + a_{k+1}}{2} + \varepsilon \tag{5.6.9}$$

Визначаємо $f(\lambda_{k+1}); f(\mu_{k+1})$ і йдемо на наступну ітерацію.

Третій крок $a_{k+1} = a_k; b_{k+1} = \mu_k; \mu_{k+1} = \lambda_k$

$$\lambda_{k+1} = \frac{b_{k+1} + a_{k+1}}{2} - \varepsilon \tag{5.6.10}$$

Визначаємо $f(\lambda_{k+1}); f(\mu_{k+1})$ і йдемо на наступну ітерацію.

5.6.3 Метод золотого перетину

Початковий етап Нехай $[a_1, b_1]$ - інтервал невизначеності, Задаємося кінцевою довжиною інтервалу невизначеності $l > 0$. $\alpha = 0.618$.
Визначаємо λ_1, μ_1

$$\mu_1 = a_1 + \alpha\,(b_1 - a_1) \qquad \lambda_1 = a_1 + cb1 - \alpha\,(b_1 - a_1) \tag{5.6.11}$$

Йдемо на першу ітерацію.

k-та ітерація і перший крок Якщо $b_k - a_k \leq l$, то оптимальний розв'язок лежить в інтервалі $[a_k, b_k]$. І іншому випадку порівнюємо $f(\lambda_k)$ та $f(\mu_k)$. Якщо $f(\lambda_k) > f(\mu_k)$, то другий крок. Інакше - третій.

Другий крок $a_{k+1} = \lambda_k; b_{k+1} = b_k; \lambda_{k+1} = \mu_k$

$$\mu_{k+1} = a_{k+1} + \alpha\,(b_{k+1} - a_{k+1}) \tag{5.6.12}$$

Визначаємо $f(\lambda_{k+1}); f(\mu_{k+1})$ і йдемо на наступну ітерацію.

Третій крок $a_{k+1} = a_k; b_{k+1} = \mu_k; \mu_{k+1} = \lambda_k$

$$\lambda_{k+1} = a_{k+1} + cb1 - \alpha\left(b_{k+1} - a_{k+1}\right) \tag{5.6.13}$$

Визначаємо $f(\lambda_{k+1}); f(\mu_{k+1})$ і йдемо на наступну ітерацію.

Приклад 5.6.1 (Задача керування запасами)**.** Плануємо роботу на 4 місяця. $d_1 = 90, d_2 = 125, d_3 = 140, d_4 = 100, S_i = 2.$
$A_i = 300, y_1 = y_5 = 0$. Запишемо математичну модель

$$\min\left(300 \cdot \sum_{j=1}^{4} \delta_j(x_j)\right) + 2\sum_{j=1}^{4} y_{j+1} \tag{5.6.14}$$

$$x_j + y_j - d_j = y_{j+1} \tag{5.6.15}$$

Де δ_j - індикатор замовлення.
Вигідно розв'язувати її у прямому напрямі) Тому саме так і розв'язуємо. Перший місяць

$$\Lambda_1(0) = 300 \tag{5.6.16}$$

Другий місяць:

$$\Lambda_2(0) = \begin{pmatrix} g_2(1) = A + 2 \cdot d_2 = 550\,(*) \\ g_2(2) = A + \Lambda_1(0) = 600 \end{pmatrix} \tag{5.6.17}$$

Третій місяць:

$$\Lambda_3(0) = \begin{pmatrix} g_3(1) = A + 2\left(d_2 + d_3\right) + 2 \cdot d_3 = 1110 \\ g_3(2) = A + \Lambda_1(0) + 2 \cdot d_3 = 880 \\ g_3(3) = A + \Lambda_2(0) = 850\,(*) \end{pmatrix} \tag{5.6.18}$$

Розділ 6

Практика

6.1 Практика

$\boxed{10.04.2014}$

Завдання 6.1.1.

$$\max x_1 + x_2 \tag{6.1.1}$$
$$2x_1 + 11x_2 \le 38 \tag{6.1.2}$$
$$x_1 + x_2 \le 7 \tag{6.1.3}$$
$$4x_1 - 5x_2 \le 5 \tag{6.1.4}$$
$$x_1, x_2 \ge 0 \tag{6.1.5}$$
$$x_1, x_2 \in \mathbb{Z} \tag{6.1.6}$$
$$d_1^0 = d_2^0 = 0 \tag{6.1.7}$$

