Київський національний університет
імені Тараса Шевченка

Ю.Д.Попов, В.І.Тюптя, В.І.Шевченко

Методичні рекомендації
до виконання лабораторних робіт з методів оптимізації на персональних комп'ютерах

Київ
Електронна бібліотека факультету кібернетики
2003

Методичні рекомендації до виконання лабораторних робіт з методів оптимізації на персональних комп'ютерах / Упорядн. Юрій Дмитрович Попов, Володимир Іванович Тюптя, Віталій Іванович Шевченко. — К.: **Електронне видання. Ел. бібліотека факультету кібернетики Київського національного університету імені Тараса Шевченка,** 2003, — 53 с.

Рецензенти: С.І. Ляшко, д-р фіз.-мат. наук;
В.Ф. Кузенко, канд. фіз.-мат. наук

Затверджено вченою радою
факультету кібернетики
25 жовтня 2000 року

Загальні рекомендації до використання програмного забезпечення

Програмне забезпечення *ПЗ–МО* складають діалогові навчаючі програми з лінійних та нелінійних методів оптимізації (математичного програмування), кожна з яких завантажується відповідною командою за допомогою таких меню.

М Е Т О Д И   О П Т И М І З А Ц І Ї
Лінійне програмування
Нелінійне програмування
Вихід в DOS

Л І Н І Й Н Е   П Р О Г Р А М У В А Н Н Я
Елементарні перетворення матриць
Геометрична інтерпретація ЗЛП
ЗЛП, симплекс-метод
ЗЛП, модифікований симплекс-метод
ЗЛП, двоїстий симплекс-метод
Транспортна задача, метод потенціалів
ТЗ з обмеженнями, метод потенціалів
ТЗ з обмеженнями, наближені методи
Задача про найкоротший шлях, метод Мінті
Задача про максимальний потік, метод Форда-Фалкерсона
Цілочисельна ЗЛП, 1-й метод Гоморі
Частково цілочисельна ЗЛП, 2-й метод Гоморі
Цілочисельна ЗЛП, 3-й метод Гоморі
Дискретна ЗЛП, метод Дальтона-Ллевеліна
Цілочисельна ЗЛП, метод віток і границь
Задача про призначення, угорський метод
Задача про призначення, метод Мака
Задача про призначення, наближені методи
Матричні ігри, метод Брауна-Робінсон
ВИХІД ДО ГОЛОВНОГО МЕНЮ

Н Е Л І Н І Й Н Е   П Р О Г Р А М У В А Н Н Я
Методи одновимірної оптимізації
Геометрична інтерпретація ЗНЛП
Квадратичний симплекс-метод
Метод найшвидшого спуску
Eureka: The Solver
ВИХІД ДО ГОЛОВНОГО МЕНЮ

Після запуску програми на дисплеї з'являється заставка з назвою задачі, що розглядається, і методу її розв'язування, після чого екран поділяється на кілька зон. В одній з них розміщене ***головне меню***. Кожне слово меню є командою вищого рівня. Вибір команди в меню здійснюється за допомогою клавіш управління курсором, або натискуванням клавіші з виділеною літерою (український алфавіт). В меню ці літери виділяються розміром і підвищеною яскравістю.

В ***робочій області*** екрану розміщується таблиця з даними задачі (симплекс-таблиця, транспортна таблиця і т. і.) або креслення, що описує задачу (графік, мережа і т. і.).

В ***рядках стану*** надається інформація про номер задачі, що розв'язується, крок алгоритму, наявність штучних змінних і т. і.

Вихід з програми здійснюється за допомогою команди «**Вихід**» головного меню.

До виходу на комп'ютер студент повинен вивчити формулювання оптимізаційної задачі та метод її розв'язування (див., наприклад, [1]). Під час роботи на комп'ютері необхідну довідкову інформацію всі програми надають, якщо виконати команду «**Довідка**» головного меню (або натиснути клавішу **F1**). До того ж, програми повністю контролюють дії користувача, дають необхідні підказки, нараховують штрафні бали при відхиленнях від алгоритму.

Кожна програма містить до *9* різних конкретних прикладів, які можна ввести для розв'язування за допомогою команди «**Дані**» головного меню. Можливе також введення даних з клавіатури.

У більшості програм передбачена демонстрація роботи в автоматичному режимі у відповідності з алгоритмом розв'язування оптимізаційної задачі. Для цього необхідно виконати команду «**Демонстрація**» головного меню.

Лабораторна робота 1.

Елементарні перетворення матриць. Метод Гаусса

Постановка задачі.

Розв'язати систему лінійних алгебраїчних рівнянь

$$
\begin{array}{l}
a_{11}x_1 + \ldots + a_{1n}\ x_n = a_{10}, \\
\ldots\ldots\ldots\ldots\ldots\ldots, \\
a_{m1}x_1 + \ldots + a_{mn}x_n = a_{m0},
\end{array}
\qquad (1.1)
$$

що означає:

1. визначити, чи сумісна дана система (тобто, чи має вона хоча б один розв'язок);

2. якщо система сумісна, то, чи має вона єдиний розв'язок, чи таких розв'язків безліч;

3. знайти розв'язок, якщо він єдиний, і знайти загальний розв'язок у випадку існування безлічі розв'язків.

Питання сумісності системи (1.1) розв'язує **теорема Кронекера-Капеллі**, формулювання якої таке:

Система лінійних алгебраїчних рівнянь (1.1) *сумісна в тому і тільки в тому випадку, коли співпадають ранги основної і розширеної матриць*

$$\boldsymbol{A} = \left\| \begin{array}{ccc} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{array} \right\|, \qquad \boldsymbol{A}^* = \left\| \begin{array}{cccc} a_{11} & \dots & a_{1n} & a_{10} \\ \dots & \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} & a_{m0} \end{array} \right\|.$$

При цьому розв'язок єдиний тоді і тільки тоді, коли $\operatorname{rang}(\boldsymbol{A})=n$. *Якщо ж* $\operatorname{rang}(\boldsymbol{A})=r<n$, *то обов'язково знайдуться такі r змінних* $x_{i_1},\dots,x_{i_r}$, *що вихідна система* (1.1) *буде еквівалентною системі вигляду*:

$$\begin{aligned} x_{i_1} \quad &+\sum b_{1j}\, x_j = b_{10}, \\ &\dots\dots\dots\dots\dots\dots, \\ x_{i_r} &+\sum b_{rj}\, x_j = b_{r0}, \end{aligned} \tag{1.2}$$

де підсумовування ведеться по $j=1,\dots,n$, *відмінних від* $i_1,\dots,i_r$.

Про систему (1.2) говорять, що вона розв'язана відносно змінних $x_{i_1},\dots,x_{i_r}$ і має канонічний вигляд.

Загальний розв'язок системи (1.2) визначається тривіально:

$$\begin{aligned} x_{i_1} &= b_{10} - \sum b_{1j} x_j, \\ &\dots\dots\dots\dots\dots\dots, \\ x_{i_r} &= b_{r0} - \sum b_{rj} x_j. \end{aligned} \tag{1.3}$$

В рівностях (1.3) змінні x_j (j відмінне від $i_1,\dots,i_r$) — вільні змінні, вони можуть набувати довільних значень. Кожна сукупність їх значень однозначно визначає значення змінних $x_{i_1},\dots,x_{i_r}$ — базисних змінних.

Метод Гаусса є основним скінченим методом розв'язування систем лінійних алгебраїчних рівнянь (1.1) і полягає у виконанні однотипних *перетворень виключення*.

На 1-у кроці методу знаходять змінну x_{i_1} у 1-у рівнянні, коефіцієнт біля якої не дорівнює нулю, і виключають цю змінну із всіх інших рівнянь. Якщо, наприклад, a_{11} не дорівнює нулю, то перетворення 1-го кроку такі: до *i*-го рівняння додати 1-е рівняння, помножене на число $-a_{i1}/a_{11}$, $i=2,\dots,m$.

В результаті перетворень 1-го кроку система (1.1) набуває вигляду:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}\ x_n &= a_{10}, \\ a'_{22}x_2 + \dots + a'_{2n}\ x_n &= a'_{20}, \\ \dots\dots\dots\dots\dots\dots\dots\dots&, \\ a'_{m2}x_2 + \dots + a'_{mn}x_n &= a'_{m0}. \end{aligned} \tag{1.4}$$

На наступному кроці ця ж послідовність дій застосовується до частини системи, що складається з системи (1.4) без 1-го рівняння і т. д. Згаданий процес продовжується доти, поки не буде отримана система трапецієвидного (зокрема, трикутного) вигляду, або не буде підтверджена ознака несумісності системи.

Описані перетворення складають прямий хід методу Гаусса. При прямому ході перетворення виконуються «зверху вниз» — від першого рівняння до останнього. Зворотний хід методу Гаусса складають перетворення виключення відокремлених при прямому ході базисних змінних, при цьому перетворення виконуються в зворотному порядку: від останнього рівняння до першого — «знизу вверх».

Прямий та зворотний хід методу Гаусса дістав назву *методу повного виключення Жордана-Гаусса.*

Крім основного призначення (як методу розв'язування систем лінійних алгебраїчних рівнянь) метод Гаусса має і інші важливі застосування.

За допомогою прямого ходу методу Гаусса можна обчислювати *чисельні визначники* (шляхом зведення визначника до трикутного вигляду).

Метод Жордана-Гаусса дозволяє знаходити *обернені матриці*.

Нехай маємо деяку невироджену квадратну матрицю:

$$\boldsymbol{A} = \begin{Vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{Vmatrix}.$$

Побудуємо розширену матрицю вигляду:

$$\begin{Vmatrix} a_{11} & \dots & a_{1n} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} & 0 & \dots & 1 \end{Vmatrix}$$

і застосуємо до неї метод Жордана-Гаусса. Якщо в результаті отримаємо матрицю вигляду:

$$\begin{Vmatrix} 1 & \dots & 0 & b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & b_{n1} & \dots & b_{nn} \end{Vmatrix},$$

то

$$\begin{Vmatrix} b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{nn} \end{Vmatrix}$$

Є оберненою матрицею до вихідної матриці $\boldsymbol{A}$.

<u>*Програмне забезпечення*</u>.

Модуль, який виконує елементарні перетворення матриць, що лежать в основі методу Гаусса, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

<u>*Завдання*</u>.

Розв'язати методом Гаусса системи лінійних алгебраїчних рівнянь, розширені матриці яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

Розв'язати методом Гаусса системи лінійних алгебраїчних рівнянь, що задані розширеними матрицями:

$$\textbf{1)} \begin{Vmatrix} 2 & -2 & 0 & 1 & -3 \\ 2 & 3 & 1 & -3 & -6 \\ 3 & 4 & -1 & 2 & 0 \\ 1 & 3 & 1 & -1 & 2 \end{Vmatrix} \quad \textbf{2)} \begin{Vmatrix} 1 & 1 & -6 & -4 & 6 \\ 3 & -1 & -6 & -4 & 2 \\ 2 & 3 & 9 & 2 & 6 \\ 3 & 2 & 3 & 8 & -7 \end{Vmatrix} \quad \textbf{3)} \begin{Vmatrix} 2 & -1 & 3 & 4 & 5 \\ 4 & -2 & 5 & 6 & 7 \\ 6 & -3 & 7 & 8 & 9 \\ 8 & -4 & 9 & 10 & 11 \end{Vmatrix}$$

Обчислити такі визначники:

4) $\begin{Vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{Vmatrix}$ **5)** $\begin{Vmatrix} 1 & 1 & 6 & 4 \\ 3 & 1 & 6 & 4 \\ 2 & 3 & 9 & 2 \\ 3 & 2 & 3 & 8 \end{Vmatrix}$ **6)** $\begin{Vmatrix} 3 & 6 & 5 & 6 & 4 \\ 5 & 9 & 7 & 8 & 6 \\ 6 & 12 & 13 & 9 & 7 \\ 2 & 5 & 4 & 5 & 3 \\ 4 & 6 & 6 & 5 & 4 \end{Vmatrix}$

Знайти обернені до таких матриць:

7) $\begin{Vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{Vmatrix}$ **8)** $\begin{Vmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 2 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & 6 \end{Vmatrix}$ **9)** $\begin{Vmatrix} 3 & 6 & 5 & 6 \\ 5 & 9 & 7 & 8 \\ 6 & 12 & 13 & 9 \\ 2 & 5 & 4 & 5 \end{Vmatrix}$

Лабораторна робота 2.

Задача лінійного програмування. Симплекс-метод

Постановка задачі лінійного програмування в стандартній формі (СЗЛП).

Знайти вектор $\boldsymbol{x} = (x_1, \dots, x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \dots + c_n x_n \tag{2.1}$$

і задовольняє систему обмежень

$$\begin{aligned} &a_{11}x_1 + \dots + a_{1n}\ x_n = a_{10}, \\ &\dots\dots\dots\dots\dots\dots, \\ &a_{m1}x_1 + \dots + a_{mn}x_n = a_{m0}, \end{aligned} \tag{2.2}$$

$$x_j \ge 0,\ j = 1, \dots, n. \tag{2.3}$$

В задачі (2.1)–(2.3) будемо називати матрицю $\boldsymbol{A} = \|a_{ij}\|$, $i=1,\dots,m,\ j=1,\dots,n,$ *матрицею умов*; вектор-стовпець $\boldsymbol{A}_j = (a_{1j}, \dots, a_{mj})^{\mathrm{T}}$, $j=1,\dots,n,$ — j-м *вектором умов*; вектор $\boldsymbol{A}_0 = (a_{10}, \dots, a_{m0})^{\mathrm{T}}$ — *вектором обмежень*.

Основні означення.

Вектор $\boldsymbol{x} = (x_1, \dots, x_n)$ називається *допустимим розв'язком ЗЛП*, якщо його компоненти задовольняють обмеження (2.2)–(2.3).

Ненульовий допустимий розв'язок $\boldsymbol{x}$ — *базисний* (*ДБР*), якщо його додатним компонентам x_j відповідають лінійно незалежні вектори умов $\boldsymbol{A}_j$. (Нульовий допустимий розв'язок будемо завжди вважати базисним).

Система m лінійно незалежних векторів умов, що включає вказані вектори $\boldsymbol{A}_j$, називається *базисом*.

Вектори базису утворюють *базисну матрицю*.

Виклад симплекс-методу.

Симплекс-метод (*СМ*) безпосередньо застосовується до розв'язування *канонічної задачі лінійного програмування* (*КЗЛП*) і здійснює цілеспрямоване перебирання допустимих базисних розв'язків (*ДБР*) (вершин *допустимого многогранника* розв'язків *ЗЛП*, що визначається обмеженнями (2.2)–(2.3)), до множини яких належить оптимальний розв'язок, якщо він існує; або визначає, що *ЗЛП* не має оптимального розв'язку.

ЗЛП — *канонічна*, якщо її обмеження (2.2) мають канонічну форму:

$$x_i + a_{i,m+1}\, x_{m+1} + \ldots + a_{in}\, x_n = a_{i0}, \quad a_{i0} \geq 0, \; i=1,\ldots,m,$$

тобто, матриця умов $\boldsymbol{A} = \| a_{ij} \|$, $i=1,\ldots,m,\; j=1,\ldots,n,$ містить в собі одиничну підматрицю розміру $m \times m$ і вектор обмежень $\boldsymbol{A}_0 = (a_{10}, \ldots, a_{m0})^{\text{T}}$ — невід'ємний.

КЗЛП елементарно визначає такі основні в лінійному програмуванні конструкції:

- деякий *ДБР* — $\boldsymbol{x}(0) = (a_{10}, \ldots, a_{m0}, 0, \ldots, 0)^{\text{T}}$;
- його базис — m-вимірні одиничні вектори $(1,0,\ldots,0)^{\text{T}}$, $(0,1,\ldots,0)^{\text{T}}$,..., $(0,\ldots,0,1)^{\text{T}}$;
- його базисну матрицю $\boldsymbol{B}$ — одиничну матрицю розміру $m \times m$;
- базисні змінні — $x_1, \ldots, x_m$.

Стандартна *ЗЛП* (2.1)–(2.3) зводиться в загальному випадку до канонічної *ЗЛП* додаванням штучних невід'ємних змінних до лівих частин обмежень (2.2) і введенням цих же змінних з досить великим коефіцієнтом $M > 0$ до цільової функції (2.1) (*М-метод*).

М-задача має вигляд:

$$L(\boldsymbol{x}) = c_1 x_1 + \ldots + c_n x_n + M(y_1 + \ldots + y_m) \to \min,$$

$$\begin{aligned} a_{11} x_1 + \ldots + a_{1n} x_n + y_1 \qquad &= a_{10}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots &, \\ a_{m1} x_1 + \ldots + a_{mn} x_n + \qquad y_m &= a_{m0}, \end{aligned}$$

$$x_j \geq 0, \; j=1,\ldots,n, \; y_i \geq 0, \; i=1,\ldots,m,$$

$$a_{i0} \geq 0, \; i=1,\ldots,m.$$

Ознака оптимальності: Якщо на деякому кроці *СМ* симплекс-різниці (див. алгоритм) *ДБР* $\boldsymbol{x}^*$ невід'ємні, то $\boldsymbol{x}^*$ — оптимальний розв'язок *КЗЛП*.

Умови відсутності розв'язку:

1. *ЗЛП* не має оптимального розв'язку, якщо на якому-небудь кроці хоча б один вектор умов $\boldsymbol{A}_j$ з від'ємною оцінкою Δ_j не має додатних компонент;
2. *ЗЛП* не має розв'язків, якщо хоча б одна штучна змінна додатна у випадку, коли виконується ознака оптимальності.

Алгоритм симплекс-методу.

На кожному кроці *СМ* виконуються такі дії (розрахункові формули наводяться лише для першого кроку).

1. Розглядається *ДБР* $\boldsymbol{x} = (a_{10}, \ldots, a_{m0}, 0, \ldots, 0)^{\text{T}}$.

   Обчислюються *відносні оцінки* (*симплекс-різниці*) Δ_j небазисних змінних x_j, $j=m+1,\ldots,n,$ за формулою:

   $$\Delta_j = c_j - (\boldsymbol{c}_б, \boldsymbol{A}_j),$$

   де $\boldsymbol{c}_б = (c_1, \ldots, c_m)^{\text{T}}$, $\boldsymbol{A}_j$ — вектор умов, що відповідає змінній x_j (відносні оцінки базисних змінних дорівнюють нулю).

   Якщо для всіх $j=1,\ldots,n,$ виконується умова $\Delta_j \geq 0$, то *ДБР* $\boldsymbol{x}$ є оптимальним розв'язком *КЗЛП*. Якщо до того ж усі штучні змінні дорівнюють нулю, то, відкидаючи їх, отримаємо оптимальний розв'язок вихідної *ЗЛП*; інакше вихідна *ЗЛП* не має розв'язків (її допустима область порожня).

Якщо існує таке j, що $\Delta_j < 0$, а вектор умов $\boldsymbol{A}_j$ не має додатних компонент, то *ЗЛП* не має оптимального розв'язку (її цільова функція $L(\boldsymbol{x})$ не обмежена знизу на допустимому многограннику розв'язків). Кінець обчислень.

2. Якщо існують індекси j, для яких $\Delta_j < 0$, а відповідні вектори умов $\boldsymbol{A}_j$ мають додатні компоненти, то знаходять k за формулою

$$k = \operatorname*{argmin}_{j:\, \Delta_j < 0} \Delta_j ,$$

і, обчислюючи для $a_{ik} > 0$ відношення $\theta_i = a_{i0}/a_{ik}$, визначають l, що задовольняє співвідношення

$$l = \operatorname*{argmin}_{i:\, a_{ik} > 0} \theta_i .$$

3. Переходять до нового *ДБР*, шляхом введення в базис вектора $\boldsymbol{A}_k$ і виведення з базису вектора $\boldsymbol{A}_l$. Такий перехід здійснюється за допомогою симплекс-перетворень (елементарних перетворень Жордана-Гаусса) з ведучим елементом a_{lk} над елементами розширеної матриці умов (2.2). Перехід до пункту 1.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого *ЗЛП* розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Геометрична інтерпретація задачі лінійного програмування здійснюється модулем, який завантажується з того ж самого розділу.

Завдання.

Розв'язати симплекс-методом задачі лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються нижче, усі змінні невід'ємні.

1)
$$\begin{aligned} &x_1 + x_2 + x_3 \to \min, \\ &x_1 - x_4 - 2x_6 = 5, \\ &x_2 + 2x_4 - 3x_5 + x_6 = 3, \\ &x_3 + 2x_4 - 5x_5 + 6x_6 = 5; \end{aligned}$$

2)
$$\begin{aligned} &2x_1 + x_2 - x_3 - x_4 \to \min, \\ &x_1 + x_2 + 2x_3 - x_4 = 2, \\ &2x_1 + x_2 - 3x_3 + x_4 = 6, \\ &x_1 + x_2 + x_3 + x_4 = 7; \end{aligned}$$

3)
$$\begin{aligned} &x_1 - 2x_2 + 3x_3 \to \min, \\ &2x_1 + 3x_2 + 4x_3 = 1, \\ &-2x_1 + x_2 + 3x_3 = 2; \end{aligned}$$

4)
$$\begin{aligned} &2x_1 - 3x_2 \to \max, \\ &5x_1 + 2x_2 \ge 10, \\ &x_1 + 3x_2 \le 12; \end{aligned}$$

5)
$$\begin{aligned} &6x_1 + 4x_2 \to \min, \\ &2x_1 + x_2 \ge 3, \\ &X_1 - x_2 \le 1; \end{aligned}$$

6)
$$\begin{aligned} &2x_1 - 4x_2 \to \min, \\ &8x_1 - 5x_2 \le 16, \\ &x_1 + 3x_2 \ge 2, \\ &2x_1 + 7x_2 \le 9; \end{aligned}$$

7)
$$\begin{aligned} &7x_1 + 5x_2 \to \max, \\ &7x_1 + 5x_2 \ge 7, \\ &7x_1 - 5x_2 \ge 35, \\ &x_1 - x_2 \le 0; \end{aligned}$$

8)
$$\begin{aligned} &3x_1 + 2x_2 \to \max, \\ &4x_1 + 2x_2 \ge 12, \\ &x_1 + 2x_2 \le 10, \\ &2x_1 + 2x_2 = 6; \end{aligned}$$

9) $4x_1 + 5x_2 + 9x_3 + 11x_4 \to \max,$
$x_1 + x_2 + x_3 + x_4 \le 15,$
$7x_1 + 5x_2 + 3x_3 + 2x_4 \le 80,$
$3x_1 + 5x_2 + 10x_3 + 15x_4 \le 60;$

10) $+x_1 - x_2 - 2x_3 + x_4 \to \max,$
$2x_1 + x_2 + x_3 - x_4 = 2,$
$-3x_1 + x_2 + 2x_3 + x_4 = 6,$
$x_1 + x_2 + x_3 + x_4 = 7;$

11) $4x_1 + x_2 - 2x_3 - x_4 - x_5 \to \min,$
$x_3 - x_4 + x_5 = 1,$
$x_2 + 2x_4 - x_5 = 1,$
$x_1 + 2x_2 + 2x_5 = 4;$

12) $x_1 + 2x_2 + 3x_3 - x_4 \to \max,$
$x_1 + 2x_2 + 3x_3 = 15,$
$2x_1 + x_2 - 3x_3 = 20,$
$x_1 + 2x_2 + x_4 = 10.$

Відповіді:

1) $\boldsymbol{x}^* = (7.1; 0; 0; 1.3; 0; 0.4)$, $L(\boldsymbol{x}^*) = 7.1$.
2) $\boldsymbol{x}^* = (0; 4.13; 0.25; 2.63)$, $L(\boldsymbol{x}^*) = 1.25$.
3) Розв’язку немає ($D = \varnothing$).
4) $\boldsymbol{x}^* = (12; 0)$, $L(\boldsymbol{x}^*) = 24$.
5) $\boldsymbol{x}^* = (1.33; 0.33)$, $L(\boldsymbol{x}^*) = 9.33$.
6) $\boldsymbol{x}^* = (0; 1.29)$, $L(\boldsymbol{x}^*) = -5.14$.
7) Розв’язку немає (цільова функція необмежена зверху на допустимій множині).
8) $\boldsymbol{x}^* = (3; 0)$, $L(\boldsymbol{x}^*) = 9$.
9) $\boldsymbol{x}^* = (10.16; 0; 2.95; 0)$, $L(\boldsymbol{x}^*) = 67.21$.
10) $\boldsymbol{x}^* = (0.25; 4.13; 0; 2.63)$, $L(\boldsymbol{x}^*) = -1.25$.
11) $\boldsymbol{x}^* = (0; 0; 0.5; 1.5; 2)$, $L(\boldsymbol{x}^*) = -4.5$.
12) Розв’язку немає ($D = \varnothing$).

Лабораторна робота 3.

Задача лінійного програмування. Модифікований симплекс-метод

Виклад модифікованого симплекс-методу.

Модифікований симплекс-метод (*МСМ*) безпосередньо застосовується до розв'язування *КЗЛП* і здійснює цілеспрямоване перебирання *ДБР*, до множини яких належить оптимальний розв'язок, якщо він існує; або визначає, що *ЗЛП* не має оптимального розв'язку.

У *МСМ* на відміну від *СМ* симплекс-перетворення застосовуються тільки до базисної частини матриці умов $\boldsymbol{A}$.

Ознака оптимальності та умови відсутності оптимального розв'язку *ЗЛП* для *МСМ* і *СМ* однакові.

Алгоритм модифікованого симплекс-методу.

На *s*-у кроці *МСМ* виконуються такі дії:

1. Обчислюються відносні оцінки (симплекс-різниці)

$$\Delta_j[s] = c_j - (\boldsymbol{u}[s],\ \boldsymbol{A}_j),$$

де $\boldsymbol{u}[s]^T = \boldsymbol{c}_б[s]^T \boldsymbol{B}_{об}[s]$ — вектор симплекс-множників, $\boldsymbol{B}_{об}[s]$ — матриця, обернена до базисної, $\boldsymbol{c}_б[s]$ — базисна частина вектора $\boldsymbol{c}$. Якщо $\Delta_j[s] \ge 0$

для всіх $j=1,\ldots,n$, то кінець обчислень: поточний базисний розв'язок оптимальний. Якщо існує j таке, що $\Delta_j[s] < 0$, то перейти до пункту 2.

2. Вибирається k за формулою

$$k = \underset{j:\,\Delta_j[s]<0}{\operatorname{argmin}} \Delta_j[s].$$

Обчислюються вектори $\boldsymbol{A}_k[s] = \boldsymbol{B}_{об}[s]\,\boldsymbol{A}_k$ та $\boldsymbol{A}_0[s] = \boldsymbol{B}_{об}[s]\,\boldsymbol{A}_0$.
Якщо $\boldsymbol{A}_k[s] \le 0$, то кінець: цільова функція необмежена знизу на допустимій множині. Якщо існують i такі, що $a_{ik}[s] > 0$, то перейти до пункту 3.

3. Обчислюються відношення $\theta_i[s] = a_{i0}[s]/a_{ik}[s]$ для всіх $a_{ik}[s] > 0$ та вибирається l так, що

$$l = \underset{i:\,a_{ik}[s]>0}{\operatorname{argmin}} \theta_i[s].$$

Вектор $\boldsymbol{A}_k$ належить ввести до базису, а вектор, що є базисним у l–му непрямому обмеженні, повинен бути виведений з базису.

4. Матриця $\boldsymbol{B}_{об}[s+1]$ обчислюється за матрицею $\boldsymbol{B}_{об}[s]$ за допомогою звичайних симплекс-перетворень з використанням ведучого елемента $a_{lk}[s]$. Подібним чином може бути визначений і вектор $\boldsymbol{A}_0[s+1]$. Для *КЗЛП* $\boldsymbol{B}_{об}[0]$ — одинична матриця розмірності $m \times m$.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого *ЗЛП* розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Завдання.

Розв'язати модифікованим симплекс-методом задачі лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються нижче, всі змінні невід'ємні.

1)
$$\begin{aligned}
&x_1 - x_2 + 3x_4 + x_5 \to \min,\\
&x_1 + 2x_2 + 3x_3 + 3x_4 + x_5 = 15,\\
&2x_1 - x_2 - 3x_3 + x_4 - x_5 = 4,\\
&2x_1 - 2x_2 - x_3 + 2x_5 = 8;
\end{aligned}$$

2)
$$\begin{aligned}
&3x_1 + x_2 + x_4 + x_5 \to \min,\\
&2x_1 - x_2 + x_4 - x_5 = 9,\\
&4x_1 - x_2 - x_3 - x_5 = 4,\\
&x_1 - x_2 - x_3 - x_5 = 6;
\end{aligned}$$

3)
$$\begin{aligned}
&x_1 - 2x_2 + 2x_3 + x_4 + 2x_5 \to \max,\\
&-x_1 + x_2 - 2x_3 + 3x_4 + x_5 = 2,\\
&-x_1 + 2x_2 - x_3 + 2x_4 = 3,\\
&2x_1 + 3x_2 + x_4 - x_5 = 6;
\end{aligned}$$

4)
$$\begin{aligned}
&x_1 + 2x_2 + x_3 - x_4 \to \min,\\
&-x_1 + 5x_2 + x_3 + x_4 + x_5 = 10,\\
&2x_1 - x_2 + x_3 - 3x_4 = 6,\\
&10x_2 + x_3 + 2x_4 + 3x_5 = 25;
\end{aligned}$$

5)
$$\begin{aligned}
&x_1 - x_2 + x_3 + x_4 - x_5 \to \min,\\
&x_1 + x_4 + 6x_5 = 9,\\
&3x_1 + x_2 - 4x_3 + 2x_5 = 2,\\
&x_1 + 2x_2 + 2x_5 = 6;
\end{aligned}$$

6)
$$\begin{aligned}
&x_1 - x_2 + x_3 + 2x_4 \to \max,\\
&x_1 + x_2 + x_3 + 2x_4 + 3x_5 = 7,\\
&x_1 + 2x_2 + 2x_3 + 3x_4 + 2x_5 = 12,\\
&2x_1 + 3x_2 + 4x_3 + 4x_4 - x_5 = 22;
\end{aligned}$$

7) $x_1 + 2x_2 - x_3 - x_4 \to \min,$
$x_1 + x_2 + 2x_3 - x_4 = 2,$
$x_1 + 2x_2 - 3x_3 + x_4 = 6,$
$x_1 + x_2 + x_3 + x_4 = 7;$

8) $x_1 - x_2 + x_3 + x_4 + x_5 - x_6 \to \min,$
$4x_1 + x_2 - 4x_3 + x_4 + 8x_6 = 15,$
$4x_1 + x_2 - 2x_3 + x_5 + 4x_6 = 8,$
$5x_1 + x_2 - 2x_3 + x_4 + x_5 + 10x_6 = 21;$

9) $x_1 + 2x_2 + 3x_3 + x_4 + x_5 \to \max,$
$x_1 + x_2 + 4x_3 - x_4 + x_5 = 1,$
$x_1 + x_2 - 2x_3 + x_4 + x_5 = 1,$
$-x_1 + x_2 - 6x_3 + x_4 + x_5 = 1;$

10) $x_1 - x_2 + x_3 - 3x_4 + 2x_5 \to \min,$
$x_1 + 2x_2 - x_3 + 2x_4 + x_5 \geq 8,$
$-x_1 - x_2 + x_3 - 3x_4 - 2x_5 = 10,$
$2x_1 - x_2 + 3x_3 - x_4 + 2x_5 \leq 4.$

Відповіді:

1) $\boldsymbol{x}^* = (5.11; 3.67; 0; 0; 2.56)$, $L(\boldsymbol{x}^*) = 4$.
2) Розв'язку немає ($D = \varnothing$).
3) Розв'язку немає (цільова функція необмежена зверху на допустимій множині).
4) $\boldsymbol{x}^* = (1; 0; 4; 0; 7)$, $L(\boldsymbol{x}^*) = 5$.
5) $\boldsymbol{x}^* = (0; 1.5; 0.63; 0; 1.5)$, $L(\boldsymbol{x}^*) = -2.38$.
6) $\boldsymbol{x}^* = (1; 0; 4; 1; 0)$, $L(\boldsymbol{x}^*) = 7$.
7) $\boldsymbol{x}^* = (4.13; 0; 1.25; 2.62)$, $L(\boldsymbol{x}^*) = 1.25$.
8) $\boldsymbol{x}^* = (0; 1; 0.83; 0; 0; 2.17)$, $L(\boldsymbol{x}^*) = -2.33$.
9) $\boldsymbol{x}^* = (0; 1; 0; 0; 0)$, $L(\boldsymbol{x}^*) = 2$.
10) Розв'язку немає ($D = \varnothing$).

Лабораторна робота 4.

Задача лінійного програмування. Двоїстий симплекс-метод

Виклад двоїстого симплекс-методу.

Двоїстий симплекс-метод (*ДСМ*) безпосередньо застосовується до розв'язування *майже канонічної задачі лінійного програмування* (*МКЗЛП*), яка формулюється таким чином:

Знайти вектор $\boldsymbol{x} = (x_1,...,x_n)$, що мінімізує лінійну функцію

$$L(\boldsymbol{x}) = c_1 x_1 + ... + c_n x_n \tag{4.1}$$

і задовольняє систему лінійних обмежень

$$x_i + a_{i,m+1} x_{m+1} + ... + a_{in} x_n = a_{i0}, \; i=1,\dots,m, \tag{4.2}$$

$$x_j \geq 0, \; j=1,\dots,n, \tag{4.3}$$

(компоненти a_{i0} вектора обмежень $\boldsymbol{A}_0$ можуть бути від'ємними) при додатковій умові: *відносні оцінки* (*симплекс-різниці*) Δ_j *змінних* x_j *невід'ємні*.

Вектор $\boldsymbol{x} = (x_1, \dots, x_n)$ називається *майже допустимим базисним розв'язком* (*МДБР*) *МКЗЛП*, якщо його компоненти задовольняють обмеження (4.2), і ненульовим компонентам x_j відповідають лінійно незалежні вектори умов $\boldsymbol{A}_j$.

Базис і базисна матриця *МДБР* визначаються подібно тому, як це робиться для *СЗЛП*.

МКЗЛП є частинним випадком *СЗЛП*. Існують методи зведення довільної *ЗЛП* до майже канонічного вигляду.

Ознака оптимальності: Якщо на деякому кроці *ДСМ* компоненти *МДБР* $\boldsymbol{x}^*$ невід'ємні, то $\boldsymbol{x}^*$ — оптимальний розв'язок *МКЗЛП*.

Ознака відсутності розв'язку: Оптимального розв'язку *МКЗЛП* не існує, якщо на якому-небудь кроці *ДСМ* в рядку з $a_{i0} < 0$ всі компоненти $a_{ij} \geq 0,\ j=1,\ldots,n$. В цьому випадку допустима множина розв'язків *МКЗЛП* порожня.

Алгоритм двоїстого симплекс-методу.

На кожному кроці *ДСМ* виконуються такі дії (розрахункові формули наводяться лише для першого кроку).

1. Розглядається *МДБР* $\boldsymbol{x} = (a_{10}, \ldots, a_{m0}, 0, \ldots, 0)$.
Обчислюються відносні оцінки (симплекс-різниці) Δ_j небазисних змінних x_j, $j=m+1,\ldots,n,$ за формулою:

$$\Delta_j = c_j - (\boldsymbol{c}_Б, \boldsymbol{A}_j),$$

де $\boldsymbol{c}_Б = (c_1,\ldots,c_m)$, $\boldsymbol{A}_j$ — вектор умов, що відповідає змінній x_j (відносні оцінки базисних змінних дорівнюють нулю).
Якщо для всіх $i=1,\ldots,m$ виконується умова $a_{i0} \geq 0$, то *МДБР* $\boldsymbol{x}$ буде оптимальним розв'язком *МКЗЛП*. Кінець обчислень.
Якщо існує таке i, що $a_{i0} < 0$, а коефіцієнти $a_{ij} \geq 0$, $j=1,\ldots,n$, то *МКЗЛП* не має допустимих розв'язків. Кінець обчислень.

2. Якщо існують індекси i, для яких $a_{i0} < 0$, а серед відповідних компонент a_{ij}, $j=1,\ldots,n$, є від'ємні, то знаходять l:

$$l = \underset{i:\, a_{i0} < 0}{\operatorname{argmin}}\, a_{i0},$$

обчислюють відношення $\gamma_j = -\Delta_j / a_{lj}$ для всіх $a_{lj} < 0$ та визначають k:

$$k = \underset{i:\, a_{lj} < 0}{\operatorname{argmin}}\, \gamma_j.$$

3. Переходять до нового *МДБР*, виключаючи з базису вектор $\boldsymbol{A}_l$ і вводячи до базису вектор $\boldsymbol{A}_k$. Згаданий перехід здійснюється за допомогою симплекс-перетворень (елементарних перетворень Жордана-Гаусса з ведучим елементом a_{lk}) над елементами розширеної матриці умов. Перехід до п. **1**.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого *ЗЛП* розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Завдання.

Розв'язати двоїстим симплекс-методом задачі лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються далі, всі змінні невід'ємні.

1) $6x_1 + 4x_2 \to \min,$
$2x_1 + x_2 \ge 3,$
$x_1 - x_2 \le 1,$
$-x_1 + 2x_2 \ge 1;$

2) $2x_1 + 3x_2 \to \min,$
$x_1 + 5x_2 \ge 16,$
$3x_1 + 2x_2 \ge 12,$
$2x_1 + 4x_2 \ge 16;$

3) $-6x_1 - 4x_2 \to \max,$
$2x_1 + x_2 \ge 3,$
$x_1 - 2x_2 \le 2,$
$3x_1 + 2x_2 \ge 1;$

4) $6x_1 + 4x_2 \to \min,$
$2x_1 + x_2 \ge 3,$
$3x_1 + 2x_2 \ge 1,$
$-x_1 - x_2 \ge 6;$

5) $7x_1 + x_2 \to \min,$
$x_1 + x_2 \ge 3,$
$5x_1 + x_2 \ge 5,$
$x_1 + 5x_2 \ge 5;$

6) $7x_1 + 10x_2 \to \min,$
$2x_1 + 28x_2 \ge 17,$
$x_1 + 2x_2 \ge 3;$
$x_1 + 17x_2 \ge 19;$

7) $x_1 + x_2 + 2x_3 \to \min,$
$2x_1 - x_2 - 3x_3 + x_4 = -3,$
$x_1 - 3x_2 - 4x_3 + x_5 = -1;$

8) $-15x_1 - 33x_2 \to \max,$
$3x_1 + 2x_2 \ge 6,$
$6x_1 + x_2 \ge 6;$

9) $x_1 + 2x_2 \to \min,$
$2x_1 + x_2 \le 18,$
$x_1 + 2x_2 \ge 14,$
$x_1 - 2x_2 \le 10;$

10) $78x_1 + 52x_2 \to \min,$
$6x_1 + 2x_2 \ge 9,$
$-10x_1 + 14x_2 \ge 13,$
$11x_1 - x_2 \ge 6;$

11) $5x_1 + 4x_2 \to \min,$
$x_1 + x_2 \le 6,$
$2x_1 + x_2 \ge 9,$
$3x_1 + x_2 \ge 11.$

Відповіді:

1) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 10$.
2) $\boldsymbol{x}^* = (2; 3)$, $L(\boldsymbol{x}^*) = 13$.
3) $\boldsymbol{x}^* = (1.5; 0)$, $L(\boldsymbol{x}^*) = -9$.
4) Розв'язку немає ($D = \varnothing$).
5) $\boldsymbol{x}^* = (0; 5)$, $L(\boldsymbol{x}^*) = 5$.
6) $\boldsymbol{x}^* = (0; 1.5)$, $L(\boldsymbol{x}^*) = 15$.
7) $\boldsymbol{x}^* = (0; 0; 1; 0; 3)$, $L(\boldsymbol{x}^*) = 2$.
8) $\boldsymbol{x}^* = (2; 0)$, $L(\boldsymbol{x}^*) = -30$.
9) $\boldsymbol{x}^* = (7.33; 3.33)$, $L(\boldsymbol{x}^*) = 14$.
10) $\boldsymbol{x}^* = (0.96; 1.62)$, $L(\boldsymbol{x}^*) = 159.12$.
11) $\boldsymbol{x}^* = (4.5; 0)$, $L(\boldsymbol{x}^*) = 22.5$.

Лабораторна робота 5.

Транспортна задача. Метод потенціалів

Постановка транспортної задачі.

В кожному з пунктів P_i, $i=1,\ldots,m$, виробляється a_i одиниць деякого однорідного продукту, а в кожному з пунктів Q_j, $j=1,\ldots,n$, споживається b_j одиниць того ж продукту. Можливе транспортування продукту із кожного пункту виробництва P_i в кожний пункт споживання Q_j. Вартість перевезення одиниці продукту з пункту P_i в пункт Q_j відома і складає c_{ij} одиниць. Вважаючи, що сумарний об'єм виробництва дорівнює сумарному об'єму споживання, потрібно скласти план перевезень продукту, що мінімізує сумарні транспортні витрати.

Математична модель транспортної задачі має вигляд:

$$L(\boldsymbol{x}) = c_{11}x_{11} + \ldots + c_{1n}x_{1n} + \ldots + c_{m1}x_{m1} + \ldots + c_{mn}x_{mn} \to \min,$$

$$x_{i1} + \ldots + x_{in} = a_i,\ i=1,\ldots,m,$$

$$x_{1j} + \ldots + x_{mj} = b_j,\ j=1,\ldots,n,$$

$$x_{ij} \geq 0,\ i=1,\ldots,m,\ j=1,\ldots,n,$$

$$a_1 + \ldots + a_m = b_1 + \ldots + b_n.$$

Остання умова визначає *збалансовану транспортну задачу.*

У запропонованій моделі назвемо вектор $\boldsymbol{x}=(x_{11},\ldots,x_{1n},\ldots,x_{m1},\ldots,x_{mn})^{\mathrm{T}}$ вектором *перевезень*, вектор $\boldsymbol{b}=(a_1,\ldots,a_m,b_1,\ldots,b_n)^{\mathrm{T}}$ — вектором *запасів-потреб*, вектор $\boldsymbol{A}_{ij}=(0,\ldots,0,1,0\ldots,0,0,\ldots,0,1,0,\ldots,0)^{\mathrm{T}}$ — вектором *комунікації* P_iQ_j (вектор $\boldsymbol{A}_{ij}$ має розмірність $m+n$, причому перша одиниця стоїть на i-у місці, а друга — на $m+j$-у) і, нарешті, вектор $\boldsymbol{c}=(c_{11},\ldots,c_{1n},\ldots,c_{m1},\ldots,c_{mn})^{\mathrm{T}}$ — вектором *транспортних витрат*.

<u>*Основні означення*</u>.

Оскільки транспортна задача є частинним випадком задачі лінійного програмування, для неї мають силу всі загальні означення останньої. Зокрема, зауважене відноситься також і до допустимого базисного розв'язку (*ДБР*), як невиродженого, так і виродженого.

Послідовність комунікацій, серед яких немає однакових, вигляду:

$$P_{i_1}Q_{j_1},\ P_{i_2}Q_{j_1},\ P_{i_2}Q_{j_2},\ldots,P_{i_s}Q_{j_s},$$

називається *маршрутом*, що зв'язує пункти P_{i_1} і Q_{j_s}.

Маршрут, до якого додана комунікація $P_{i_1}Q_{j_s}$, називається *замкненим маршрутом* (*циклом*).

Комунікація P_iQ_j називається *основною комунікацією* розв'язку $\boldsymbol{x}$, якщо відповідна їй компонента розв'язку $x_{ij}>0$.

Подібні означення мають місце і для клітинок транспортної таблиці.

<u>*Властивості транспортної задачі*</u>.