Знадемо множину допустимих значень для x_1 та x_2 з обмежень. Вона має бути найширшою.

$$x_1^0 = \{0, 7\} \tag{6.1.8}$$
$$x_2^0 = \{0, 3\} \tag{6.1.9}$$

Відсіюємо недопустимі варіанти

$$\omega_1(x_1): \quad 2x_1 > 38 - \min\{11x_2\} \tag{6.1.10}$$
$$x_1 > 7 - \min\{x_2\} \tag{6.1.11}$$
$$4x_1 > 5 + \max\{5x_2\} \tag{6.1.12}$$
$$x_1^0 = \{0, 5\} \tag{6.1.13}$$
$$\omega_1(x_2): \quad 11x_1 > 38 - \min\{2x_1\} \tag{6.1.14}$$
$$x_2 > 7 - \min\{x_1\} \tag{6.1.15}$$
$$5x_2 > 5 - \min\{4x_2\} \tag{6.1.16}$$
$$\tag{6.1.17}$$

Переходимо до відсіву значень за значеннями цільової функції

$$\omega_2: \quad f_1^* = \frac{1}{2}(\min f(x) + \max f(x)) \tag{6.1.18}$$

$$f_1^* = \frac{1}{2} \cdot 8 = 4 \tag{6.1.19}$$

$$x_1 < f_1^* - \max\{x_2\} \tag{6.1.20}$$

$$x_1 < 1 \Rightarrow x_1^0 = \{1, 5\} \tag{6.1.21}$$

$$x_2 < f_1^* - \max\{x_1\} \tag{6.1.22}$$

$$x_2 < -1 \tag{6.1.23}$$

Отже, за x_2 немає відсіву і йдемо на процедуру ω_1. Зауважимо те, що змінилася лише нижня границя x_1, тому має сенс робити цю процедуру лише для x_2.

$$\omega_1(x_2): \quad x_2 > \frac{36}{11} \tag{6.1.24}$$

$$x_2 > 6 \tag{6.1.25}$$

$$x_2 < -\frac{1}{4} \tag{6.1.26}$$

Відсіву не відбулося. Далі робити відсів немає сенсу.
Переходимо на процедуру ω_2 і вводимо новий поріг відсіву:

$$\omega_2: \quad f_2^* = \frac{1}{2}(f_1^* + \max f(x)) = \frac{1}{2}(4 + 8) = 6 \tag{6.1.27}$$

$$x_1 < 6 - \max\{x_2\} \tag{6.1.28}$$

$$x_1 < 3 \Rightarrow x_1^0 = \{3, 5\} \tag{6.1.29}$$

$$x_2 < 6 - \max\{x_1\} \tag{6.1.30}$$

$$x_2 < 1 \Rightarrow x_2^0 = \{1, 3\} \tag{6.1.31}$$

Аналогічно попереднім роздумам переходимо до процедури ω_1, враховуючи те, що змінилися обидві множини і лише з нижніми границями. Тобто, перевіряємо лише там, де мінімум.

$$\omega_1(x_1): \quad x_1 > \frac{27}{2} \tag{6.1.32}$$

$$x_1 > 6 \tag{6.1.33}$$

$$\omega_1(x_2): \quad x_2 > \frac{32}{11} \Rightarrow x_2^0 = \{1, 2\} \tag{6.1.34}$$

$$x_2 > 4 \tag{6.1.35}$$

$$x_2 < \frac{7}{5} \Rightarrow x_2^0 = 2 \tag{6.1.36}$$

Оскільки множини змінилося, то ми повторюємо процедур ω_1 лише для x_1.

$$\omega_1(x_1): \quad x_1 > 8 \tag{6.1.37}$$

$$x_1 > 5 \tag{6.1.38}$$

$$x_1 > \frac{15}{4} \Rightarrow x_1^0 = 3 \tag{6.1.39}$$

6.2 Практика

24.04.2014

Завдання 6.2.1.

$$\max x_1 + 2x_2 - x_1^2 - x_2^2 \tag{6.2.1}$$
$$x_1 + 2x_2 \leq 16 \tag{6.2.2}$$
$$x_1 + x_2 = 8 \tag{6.2.3}$$
$$x_1, x_2 \leq 0 \tag{6.2.4}$$

Приведемо до канонічного виду:

$$-x_1 - 2x_1 + 16 \geq 0 \tag{6.2.5}$$
$$x_1 + x_2 - 8 = 0 \tag{6.2.6}$$
$$\lambda_1 > 0 \tag{6.2.7}$$

Введемо функцію Лапласа

$$L(x, \Lambda) = f(x) + \lambda_1(-x_1 - 2x_2 + 16) + \lambda_2(x_1 + x_2 - 8) \tag{6.2.8}$$

Перевіримо на знаковизначеність:

$$\Delta_1 = f_{11} = \frac{\partial^2 f}{\partial {x_1}^2} = -2 < 0 \tag{6.2.9}$$
$$\Delta_2 = \begin{vmatrix} -2 & 0 \\ 0 & -2 \end{vmatrix} = 4 > 0 \tag{6.2.10}$$

Отже, вона від'ємновизначена.

$$\frac{\partial L}{\partial x_1} = 1 - 2x_1 - -\lambda_1 + \lambda_2 \leq 0 \tag{6.2.11}$$
$$\frac{\partial L}{\partial x_2} = 2 - 2x_2 - 2\lambda_1 + \lambda_2 \leq 0 \tag{6.2.12}$$
$$\frac{\partial L}{\partial x_1} \cdot x_1 = 0 \tag{6.2.13}$$
$$\frac{\partial L}{\partial x_2} \cdot x_2 = 0 \tag{6.2.14}$$
$$\frac{\partial L}{\partial \lambda_1} = -x_1 - 2x + 2 + 16 \geq 0 \tag{6.2.15}$$
$$\frac{\partial L}{\partial \lambda_1} \cdot \lambda_1 = 0 \tag{6.2.16}$$
$$\frac{\partial L}{\partial \lambda_2} = x_1 + x_2 - 8 = 0 \tag{6.2.17}$$

Отримаємо таку систему:

$$2x_1 + \lambda_1 - \lambda_2 - v_1 = 1 \tag{6.2.18}$$
$$2x_2 + 2\lambda_1 - \lambda_2 - x_2 = 2 \tag{6.2.19}$$
$$x_1 + 2x_2 + w_1 = 16 \tag{6.2.20}$$
$$x_1 + x_2 = 8 \tag{6.2.21}$$
$$x_1 v_1 = 0 \tag{6.2.22}$$
$$x_2 v_2 = 0 \tag{6.2.23}$$
$$\lambda_1 w_1 = 0 \tag{6.2.24}$$
$$\tag{6.2.25}$$

Оскільки на λ_2 немає обмежень у знаку, запишемо $\lambda_2 = \lambda_3 - \lambda_4, \lambda_3, \lambda_4 \geq 0$.
Перепишемо систему:

$$2x_1 + \lambda_1 - \lambda_3 + \lambda_4 - v_1 = 1 \tag{6.2.26}$$
$$2x_2 + 2\lambda_1 - \lambda_3 + \lambda_4 - x_2 = 2 \tag{6.2.27}$$
$$x_1 + 2x_2 + w_1 = 16 \tag{6.2.28}$$
$$x_1 + x_2 = 8 \tag{6.2.29}$$
$$x_1 v_1 = 0 \tag{6.2.30}$$
$$x_2 v_2 = 0 \tag{6.2.31}$$
$$\lambda_1 w_1 = 0 \tag{6.2.32}$$
$$\tag{6.2.33}$$

А далі це розв'язуємо через симплекс-таблицю.
Або через метод перебору.
Подумаємо з:

$$\frac{\partial L}{\partial x_1} = 1 - 2x_1 - -\lambda_1 + \lambda_2 \leq 0 \tag{6.2.34}$$
$$\frac{\partial L}{\partial x_2} = 2 - 2x_2 - 2\lambda_1 + \lambda_2 \leq 0 \tag{6.2.35}$$
$$\frac{\partial L}{\partial x_1} \cdot x_1 = 0 \tag{6.2.36}$$
$$\frac{\partial L}{\partial x_2} \cdot x_2 = 0 \tag{6.2.37}$$
$$\frac{\partial L}{\partial \lambda_1} = -x_1 - 2x + 2 + 16 \geq 0 \tag{6.2.38}$$
$$\frac{\partial L}{\partial \lambda_1} \cdot \lambda_1 = 0 \tag{6.2.39}$$
$$\frac{\partial L}{\partial \lambda_2} = x_1 + x_2 - 8 = 0 \tag{6.2.40}$$