1. Збалансована транспортна задача завжди допустима і має оптимальний розв'язок.
2. Ранг матриці $\boldsymbol{A}$ обмежень транспортної задачі дорівнює $m+n-1$, внаслідок чого допустимий базисний розв'язок задачі містить не більше $m+n-1$ ненульових перевезень x_{ij}.
3. Якщо в транспортній задачі всі числа $a_i,\ i=1,\ldots,m,\ b_j,\ j=1,\ldots,n,$ — цілі, то хоча б один оптимальний розв'язок задачі — цілочисельний.

<u>*Основні теореми*</u>.

1. *Розв'язок транспортної задачі базисний, якщо з його основних комунікацій неможливо скласти замкнений маршрут* (*цикл*).
2. *ДБР* $\boldsymbol{x}=(x_{ij},\ i=1,\ldots,m,\ j=1,\ldots,n)$ *оптимальний тоді і тільки тоді, коли існують потенціали* u_i, v_j *такі, що*

$v_j - u_i = c_{ij}$, *якщо* x_{ij} – базисне перевезення,
$v_j - u_i \leq c_{ij}$, *якщо* x_{ij} – небазисне перевезення.

Методи пошуку вихідного ДБР.

Метод північно-західного кута.

Метод складається з однотипних кроків, тому його формальний виклад дамо лише для 1-го кроку. Заповнюємо північно-західну клітинку таблиці, покладаючи $x_{11} = \min\{a_1, b_1\}$. Можливі три випадки:

1. $a_1 < b_1$, тоді $x_{11} = a_1$ і викреслюється 1-й рядок таблиці;
2. $a_1 > b_1$, тоді $x_{11} = b_1$ і викреслюється 1-й стовпець таблиці;
3. $a_1 = b_1$, тоді $x_{11} = a_1 = b_1$ і викреслюється як 1-й рядок, так і 1-й стовпець. В останньому випадку в одну з викреслених клітинок заноситься нульове базисне перевезення (відповідний вихідний допустимий базисний розв'язок буде виродженим).

В усіх випадках після заповнення базисної клітинки об'єми запасів a_1 і потреб b_1 зменшують на величину, що дорівнює x_{11}. Кінець кроку.

На кожному з наступних кроків розглядаються лише невикреслені клітини транспортної таблиці, тобто викреслені рядки і стовпці ігноруються.

Метод мінімального елемента.

Метод відрізняється від попереднього тим, що замість північно-західної клітинки на кожному кроці вибирається клітинка з найменшим значенням транспортних витрат c_{ij}.

Метод викреслювання.

Метод застосовується при побудові циклу. На кожному кроці методу в транспортній таблиці викреслюється або рядок, або стовпець, які в подальшому ігноруються. Викреслювати належить рядки (стовпці), які мають не більше однієї базисної клітинки. Невикреслені клітинки транспортної таблиці утворюють цикл.

Алгоритм методу потенціалів.

1. Знаходиться початковий допустимий базисний розв'язок (*ДБР*), наприклад, за допомогою одного із розглянутих вище методів.

2. Надалі метод потенціалів складається з однотипних кроків, на кожному з яких:

   i) Обчислюються потенціали рядків u_i, $i=1,...,m$, і стовпців v_j, $j=1,...,n$, транспортної таблиці як розв'язок системи $v_j - u_i = c_{ij}$, де i та j приймають такі значення, що клітинки (i,j) — базисні.

   ii) Обчислюються оцінки змінних x_{ij} для всіх небазисних клітинок (i,j) за формулою $\Delta_{ij} = c_{ij} - v_j + u_i$ (оцінки базисних змінних — нульові).

   iii) Знайдені оцінки Δ_{ij} перевіряються на невід'ємність. Якщо всі $\Delta_{ij} \geq 0$, $i=1,\dots,m$, $j=1,\dots,n$, то поточний *ДБР* оптимальний. Інакше переходять до поліпшення *ДБР* (пункт iv)).

   iv) Визначають клітинку (k,l) з мінімальною від'ємною оцінкою, і приєднують її до сукупності базисних. Знаходять цикл (наприклад, методом викреслювання). Поділяють цикл на додатний і від'ємний півцикли, послідовно позначаючи клітинки – вершини циклу знаками «+» і «–«, починаючи з клітинки (k,l), яку першою відносять до додатного півциклу, наступну за нею — до від'ємного, третю — до додатного і т. д. Серед клітинок від'ємного півциклу визначають клітинку (s,r) з мінімальною величиною перевезення x_{ij} (якщо таких клітинок кілька, то вибирають тільки одну з них). Покладають $\theta = x_{sr}$. Збільшують на значення θ перевезення x_{ij} в клітинках додатного півциклу і зменшують їх на те ж значення в клітинках від'ємного півциклу. В результаті здійснення вказаних процедур клітинка (k,l) вводиться до сукупності базисних, а клітинка (s,r) перестає бути базисною (на ній розривають цикл). Кінець кроку.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого транспортна задача розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом потенціалів транспортні задачі, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі:

1)

a\b	*25*	*40*	*50*	*35*	*45*
20	*7*	*3*	*4*	*8*	*6*
60	*5*	*7*	*2*	*3*	*5*
45	*1*	*4*	*5*	*2*	*6*
70	*3*	*4*	*2*	*7*	*8*

2)

a\b	*35*	*30*	*50*	*25*	*65*
50	*8*	*6*	*7*	*3*	*4*
50	*7*	*4*	*9*	*3*	*4*
55	*6*	*1*	*4*	*5*	*2*
50	*7*	*8*	*3*	*4*	*2*

3)

a\b	*10*	*40*	*20*	*60*	*20*
30	*5*	*1*	*5*	*2*	*4*
70	*5*	*7*	*6*	*3*	*2*
25	*1*	*5*	*4*	*2*	*6*
25	*1*	*6*	*3*	*3*	*5*

4)

a\b	*70*	*40*	*30*	*60*	*50*
20	*6*	*1*	*7*	*3*	*3*
90	*7*	*4*	*4*	*8*	*4*
80	*8*	*2*	*3*	*5*	*7*
60	*3*	*4*	*2*	*8*	*5*

5)

a\b	*30*	*90*	*80*	*20*	*30*
95	*2*	*8*	*4*	*6*	*3*
55	*3*	*2*	*5*	*2*	*6*
40	*6*	*5*	*8*	*7*	*4*
60	*3*	*4*	*4*	*2*	*1*

6)

a\b	*10*	*30*	*25*	*15*	*20*
20	*9*	*1*	*5*	*7*	*1*
15	*2*	*8*	*4*	*8*	*1*
45	*2*	*3*	*2*	*8*	*5*
20	*6*	*1*	*3*	*4*	*7*

7)

a\b	*13*	*13*	*13*	*13*	*28*
28	*8*	*4*	*6*	*3*	*1*
13	*9*	*3*	*8*	*5*	*7*
19	*7*	*3*	*5*	*9*	*8*
20	*2*	*1*	*4*	*5*	*7*

8)

a\b	*11*	*13*	*26*	*10*	*10*
24	*9*	*1*	*3*	*2*	*7*
12	*6*	*9*	*4*	*1*	*5*
18	*9*	*1*	*2*	*8*	*5*
16	*3*	*3*	*9*	*6*	*8*

9)

a\b	*10*	*35*	*15*	*25*	*35*
30	*7*	*3*	*1*	*5*	*4*
25	*7*	*5*	*8*	*3*	*2*
45	*6*	*4*	*8*	*3*	*2*
20	*3*	*1*	*7*	*6*	*2*

10)

a\b	*30*	*80*	*65*	*35*	*40*
60	*8*	*2*	*4*	*9*	*1*
55	*7*	*5*	*5*	*3*	*6*
85	*9*	*4*	*6*	*2*	*7*
50	*5*	*3*	*2*	*6*	*4*

Відповіді:

1) $L(\boldsymbol{x}^*) = 575$. **2)** $L(\boldsymbol{x}^*) = 710$. **3)** $L(\boldsymbol{x}^*) = 360$. **4)** $L(\boldsymbol{x}^*) = 910$. **5)** $L(\boldsymbol{x}^*) = 750$.
6) $L(\boldsymbol{x}^*) = 200$. **7)** $L(\boldsymbol{x}^*) = 209$. **8)** $L(\boldsymbol{x}^*) = 184$. **9)** $L(\boldsymbol{x}^*) = 285$. **10)** $L(\boldsymbol{x}^*) = 785$.

Лабораторна робота 6.

Транспортна задача з обмеженими пропускними спроможностями. Метод потенціалів

Постановка транспортної задачі з обмеженими пропускними спроможностями (ТЗО).

В пункті P_i ($i=1,...,m$) виробляється a_i одиниць деякого продукту, а в пункті Q_j ($j=1,...,n$) споживається b_j одиниць того ж продукту. Транспортні витрати на перевезення одиниці продукту з пункту P_i в пункт Q_j становлять c_{ij} одиниць. Комунікація $P_i\,Q_j$ має пропускну спроможність r_{ij}. Вважаючи, що сумарний об'єм виробництва дорівнює сумарному об'єму споживання, потрібно скласти план перевезень продукту, що мінімізує сумарні транспортні витрати.

Математична модель *ТЗО* має вигляд:

$$L(\boldsymbol{x}) = c_{11}x_{11} + ... + c_{1n}x_{1n} + ... + c_{m1}x_{m1} + ... + c_{mn}x_{mn} \to \min,$$

$$x_{i1} + ... + x_{in} = a_i,\ i=1,\dots,m,$$

$$x_{1j} + ... + x_{mj} = b_j,\ j=1,\dots,n,$$

$$0 \le x_{ij} \le r_{ij},\ i=1,\dots,m,\ j=1,\dots,n,$$

$$a_1 + ... + a_m = b_1 + ... + b_n.$$

Остання умова визначає *збалансовану ТЗО*.

Окрім векторів перевезень $\boldsymbol{x}$, виробництва-споживання $\boldsymbol{b}$, комунікацій $\boldsymbol{A}_{ij}$ та транспортних витрат $\boldsymbol{c}$, означення яких наведені при формулюванні звичайної збалансованої транспортної задачі (див. лаб. роб. 5), уведемо вектор обмежень пропускних спроможностей комунікацій

$$\boldsymbol{r} = (r_{11},\dots,r_{1n},\dots,r_{m1},\dots,r_{mn}).$$

Поряд з векторами $\boldsymbol{x}$, $\boldsymbol{c}$, $\boldsymbol{r}$ будемо також розглядати і матриці $\boldsymbol{X} = \|x_{ij}\|$, $\boldsymbol{C} = \|c_{ij}\|$, $\boldsymbol{R} = \|r_{ij}\|$, $i=1,...,m,\ j=1,...,n.$

Основні означення.

Допустимий розв'язок *ТЗО* $\boldsymbol{x} = \|x_{ij}\|$, $i=1,\dots,m,\ j=1,\dots,n,$ базисний, якщо його перевезенням x_{ij}, які задовольняють умову $0 < x_{ij} < r_{ij}$, відповідає система лінійно незалежних векторів комунікацій $\boldsymbol{A}_{ij}$.

Послідовність різних комунікацій

$$P_{i_1}Q_{j_1},\ P_{i_2}Q_{j_1},\ P_{i_2}Q_{j_2},\ \dots,\ P_{i_s}Q_{j_s},$$

називається *маршрутом*, що зв'язує пункти P_{i_1} і Q_{j_s}.

Маршрут, до якого додана комунікація $P_{i_1}Q_{j_s}$, називається *замкненим маршрутом* (*циклом*).

Комунікація $P_i\,Q_j$ називається *основною комунікацією* розв'язку $\boldsymbol{x}$, якщо для цього розв'язку $0 < x_{ij} < r_{ij}$.

Аналогічні означення справедливі і для клітинок транспортної таблиці.

Властивості ТЗО та основні теореми.

1. Ранг складеної з векторів $\boldsymbol{A}_{ij}$ матриці $\boldsymbol{A}$ обмежень транспортної задачі дорівнює $m+n-1$, звідки випливає, що допустимий базисний розв'язок задачі (якщо він існує) має не більше $m+n-1$ перевезень x_{ij}, які задовольняють умову $0 < x_{ij} < r_{ij}$.

2. *Розв'язок ТЗО є базисним, якщо з його основних комунікацій неможливо скласти замкнений маршрут* (*цикл*).

3. *ДБР* $\boldsymbol{x}=(x_{ij},\ i=1,\dots,m,\ j=1,\dots,n)$ *оптимальний тоді і тільки тоді, коли існують потенціали* u_i, v_j *такі, що*

$v_j - u_i = c_{ij}$, *якщо* x_{ij} – базисне перевезення,
$v_j - u_i \le c_{ij}$, *якщо* $x_{ij} = 0$, небазисне перевезення,
$v_j - u_i \ge c_{ij}$, *якщо* $x_{ij} = r_{ij}$, небазисне перевезення.

Метод пошуку вихідного ДБР.

Вихідний *ДБР ТЗО* (на відміну від *ТЗ* без обмежень), якщо він існує, знаходиться суттєво більш складно. Його пошук проводиться в два етапи.

Перший етап (попередній) нагадує метод мінімального елемента в *ТЗ* без обмежень і в загальному випадку не дає допустимого розв'язку.

Другий етап містить ряд ітерацій методу потенціалів, який застосовується до деякої *розширеної ТЗО*, побудованої за результатами першого етапу.

Вихідний ДБР ТЗО. I етап.

На множині невикреслених клітинок транспортної таблиці знаходять клітинку (i_1, j_1) з мінімальними транспортними витратами $c_{i_1 j_1}$.

Покладають $x_{i_1 j_1} = \min\{a_{i_1}, b_{j_1}, r_{i_1 j_1}\}$.

Якщо $x_{i_1 j_1} = a_{i_1}$, то викреслюють i_1-й рядок транспортної таблиці.

Якщо $x_{i_1 j_1} = b_{j_1}$, то викреслюють j_1-й стовпець транспортної таблиці.

Якщо $x_{i_1 j_1} = r_{i_1 j_1}$ ($r_{i_1 j_1} < a_{i_1}$, $r_{i_1 j_1} < b_{j_1}$), то викреслюють тільки клітинку (i_1, j_1) транспортної таблиці.

Якщо $x_{i_1 j_1} = a_{i_1} = b_{j_1}$, то у довільну, невикреслену на попередніх кроках клітинку, яка лежить або в i_1-у рядку або в j_1-у стовпці, заносять нульове базисне перевезення.

Після заповнення клітинки у всіх випадках величини a_{i_1} та b_{j_1} зменшуються на $x_{i_1 j_1}$.

Указані дії виконують доти, поки не будуть викреслені всі клітинки транспортної таблиці.

Вихідний ДБР ТЗО. II етап.

Нехай $\boldsymbol{X} = \|x_{ij}\|$, $i=1,\dots,m,\ j=1,\dots,n,$ — матриця перевезень, побудована на першому етапі. Покладемо

$$x_{i,n+1} = a_i - (x_{i1} + \dots + x_{in}),\ i=1,\dots,m,$$

$$x_{m+1,j} = b_j - (x_{1j} + \dots + x_{mj}),\ j=1,\dots,n.$$

Позначимо

$$\omega = x_{1,n+1} + ... + x_{m,n+1} = x_{m+1,1} + ... + x_{m+1,n}.$$

Якщо $\omega = 0$, то $\boldsymbol{x}$ — вихідний *ДБР*.

Якщо $\omega > 0$, то вихідна *ТЗ* розширюється за рахунок фіктивних пунктів виробництва P_{m+1} та споживання Q_{n+1} з $a_{m+1} = b_{n+1} = \omega$, де

$$c_{i,n+1} = M, \; r_{i,n+1} = \infty, \; i=1,...,m,$$

$$c_{m+1,j} = M, \; r_{m+1,j} = \infty, \; j=1,...,n,$$

$$c_{m+1,n+1} = 0, \; r_{m+1,n+1} = \infty,$$

M — досить велике додатне число, ∞ — нескінченність.

Нерозподілені на першому етапі залишки продукту (як по об'ємах виробництва, так і по об'ємах споживання) розподіляються по фіктивних пунктах P_{m+1} та Q_{n+1}. Відповідні заповнені клітинки вважаються базисними (оскільки пропускні спроможності відповідних їм комунікацій необмежені) і приєднуються до сукупності базисних клітинок заповнених на першому етапі. Якщо після цього загальна кількість базисних клітинок не рівна $(m+1)+(n+1)-1$, то множину базисних клітинок доповнюють до цього числа за рахунок незаповнених клітинок або заповнених до пропускної спроможності, але так, щоб розширена множина базисних клітинок не містила циклів.

Розширена ТЗ розв'язується методом потенціалів.

Якщо в оптимальному розв'язку $x^*_{m+1,n+1} = \omega$, то, відкидаючи фіктивні пункти, отримаємо вихідний *ДБР*, інакше *ТЗО* не має розв'язків.

Потенціали.

Потенціали рядків u_i, $i=1,...,m$, та стовпців v_j, $j=1,...,n$, визначаються як розв'язок системи $v_j - u_i = c_{ij}$, де i та j приймають такі значення, що клітинки (i,j) — базисні.

Вказана система містить $m+n-1$ рівнянь (за числом базисних клітинок) та $m+n$ змінних. Тому одна із змінних задається довільно (наприклад, $u_1 = 0$).

Оцінки.

Оцінки Δ_{ij} змінних x_{ij} для всіх небазисних клітинок обчислюються за формулою $\Delta_{ij} = c_{ij} - v_j + u_i$ (оцінки базисних змінних — нульові).

Поточний *ДБР* $\boldsymbol{X} = \|x_{ij}\|$, $i=1,...,m$, $j=1,...,n$, оптимальний, коли

$\Delta_{ij} = 0$, *якщо* x_{ij} – базисне перевезення,

$\Delta_{ij} \geq 0$, *якщо* $x_{ij} = 0$, небазисне перевезення,

$\Delta_{ij} \leq 0$, *якщо* $x_{ij} = r_{ij}$, небазисне перевезення.

Цикл. Метод викреслювання.

При побудові циклу застосовується метод викреслювання. На кожному кроці методу в транспортній таблиці викреслюється або рядок, або стовпець, які в подальшому ігноруються. Закресленню підлягають рядки (стовпці), які містять не більше однієї базисної клітинки. Невикреслені клітинки транспортної таблиці утворюють цикл.

Новий ДБР.

Серед всіх клітинок (i,j), для яких не виконується критерій оптимальності, обирають клітинку з найбільшим модулем оцінки Δ_{ij}. Позначимо таку клітинку через (i_0,j_0).

Нехай клітинка (i_0,j_0), для якої $x_{i_0 j_0} = 0$, $\Delta_{i_0 j_0} < 0$, приєднується до сукупності базисних клітинок.

Знаходиться цикл, що утворюється цими клітинками. Цикл розбивається на додатний (C^+) та від'ємний (C^-) півцикли, клітинки яких чергуються одна з одною, причому клітинка (i_0,j_0) відноситься до додатного півциклу. Обчислюються величини

$$\theta_1 = \min\{x_{ij}\} \qquad \text{по } C^-,$$

$$\theta_2 = \min\{r_{ij} - x_{ij}\} \qquad \text{по } C^+,$$

$$\theta = \min\{\theta_1, \theta_2\}.$$

Збільшують на значення θ перевезення x_{ij} в клітинках півциклу C^+ і зменшують їх на те ж значення в клітинках C^-.

Для клітинки (i_0,j_0) такої, що $x_{i_0 j_0} = r_{i_0 j_0}$, $\Delta_{i_0 j_0} > 0$, відміна полягає в тому, що вона відноситься до від'ємного півциклу.

В результаті виконання вказаних процедур клітинка (i_0,j_0) вводиться до множини базисних, а клітинка, пов'язана з θ, стає небазисною. Якщо θ досягається на клітинці (i_0,j_0), то множина базисних клітинок не змінюється після перерозподілу перевезень вздовж циклу на сталу θ і новий *ДБР* матиме ту ж саму систему потенціалів і ті ж самі оцінки, що і попередній. Тому у цьому випадку після обчислення нового *ДБР* безпосередньо переходять до перевірки його на оптимальність.

Алгоритм методу потенціалів.

1. Будується вихідний *ДБР*.

2. Далі метод потенціалів складається з однотипних кроків, на кожному з яких:

i) Обчислюються потенціали u_i, $i=1,...,m$, та v_j, $j=1,...,n$;

ii) Обчислюються оцінки Δ_{ij} змінних x_{ij}, $i=1,...,m$, $j=1,...,n$;

iii) Аналізуються знайдені оцінки Δ_{ij}. Якщо $\Delta_{ij} \geq 0$ для $x_{ij} = 0$ та $\Delta_{ij} \leq 0$ для $x_{ij} = r_{ij}$, то поточний *ДБР* оптимальний. В іншому випадку переходять до покращання поточного *ДБР* (п. п. iv) та v)).

iv) Будується цикл.

v) Знаходиться новий *ДБР*.

Крок закінчений. Перехід до пункту i).

Програмне забезпечення.

Навчаючий модуль, за допомогою якого транспортна задача з обмеженими пропускними спроможностями комунікацій розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Завдання.