З цього можна отримати систему:

$$x_1, x_2, \lambda_1 \neq 0 \tag{6.2.41}$$
$$-2x_1 - \lambda_1 + \lambda_2 = -1 \tag{6.2.42}$$
$$2x_2 + 2\lambda_1 - \lambda_2 = 2 \tag{6.2.43}$$
$$x_1 + 2x_2 = 16 \tag{6.2.44}$$
$$x_1 + x_2 = 8 \tag{6.2.45}$$

Отримали: $\lambda_1 = -15$. Отже, нехай $\lambda_1 = 0, x_1, x_2 \neq 0$

$$-2x_1 + \lambda_2 = -1 \tag{6.2.46}$$
$$2x_2 - \lambda_2 = 2 \tag{6.2.47}$$
$$x_1 + 2x_2 = 16 \tag{6.2.48}$$
$$x_1 + x_2 = 8 \tag{6.2.49}$$

Отримали з цієї системи: $x_2 = \dfrac{17}{4}, x_1 = \dfrac{15}{2}$

6.3 Практика

22.05.2014

Завдання 6.3.1.

$$\min{(x_1 - 6)}^2 + {(x_2 - 2)}^2 \tag{6.3.1}$$
$$-x_1 + 2x_2 \leq 4 \tag{6.3.2}$$
$$3x_1 + 2x_2 \leq 12 \tag{6.3.3}$$
$$-x_1 \leq 0 \tag{6.3.4}$$
$$-x_2 \leq 0 \tag{6.3.5}$$

Спочатку визначаємо допустиму точку.

Завдання 6.3.2.

$$\min 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \tag{6.3.6}$$
$$x_1 + 5x_2 \leq 5 \tag{6.3.7}$$
$$2x_1^2 - x_2 \leq 0 \tag{6.3.8}$$
$$-x_1 \leq 0 \tag{6.3.9}$$
$$-x_2 \leq 0 \tag{6.3.10}$$

Починаємо з точки (0,0)

$$\bigtriangledown f = (4x_1 - 2x_2 - 4; 4x_2 - 2x_1 - 6) \tag{6.3.11}$$

$$\bigtriangledown f(x') = (-4, -6) \tag{6.3.12}$$
$$\bigtriangledown g_2(x) = (4x_1, -1) \tag{6.3.13}$$
$$\bigtriangledown (x') = (0, -1) \tag{6.3.14}$$

Записуємо нову задачу:

$$\min z \tag{6.3.15}$$
$$-4s_1 - 6s_2 - z \leq 0 \tag{6.3.16}$$
$$-s_2 - z \leq 0 \tag{6.3.17}$$
$$-s_1 - z \leq 0 \tag{6.3.18}$$
$$|s_{1,2}| \leq 1 \tag{6.3.19}$$

Отримали розв'язок $s_1 = s_2 = 1, z = -1$.

$$\min f(x' + \lambda s') = 2\lambda^2 s_1^2 + 2\lambda^2 s_2^2 - 2\lambda^2 s_1 s_2 - 4\lambda - 6\lambda = 2\lambda^2 - 10\lambda \tag{6.3.20}$$

А з обмеженнями вийшло таке:

$$\lambda \leq \frac{5}{6} \tag{6.3.21}$$

З другого:

$$2\lambda^2 - \lambda \leq 0 \Rightarrow \lambda \leq \frac{1}{2} \tag{6.3.22}$$

Нова точка $\lambda = \left(\frac{1}{2}, \frac{1}{2}\right)$

6.3.1 Геометричне програмування

Завдання 6.3.3.

$$\min g_0(t) = 40t_1 t_2^{-1/2} t_3^{-1} + 20t_1 t_3 + 20t_1 t_2 t_3 \tag{6.3.23}$$
$$g_1(t) = \frac{1}{3} t_1^{-2} t_2^{-2} \frac{4}{3} t_1^{1/2} t_3^{-1} \leq 1 \tag{6.3.24}$$

Запишемо двоїсту задачу:

$$\max \left(\frac{40}{\delta_1}\right)^{\delta_1} \left(\frac{20}{\delta_2}\right)^{\delta_2} \left(\frac{20}{\delta_3}\right)^{\delta_3} \left(\frac{1}{3\delta_4}\right)^{\delta_4} \left(\frac{4}{3\delta_5}\right)^{\delta_5} (\delta_4 + \delta_4)^{\delta_4 + \delta_5} \tag{6.3.25}$$
$$\delta_1 + \delta_2 + \delta_3 = 1 \tag{6.3.26}$$
$$-\delta_1 + \delta_2 + \delta_3 - 2\delta_4 = 0 \tag{6.3.27}$$
$$-\frac{1}{2}\delta_1 + \delta_3 - 2\delta_4 + \frac{1}{2}\delta_5 = 0 \tag{6.3.28}$$
$$-\delta_1 + \delta_2 + \delta_3 - \delta_5 = 0 \tag{6.3.29}$$

Обчислимо степінь складності задачі:

$$d = n - m - 1 = 5 - 3 - 1 = 1 \tag{6.3.30}$$

Записуємо всі обмеження у табличку:

A_0	δ_1	δ_2	δ_3	δ_4	δ_5
1	1	1	1	0	0
0	-1	1	1	-2	0
0	$-\frac{1}{2}$	0	1	-2	$\frac{1}{2}$
0	-1	1	1	0	-1

Приводимо методом Гауса чотири вектора до базисного виду і отримуємо таку таблицю:

A_0	δ_1	δ_2	δ_3	δ_4	δ_5
$\frac{1}{2}$	1	0	0	1	0
$\frac{1}{4}$	0	1	0	$-\frac{1}{2}$	0
0	0	0	0	-2	1
$\frac{1}{4}$	0	0	1	$-\frac{1}{2}$	0

$$b^o = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{4} \\ 0 \\ 0 \end{pmatrix} \tag{6.3.31}$$

$$\delta_1^* = \frac{1}{2} - r \tag{6.3.32}$$

$$\delta_2^* = \frac{1}{4} + \frac{1}{2}r \tag{6.3.33}$$

$$\delta_3^* = 2r \tag{6.3.34}$$

$$\delta_4^* = r \tag{6.3.35}$$

$$\delta_5^* = \frac{1}{4} + \frac{1}{2}r \tag{6.3.36}$$

Складаємо максимізуюче рівняння:

$$K_j = \prod S_i(r)^{b_i^{(j)}} \prod \Lambda_k(2) \tag{6.3.37}$$

$$K_j = 40^{-1} \cdot 20^{\frac{1}{2}} \cdot 20^{\frac{1}{2}} \cdot \left(\frac{1}{3}\right)^1 \cdot \left(\frac{4}{3}\right)^2 = \frac{8}{27} \tag{6.3.38}$$

$$\frac{8}{27} = \left(\frac{1}{2} - r\right)^{-1} \left(\frac{1}{4} + \frac{1}{2}r\right)^{\frac{1}{2}} \left(\frac{1}{4} + \frac{1}{2}r\right)^{\frac{1}{2}} (r)^1 (2r)^2 (3r) \tag{6.3.39}$$

Отримали дивний результат: $r = 0.3$. Тепер обчислимо δ_i:

$$\delta_1 = 0.2 \quad \delta_2 = 0.4$$
$$\delta_3 = 0.4 \quad \delta_4 = 0.3$$
$$\delta_5 = 0.6$$

Складемо систему рівнянь:

$$40t_1^{-1}t_2^{-1/2}t_3^{-1} = 0.2 \cdot 100 = 20 \tag{6.3.40}$$

$$20t_1t_3 = 40 \tag{6.3.41}$$

$$20t_1t_2t_3 = 40 \tag{6.3.42}$$

$$\tfrac{1}{3}t_1^{-2}t_2^{-2} = \tfrac{1}{3} \tag{6.3.43}$$

$$\tfrac{4}{3}t_2^{1/2}t_3^{-1} = \tfrac{2}{3} \tag{6.3.44}$$

На екзамені можна довести до цього місця і кинути.