Розв'язати методом потенціалів транспортні задачі, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі:

1) $\boldsymbol{C} = \begin{Vmatrix} 6 & 2 & 5 & 3 \\ 3 & 6 & 9 & 7 \\ 7 & 1 & 6 & 5 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 16 & 34 & 15 & 19 \\ 13 & 17 & 12 & 4 \\ 17 & 19 & 2 & 4 \end{Vmatrix}$,

$\boldsymbol{a} = (74, 33, 19)$, $\boldsymbol{b} = (28, 70, 15, 13)$;

2) $\boldsymbol{C} = \begin{Vmatrix} 2 & 1 & 5 & 5 \\ 2 & 3 & 7 & 4 \\ 9 & 5 & 7 & 1 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 13 & 10 & 31 & 15 \\ 20 & 28 & 7 & 22 \\ 19 & 6 & 15 & 8 \end{Vmatrix}$,

$\boldsymbol{a} = (63, 72, 17)$, $\boldsymbol{b} = (33, 40, 43, 36)$;

3) $\boldsymbol{C} = \begin{Vmatrix} 3 & 6 & 6 & 4 \\ 3 & 2 & 2 & 6 \\ 1 & 4 & 10 & 7 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 5 & 8 & 6 & 4 \\ 29 & 24 & 15 & 17 \\ 20 & 5 & 10 & 6 \end{Vmatrix}$,

$\boldsymbol{a} = (13, 79, 34)$, $\boldsymbol{b} = (49, 30, 22, 25)$;

4) $\boldsymbol{C} = \begin{Vmatrix} 5 & 1 & 2 & 7 \\ 6 & 1 & 8 & 7 \\ 2 & 9 & 4 & 10 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 4 & 12 & 5 & 5 \\ 20 & 10 & 37 & 10 \\ 25 & 10 & 10 & 15 \end{Vmatrix}$,

$\boldsymbol{a} = (20, 76, 54)$, $\boldsymbol{b} = (40, 30, 52, 28)$;

5) $\boldsymbol{C} = \begin{Vmatrix} 9 & 6 & 6 & 1 \\ 8 & 10 & 9 & 2 \\ 5 & 9 & 2 & 6 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 15 & 10 & 34 & 7 \\ 31 & 23 & 20 & 11 \\ 11 & 15 & 4 & 10 \end{Vmatrix}$,

$\boldsymbol{a} = (64, 75, 21)$, $\boldsymbol{b} = (57, 33, 44, 26)$;

6) $\boldsymbol{C} = \begin{Vmatrix} 6 & 4 & 8 & 5 \\ 2 & 8 & 3 & 2 \\ 1 & 7 & 2 & 8 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 20 & 5 & 15 & 12 \\ 45 & 20 & 21 & 19 \\ 8 & 20 & 12 & 4 \end{Vmatrix}$,

$\boldsymbol{a} = (42, 99, 27)$, $\boldsymbol{b} = (68, 40, 30, 30)$;

7) $\boldsymbol{C} = \begin{Vmatrix} 5 & 6 & 10 & 3 \\ 6 & 4 & 7 & 2 \\ 8 & 8 & 3 & 7 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 37 & 20 & 7 & 21 \\ 5 & 26 & 7 & 11 \\ 5 & 20 & 25 & 10 \end{Vmatrix}$,

$\boldsymbol{a} = (78, 37, 53)$, $\boldsymbol{b} = (38, 60, 30, 40)$;

8) $\boldsymbol{C} = \begin{Vmatrix} 6 & 1 & 9 & 3 \\ 9 & 2 & 9 & 7 \\ 3 & 2 & 10 & 6 \end{Vmatrix}$, $\boldsymbol{R} = \begin{Vmatrix} 16 & 30 & 21 & 30 \\ 18 & 4 & 5 & 11 \\ 30 & 4 & 29 & 15 \end{Vmatrix}$,

$\boldsymbol{a} = (77, 24, 70)$, $\boldsymbol{b} = (56, 30, 40, 45)$;

9) $\quad \boldsymbol{C} = \begin{Vmatrix} 8 & 4 & 6 & 8 \\ 8 & 9 & 9 & 6 \\ 9 & 10 & 4 & 7 \end{Vmatrix}, \quad \boldsymbol{R} = \begin{Vmatrix} 12 & 20 & 30 & 20 \\ 25 & 4 & 3 & 2 \\ 33 & 10 & 30 & 10 \end{Vmatrix},$

$\boldsymbol{a} = (72, 29, 68)$, $\boldsymbol{b} = (65, 24, 50, 30)$;

10) $\quad \boldsymbol{C} = \begin{Vmatrix} 1 & 3 & 8 & 7 \\ 9 & 4 & 5 & 10 \\ 9 & 7 & 2 & 8 \end{Vmatrix}, \quad \boldsymbol{R} = \begin{Vmatrix} 10 & 26 & 23 & 8 \\ 6 & 18 & 30 & 5 \\ 9 & 2 & 25 & 3 \end{Vmatrix},$

$\boldsymbol{a} = (53, 45, 38)$, $\boldsymbol{b} = (21, 30, 75, 10)$.

Відповіді:

1) $L(\boldsymbol{x}^*) = 444$. **2)** $L(\boldsymbol{x}^*) = 538$. **3)** $L(\boldsymbol{x}^*) = 413$. **4)** $L(\boldsymbol{x}^*) = 837$. **5)** $L(\boldsymbol{x}^*) = 1091$.
6) $L(\boldsymbol{x}^*) = 700$. **7)** $L(\boldsymbol{x}^*) = 800$. **8)** $L(\boldsymbol{x}^*) = 885$. **9)** $L(\boldsymbol{x}^*) = 1134$. **10)** $L(\boldsymbol{x}^*) = 649$.

Лабораторна робота 7.

Задача про найкоротший шлях на мережі. Метод Мінті

Постановка задачі про найкоротший шлях на мережі.

На мережі, що задається графом $(\boldsymbol{I}, \boldsymbol{U})$, де $\boldsymbol{I}$ — множина вершин, $\boldsymbol{U}$ — множина дуг, з визначеною на ній функцією вартості c_{ij} ((i,j) — дуга з $\boldsymbol{U}$), для фіксованих i_1 та i_s знайти шлях

$$L = ((i_1,i_2),(i_2,i_3),\dots,(i_{s-1},i_s))$$

із вершини i_1 у вершину i_s, довжина якого

$$C(L) = c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_{s-1} i_s}$$

найменша.

Алгоритм методу Мінті.

Методом Мінті розв'язується задача побудови дерева найкоротших шляхів на мережі з коренем у фіксованій вершині i_1.

Алгоритм складається із скінченного числа кроків, на кожному з яких позначаються вершини мережі і виділяються дуги, що належать дереву найкоротших шляхів.

Нехай $\boldsymbol{P}_r$ — множина вершин, позначених на кроці r, а $\boldsymbol{I}_r$ — множина вершин, позначених за r кроків.

КРОК 0. Корінь дерева (вершина i_1) позначається сталою $h_1 = 0$; $i_1(h_1) = i_1(0)$. Після нульового кроку $\boldsymbol{P}_0 = \{i_1(0)\}$, $\boldsymbol{I}_0 = \{i_1(0)\}$.

Нехай виконано r кроків, за які побудована множина

$$\boldsymbol{I}_r = \{i_1(0),\dots,i_k(h_k),\dots\}$$

позначених вершин $i_k(h_k)$, кожній з яких поставлене у відповідність число h_k (чисельно рівне довжині найкоротшого шляху із вершини i_1 у вершину i_k).

КРОК r+1. Будується розріз мережі, який породжується множиною позначених вершин $\boldsymbol{I}_r$, і визначається множина $\boldsymbol{J}_r = \{\dots,i_m,\dots\}$ непозначених вершин i_m мережі, в які заходять дуги розрізу. Для кожної дуги (i_k, i_m) розрізу обчислюють суму $h_k + c_{i_k i_m}$ і позначають ті з дуг, для яких ця сума мінімальна. Потім виділяють позначені дуги так, щоб в кожну непозначену вершину множини $\boldsymbol{J}_r$, в яку заходять позначені дуги розрізу, заходила б тільки одна виділена дуга. Після виділення дуг позначають вершини — кінці виділених дуг. Величина позначки рівна мінімальній із сум $h_k + c_{i_k i_m}$, обчислених для всіх дуг розрізу. Об'єднуючи множину $\boldsymbol{I}_r$ з множиною $\boldsymbol{P}_{r+1}$ вершин, позначених на ($r+1$)-у кроці, отримують множину $\boldsymbol{I}_{r+1}$ вершин, позначених за ($r+1$) кроків. Переходять до наступного кроку, якщо існує розріз, що породжується множиною $\boldsymbol{I}_{r+1}$.

Указаний процес продовжують доти, поки можливе розширення множини позначених вершин.

Якщо деяка вершина i_n мережі залишилась непозначеною після закінчення процедури Мінті, то шляху, що починається у вершині i_1 і закінчується у вершині i_n, не існує.

<u>*Програмне забезпечення.*</u>

Навчаючий модуль, за допомогою якого задача про найкоротший шлях на мережі розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

<u>*Завдання.*</u>

Розв'язати методом Мінті задачі про найкоротший шлях на мережі, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі, в кожній з яких мережа задається числом вершин N і матрицею $\boldsymbol{L}$, що описує множину дуг мережі (кожний стовпець цієї матриці відповідає існуючій дузі мережі, при цьому, перший елемент стовпця є початок дуги, другий — кінець дуги, третій — довжина дуги):

1) $N = 10$, $\boldsymbol{L} =$

$$\begin{Vmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 & 5 & 6 & 6 & 7 & 7 & 8 & 9 \\ 2 & 3 & 4 & 3 & 5 & 6 & 4 & 6 & 6 & 7 & 8 & 9 & 10 & 5 & 9 & 6 & 9 & 10 & 10 \\ 5 & 10 & 20 & 5 & 20 & 17 & 12 & 17 & 5 & 4 & 10 & 6 & 20 & 7 & 5 & 1 & 5 & 11 & 15 \end{Vmatrix}$$

2) $N = 10$, $\boldsymbol{L} =$

$$\begin{Vmatrix} 1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 7 & 8 & 8 & 9 & 9 \\ 2 & 3 & 4 & 5 & 2 & 7 & 8 & 3 & 7 & 9 & 10 & 5 & 10 & 6 & 8 & 9 & 4 & 9 & 6 & 10 \\ 20 & 12 & 9 & 20 & 12 & 20 & 7 & 20 & 2 & 10 & 18 & 10 & 18 & 1 & 7 & 16 & 18 & 17 & 20 & 8 \end{Vmatrix}$$

3) $N = 10$, $\boldsymbol{L} =$

$$\begin{Vmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 4 & 4 & 4 & 4 & 5 & 5 & 6 & 7 & 7 & 7 & 8 & 8 & 9 \\ 2 & 3 & 4 & 3 & 5 & 6 & 5 & 3 & 5 & 7 & 9 & 6 & 7 & 8 & 6 & 8 & 9 & 9 & 10 & 10 \\ 18 & 20 & 19 & 8 & 2 & 11 & 2 & 7 & 1 & 6 & 12 & 9 & 5 & 12 & 6 & 15 & 6 & 16 & 2 & 4 \end{Vmatrix}$$

4) $N = 10$, $\boldsymbol{L} =$

$$\begin{Vmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 4 & 5 & 5 & 5 & 6 & 6 & 7 & 8 & 9 \\ 2 & 3 & 4 & 4 & 5 & 6 & 2 & 4 & 5 & 7 & 8 & 7 & 9 & 10 & 5 & 9 & 8 & 10 & 10 \\ 15 & 2 & 20 & 9 & 20 & 17 & 13 & 12 & 15 & 8 & 5 & 4 & 3 & 7 & 7 & 11 & 5 & 6 & 4 \end{Vmatrix}$$

5) $N = 10$, $\boldsymbol{L} =$

$$\begin{Vmatrix} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 6 & 2 & 6 & 7 & 8 & 5 & 9 & 6 & 9 & 7 & 9 & 9 & 10 & 7 & 10 & 8 \\ 13 & 2 & 8 & 18 & 2 & 11 & 13 & 20 & 20 & 10 & 20 & 4 & 20 & 20 & 20 & 11 & 20 & 11 & 14 & 5 \end{Vmatrix}$$

6) $N = 10$, $\mathbf{L} =$

$$\left\| \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 & 9 \\ 2 & 3 & 4 & 5 & 3 & 6 & 4 & 6 & 7 & 10 & 7 & 8 & 4 & 8 & 10 & 9 & 10 & 7 & 9 & 10 \\ 10 & 18 & 20 & 20 & 8 & 15 & 9 & 7 & 17 & 12 & 8 & 10 & 7 & 15 & 5 & 10 & 19 & 20 & 10 & 2 \end{matrix} \right\|$$

Лабораторна робота 8.

Задача про максимальний потік на мережі. Метод Форда-Фалкерсона

Постановка задачі про максимальний потік на мережі.

На мережі, що задається графом $(\mathbf{I}, \mathbf{U})$, де $\mathbf{I}$ — множина вершин, $\mathbf{U}$ — множина дуг, з визначеною на ній функцією пропускних спроможностей r_{ij} ((i,j) — дуга з $\mathbf{U}$), зафіксовані дві вершини — i_1 та i_n. Вершина i_1 (*джерело*) має інтенсивність d, вершина i_n (*стік*) — інтенсивність $-d$, всі інші вершини *нейтральні*. Потрібно знайти максимальну інтенсивність джерела d, при якій мережа допускає потік. Потік, що відповідає такому максимальному значенню інтенсивності d^*, називається *максимальним потоком*, а саме значення d^* — *величиною цього потоку*.

Алгоритм Форда-Фалкерсона застосовується для побудови максимального потоку на мережі із заданої початкової вершини-джерела в задану кінцеву вершину-стік.

Будемо вважати, що вершиною-джерелом є *1*-а вершина, вершиною-стоком — вершина з номером n.

Вхідні дані.

Для роботи алгоритму необхідно задати наступну інформацію:

1. Число вершин мережі.

2. *Матрицю суміжностей* $\mathbf{C}$, елементи якої c_{ij} визначаються співвідношеннями:

$c_{ij} = 1$, якщо існує дуга (i,j);
$c_{ij} = 0$, якщо дуга (i,j) відсутня.

3. Величини r_{ij} *пропускних спроможностей* дуг (i,j).

4. *Початковий потік* на мережі, тобто величини x_{ij}, що задовольняють умови:

a) $0 \le x_{ij} \le r_{ij}$;
b) Для довільної вершини (крім першої і останньої) потік, що входить у вершину, дорівнює потоку, що виходить з неї.

Виклад алгоритму Форда-Фалкерсона.

Алгоритм складається з послідовних ітерацій, які проводяться доти, поки не буде виконуватись ознака оптимальності.

В кожній ітерації можна виділити два етапи.

Етап 1 (*етап виставлення позначок*).

1. В кожний момент часу кожна вершина може знаходитись в одному з трьох положень:

а) не позначена;
б) позначена, але не проглянута;

в) позначена і проглянута.

Загальний вигляд позначки j-ї вершини: $[i+, \theta_j]$, або $[i-, \theta_j]$, де i — номер деякої позначеної вершини, при проглядані якої була позначена вершина j. Вершина 1 завжди позначена та має позначку $[1+, \theta_1]$, де θ_1 рівне $+\infty$ (нескінченності).

2. Проглядання довільної позначеної вершини j полягає в наступному:

а) довільна непозначена вершина k, для якої існує дуга (j, k) і має місце нерівність $x_{jk} < r_{jk}$, отримує позначку вигляду $[j+, \theta_k]$, де

$$\theta_k = \min \{\theta_j, r_{jk} - x_{jk}\};$$

б) довільна непозначена вершина k, для якої існує дуга (k, j) і має місце умова $x_{kj} > 0$, отримує позначку вигляду $[j-, \theta_k]$, де

$$\theta_k = \min \{\theta_j, x_{kj}\}.$$

3. Процес виставлення позначок закінчується в одному з двох випадків:

а) вершина з номером n позначена;

б) умова а) не виконується, але жодної вершини більше позначити не можна.

В першому випадку переходимо до етапу зміни потоку на мережі. В другому — до визначення мінімального розрізу мережі і максимального потоку.

Етап 2 (*етап зміни потоку*).

Потік в мережі змінюється на величину θ_n за таким правилом.

Якщо вершина n має позначку $[k+, \theta_n]$, де k — номер деякої вершини, то $x_{kn} = x_{kn} + \theta_n$. Якщо позначка має вигляд $[k-, \theta_n]$, то $x_{nk} = x_{nk} - \theta_n$. Переходимо до вершини з номером k. Якщо позначка вершини k має вигляд $[j+, \theta_k]$, то $x_{jk} = x_{jk} + \theta_n$; якщо вона дорівнює $[j-, \theta_k]$, то $x_{kj} = x_{kj} - \theta_n$. Подібні дії продовжуємо доти, поки не буде досягнута початкова вершина. В цьому випадку всі старі позначки витираємо і повертаємось до першого етапу.

В результаті роботи згідно алгоритму визначаються *мінімальний розріз мережі* і, як наслідок, *максимальний потік*.

Мінімальний розріз визначається сукупністю дуг, що виходять з множини позначених вершин і заходять у множину непозначених.

Величина максимального потоку рівна сумарній пропускній спроможності мінімального розрізу.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача про максимальний потік на мережі розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом Форда-Фалкерсона задачі про максимальний потік на мережі, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі, в кожній з яких мережа задається матрицями суміжностей ***C*** і пропускних спроможностей ***R***. У всіх задачах джерелом вважається вершина *1*, а стоком — вершина *6*.

1)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 10 & 18 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14 & 1 \\ 0 & 10 & 0 & 17 & 0 & 0 \\ 0 & 9 & 0 & 0 & 22 & 12 \\ 0 & 0 & 0 & 0 & 0 & 18 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix};$$

2)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 11 & 33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 4 & 0 \\ 0 & 5 & 0 & 15 & 23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 23 \\ 0 & 0 & 0 & 11 & 0 & 22 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix};$$

3)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 20 & 21 & 22 & 0 & 0 \\ 0 & 0 & 5 & 13 & 14 & 0 \\ 0 & 0 & 0 & 0 & 12 & 28 \\ 0 & 0 & 12 & 0 & 11 & 10 \\ 0 & 0 & 0 & 0 & 0 & 17 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix};$$

4)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 5 & 19 & 27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 17 & 15 \\ 0 & 23 & 0 & 0 & 7 & 8 \\ 0 & 5 & 12 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 28 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix};$$

5)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 24 & 33 & 6 & 12 & 0 \\ 0 & 0 & 10 & 15 & 7 & 0 \\ 0 & 0 & 0 & 11 & 15 & 17 \\ 0 & 0 & 0 & 0 & 12 & 26 \\ 0 & 0 & 0 & 0 & 0 & 25 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix};$$

6)

$$C = \begin{Vmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}, \qquad R = \begin{Vmatrix} 0 & 19 & 16 & 8 & 18 & 0 \\ 0 & 0 & 20 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 32 & 42 \\ 0 & 0 & 45 & 0 & 0 & 2 \\ 0 & 0 & 0 & 31 & 0 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{Vmatrix}.$$

Відповіді:

1) $d^* = 28$. **2)** $d^* = 44$. **3)** $d^* = 55$. **4)** $d^* = 51$. **5)** $d^* = 68$. **6)** $d^* = 61$.

Лабораторна робота 9.

Задача цілочисельного лінійного програмування. Метод Гоморі-1

Постановка цілочисельної задачі лінійного програмування.

Знайти вектор $\boldsymbol{x}=(x_1,\ldots,x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \ldots + c_n x_n \tag{9.1}$$

і задовольняє систему обмежень

$$\begin{array}{l} a_{11}x_1 + \ldots + a_{1n}\ x_n = a_{10}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ a_{m1}x_1 + \ldots + a_{mn}x_n = a_{m0}, \end{array} \tag{9.2}$$

$$x_j \geq 0,\ j=1,\ldots,n, \tag{9.3}$$

$$x_j \text{ — цілі, } j=1,\ldots,n. \tag{9.4}$$

Виклад методу Гоморі-1.

Метод Гоморі-1 є одним з методів відтинання, ідея яких така.

Розв'язується допоміжна *ЗЛП* (9.1)–(9.3), яку отримують з вихідної *ЦЗЛП* (9.1)–(9.4) відкиданням умови цілочисельності змінних (9.4). Якщо оптимальний розв'язок допоміжної *ЗЛП* — цілочисельний, то він буде і розв'язком вихідної *ЦЗЛП*. Якщо ж отриманий розв'язок допоміжної задачі не є цілочисельним, то від розв'язаної *ЗЛП* переходять до нової допоміжної *ЗЛП* приєднанням лінійного обмеження, яке задовольняють цілочисельні розв'язки вихідної *ЦЗЛП* і яке не задовольняє отриманий нецілочисельний розв'язок допоміжної *ЗЛП*. Згадане додаткове лінійне обмеження визначає деяку відтинаючу площину і називається *правильним відтином*. Приєднання нових правильних відтинів до початкової допоміжної *ЗЛП* здійснюється доти, поки на деякому кроці не буде отриманий цілочисельний розв'язок допоміжної задачі, який, очевидно, буде оптимальним розв'язком вихідної *ЦЗЛП*. В методі Гоморі-1 правильний відтин будується таким чином.

Нехай на останній ітерації симплекс-методу при розв'язуванні допоміжної *ЗЛП* непрямі обмеження цієї задачі набули вигляду:

$$x_i + a'_{i,m+1}x_{m+1} + \ldots + a'_{in}x_n = a'_{i0},\ i=1,\ldots,m,$$

і, таким чином, розв'язком допоміжної *ЗЛП* буде вектор

$$\boldsymbol{x} = (a'_{10},\ldots,\ a'_{m0}, 0,\ldots,0\,).$$

Нехай існує номер r такий, що a'_{r0} — дріб, і, як завжди, $\{z\}$ — дробова частина z. Тоді правильний відтин методу Гоморі-1 задається нерівністю:

$$\{\,a'_{r,m+1}\}\,x_{m+1} + \ldots + \{\,a'_{rn}\}\,x_n \geq \{\,a'_{r0}\}\,.$$

Алгоритм методу Гоморі-1.

1. Розв'язуємо допоміжну *ЗЛП* (9.1)–(9.3). Нехай $\boldsymbol{x}(0)$ — її оптимальний розв'язок. Якщо оптимальний розв'язок не існує, то вихідна *ЦЗЛП* також не має оптимального розв'язку.

2. Нехай на s-й ітерації розв'язана допоміжна *ЗЛП*, що має M обмежень і N змінних, $\boldsymbol{x}(s)$ — її оптимальний розв'язок.

Будемо вважати, що $\boldsymbol{x}(s)$ визначається канонічними обмеженнями останньої ітерації, тобто:

$$x_i \ + b_{i,M+1} x_{M+1} + \ldots + b_{iN} x_N = b_{i0}, \ \ i=1,\ldots,M,$$

Звідки

$$\boldsymbol{x}(s) = (\, b_{10}, \ldots, b_{M0}, 0, \ldots, 0\,).$$

3. Якщо b_{i0} $(i=1,\ldots,M)$ — цілі, то — кінець, $\boldsymbol{x}(s)$ є оптимальним розв'язком вихідної *ЦЗЛП*. Якщо існує хоча б одне i таке, що b_{i0} — дріб, то перехід до пункту 4.