Розділ 7

Додатки

7.1 Література

7.2 Теоретичні питання

7.3 Практичні задачі

Список задач з посиланнями на розв'язки:

1. ~~Багатокритеріальні задачі~~
2. Задача комівояжера (Приклад 2.3.2)
3. Задача на метод гілок та меж (Приклад 7.3.1).
4. Задачі на метод послідовного аналізу та відсіву варіантів (Приклад 6.1.1)
5. Задачі на метод Куна-Такера
6. Задачі на метод можливих напрямів з лінійними обмеженнями
7. Задачі на метод можливих напрямів з нелінійними обмеженнями
8. Задача розподілу ресурсів
9. Задача керування ресурсами
10. ~~Геометричні задачі~~

Приклад 7.3.1. Роз'язати методом гілок та меж таку задачу:

$$\max x_1 + x_2 \tag{7.3.1}$$
$$6x_1 + 5x_2 \leq 20 \tag{7.3.2}$$
$$2x_1 + 3x_2 \leq 10 \tag{7.3.3}$$
$$x_1, x_2 \geq 0 \tag{7.3.4}$$
$$x_1, x_2 \in \mathbb{Z} \tag{7.3.5}$$

Приведемо до канонічного виду:

$$\max x_1 + x_2 \tag{7.3.6}$$
$$6x_1 + 5x_2 + x_3 = 20 \tag{7.3.7}$$
$$2x_1 + 3x_2 + x_4 = 10 \tag{7.3.8}$$
$$x_1, x_2 \geq 0 \tag{7.3.9}$$

Спочатку знаходимо оптимальний розв'язок задачі лінійного програмування за допомогою симплекс методу.

			1	1		
c	B_x	A_0	A_1	A_2	A_3	A_4
1	x_1	$\frac{5}{4}$	1	0	$\frac{3}{8}$	$\frac{5}{8}$
1	x_2	$\frac{5}{2}$	0	1	$-\frac{1}{4}$	$\frac{3}{4}$
	Δ	$\frac{15}{4}$	0	0	$\frac{1}{8}$	$\frac{11}{8}$

(7.3.10)

Обчислюємо оцінку на множині

$$G_0: \quad \xi\left(G_0\right) = \left[f(x^0)\right] = \left[\frac{15}{4}\right] = 3 \tag{7.3.11}$$

Оскільки x_1 не цілий, то розбиваємо по цій компоненті

$$G_0: \quad G_0 = G_1^1 \cup G_2^1, \quad G_1^1 = G_0 \cap \{x_1 \leq 1\} \tag{7.3.12}$$

Запишемо обмеження $x_1 \leq 1$ у вигляді $x_1 + x_5 = 1$
Дописуємо рядок x_5 і стовпець $A_5 = (0, 0, 1)$ до симплекс таблиці. Віднімаємо від x_5 рядок x_1 і отримуємо рядок x_5' в таблиці. Далі розв'язуємо задачу двоїстим симплекс методом:

			1	1			
c	B_x	A_0	A_1	A_2	A_3	A_4	A_5
1	x_1	$\frac{5}{4}$	1	0	$\frac{3}{8}$	$\frac{5}{8}$	0
1	x_2	$\frac{5}{2}$	0	1	$-\frac{1}{4}$	$\frac{3}{4}$	0
	x_5	1	1	0	0	0	1
	x_5'	$-\frac{1}{4}$	0	0	$-\frac{3}{8}$	$-\frac{5}{8}$	1

(7.3.13)

Виконавши ітерацію симплекс методом отримуємо новий розв'язок $x_1^1 = \left[1; \frac{8}{3}\right]$.

Цей розв'язок також не цілочисельний, його оцінка $\xi\left(G_1^1\right) = 3$. Тепер знаходимо розв'язок на множині $G_1^2 = G_0 \cap (x_1 \geq 2)$. Це виконується аналогічним чином до попереднього кроку з x_5. Отримали нерозв'язну задачу. Отже, $\xi\left(G_2^1\right) = -\infty$. Інші ітерації виконуються так само.