4. Знаходимо $r = \min\{i\}$ по всіх i таких, що b_{i0} — дріб та будуємо додаткове обмеження

$$x_{N+1} - \{b_{r,M+1}\} x_{M+1} - \ldots - \{b_{rN}\} x_N = -\{b_{r0}\},$$

де $x_{N+1} \ge 0$ — додаткова змінна.

5. Розширюємо симплекс-таблицю за рахунок ($M+1$)-го рядка (додаткове обмеження) та ($N+1$)-го стовпця, що відповідає додатковій змінній x_{N+1}.

6. Розв'язуємо розширену таким чином *ЗЛП* двоїстим симплекс-методом і переходимо до пункту 2 з заміною s на $s+1$. Якщо при цьому на якій-небудь ітерації двоїстого симплекс-методу одна з додаткових змінних задачі повторно стає базисною, то виключаються з подальшого розгляду відповідні їй рядок і стовпець.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого повністю цілочисельна задача лінійного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом Гоморі-1 задачі цілочисельного лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються нижче, всі змінні невід'ємні та цілі.

1)
$$\begin{aligned} &5x_1 + 6x_2 + 6x_3 \to \min,\\ &2x_1 + 4x_2 \ge 10,\\ &3x_1 + 2x_2 + 2x_3 \ge 10; \end{aligned}$$

2)
$$\begin{aligned} &6x_1 + 4x_2 \to \min,\\ &2x_1 + x_2 \ge 3,\\ &x_1 - x_2 \le 1; \end{aligned}$$

3)
$$\begin{aligned} &6x_1 + 4x_2 \to \min,\\ &2x_1 + x_2 \ge 3,\\ &x_1 - 2x_2 \le 2,\\ &3x_1 + 2x_2 \ge 1; \end{aligned}$$

4)
$$\begin{aligned} &x_1 + x_2 \to \min,\\ &-x_1 - x_2 + x_5 = -1,\\ &-2x_1 + 2x_2 + x_3 = -2,\\ &-4x_1 + 2x_2 + x_4 = -1; \end{aligned}$$

5)
$$\begin{aligned} &6x_1 + 4x_2 \to \min,\\ &2x_1 + x_2 \ge 3,\\ &x_1 - x_2 \ge 1; \end{aligned}$$

6)
$$\begin{aligned} &2x_1 + 3x_2 \to \min,\\ &x_1 + 2x_2 \ge 16,\\ &2x_1 + x_2 \ge 16; \end{aligned}$$

7) $5x_1 - 3x_2 \to \max,$
$3x_1 + 2x_2 \geq 6,$
$2x_1 - 3x_2 \geq -6,$
$x_1 - x_2 \leq 4,$

8) $5x_2 + 7x_4 \to \min,$
$-10x_2 + x_3 + x_4 = -16,$
$x_1 - 3x_2 - 3x_4 = -12,$
$-6x_2 - 2x_4 + x_5 = -17;$

9) $-5x_4 - 7x_5 \to \max,$
$x_1 - x_4 - 2x_5 = -7,$
$-x_3 + 3x_4 - 6x_5 = -3,$
$x_2 - x_4 - 4x_5 = -11;$

10) $8x_1 + 2x_2 \to \max,$
$x_1 - 4x_2 + x_3 = 4,$
$-4x_1 + x_2 + x_4 = 4,$
$x_1 + x_2 + x_5 = 6.$

Відповіді:

1) $\boldsymbol{x}^* = (3; 1; 0)$, $L(\boldsymbol{x}^*) = 21$.
2) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 10$.
3) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 10$.
4) $\boldsymbol{x}^* = (1; 0; 0; 3; 0)$, $L(\boldsymbol{x}^*) = 1$.
5) $\boldsymbol{x}^* = (2; 0)$, $L(\boldsymbol{x}^*) = 12$.
6) $\boldsymbol{x}^* = (6; 5)$, $L(\boldsymbol{x}^*) = 27$.
7) $\boldsymbol{x}^* = (18; 14)$, $L(\boldsymbol{x}^*) = 48$.
8) $\boldsymbol{x}^* = (0; 4; 24; 0; 7)$, $L(\boldsymbol{x}^*) = 20$.
9) $\boldsymbol{x}^* = (0; 0; 0; 3; 2)$, $L(\boldsymbol{x}^*) = -29$.
10) $\boldsymbol{x}^* = (5; 1; 0; 0; 0)$, $L(\boldsymbol{x}^*) = 42$.

Лабораторна робота 10.

Задача частково цілочисельного лінійного програмування. Метод Гоморі-2

Постановка частково цілочисельної задачі лінійного програмування (ЧЦЗЛП).

Знайти вектор $\boldsymbol{x} = (x_1, \ldots, x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \ldots + c_n x_n \tag{10.1}$$

і задовольняє систему обмежень

$$\begin{aligned} &a_{11}x_1 + \ldots + a_{1n}\ x_n = a_{10}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ &a_{m1}x_1 + \ldots + a_{mn}x_n = a_{m0}, \end{aligned} \tag{10.2}$$

$$x_j \geq 0,\ j = 1, \ldots, n, \tag{10.3}$$

$$x_j \text{ — цілі},\ j = 1, \ldots, p,\ (p \leq n). \tag{10.4}$$

Виклад методу Гоморі-2.

Метод Гоморі-2, як і метод Гоморі-1, є одним з методів відтинання і полягає в наступному.

Розв'язується допоміжна *ЗЛП* (10.1)–(10.3), яку отримують з вихідної ЗЛП (10.1)–(10.4) відкиданням умови цілочисельності змінних (10.4). Якщо її розв'язок задовольняє умову (10.4), то він же є і розв'язком вихідної *ЧЦЗЛП*. Інакше від розв'язаної ЗЛП переходять до нової допоміжної *ЗЛП* приєднанням лінійного обмеження, яке задовольняють цілочисельні (у розумінні умов (10.4)) розв'язки вихідної *ЧЦЗЛП*, але не задовольняє отриманий нецілочисельний розв'язок вихідної *ЗЛП*. Згадане додаткове обмеження визначає деяку відтинаючу площину і називається *правильним відтином*. Приєднання нових правильних відтинів здійснюється доти, поки на деякому кроці не буде отримано цілочисельний (у розумінні умов (10.4)) розв'язок допоміжної задачі, який і є оптимальним розв'язком вихідної *ЧЦЗЛП*. В методі Гоморі-2 правильний відтин будується так.

Нехай на останній ітерації симплекс-методу при розв'язуванні допоміжної *ЗЛП* її непрямі обмеження набули вигляду:

$$x_i + Q_{i,m+1} x_{m+1} + \ldots + Q_{in} x_n = Q_{i0}, \quad i=1,\ldots,m,$$

і, значить, розв'язком допоміжної *ЗЛП* є вектор

$$\boldsymbol{x} = (Q_{10}, \ldots, Q_{m0}, 0, \ldots, 0).$$

Нехай існує номер r ($r \le p$) такий, що Q_{r0} — неціле, і $\{z\}$ — дробова частина z. Тоді правильний відтин методу Гоморі-2 має вигляд:

$$x_{n+1} - D_{r,m+1} x_{m+1} - \ldots - D_{rn} x_n = -\{Q_{r0}\}, \tag{10.5}$$

де $x_{n+1} \ge 0$ — додаткова змінна, та

$$D_{rj} = \begin{cases} \{Q_{rj}\}, & \text{при } j \le p,\ \{Q_{rj}\} \le \{Q_{r0}\}, \\ \{Q_{r0}\}(1-\{Q_{rj}\})/(1-\{Q_{r0}\}), & \text{при } j \le p,\ \{Q_{rj}\} > \{Q_{r0}\}, \\ Q_{rj}, & \text{при } j > p,\ Q_{rj} \ge 0, \\ \{Q_{r0}\}(-Q_{rj})/(1-\{Q_{r0}\}), & \text{при } j > p,\ Q_{rj} < 0. \end{cases} \tag{10.6}$$

<u>*Алгоритм методу Гоморі-2.*</u>

1. Розв'язуємо допоміжну ЗЛП (10.1)–(10.3). Нехай $\boldsymbol{x}(0)$ — її оптимальний розв'язок. Якщо ця задача не має розв'язку, то вихідна *ЧЦЗЛП* також не має розв'язку.

2. Нехай на s-й ітерації розв'язана допоміжна *ЗЛП*, що має M обмежень і N змінних, $\boldsymbol{x}(s)$ — її оптимальний розв'язок. Припустимо, що $\boldsymbol{x}(s)$ визначається канонічними обмеженнями останньої ітерації, а саме:

$$x_i + Q_{i,M+1} x_{M+1} + \ldots + Q_{iN} x_N = Q_{i0}, \quad i=1,\ldots,M,$$

звідки випливає, що

$$\boldsymbol{x}(s) = (Q_{10}, \ldots, Q_{M0}, 0, \ldots, 0).$$

3. Якщо Q_{i0} ($i=1,\ldots,p$) — цілі, то кінець: $\boldsymbol{x}(s)$ є оптимальним розв'язком вихідної *ЧЦЗЛП*. Якщо існує хоча б одне i таке, що Q_{i0} — неціле ($i=1,\ldots,p$), то перехід до пункту 4.

4. Знаходимо $r = \min\{i\}$ по всіх i ($i=1,\ldots,p$) таких, що Q_{i0} — неціле та будуємо додаткове обмеження за формулами (10.5)–(10.6) при $m = M$ та $n = N$.

5. Розширюємо симплекс-таблицю за рахунок ($M+1$)-го рядка (додаткове обмеження) та ($N+1$)-го стовпця, що відповідає додатковій змінній x_{N+1}.

6. Розв'язуємо розширену *ЗЛП* за допомогою двоїстого симплекс-методу (*ДСМ*) і переходимо до пункту 2 з заміною s на $s+1$, M на $M+1$, N на $N+1$. Якщо на деякій ітерації *ДСМ* одна з додаткових змінних задачі знову стає базисною, то з подальшого розгляду виключаються відповідні їй рядок та стовпець і при переході до пункту 2 замінюється лише s на $s+1$.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого частково цілочисельна задача лінійного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Завдання.

Розв'язати методом Гоморі-2 задачі частково цілочисельного лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

1)
$$\begin{aligned} x_1 + 8x_2 &\to \max, \\ 3x_1 + x_2 &\le 9, \\ 0.16x_1 + x_2 &\le 1.9, \\ x_j \ge 0,\ x_j &\text{ — ціле},\ j = 1,2; \end{aligned}$$

2)
$$\begin{aligned} -6x_1 - x_2 &\to \min, \\ -2.9x_1 + 6x_2 &\le 17.4, \\ 3x_1 - x_2 &\le 1, \\ x_j \ge 0,\ x_j &\text{ — ціле},\ j = 1,2; \end{aligned}$$

3)
$$\begin{aligned} 0.25x_1 + x_2 &\to \max, \\ 0.5x_1 + x_2 &\le 1.75, \\ x_1 + 0.3x_2 &\le 1.5, \\ x_j \ge 0,\ x_j &\text{ — ціле},\ j = 1,2; \end{aligned}$$

4)
$$\begin{aligned} -2x_1 - 4x_2 &\to \min, \\ 2x_1 + x_2 &\le 19.33, \\ x_1 + 3x_2 &\le 10, \\ x_j \ge 0,\ x_j &\text{ — ціле},\ j = 1,2; \end{aligned}$$

5)
$$\begin{aligned} x_1 + x_2 &\to \max, \\ 2x_1 + 11x_2 &\le 38, \\ x_1 + x_2 &\le 7, \\ 4x_1 - 5x_2 &\le 5, \\ x_j \ge 0,\ j = 1,2,\ x_2 &\text{ — ціле}; \end{aligned}$$

6)
$$\begin{aligned} x_1 + x_2 &\to \max, \\ 2x_1 + 11x_2 &\le 38, \\ x_1 + x_2 &\le 7, \\ 4x_1 - 5x_2 &\le 5, \\ x_j \ge 0,\ j = 1,2,\ x_1 &\text{ — ціле}; \end{aligned}$$

7)
$$\begin{aligned} x_1 &\to \max, \\ x_1 + 3x_2 &\le 12, \\ 3x_1 - 8x_2 &\le 24, \\ x_j \ge 0,\ j = 1,2,\ x_1 &\text{ — ціле}; \end{aligned}$$

8)
$$\begin{aligned} -8x_1 - 6x_2 &\to \min, \\ 3x_1 + 5x_2 + x_3 &= 11, \\ 4x_1 + x_2 + x_4 &= 8, \\ x_j \ge 0,\ j = 1,2,\ x_1 &\text{ — ціле}. \end{aligned}$$

Відповіді:

1) $\boldsymbol{x}^* = (2; 1)$, $L(\boldsymbol{x}^*) = 10$.
2) $\boldsymbol{x}^* = (1; 3)$, $L(\boldsymbol{x}^*) = -9$.
3) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 1.25$.
4) $\boldsymbol{x}^* = (7; 1)$, $L(\boldsymbol{x}^*) = -18$.
5) $\boldsymbol{x}^* = (3.75; 2)$, $L(\boldsymbol{x}^*) = 5.75$.
6) $\boldsymbol{x}^* = (4; 2.73)$, $L(\boldsymbol{x}^*) = 6.73$.
7) $\boldsymbol{x}^* = (9; 0.38)$, $L(\boldsymbol{x}^*) = 9$.
8) $\boldsymbol{x}^* = (1; 1.6; 0; 2.41)$, $L(\boldsymbol{x}^*) = -17.6$.

Лабораторна робота 11.

Задача цілочисельного лінійного програмування. Метод Гоморі-3

Постановка цілочисельної задачі лінійного програмування (ЦЗЛП).

Знайти вектор $\boldsymbol{x}=(x_1,\dots,x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \dots + c_n x_n \tag{11.1}$$

і задовольняє систему обмежень

$$\begin{aligned} &a_{11}x_1 + \dots + a_{1n}\ x_n = a_{10},\\ &\dots\dots\dots\dots\dots\dots,\\ &a_{m1}x_1 + \dots + a_{mn}x_n = a_{m0}, \end{aligned} \tag{11.2}$$

$$x_j \ge 0,\ j=1,\dots,n, \tag{11.3}$$

$$x_j \text{ — цілі, } j=1,\dots,n. \tag{11.4}$$

Виклад методу Гоморі-3.

Нехай обмеження (11.2) *ЗЛП* (11.1)–(11.3) приведені до майже канонічного вигляду:

$$x_i\ +\alpha_{i,m+1}x_{m+1}+\dots+\alpha_{in}x_n=\alpha_{i0},\ \ i=1,\dots,m, \tag{11.5}$$

де α_{ij}, $i=1,\dots,m,\ j=0,m+1,\dots,n,$ — цілі та симплекс-різниці $\Delta_j \ge 0,\ j=1,\dots,n$.

Якщо $\alpha_{i0} \ge 0$, $i=1,\dots,m,$ то обмеження (11.5) визначають оптимальний розв'язок *ЦЗЛП* (11.1)–(11.4), інакше визначається деякий цілочисельний майже допустимий базисний розв'язок (*МДБР*) вихідної *ЗЛП*. Можна було б, звичайно, вибрати один з індексів *i*, для якого $\alpha_{i0} < 0$, та виконати ітерацію двоїстого симплекс-методу. Проте в цьому випадку цілочисельність нових параметрів була б, взагалі кажучи, порушена через необхідність ділення на ведучий елемент перетворення. Цього можна уникнути лише тоді, коли ведучий елемент дорівнює –*1*. Виявляється, що можна побудувати додаткове обмеження, якому задовольняють всі цілочисельні розв'язки *ЦЗЛП* і яке разом з тим визначає ведучий рядок перетворення, що має ведучий елемент –*1*. Будується воно за *l*-м обмеженням системи (11.5), для якого α_{l0} мінімальне серед від'ємних α_{i0}, і має вигляд:

$$x_{n+1} +\alpha_{m+1,m+1}x_{m+1} +\dots+\alpha_{m+1,n}x_n =\alpha_{m+1,0},$$

де $x_{n+1} \ge 0$ — додаткова змінна,

$$\alpha_{m+1,j} = [\alpha_{lj}/\alpha],\ j=0,m+1,\dots,n,$$
$$\alpha = \max\{-\alpha_{lj}\},\ j=m+1,\dots,n.$$

Отже, наявна симплекс-таблиця розширюється за рахунок ($m+1$)-го рядка з елементами $\alpha_{m+1,j}$ ($\alpha_{m+1,j}=0$ при $j=1,\dots,m$) та одиничного стовпця $\boldsymbol{A}_{n+1}$, що відповідає додатковій змінній x_{n+1}. Потім виконується ітерація двоїстого симплекс-методу з ($m+1$)-м ведучим рядком.

Алгоритм методу Гоморі-3.

1. Зводимо вихідну *ЗЛП* (11.1)–(11.3) до майже канонічної *ЗЛП* (*МКЗЛП*) з цілочисельними коефіцієнтами, що визначає цілочисельний *МДБР* $\boldsymbol{x}(0)$, для якого $\Delta_j \geq 0$, $j=1,\dots,n$.

2. Нехай на *s*-й ітерації отримана повністю цілочисельна *МКЗЛП*

$$x_i + \alpha_{i,M+1} x_{M+1} + \dots + \alpha_{iN} x_N = \alpha_{i0}, \quad i=1,\dots,M,$$

що визначає цілочисельний *N*-вимірний *МДБР*

$$\boldsymbol{x}(s) = (\alpha_{10},\dots,\alpha_{M0},0,\dots,0).$$

3. Якщо $\alpha_{i0} \geq 0$, $i=1,\dots,M,$ то кінець: $\boldsymbol{x}(s)$ — оптимальний розв'язок *ЦЗЛП*. Інакше

4. Якщо для деякого *i* такого, що $\alpha_{i0} < 0$, $\alpha_{ij} \geq 0$, $j=1,\dots,N$, то кінець: вихідна *ЦЗЛП* не має допустимих розв'язків. Якщо таких *i* немає, то

5. Знаходимо $l = \text{argmin}\{\alpha_{i0}\}$, де i: $\alpha_{i0} < 0$. Визначаємо $\alpha = \max\{-\alpha_{lj}\}$, $j=M+1,\dots,N,$ і будуємо додаткове обмеження

$$x_{N+1} + \alpha_{M+1,M+1} x_{M+1} + \dots + \alpha_{M+1,N} x_N = \alpha_{M+1,0},$$

де $x_{N+1} \geq 0$ — додаткова змінна, $\alpha_{M+1,j}$ — ціла частина відношення α_{lj}/α, $j=0,M+1,\dots,N.$

6. Розширюємо симплекс-таблицю за рахунок (*M+1*)-го рядка (додаткове обмеження) та (*N+1*)-го стовпця, що відповідає додатковій змінній x_{N+1}.

7. Знаходимо $k = \text{argmin}\{\Delta_j\}$, де j: $\alpha_{M+1,j} < 0$. Переходимо до нового цілочисельного *МДБР* $\boldsymbol{x}(s+1)$, виконуючи симплекс-перетворення з ведучими рядком *M+1* та стовпцем *k*. Якщо *k* — індекс однієї з додаткових змінних, то переходимо до п. 8, інакше — до п. 3, заміняючи *M* на *M+1*, *N* на *N+1*, *s* на *s+1*.

8. Виключаємо з подальшого розгляду (викреслюємо) *k*-й стовпець та (*M+1*)-й (останній) рядок симплекс-таблиці, перенумеровуємо решту додаткових змінних для збереження неперервної нумерації всіх змінних задачі і переходимо до п. 3, заміняючи *s* на *s+1*.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого цілочисельна задача лінійного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом Гоморі-3 задачі цілочисельного лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються нижче, всі змінні невід'ємні та цілі.

1)
$$\begin{aligned} -x_1 - 5x_2 &\to \max, \\ 12x_1 - 3x_2 &\geq 8, \\ -3x_1 + 9x_2 &\geq 8, \\ -x_1 + 3x_2 &\geq 3; \end{aligned}$$

2)
$$\begin{aligned} 3x_1 + 2x_2 &\to \min, \\ -3x_2 &\leq 2, \\ -2x_1 + 2x_2 &\leq 2, \\ 2x_1 - x_2 &\geq 1; \end{aligned}$$

3)
$$\begin{aligned} x_1 + 3x_2 &\to \min, \\ x_1 + 6x_2 &\geq 5, \\ -5x_1 + 19x_2 &\geq 13, \\ -3x_1 + 6x_2 &\geq 2; \end{aligned}$$

4) $7x_1 + 12x_2 \to \min,$
$4x_1 - 23x_2 \le -14,$
$-12x_1 + 2x_2 \le -9,$
$-13x_1 - x_2 \le -10;$

5) $-3x_1 - 8x_2 \to \max,$
$-2x_1 + 10x_2 \ge 4,$
$-3x_1 - 5x_2 \le -10,$
$2x_1 - x_2 \ge 2;$

6) $10x_1 + 7x_2 \to \min,$
$2x_1 - 12x_2 \le -9,$
$x_1 + 4x_2 \ge 10,$
$4x_1 - 2x_2 \ge 14;$

7)
$$\begin{aligned} & -3x_4 - 2x_5 \to \max, \\ x_1 & -5x_4 + x_5 = -15, \\ x_3 & + x_4 - 8x_5 = -9, \\ x_2 & - x_4 - 10x_5 = -19; \end{aligned}$$

8)
$$\begin{aligned} & 3x_2 + x_5 \to \min, \\ & -2x_2 + x_3 - 18x_5 = -20, \\ & x_1 + x_2 - 15x_5 = -7, \\ & -5x_2 + x_4 + x_5 = -9. \end{aligned}$$

<u>Відповіді:</u>

1) $\boldsymbol{x}^* = (2; 2)$, $L(\boldsymbol{x}^*) = -12$.
2) $\boldsymbol{x}^* = (1; 0)$, $L(\boldsymbol{x}^*) = 3$.
3) $\boldsymbol{x}^* = (0; 1)$, $L(\boldsymbol{x}^*) = 3$.
4) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 19$.
5) $\boldsymbol{x}^* = (2; 1)$, $L(\boldsymbol{x}^*) = -14$.
6) $\boldsymbol{x}^* = (5; 2)$, $L(\boldsymbol{x}^*) = 64$.
7) $\boldsymbol{x}^* = (3; 5; 3; 4; 2)$, $L(\boldsymbol{x}^*) = -16$.
8) $\boldsymbol{x}^* = (6; 2; 2; 0; 1)$, $L(\boldsymbol{x}^*) = 7$.

<u>Лабораторна робота 12.</u>

Задача частково дискретного лінійного програмування. Метод Дальтона-Ллевеліна

<u>Постановка частково дискретної задачі лінійного програмування (ЧДЗЛП).</u>

Знайти вектор $\boldsymbol{x} = (x_1, \ldots, x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \ldots + c_n x_n \tag{12.1}$$

і задовольняє систему обмежень

$$\begin{aligned} & a_{11} x_1 + \ldots + a_{1n} x_n = a_{10}, \\ & \ldots\ldots\ldots\ldots\ldots\ldots, \\ & a_{m1} x_1 + \ldots + a_{mn} x_n = a_{m0}, \end{aligned} \tag{12.2}$$

$$x_j \ge 0,\ j = 1, \ldots, n, \tag{12.3}$$

$$x_j \in \{X_{j1}, \ldots, X_{j,nj}\},\ j = 1, \ldots, p,\ (p \le n), \tag{12.4}$$

де $0 = X_{j1} < \ldots < X_{j,nj}$.

<u>Виклад методу Дальтона-Ллевеліна.</u>

Метод Дальтона-Ллевеліна, як і методи Гоморі, є одним з *методів відтинання* і полягає в наступному.

Розв'язується допоміжна *ЗЛП* (12.1)–(12.3), яку отримують з вихідної задачі (12.1)–(12.4) відкиданням умови дискретності змінних (12.4). Якщо її розв'язок задовольняє умову (12.4), то він же є і розв'язком вихідної *ЧДЗЛП*. Інакше від розв'язаної *ЗЛП* переходять до нової допоміжної *ЗЛП* приєднанням лінійного обмеження, яке задовольняють дискретні (в розумінні умов (12.4)) розв'язки вихідної *ЧДЗЛП*, але не задовольняє отриманий розв'язок вихідної *ЗЛП*. Це додаткове обмеження визначає деяку відтинаючу площину і зветься *правильним відтином.*

Приєднання нових правильних відтинів до початкової допоміжної *ЗЛП* здійснюється доти, поки на деякому кроці не буде отриманий дискретний розв'язок допоміжної *ЗЛП*, який, очевидно, буде і оптимальним розв'язком вихідної *ЧДЗЛП.*

В методі Дальтона-Ллевеліна правильний відтин будується таким чином. Нехай на останній ітерації симплекс-методу при розв'язуванні допоміжної *ЗЛП* її непрямі обмеження набули вигляду:

$$x_i + Q_{i,m+1}x_{m+1} + \ldots + Q_{in}x_n = Q_{i0},\ i=1,\ldots,m,$$

і, отже, розв'язком допоміжної *ЗЛП* є вектор

$$\boldsymbol{x} = (Q_{10},\ldots,Q_{m0},0,\ldots,0).$$

Нехай існує номер r ($r \le p$) такий, що Q_{r0} не задовольняє (12.4), до того ж $X_{rt} < Q_{r0} < X_{r,t+1}$ для деякого $t=1,\ldots,n_r-1$.

Тоді правильний відтин методу Дальтона-Ллевеліна має вигляд:

$$x_{n+1} - D_{r,m+1}x_{m+1} - \ldots - D_{rn}x_n = -(Q_{r0} - X_{rt}), \qquad (12.5)$$

де $x_{n+1} \ge 0$ — додаткова змінна,

$$D_{rj} = \begin{cases} 0, \text{ якщо } x_j - \text{базисна змінна,} & \text{інакше} \\ Q_{rj}, & \text{якщо } Q_{rj} \ge 0, \\ (-Q_{rj})(Q_{r0} - X_{rt})/(X_{rt+1} - Q_{r0}), & \text{якщо } Q_{rj} < 0. \end{cases} \qquad (12.6)$$

<u>*Алгоритм методу Дальтона-Ллевеліна.*</u>

1. Розв'язуємо допоміжну *ЗЛП* (12.1)–(12.3). Нехай $\boldsymbol{x}(0)$ — її оптимальний розв'язок. Якщо ця задача не має розв'язку, то вихідна *ЧДЗЛП* також не має розв'язку.

2. Нехай на s-й ітерації розв'язана допоміжна *ЗЛП*, що має M обмежень та N змінних, $\boldsymbol{x}(s)$ — її оптимальний розв'язок.
Припустимо, що $\boldsymbol{x}(s)$ визначається за канонічними обмеженнями останньої ітерації, а саме:

$$x_i + Q_{i,M+1}x_{M+1} + \ldots + Q_{iN}x_N = Q_{i0},\ i=1,\ldots,M,$$

звідки випливає, що

$$\boldsymbol{x}(s) = (Q_{10},\ldots,Q_{M0},0,0,\ldots,0).$$

3. Якщо Q_{i0} ($i=1,\ldots,p$) задовольняє умову (12.4), то кінець: $\boldsymbol{x}(s)$ є розв'язком вихідної *ЧДЗЛП*. Інакше

4. Знаходимо $r = \min\{i\}$ по всіх i ($i=1,\ldots,p$) таких, що при деякому t ($t=1,\ldots,n_i-1$) $X_{it} < Q_{i0} < X_{i,t+1}$, і будуємо додаткове обмеження за формулами (12.5)–(12.6) при $m = M$ та $n = N$.

5. Розширюємо симплекс-таблицю за рахунок ($M+1$)-го рядка (додаткове обмеження) та ($N+1$)-го стовпця, що відповідає додатковій змінній x_{N+1}.

6. Розв'язуємо розширену *ЗЛП* за допомогою двоїстого симплекс-методу (*ДСМ*) і переходимо до пункту 2 з заміною s на $s+1$, M на $M+1$, N на $N+1$. Якщо на деякій ітерації *ДСМ* одна з додаткових змінних задачі знову стає базисною, то з подальшого розгляду виключаються відповідні їй рядок та стовпець і при переході до пункту 2 замінюється лише *s* на *s+1*.

<u>Програмне забезпечення.</u>

Навчаючий модуль, за допомогою якого задача частково дискретного лінійного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

<u>Завдання.</u>

Розв'язати методом Дальтона-Ллевеліна задачі частково дискретного лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

В усіх задачах, що пропонуються нижче, всі змінні невід'ємні.

1) $$\begin{aligned} &2x_1 + 3x_2 \to \min, \\ &-3x_1 - 2x_2 \le -6, \\ &x_1 + 4x_2 \ge 4, \\ &x_1 \in \{0,1,3,4\}, \\ &x_2 \in \{0,2,3,5\}; \end{aligned}$$

2) $$\begin{aligned} &5x_1 + 3x_2 \to \max, \\ &3x_1 + 5x_2 \le 15, \\ &5x_1 + 2x_2 \le 10, \\ &x_1 \in \{0,1,2\}, \\ &x_2 \in \{0,2,3\}; \end{aligned}$$

3) $$\begin{aligned} &x_1 + x_2 \to \max, \\ &x_1 + 2x_2 \le 10, \\ &2x_1 + x_2 \le 10, \\ &x_1 \in \{0,3,4\}, \\ &x_2 \in \{0,1,2,4\}; \end{aligned}$$

4) $$\begin{aligned} &-2x_1 + 3x_2 \to \max, \\ &4x_1 + 5x_2 \le 20, \\ &2x_1 + x_2 \ge 6, \\ &x_1 \in \{0,2,4\}, \\ &x_2 \in \{0,1,2,5\}; \end{aligned}$$

5) $$\begin{aligned} &-x_1 - 3x_2 \to \min, \\ &x_1 + x_2 \le 5, \\ &-x_1 + 2x_2 \ge -4, \\ &x_1 \in \{0,2,4\}, \\ &x_2 \in \{0,1,3,6\}; \end{aligned}$$

6) $$\begin{aligned} &x_1 - 2x_2 \to \min, \\ &x_1 - x_2 \le 2, \\ &x_1 + x_2 \le 3, \\ &x_1 \in \{0,1,2,4\}, \\ &x_2 \in \{0,2,4\}; \end{aligned}$$

7) $$\begin{aligned} &x_1 - x_2 \to \max, \\ &2x_1 - 2x_2 \ge -5, \\ &2x_1 + 2x_2 \le 7, \\ &x_1 \in \{0,2,3\}, \\ &x_2 \in \{0,1,3\}; \end{aligned}$$

8) $$\begin{aligned} &-x_1 - x_2 \to \min, \\ &-x_1 + 4x_2 \le 12, \\ &4x_1 - x_2 \le 12, \\ &x_1 \in \{0,1,4\}, \\ &x_2 \in \{0,1,3,5\}; \end{aligned}$$

9) $$\begin{aligned} &3x_1 + 4x_2 \to \max, \\ &-x_1 + 2x_2 \le 4, \\ &3x_1 - x_2 \le 7, \\ &x_1 \in \{0,1,3,5\}, \\ &x_2 \in \{0,2,3,5\}. \end{aligned}$$

<u>Відповіді:</u>

1) $\boldsymbol{x}^* = (1;\ 2)$, $L(\boldsymbol{x}^*) = 8$.
2) $\boldsymbol{x}^* = (1;\ 2)$, $L(\boldsymbol{x}^*) = 11$.
3) $\boldsymbol{x}^* = (4;\ 2)$, $L(\boldsymbol{x}^*) = 6$.
4) $\boldsymbol{x}^* = (2;\ 2)$, $L(\boldsymbol{x}^*) = 2$.
5) $\boldsymbol{x}^* = (2;\ 3)$, $L(\boldsymbol{x}^*) = -11$.
6) $\boldsymbol{x}^* = (0;\ 2)$, $L(\boldsymbol{x}^*) = -4$.
7) $\boldsymbol{x}^* = (2;\ 0)$, $L(\boldsymbol{x}^*) = 2$.
8) $\boldsymbol{x}^* = (1;\ 3)$, $L(\boldsymbol{x}^*) = -4$.
9) $\boldsymbol{x}^* = (3;\ 3)$, $L(\boldsymbol{x}^*) = 21$.

Лабораторна робота 13.

Задача цілочисельного лінійного програмування. Метод віток і границь

Постановка цілочисельної задачі лінійного програмування (ЦЗЛП).

Знайти вектор $\boldsymbol{x}=(x_1,\dots,x_n)$, що мінімізує цільову функцію

$$L(\boldsymbol{x}) = c_1 x_1 + \dots + c_n x_n \tag{13.1}$$

і задовольняє систему обмежень

$$\begin{array}{l} a_{11}x_1 + \dots + a_{1n}\ x_n\ R_1\ \ a_{10}, \\ \dots\dots\dots\dots\dots\dots\dots\dots, \\ a_{m1}x_1 + \dots + a_{mn}x_n\ R_m\ a_{m0}, \end{array} \tag{13.2}$$

$$x_j \geq 0,\ j=1,\dots,n, \tag{13.3}$$

$$x_j \text{ — цілі, } j=1,\dots,n. \tag{13.4}$$

де символ R_i $(i=1,\dots,m)$ замінює один із знаків: $\leq$, $=$, $\geq$.

Вважається також, що многогранна множина, яка визначається співвідношеннями (13.2)–(13.3), обмежена, а коефіцієнти c_j цільової функції (13.1) — цілі числа.

Далі наводиться *алгоритм методу Ленд-Дойг*, який являє собою реалізацію *методу віток та границь* для сформульованої вище задачі.

Виклад методу Ленд-Дойг.

Розв'язується допоміжна *ЗЛП* (13.1)–(13.3), яка отримана з вихідної *ЦЗЛП* (13.1)–(13.4) відкиданням умови цілочисельності змінних (13.4) (вітка $\boldsymbol{0;1}$). Якщо її розв'язок $\boldsymbol{x}(\boldsymbol{0;1})$ — цілочисельний, то він же є і розв'язком вихідної *ЦЗЛП*. Інакше величина $\xi(\boldsymbol{0;1}) = L(\boldsymbol{x}(\boldsymbol{0;1}))$ дає нижню оцінку (границю) цільової функції *ЦЗЛП* на множині $D(\boldsymbol{0;1}) = D$, що визначається співвідношеннями (13.2), (13.3).

Нехай деяка координата $x_j(\boldsymbol{0;1})$ $(j=1,\dots,n)$ розв'язку $\boldsymbol{x}(\boldsymbol{0;1})$ не є цілочисельною. В цьому випадку здійснюється розгалуження множини $D(\boldsymbol{0;1})$ на дві підмножини $D(\boldsymbol{1;1})$ і $D(\boldsymbol{1;2})$ додаванням до обмежень, що задають $D(\boldsymbol{0;1})$, обмежень $x_j \leq [x_j(\boldsymbol{0;1})]$ та $x_j \geq [x_j(\boldsymbol{0;1})]+1$ відповідно, де $[z]$ — ціла частина числа z. Далі розв'язуються нові допоміжні *ЗЛП* з обмеженнями, які визначаються підмножинами $D(\boldsymbol{1;1})$ та $D(\boldsymbol{1;2})$, знаходяться границі $\xi(\boldsymbol{1;1})$ та $\xi(\boldsymbol{1;2})$ і т. д.

Для подальшого розгалуження обирається *перспективна множина* $D(\boldsymbol{k;r})$ з найменшою границею $\xi(\boldsymbol{k;r})$. Процес продовжується доти, поки не буде отримано розв'язок, який задовольняє умову цілочисельності і для якого виконується ознака оптимальності (див. п. 4 алгоритму). Внаслідок обмеженості допустимої множини *ЗЛП* (скінченності допустимої множини *ЦЗЛП*) метод Ленд-Дойг скінченний.

Алгоритм методу Ленд-Дойг.

1. Визначаються множини $D(\boldsymbol{k;r})$ умовами (13.2), (13.3) і додатковими обмеженнями, які виникають в процесі розгалуження (див. пункт 5). На ***0***-у кроці покладаємо $D(\boldsymbol{0;1}) = D$, де D задається умовами (13.2), (13.3).

2. Розв'язуються допоміжні *ЗЛП* на множинах $D(\boldsymbol{k;r})$. Нехай $\boldsymbol{x}(\boldsymbol{k;r})$ — оптимальні розв'язки вказаних *ЗЛП*.

3. Обчислюються границі на множинах $D(\boldsymbol{k;r})$ за формулою $\xi(\boldsymbol{k;r}) = \,]L(\boldsymbol{x}(\boldsymbol{k;r}))[$, де $]z[$ — найменше ціле число, не менше z.
4. Якщо існують k, l такі, що $\boldsymbol{x}(\boldsymbol{k;l})$ — цілочисельний розв'язок та для всіх віток $\boldsymbol{r}$ на $\boldsymbol{k}$-у кроці виконуються співвідношення

$$L(\boldsymbol{x}(\boldsymbol{k;l})) = \xi(\boldsymbol{k;l}) \le \xi(\boldsymbol{k;r}),$$

то $\boldsymbol{x}^* = \boldsymbol{x}(\boldsymbol{k;l})$ — оптимальний розв'язок *ЦЗЛП*.
5. Розгалуження здійснюється по нецілочисельній компоненті $x_j(\boldsymbol{k;r})$ (з мінімальним j) розв'язку $\boldsymbol{x}(\boldsymbol{k;r})$, що відповідає перспективній вітці $\boldsymbol{k;r}$ (якщо таких віток декілька, то вибирається вітка з мінімальним $\boldsymbol{r}$), додаванням до $D(\boldsymbol{k;r})$ однієї з підмножин $x_j \le [x_j(\boldsymbol{k;r})]$ або $x_j \ge [x_j(\boldsymbol{k;r})]+1$.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача цілочисельного лінійного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО.*

Завдання.

Розв'язати методом віток та границь задачі цілочисельного лінійного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі (без використання програмного забезпечення).

В усіх задачах виконуються умови: $x_j \ge 0$, x_j — ціле, $j=1,2$.

1) $x_1 + x_2 \to \max$,
$2x_1 + 11x_2 \le 38$,
$x_1 + x_2 \le 7$,
$4x_1 - 5x_2 \le 5$;

2) $-2x_1 - x_2 \to \min$,
$6x_1 + 4x_2 \le 24$,
$x_1 - x_2 \le 3$,
$-x_1 + 3x_2 \le 3$;

3) $6x_1 + 4x_2 \to \min$,
$2x_1 + x_2 \ge 3$,
$x_1 - 2x_2 \le 2$,
$3x_1 + 2x_2 \ge 1$;

4) $x_1 + x_2 \to \min$,
$-x_1 - x_2 \le -1$,
$-2x_1 + 2x_2 \le -2$,
$-4x_1 + 2x_2 \le -1$;

5) $5x_1 + 7x_2 \to \min$,
$-10x_1 + x_2 \le -16$,
$-3x_1 - 3x_2 \le -12$,
$-6x_1 - 2x_2 \le -17$,

6) $2x_1 - x_2 \to \max$,
$2x_1 + x_2 \le 8$,
$x_1 + 3x_2 \ge 6$,
$3x_1 + x_2 \ge 3$;

7) $x_1 + 4x_2 \to \min$,
$x_1 - 3x_2 \le -1$,
$8x_1 - x_2 \ge 6$,
$3x_1 - 11x_2 \le -7$;

8) $-5x_1 - 7x_2 \to \max$,
$2x_1 + x_2 \ge 3$,
$20x_1 - x_2 \ge 15$,
$x_1 - x_2 \le -2$;

9) $x_1 + 2x_2 \to \min$,
$2x_1 + 2x_2 \ge 5$,
$x_1 - 3x_2 \le -1$,
$x_1 - x_2 \ge 2$.

Відповіді:

1) $\boldsymbol{x}^* = (3; 2)$ або $\boldsymbol{x}^* = (2; 3)$, $L(\boldsymbol{x}^*) = 5$.
2) $\boldsymbol{x}^* = (3; 1)$, $L(\boldsymbol{x}^*) = -7$.
3) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 10$.
4) $\boldsymbol{x}^* = (1; 0)$, $L(\boldsymbol{x}^*) = 1$.
5) $\boldsymbol{x}^* = (4; 0)$, $L(\boldsymbol{x}^*) = 20$.
6) $\boldsymbol{x}^* = (3; 1)$, $L(\boldsymbol{x}^*) = 5$.

7) $\boldsymbol{x}^* = (1; 1)$, $L(\boldsymbol{x}^*) = 5$.
8) $\boldsymbol{x}^* = (1; 3)$, $L(\boldsymbol{x}^*) = -26$.
9) $\boldsymbol{x}^* = (4; 2)$, $L(\boldsymbol{x}^*) = 8$.

Лабораторна робота 14.

Задача про призначення. Угорський метод

Постановка задачі про призначення.

Знайти вектор (матрицю) $\boldsymbol{X} = (x_{ij},\ i,j=1,\ldots,n)$, що мінімізує цільову функцію

$$L(\boldsymbol{X}) = c_{11}x_{11} + \ldots + c_{1n}x_{1n} + \ldots + c_{n1}x_{n1} + \ldots + c_{nn}x_{nn} \qquad (14.1)$$

і задовольняє систему обмежень

$$x_{i1} + \ldots + x_{in} = 1,\ i=1,\ldots,n, \qquad (14.2)$$

$$x_{1j} + \ldots + x_{nj} = 1,\ j=1,\ldots,n, \qquad (14.3)$$

$$x_{ij} = 0 \text{ або } 1,\ i,j=1,\ldots,n, \qquad (14.4)$$

де c_{ij} — витрати, пов'язані з використанням i-го виконавця для виконання j-ї роботи ($i,j=1,\ldots,n$). Елементи c_{ij} утворюють матрицю витрат $\boldsymbol{C}$.

Задача (14.1)–(14.4) розв'язується *угорським методом*, що грунтується на тому факті, що віднімання числа a_i, $i=1,\ldots,n$, від кожного елемента i-го рядка та числа b_j, $j=1,\ldots,n$, від кожного елемента j-го стовпця матриці витрат $\boldsymbol{C}$ не змінює множини оптимальних призначень. В цьому розумінні можна говорити, що вказані дії перетворюють матрицю витрат $\boldsymbol{C}$ в еквівалентну їй. В алгоритмі, що дається нижче, матриці, еквівалентні вихідній матриці витрат $\boldsymbol{C}$, називаються просто матрицями витрат.

Алгоритм угорського методу.

1. Віднімаємо в матриці $\boldsymbol{C}$ від кожного елемента i-го рядка мінімальний елемент цього рядка ($i=1,\ldots,n$).

2. Віднімаємо від кожного елемента j-го стовпця перетвореної матриці витрат його мінімальний елемент ($j=1,\ldots,n$). В результаті виконання двох пунктів кожний рядок та кожний стовпець матриці витрат мають принаймні один ***0***.

3. Проглядаємо послідовно рядки матриці витрат, починаючи з першого. Якщо рядок має лише один непозначений ***0***, позначаємо його позначкою ***** та закреслюємо (за допомогою позначки **^**) решту нулів в цьому ж стовпці. ***0*** вважається позначеним, якщо він має позначку *****. Повторюємо ці дії, поки кожний рядок не буде мати непозначених нулів, або буде мати їх принаймні два.

4. Дії п. 3 повторюємо для всіх стовпців матриці витрат.

5. Дії п. п. 3 та 4 повторюємо послідовно (якщо необхідно) поки не трапиться один з трьох можливих випадків:
   i) кожний рядок має призначення (має ***0*** з позначкою *****);
   ii) є принаймні два непозначених нулі в деяких рядках і деяких стовпцях матриці витрат;
   iii) немає непозначених нулів і повне призначення ще не отримане (число нулів з позначкою ***** менше n).

6. У випадку i) задача про оптимальні призначення розв'язана: x_{ij}, що відповідають ***0****, дорівнюють *1*, решта — *0*, кінець. У випадку ii) довільно вибираємо один з непозначених нулів, позначаємо його позначкою *, закреслюємо решту нулів в тому ж рядку і в тому ж стовпці і повертаємося до п. 3. У випадку iii) переходимо до п. 7.

7. Позначаємо позначкою **#** рядки, для яких не отримане призначення (в яких немає $\boldsymbol{0^*}$). Такі рядки вважаємо *позначеними*, решту — *непозначеними*. Аналогічно називаються і стовпці матриці витрат.

8. Позначаємо позначкою **#** ще непозначені стовпці, які мають закреслений $\boldsymbol{0}$ (позначений позначкою **^**) у позначених рядках.

9. Позначаємо позначкою **#** ще непозначені рядки, які мають призначення (тобто $\boldsymbol{0^*}$) у позначених стовпцях.

10. Повторюємо дії п. п. 8 та 9 доти, поки більше не можна буде позначити жодного рядка та стовпця матриці витрат.

11. Викреслюємо (за допомогою позначки **&**) непозначені рядки і позначені стовпці матриці витрат.

12. Знаходимо мінімальний невикреслений елемент матриці витрат, віднімаємо його від елементів кожного з невикреслених рядків, додаємо до елементів всіх викреслених стовпців і переходимо до п. 3. При цьому позначки елементів матриці витрат (***** та **^**) втрачають свою силу.

Зауваження.

1. Якщо в задачі про оптимальні призначення (14.1)–(14.4) цільову функцію (14.1) треба максимізувати, то для її розв'язування можна застосувати угорський метод, замінивши матрицю $\boldsymbol{C}$ на $-\boldsymbol{C}$.

2. За означенням в задачі про оптимальні призначення матриця витрат квадратна. Якщо матриця $\boldsymbol{C}$ не є квадратною, то вона перетворюється до такої додаванням необхідного числа додаткових рядків або стовпців з відповідними елементами $c_{ij} = 0$. В 1-у випадку роботи, що отримали оптимальні призначення в додаткових рядках, залишаються без виконавців. В 2-у — виконавці, які отримали оптимальні призначення в додаткових стовпцях, залишаються без роботи.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача про оптимальні призначення розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати угорським методом задачі про оптимальні призначення, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі з матрицями витрат $\boldsymbol{C}$:

1) $\left\| \begin{matrix} 3 & 4 & 2 & 8 & 1 & 7 & 3 \\ 2 & 3 & 13 & 9 & 1 & 6 & 2 \\ 12 & 4 & 12 & 5 & 3 & 1 & 4 \\ 5 & 6 & 1 & 7 & 11 & 8 & 6 \\ 11 & 4 & 10 & 10 & 5 & 13 & 7 \\ 9 & 6 & 11 & 12 & 7 & 1 & 2 \\ 2 & 4 & 8 & 5 & 9 & 3 & 10 \end{matrix} \right\|$ **2)** $\left\| \begin{matrix} 6 & 2 & 15 & 2 & 4 & 9 & 5 \\ 12 & 11 & 1 & 13 & 8 & 11 & 13 \\ 3 & 2 & 12 & 9 & 10 & 14 & 1 \\ 7 & 1 & 3 & 4 & 5 & 6 & 8 \\ 8 & 9 & 14 & 3 & 11 & 18 & 12 \\ 1 & 7 & 5 & 6 & 15 & 16 & 2 \\ 13 & 10 & 4 & 7 & 10 & 16 & 17 \end{matrix} \right\|$

3) $\left\| \begin{matrix} 5 & 7 & 5 & 12 & 11 & 6 & 4 \\ 10 & 9 & 6 & 10 & 12 & 16 & 4 \\ 9 & 3 & 2 & 8 & 13 & 14 & 8 \\ 13 & 1 & 7 & 11 & 7 & 18 & 19 \\ 1 & 7 & 12 & 8 & 3 & 1 & 5 \\ 3 & 11 & 13 & 9 & 14 & 20 & 21 \\ 10 & 2 & 6 & 6 & 15 & 15 & 22 \end{matrix} \right\|$ **4)** $\left\| \begin{matrix} 21 & 7 & 2 & 12 & 15 & 2 & 17 \\ 23 & 15 & 24 & 20 & 12 & 5 & 11 \\ 17 & 24 & 4 & 17 & 2 & 22 & 15 \\ 19 & 7 & 8 & 1 & 13 & 14 & 4 \\ 15 & 6 & 6 & 14 & 19 & 3 & 16 \\ 23 & 6 & 5 & 19 & 15 & 11 & 19 \\ 16 & 18 & 22 & 22 & 1 & 1 & 7 \end{matrix} \right\|$

5) $\left\| \begin{matrix} 2 & 4 & 5 & 10 & 4 & 6 & 8 \\ 3 & 6 & 4 & 13 & 6 & 7 & 9 \\ 4 & 7 & 10 & 5 & 10 & 4 & 5 \\ 6 & 5 & 12 & 4 & 7 & 5 & 4 \\ 7 & 4 & 13 & 6 & 6 & 7 & 5 \\ 10 & 8 & 5 & 2 & 8 & 9 & 10 \\ 11 & 9 & 4 & 8 & 4 & 5 & 4 \end{matrix} \right\|$ **6)** $\left\| \begin{matrix} 7 & 9 & 4 & 6 & 4 & 12 & 3 \\ 2 & 4 & 4 & 7 & 8 & 8 & 5 \\ 4 & 5 & 6 & 5 & 12 & 7 & 3 \\ 3 & 6 & 8 & 4 & 6 & 6 & 7 \\ 5 & 10 & 9 & 3 & 8 & 5 & 4 \\ 6 & 9 & 5 & 10 & 9 & 6 & 7 \\ 7 & 4 & 3 & 6 & 2 & 5 & 4 \end{matrix} \right\|$

Розв'язати угорським методом задачі про оптимальні призначення, для яких задані матриці ефективностей:

7) $\left\| \begin{matrix} 6 & 7 & 8 & 9 & 10 & 12 & 5 \\ 2 & 3 & 4 & 8 & 9 & 16 & 8 \\ 9 & 6 & 3 & 6 & 8 & 9 & 3 \\ 5 & 7 & 6 & 7 & 10 & 7 & 8 \\ 4 & 8 & 18 & 8 & 6 & 2 & 3 \\ 12 & 9 & 2 & 10 & 8 & 4 & 5 \\ 3 & 10 & 3 & 5 & 4 & 3 & 8 \end{matrix} \right\|$ **8)** $\left\| \begin{matrix} 12 & 4 & 8 & 9 & 12 & 4 & 5 \\ 6 & 5 & 10 & 7 & 3 & 6 & 8 \\ 3 & 10 & 4 & 12 & 5 & 6 & 10 \\ 11 & 12 & 16 & 5 & 7 & 8 & 12 \\ 6 & 7 & 4 & 6 & 7 & 2 & 1 \\ 12 & 5 & 7 & 12 & 9 & 4 & 5 \\ 4 & 6 & 8 & 4 & 3 & 6 & 7 \end{matrix} \right\|$

9) $\left\| \begin{matrix} 6 & 3 & 5 & 4 & 6 & 7 & 8 \\ 5 & 2 & 4 & 3 & 8 & 9 & 10 \\ 4 & 3 & 3 & 5 & 4 & 6 & 7 \\ 7 & 5 & 3 & 2 & 1 & 4 & 5 \\ 8 & 6 & 6 & 10 & 12 & 5 & 4 \\ 12 & 7 & 8 & 7 & 4 & 8 & 9 \\ 4 & 8 & 9 & 6 & 7 & 4 & 3 \end{matrix} \right\|$ **10)** $\left\| \begin{matrix} 4 & 3 & 5 & 4 & 8 & 9 & 10 \\ 7 & 6 & 2 & 5 & 12 & 13 & 4 \\ 6 & 5 & 1 & 6 & 7 & 8 & 9 \\ 7 & 4 & 6 & 10 & 4 & 7 & 3 \\ 4 & 3 & 10 & 8 & 5 & 3 & 2 \\ 3 & 8 & 12 & 6 & 3 & 6 & 12 \\ 5 & 9 & 4 & 7 & 8 & 9 & 1 \end{matrix} \right\|$

Відповіді:

1) $L(\boldsymbol{x}^*) = 16$. **2)** $L(\boldsymbol{x}^*) = 24$. **3)** $L(\boldsymbol{x}^*) = 25$. **4)** $L(\boldsymbol{x}^*) = 38$. **5)** $L(\boldsymbol{x}^*) = 24$.
6) $L(\boldsymbol{x}^*) = 25$. **7)** $L(\boldsymbol{x}^*) = 81$. **8)** $L(\boldsymbol{x}^*) = 73$. **9)** $L(\boldsymbol{x}^*) = 60$. **10)** $L(\boldsymbol{x}^*) = 68$.

Лабораторна робота 15.

Задача про призначення. Метод Мака

Постановка задачі така ж сама, як і в попередньому розділі (14.1–14.4).

Алгоритм методу Мака.

1. Позначаємо мінімальний елемент рядка позначкою *****. Якщо мінімальних елементів декілька, позначаємо будь-який з них.

2. Дії п. 1 повторюємо для всіх рядків матриці витрат.

3. Якщо рядок має ще один мінімальний елемент, проглядаємо стовпець, до якого цей елемент належить. Можливі випадки:
   i) Стовпець не має позначених елементів;
   ii) Стовпець має принаймні один позначений елемент.
4. У випадку i) позначаємо мінімальний елемент рядка позначкою *. Всі інші позначки в цьому рядку знімаються. У випадку ii) позначаємо мінімальний елемент рядка позначкою **^**, якщо елемент цього рядка з позначкою * не є єдиним позначеним елементом у своєму стовпці.
5. Дії п. п. 3 та 4 повторюємо послідовно для всіх рядків, що мають більше одного мінімального елемента.
6. Якщо кожний стовпець матриці витрат має елемент з позначкою *, тоді задача про оптимальні призначення розв'язана. Інакше переходимо до наступного пункту.
7. Позначаємо (позначкою **&**) стовпці, що мають більше одного позначеного елемента. Вони утворюють множину ***B***, інші стовпці матриці витрат утворюють множину ***A***.
8. Проглядаємо послідовно рядки матриці витрат, починаючи з першого, і знаходимо рядок, в якому елемент з позначкою * належить множині ***B***.
9. Знаходимо для рядка мінімальну різницю між елементами множини ***A*** і елементом з позначкою *.
10. Дії п. п. 8 та 9 повторюємо послідовно для всіх рядків, що мають властивості, які вказані в п. 8.
11. Вибираємо найменшу з мінімальних різниць.
12. Додаємо це число до кожного елемента множини ***B***.
13. Повертаємось до п. 3.

Зауваження ті ж самі, як в попередньому розділі.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача про оптимальні призначення розв'язується в діалозі з користувачем за викладеним алгоритмом, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом Мака задачі про оптимальні призначення, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також задачі, матриці витрат ***C*** яких задані в попередньому розділі.

Лабораторна робота 16.

Матричні ігри. Зв’язок з задачею лінійного програмування. Метод Брауна-Робінсон

Постановка матричної гри двох осіб з нульовою сумою.

Знайти ціну гри та оптимальні змішані стратегії гравців для матричної гри двох осіб з нульовою сумою і заданою платіжною матрицею $\boldsymbol{C}=||c_{ij}||$, $i=1,\dots,m$, $j=1,\dots,n$ (гравця $\boldsymbol{I}_1$ гравцю $\boldsymbol{I}_2$).

Основні означення і теореми.

Змішані стратегії гравців $\boldsymbol{I_1}$ та $\boldsymbol{I_2}$ — це вектори $\boldsymbol{x}=(x_1,\dots,x_m)$ і $\boldsymbol{y}=(y_1,\dots,y_n)'$, компоненти яких задовольняють умови:

$$x_i \geq 0,\ i=1,\dots,m,\ x_1+\dots+x_m=1,$$
$$y_j \geq 0,\ j=1,\dots,n,\ y_1+\dots+y_n=1.$$

Можна вважати, що числа x_i ($i=1,\dots,m$) та y_j ($j=1,\dots,n$) є не що інше, як ймовірності вибору i-ї та j-ї стратегій, відповідно, гравцями $\boldsymbol{I_1}$ та $\boldsymbol{I_2}$ (i-го рядка матриці $\boldsymbol{C}$ першим гравцем та j-го стовпця цієї ж матриці другим гравцем).

Функція $F(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{x}\boldsymbol{C}\boldsymbol{y}'$ називається *математичним сподіванням платежу* гравця $\boldsymbol{I_1}$ гравцю $\boldsymbol{I_2}$ (*середнім виграшем гравця* $\boldsymbol{I_2}$).

Точка $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ ($\boldsymbol{x}^*\in X$, $\boldsymbol{y}^*\in Y$) називається *сідловою точкою* функції $f(\boldsymbol{x},\boldsymbol{y})$, $\boldsymbol{x}\in X$, $\boldsymbol{y}\in Y$, якщо для довільних $\boldsymbol{x}\in X$, $\boldsymbol{y}\in Y$ має місце нерівність

$$f(\boldsymbol{x}^*,\boldsymbol{y}) \leq f(\boldsymbol{x}^*,\boldsymbol{y}^*) \leq f(\boldsymbol{x},\boldsymbol{y}^*).$$

Теорема. *Нехай задана дійсна функція $f(\boldsymbol{x},\boldsymbol{y})$, $\boldsymbol{x}\in X$, $\boldsymbol{y}\in Y$, для якої існують*

$$\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x},\boldsymbol{y}), \qquad \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y}).$$

Для того, щоб виконувалось співвідношення

$$\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x},\boldsymbol{y}) = \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y}).$$

необхідно і достатньо, щоб функція $f(\boldsymbol{x},\boldsymbol{y})$ мала сідлову точку.

Теорема. *Функція $F(\boldsymbol{x},\boldsymbol{y})$ (середній виграш гравця $\boldsymbol{I_2}$) завжди має сідлову точку.*

Компоненти $\boldsymbol{x}^*$, $\boldsymbol{y}^*$ сідлової точки $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ функції $F(\boldsymbol{x},\boldsymbol{y})$ визначають *оптимальні змішані стратегії* гравців $\boldsymbol{I_1}$ та $\boldsymbol{I_2}$, відповідно, а *ціна гри* v визначається співвідношенням:

$$v = \min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} F(\boldsymbol{x},\boldsymbol{y}) = \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} F(\boldsymbol{x},\boldsymbol{y}) = F(\boldsymbol{x}^*,\boldsymbol{y}^*),$$

$$X = \{\, \boldsymbol{x} = (x_1,\dots,x_m) : x_i \geq 0,\ i=1,\dots,m,\ x_1+\dots+x_m = 1\},$$

$$Y = \{\, \boldsymbol{y} = (y_1,\dots,y_n)' : y_j \geq 0,\ j=1,\dots,n,\ y_1+\dots+y_n = 1\}.$$

Теорема. *Задача визначення оптимальних змішаних стратегій $\boldsymbol{x}^*$ та $\boldsymbol{y}^*$ гравців $\boldsymbol{I_1}$ та $\boldsymbol{I_2}$ еквівалентна парі двоїстих задач лінійного програмування*:

$$v \to \min,$$
$$c_{1j}x_1+\dots+c_{mj}x_m \leq v,\ j=1,\dots,n,$$
$$x_1+\dots+x_m = 1,\ x_i \geq 0,\ i=1,\dots,m;$$

$$v \to \max,$$
$$c_{i1}y_1+\dots+c_{in}y_n \geq v,\ i=1,\dots,m,$$
$$y_1+\dots+y_n = 1,\ y_j \geq 0,\ j=1,\dots,n.$$

Метод Брауна-Робінсон.

Метод Брауна-Робінсон являє собою ітеративний метод розв'язування матричної гри, з кожним кроком якого зв'язується деяка фіктивна гра, що розігрується в чистих стратегіях.

Нехай на *s*-у кроці отримані вектори

$$\boldsymbol{M}(s) = (m_1(s),\ldots,m_m(s)), \quad \boldsymbol{N}(s) = (n_1(s),\ldots,n_n(s)),$$

компоненти яких $m_i(s)$ та $n_j(s)$ відповідно рівні кількостям вибору *i*-ї та *j*-ї чистих стратегій першим та другим гравцями на попередніх кроках. Указані вектори визначають очевидно частоти вибору відповідних чистих стратегій гравців. Розглядаються також вектори відносних частот

$$\boldsymbol{x}(s) = (x_1(s),\ldots,x_m(s)) \text{ та } \boldsymbol{y}(s) = (y_1(s),\ldots,y_n(s))',$$

де $x_i(s) = m_i(s)/s,\ i=1,\ldots,m,\ \ y_j(s) = n_j(s)/s,\ j=1,\ldots,n.$

На *s*-у кроці величина

$$c_{i1}\, y_1(s) +\ldots+ c_{in}\, y_n(s), \quad i=1,\ldots,m,$$

визначає середній платіж першого гравця другому при умові, що перший гравець вибирає *i*-й рядок, величина

$$c_{1j}\, x_1(s) +\ldots+ c_{mj}\, x_m(s), \quad j=1,\ldots,n, \text{ —}$$

середній виграш другого гравця при умові, що другий гравець вибирає *j*-й стовпець.

На кожному кроці перший гравець вибирає рядок, якому відповідає мінімальне значення вказаного середнього платежу; другий гравець вибирає стовпець, який відповідає максимальному значенню вказаного середнього виграшу.

Якщо вказані оптимуми досягаються більш ніж для одного рядка (для першого гравця) або більш ніж для одного стовпця (для другого гравця), то вибирається рядок або стовпець з мінімальним номером. Після виконання гравцями вказаних дій переобчислюються всі згадані величини.

Теорема Брауна-Робінсон.

При необмеженому зростанні s величини $\boldsymbol{x}(s)$ та $\boldsymbol{y}(s)$ прямують до оптимальних змішаних стратегій $\boldsymbol{x}^*$ та $\boldsymbol{y}^*$ першого та другого гравців відповідно.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого матрична гра розв'язується в діалозі з користувачем за алгоритмом Брауна-Робінсон, завантажується з розділу «ЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом Брауна-Робінсон матричні ігри, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9).

Шляхом зведення до пари двоїстих задач лінійного програмування та методом Брауна-Робінсон знайти оптимальні змішані стратегії та ціну гри матричних ігор з такими платіжними матрицями:

1) $\begin{Vmatrix} -8 & 2 \\ 1 & -8 \end{Vmatrix}$, **2)** $\begin{Vmatrix} 2 & -5 \\ -4 & 3 \end{Vmatrix}$, **3)** $\begin{Vmatrix} -1 & 5 \\ 6 & -4 \end{Vmatrix}$, **4)** $\begin{Vmatrix} 5 & -6 \\ -3 & 0 \end{Vmatrix}$, **5)** $\begin{Vmatrix} -3 & 1 \\ 2 & -1 \end{Vmatrix}$,

6) $\begin{Vmatrix} -3 & 3 & 3 \\ -5 & 5 & -3 \\ 3 & -9 & 2 \end{Vmatrix}$, **7)** $\begin{Vmatrix} 0 & 1 & 6 \\ 7 & 1 & 3 \\ 1 & 2 & 0 \end{Vmatrix}$, **8)** $\begin{Vmatrix} 1 & 0 & -1 \\ 0 & 2 & 1 \\ 1 & -1 & 3 \end{Vmatrix}$, **9)** $\begin{Vmatrix} 1 & 0 & -1 \\ -1 & 2 & 3 \\ -2 & -3 & 3 \end{Vmatrix}$.

Лабораторна робота 17.

Методи одновимірної оптимізації

Метод золотого перерізу.

Метод золотого перерізу (*МЗП*) застосовується для пошуку мінімуму унімодальної функції однієї змінної $y=F(x)$, що задана на проміжку $[A,B]$. Алгоритм методу реалізується у вигляді послідовності кроків, на кожному з яких здійснюється звуження інтервалу, що містить точку мінімуму.

На початку обчислень покладають $A\{0\} = A$, $B\{0\} = B$.

На *s*-у кроці визначають величини

$$L\{s\} = B\{s\} - G(B\{s\} - A\{s\}),$$
$$R\{s\} = A\{s\} + G(B\{s\} - A\{s\}),$$

де стала $G = (\sqrt{5} - 1)/2 \approx 0.618$. Покладають

$$A\{s+1\} = A\{s\},\quad B\{s+1\} = R\{s\},\ \text{якщо}\ F(L\{s\}) \le F(R\{s\}),$$
$$A\{s+1\} = L\{s\},\quad B\{s+1\} = B\{s\},\ \text{якщо}\ F(L\{s\}) > F(R\{s\}).$$

Ітерації продовжують доти, поки не буде виконуватись нерівність

$$B\{s\} - A\{s\} \le \varepsilon,$$

де $\varepsilon > 0$ — задане число, яке визначає похибку розв'язку задачі.

На кожному кроці *МЗП*, починаючи з 1-го, обчислюється лише одне значення функції $F(x)$, тому що одна з точок золотого перерізу на попередньому кроці здійснює золотий переріз проміжку на наступному кроці.

За наближений розв'язок задачі приймають

$$x^* = (A\{s\} + B\{s\})/2,\quad y^* = F(x^*).$$

При розв'язуванні задачі максимізації функції $F(x)$ необхідно замінити її на функцію $-F(x)$.

Метод випадкового пошуку.

Метод випадкового пошуку застосовується для знаходження мінімуму (максимуму) довільної функції $y=F(\boldsymbol{x})$, що задана в будь-якій допустимій області D.

В програмі розглядається реалізація даного методу для функції однієї змінної.

Довільна функція $F(x)$ задана на проміжку $[A,B]$. За допомогою давача випадкових чисел, рівномірно розподілених на проміжку $[0,1]$, будується послідовність випадкових чисел $x\{k\}$, $k=1,\ldots,N$, рівномірно розподілених на проміжку $[A,B]$. Обчислюються та порівнюються між собою значення функції $F(x)$ в точках $x\{k\}$. Мінімальне з них приймається за оцінку мінімуму функції $F(x)$ на проміжку $[A,B]$.

Якщо N прямує до нескінченності, отримана оцінка за ймовірністю збігається до глобального мінімуму функції, що розглядається.

При розв'язуванні задачі максимізації функції $F(x)$ необхідно замінити її на функцію $-F(x)$.

Метод Фібоначчі.

Метод Фібоначчі (*МФ*) застосовується для пошуку мінімуму унімодальної функції однієї змінної $y=F(x)$, що задана на проміжку $[A,B]$.

Алгоритм методу реалізується у вигляді послідовності кроків, на кожному з яких здійснюється звуження інтервалу, що містить точку мінімуму.

На початку обчислень покладають $A\{0\} = A$, $B\{0\} = B$.

На *s*-у кроці визначають величини

$$L\{s\} = B\{s\} - G\{s\}\,(B\{s\} - A\{s\}),$$
$$R\{s\} = A\{s\} + G\{s\}\,(B\{s\} - A\{s\}),$$

де $G\{s\}=Fi(N-s-1)/Fi(N-s)$, $s=0,\ldots,N-3$, $G\{N-2\}=(1+\varepsilon)/2$ або $(1-\varepsilon)/2$, N — задане число ітерацій, $\varepsilon>0$, $Fi(j)$ — числа Фібоначчі, що задаються рекурентним співвідношенням

$$Fi(j) = Fi(j-1) + Fi(j-2),\ j \geq 2,\ Fi(0) = Fi(1) = 1.$$

Покладають

$$A\{s+1\} = A\{s\},\quad B\{s+1\} = R\{s\},\ \text{якщо}\ F(L\{s\}) \leq F(R\{s\}),$$
$$A\{s+1\} = L\{s\},\quad B\{s+1\} = B\{s\},\ \text{якщо}\ F(L\{s\}) > F(R\{s\}).$$

За наближений розв'язок задачі приймають

$$x^* = (A\{N\} + B\{N\})/2,\quad y^* = F(x^*).$$

Для *МФ* у випадку заздалегідь фіксованого числа ітерацій довжина кінцевого інтервалу пошуку мінімальна.

При розв'язуванні задачі максимізації функції $F(x)$ необхідно замінити її на функцію $-F(x)$.

Метод дихотомії.

Метод дихотомії (*МД*) застосовується для пошуку мінімуму унімодальної функції однієї змінної $y=F(x)$, що задана на проміжку $[A,B]$.

Алгоритм методу реалізується у вигляді послідовності кроків, на кожному з яких здійснюється звуження інтервалу, що містить точку мінімуму.

На початку обчислень покладають $A\{0\} = A$, $B\{0\} = B$.

На *s*-у кроці визначають величини

$$L\{s\} = (A\{s\} + B\{s\} - \Delta)/2,$$
$$R\{s\} = (A\{s\} + B\{s\} + \Delta)/2,$$

де $\Delta > 0$ — достатньо мале число. Покладають

$$A\{s+1\} = A\{s\},\quad B\{s+1\} = R\{s\},\ \text{якщо}\ F(L\{s\}) \leq F(R\{s\}),$$

$A\{s+1\} = L\{s\}, \quad B\{s+1\} = B\{s\},$ якщо $F(L\{s\}) > F(R\{s\})$.

Ітерації продовжують доти, поки не буде виконуватись нерівність

$B\{s\} - A\{s\} \le \varepsilon$,

де $\varepsilon > 0$ — задане число, яке визначає похибку розв'язку задачі.

За наближений розв'язок задачі приймають

$x^* = (A\{s\} + B\{s\})/2, \quad y^* = F(x^*)$.

При розв'язуванні задачі максимізації функції $F(x)$ необхідно замінити її на функцію $-F(x)$.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задачі одновимірної оптимізації розв'язуються в діалозі з користувачем за викладеними алгоритмами, завантажується з розділу «НЕЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методами золотого перетину, випадкового пошуку, Фібоначчі та дихотомії задачі одновимірної оптимізації, умови яких задаються модулем за допомогою команди «Дані» головного меню, а також знайти найбільше та найменше значення наступних функцій на вказаних проміжках:

1) $F(x) = |||x^2-1|-1|-1|$, $\quad x \in [-2, 2]$;

2) $F(x) = 2x^3-3x^2-12x+1+\sqrt{(4 - x^2)(1 + 2x^2)}$, $\quad x \in [-2, 2]$;

3) $F(x) = (x+1)^2 \ln(x+1)+x \exp(-x)$, $\quad x \in [0, e]$;

4) $F(x) = \sin(x) \sin(2x) + \arccos(x^2)$, $\quad x \in [-0.75, 0.75]$;

5) $F(x) = \mathrm{arctg}(x) - \ln(x)/2$, $\quad x \in [0.65, 1.75]$;

6) $F(x) = x+ \sqrt{x} + \exp(x)\, x^2$, $\quad x \in [0, 4]$;

7) $F(x) = 4x/(x^2 + 4)-[x^2 (x-2)]^{(2/5)}$, $\quad x \in [-2, 2]$.

Лабораторна робота 18.

Задача опуклого квадратичного програмування. Квадратичний симплекс-метод.

Постановка задачі нелінійного програмування.

Знайти вектор $\boldsymbol{x} = (x_1, \ldots, x_n)$, що мінімізує (максимізує) функцію

$$F(\boldsymbol{x}) = F(x_1, \ldots, x_n) \tag{18.1}$$

і задовольняє систему обмежень

$$G_i(x_1, \ldots, x_n)\ R_i\ 0, \quad i=1,\ldots,m, \tag{18.2}$$

де символ R_i замінює один із знаків $\le$, $=$, $\ge$.

Якщо хоча б одна з функцій F, G_i $(i=1,\ldots,m)$ є нелінійною, то вказана постановка визначає *задачу нелінійного програмування* (*ЗНЛП*).

Сукупність точок (векторів) $\boldsymbol{x}$, що задовольняють (18.2), зветься *допустимою областю* (*множиною*) і позначається через D.

Довільна точка D зветься *допустимим розв'язком* (*точкою, планом, вектором*).

Функція $F(\boldsymbol{x})$ співвідношення (18.1) зветься *цільовою функцією*.

Постановка задачі опуклого програмування.

Знайти вектор $\boldsymbol{x} = (x_1, \ldots, x_n)$, що мінімізує цільову функцію

$$F(\boldsymbol{x}) = F(x_1, \ldots, x_n) \tag{18.3}$$

і задовольняє систему обмежень

$$G_i(x_1, \ldots, x_n) \le 0, \ i=1,\ldots,m, \tag{18.4}$$

$$x_j \ge 0, \ j=1,\ldots,n, \tag{18.5}$$

де функції F, G_i $(i=1,\ldots,m)$ — опуклі.

Таким чином, для задачі опуклого програмування (*ЗОП*) цільова функція (18.3) — опукла, обмеження (18.4)–(18.5) визначають опуклу допустиму множину.

Постановка задачі опуклого квадратичного програмування.

Знайти вектор $\boldsymbol{x}$, що мінімізує цільову функцію

$$F(\boldsymbol{x}) = \boldsymbol{x}\, \boldsymbol{D}\, \boldsymbol{x}^T + \boldsymbol{c}\, \boldsymbol{x}^T \tag{18.6}$$

і задовольняє систему обмежень

$$\boldsymbol{A}\, \boldsymbol{x}^T \le \boldsymbol{b}^T, \ \ \boldsymbol{x} \ge \boldsymbol{0}, \tag{18.7}$$

де $\boldsymbol{c} = (c_1, \ldots, c_n)$, $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{D} = \| d_{kl} \|$, $k,l=1,\ldots n$, $\boldsymbol{A} = \| a_{ij} \|$, $i=1,\ldots,m$, $j=1,\ldots,n$, $\boldsymbol{b} = (b_1, \ldots, b_m)$, матриця $\boldsymbol{D}$ симетрична та невід'ємно визначена.

Таким чином, для задачі опуклого квадратичного програмування (*ЗОКП*) цільова функція (18.6) — опукла квадратична, обмеження (18.7) — лінійні і визначають опуклу многогранну допустиму множину.

Зауважимо, що для випадку $n=2$, $m=1$ *ЗОКП* має вигляд:

$$d_{11} x_1^2 + d_{22} x_2^2 + 2\, d_{12} x_1 x_2 + c_1 x_1 + c_2 x_2 \to \min,$$
$$a_{11} x_1 + a_{12} x_2 \le a_{10}, \ \ x_j \ge 0, \ j=1,2,$$

до того ж $\boldsymbol{D} = \| d_{kl} \|$, $k,l=1,2$, $d_{12} = d_{21}$.

Основні твердження.

Означення. Множина W зветься опуклою, якщо для довільних точок $\boldsymbol{x}, \boldsymbol{y} \in W$, та $t \in [0, 1]$ виконується $t\boldsymbol{x} + (1-t)\boldsymbol{y} \in W$.

Означення. Функція $H(\boldsymbol{x})$, $\boldsymbol{x} \in W \subset \boldsymbol{E}^n$, де W — опукла множина, зветься опуклою, якщо для довільних точок $\boldsymbol{x}, \boldsymbol{y} \in W$ та $t \in [0, 1]$ виконується нерівність $H(t\boldsymbol{x} + (1-t)\boldsymbol{y}) \le tH(\boldsymbol{x}) + (1-t)H(\boldsymbol{y})$.

Теорема. *Множина* $\{\boldsymbol{x} : H(\boldsymbol{x}) \le 0\}$, *де функція* $H(\boldsymbol{x})$ *— опукла, є опуклою.*

Теорема. *Перетин опуклих множин є опуклою множиною.*

Теорема. *Квадратична функція* $F(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{D}\boldsymbol{x}^T + \boldsymbol{c}\boldsymbol{x}^T$ *опукла, якщо матриця* $\boldsymbol{D}$ *— невід'ємно визначена.*

Теорема. *Матриця* $\boldsymbol{D}$ *— невід'ємно визначена, якщо всі її діагональні мінори невід'ємні*:

$$\det(\boldsymbol{D}(k)) \ge 0, \ \ \boldsymbol{D}(k) = \| d_{ij} \|, \ \ i,j=1,\ldots k.$$

Теорема Куна-Таккера-1.

Нехай допустима область $D=\{\boldsymbol{x}\in \boldsymbol{E}^n: G_i(\boldsymbol{x})\le 0,\ i=1,\dots,m,\ \boldsymbol{x}\ge 0\}$ *задачі опуклого програмування* (18.3)–(18.5) *задовольняє умову Слейтера: існує* $\boldsymbol{x}\in D$ *таке, що для всіх* $i=1,\dots,m$ $G_i(\boldsymbol{x})<0$, *тобто множина D має внутрішні точки.*

Тоді для того, щоб вектор $\boldsymbol{x}^*$ *був оптимальним розв'язком задачі опуклого програмування, необхідно і достатньо, щоб існував вектор* $\boldsymbol{u}^*\ge 0$ *такий, що пара* $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ *є сідловою точкою функції Лагранжа:*

$$L(\boldsymbol{x},\boldsymbol{u})= F(\boldsymbol{x})+(u_1\ G_1(\boldsymbol{x})\ +\dots+\ u_m\ G_m(\boldsymbol{x})),$$
$$\boldsymbol{x}=(x_1,\dots,x_n)\ge 0,\quad \boldsymbol{u}=(u_1,\dots,u_m)\ge 0.$$

Кажуть, що точка $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ є *сідловою точкою* функції $L(\boldsymbol{x},\boldsymbol{u})$, якщо для всіх $\boldsymbol{x}\ge 0$, $\boldsymbol{u}\ge 0$ мають місце нерівності

$$L(\boldsymbol{x}^*,\boldsymbol{u}) \le L(\boldsymbol{x}^*,\boldsymbol{u}^*) \le L(\boldsymbol{x},\boldsymbol{u}^*).$$

Теорема Куна-Таккера-2.

Для неперервно диференційовних функцій $F(\boldsymbol{x})$, $G_i(\boldsymbol{x})$, $i=1,\dots,m$, теорема Куна-Таккера може бути переформульована так.

Вектор $\boldsymbol{x}^*$ *є оптимальним розв'язком задачі опуклого програмування* (18.3) –(18.5) *тоді і лише тоді, коли існує вектор* $\boldsymbol{u}^*\ge 0$ *такий, що для пари* $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ *виконуються умови:*

$$\nabla_x L(\boldsymbol{x}^*,\boldsymbol{u}^*)\ge \boldsymbol{0},\quad \nabla_x L(\boldsymbol{x}^*,\boldsymbol{u}^*)(\boldsymbol{x}^*)^T=\boldsymbol{0},\quad j=1,\dots,n,$$
$$\nabla_u L(\boldsymbol{x}^*,\boldsymbol{u}^*)\le \boldsymbol{0},\quad \nabla_u L(\boldsymbol{x}^*,\boldsymbol{u}^*)(\boldsymbol{u}^*)^T=\boldsymbol{0},\quad i=1,\dots,m.$$

Для задачі (18.6)–(18.7) опуклого квадратичного програмування останні співвідношення набувають вигляду

$$\boldsymbol{c}+2\boldsymbol{x}\boldsymbol{D}+\boldsymbol{u}\boldsymbol{A} \ge \boldsymbol{0}, \tag{18. 8}$$
$$(\boldsymbol{c}+2\boldsymbol{x}\boldsymbol{D}+\boldsymbol{u}\boldsymbol{A})\boldsymbol{x}^T=\boldsymbol{0}, \tag{18. 9}$$
$$\boldsymbol{x}\boldsymbol{A}^T-\boldsymbol{b} \le \boldsymbol{0}, \tag{18.10}$$
$$(\boldsymbol{x}\boldsymbol{A}^T-\boldsymbol{b})\boldsymbol{u}^T=\boldsymbol{0}, \tag{18.11}$$
$$\boldsymbol{x}\ge \boldsymbol{0},\quad \boldsymbol{u}\ge \boldsymbol{0}.$$

Допоміжна ЗЛП для ЗОКП.

Якщо ввести вектори $\boldsymbol{y}=(y_1,\dots,y_n)\ge \boldsymbol{0}$ та $\boldsymbol{v}=(v_1,\dots,v_m)\ge \boldsymbol{0}$, то співвідношення (18.8)–(18.11) матимуть вигляд:

$$\boldsymbol{c} + 2\,\boldsymbol{x}\,\boldsymbol{D} + \boldsymbol{u}\,\boldsymbol{A} - \boldsymbol{y} = \boldsymbol{0}, \tag{18.12}$$
$$\boldsymbol{x}\,\boldsymbol{A}^T - \boldsymbol{b} + \boldsymbol{v} = \boldsymbol{0}, \tag{18.13}$$
$$\boldsymbol{y}\,\boldsymbol{x}^T = \boldsymbol{0},\quad \boldsymbol{v}\,\boldsymbol{u}^T = \boldsymbol{0}, \tag{18.14}$$
$$\boldsymbol{x}\ge \boldsymbol{0},\quad \boldsymbol{u}\ge \boldsymbol{0},\quad \boldsymbol{y}\ge \boldsymbol{0},\quad \boldsymbol{v}\ge \boldsymbol{0}.$$

Зауважимо, що для випадку $n=2$, $m=1$ матимемо

$$\begin{aligned}
&2\,d_{11}x_1 + 2\,d_{12}x_2 + a_{11}u - y_1 &&= -c_1,\\
&2\,d_{12}x_1 + 2\,d_{22}x_2 + a_{12}u \quad - y_2 &&= -c_2,\\
&a_{11}x_1 + \quad a_{12}x_2 \qquad\qquad\qquad + v &&= a_{10},\\
&x_1 y_1 = 0,\ x_2 y_2 = y_2,\ u\,v = 0,\\
&x_1,\ x_2,\ u,\ y_1,\ y_2,\ v \ge 0.
\end{aligned}$$

Система (18.12)–(18.13) складається з $n+m$ лінійних рівнянь відносно $2(m+n)$ невідомих x_j, y_j $(j=1,\dots,n)$, u_i, v_i $(i=1,\dots,m)$.

Крім того, як випливає з умов (18.14), повинно бути:

якщо $x_j > 0$, то $y_j = 0$, (18.15)
якщо $y_j > 0$, то $x_j = 0$, (18.16)
якщо $u_i > 0$, то $v_i = 0$, (18.17)
якщо $v_i > 0$, то $u_i = 0$. (18.18)

Отже, шуканим розв'язком системи (18.12)–(18.13) може бути довільний невід'ємний базисний її розв'язок, але такий, що змінні x_j та y_j (а також u_i та v_i) з однаковими індексами не можуть бути водночас базисними. Для відшукання такого розв'язку можна застосувати будь-який із відомих методів *ЛП*, зокрема, *метод штучного базису*. З цією метою запишемо систему (18.12)–(18.13) у вигляді

$$2\,\boldsymbol{x}\,\boldsymbol{D} + \boldsymbol{u}\,\boldsymbol{A} - \boldsymbol{y} = -\,\boldsymbol{c}, \quad (18.19)$$
$$\boldsymbol{x}\,\boldsymbol{A}^T + \boldsymbol{v} = \boldsymbol{b}. \quad (18.20)$$

Не обмежуючи загальності, будемо вважати, що праві частини цієї системи невід'ємні: $-\boldsymbol{c} \geq 0$, $\boldsymbol{b} \geq 0$ (оскільки, в іншому випадку відповідні рівняння, їх праву і ліву частину, досить помножити на -1). Згідно методу штучного базису в кожне рівняння системи (18.19)–(18.20), яке не містить базисної змінної, вводимо штучну змінну. Оскільки змінні v_i ($i=1,\dots,m$) можна вважати базисними, то штучні змінні $\boldsymbol{z} = (z_1, \dots, z_n) \geq 0$ вводимо тільки в рівняння (18.19) і розглядаємо допоміжну *КЗЛП*

$$\boldsymbol{z}\,\boldsymbol{i}^T \to \min, \quad (18.21)$$
$$2\,\boldsymbol{x}\,\boldsymbol{D} + \boldsymbol{u}\,\boldsymbol{A} - \boldsymbol{y} + \boldsymbol{z} = -\,\boldsymbol{c}, \quad (18.22)$$
$$\boldsymbol{x}\,\boldsymbol{A}^T + \boldsymbol{v} = \boldsymbol{b}, \quad (18.23)$$
$$\boldsymbol{x} \geq 0,\ \boldsymbol{u} \geq 0,\ \boldsymbol{y} \geq 0,\ \boldsymbol{v} \geq 0,\ \boldsymbol{z} \geq 0, \quad (18.24)$$

де $\boldsymbol{i} = (1, 1, \dots, 1)$ — n-вимірний одиничний вектор.

Квадратичний симплекс-метод.

Задача (18.21)–(18.24) розв'язується симплекс-методом.

Якщо знайдений *ДБР* цієї задачі задовольняє *умови доповнюючої нежорсткості* (18.15)–(18.18), то він визначає оптимальний розв'язок вихідної *ЗОКП*. Інакше треба перейти до нового *ДБР*. При цьому до базисних включається нова змінна з нульовою оцінкою.

Симплекс-метод з умовами (18.15)–(18.18) для розв'язування допоміжної *КЗЛП* (18.21)–(18.24), побудованої на основі задачі опуклого квадратичного програмування (18.6)–(18.7), і називають *квадратичним симплекс-методом* (алгоритм звичайного симплекс-методу, а також всі зв'язані з ним означення і твердження наведені в методичних вказівках до виконання лабораторної роботи №2).

Якщо в оптимальному розв'язку допоміжної *КЗЛП* (18.21)–(18.24) всі штучні змінні z_j ($j=1,\dots,n$) приймають нульові значення, то, відкидаючи їх, отримаємо *ДБР* системи (18.19)–(18.20). Та його частина, яка відповідає змінним початкової задачі опуклого квадратичного програмування (18.6)–(18.7), і буде її оптимальним розв'язком.

Якщо ж в оптимальному розв'язку допоміжної *КЗЛП* (18.21)–(18.24) значення хоча б однієї із штучних змінних відмінне від нуля, то система (18.19)–(18.20) розв'язків не має, а, отже, множина сідлових точок функції Лагранжа початкової задачі опуклого квадратичного програмування порожня.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача опуклого квадратичного програмування розв'язується в діалозі з користувачем за викладеним алгоритмом, завнтажується з розділу «НЕЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Геометрична інтерпретація задачі нелінійного програмування здійснюється модулем, який завантажується з того ж самого розділу.

Завдання.

Розв'язати квадратичним симплекс-методом задачі опуклого квадратичного програмування, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9), а також наступні задачі.

У всіх задачах виконуються умови: $x_1 \geq 0$, $x_2 \geq 0$.

1) $x_1^2 + x_2^2 - 2x_1 - 4x_2 \to \min$, $2x_1 + 3x_2 \leq 6$;

2) $-2x_1^2 - 3x_2^2 + 4x_1x_2 + 12x_2 \to \max$, $3x_1 + 4x_2 \leq 12$;

3) $2x_1^2 + x_2^2 - x_1x_2 - x_1 \to \min$, $x_1 + 2x_2 \leq 1$;

4) $x_1^2 + 3x_2^2 - x_1 - 2x_2 \to \min$, $x_1 + 4x_2 \leq 7$;

5) $-2x_1^2 - 3x_2^2 + 16x_1 + 24x_2 \to \max$, $2x_1 + x_2 \leq 4$;

6) $x_1^2 + x_2^2 - 3x_1 - 8x_2 \to \min$, $x_1 + 2x_2 \leq 4$;

7) $-x_1^2 - x_2^2 + x_1 + 2x_2 \to \max$, $x_1 + 2x_2 \leq 16$;

8) $-x_1^2 - x_2^2 + x_1x_2 + 5x_1 + 2x_2 \to \max$, $2x_1 + 3x_2 \leq 15$.

Відповіді:

1) $\boldsymbol{x}^*$ = (0.69; 1.54). **2)** $\boldsymbol{x}^*$ = (1.23; 2.07). **3)** $\boldsymbol{x}^*$ = (0.29; 0.14). **4)** $\boldsymbol{x}^*$ = (0.5; 0.33).
5) $\boldsymbol{x}^*$ = (0.57; 2.86). **6)** $\boldsymbol{x}^*$ = (0.4; 1.8). **7)** $\boldsymbol{x}^*$ = (0.5; 1). **8)** $\boldsymbol{x}^*$ = (3.63; 2.58).

Лабораторна робота 19.

Задача безумовної оптимізації. Метод найшвидшого спуску

Постановка задачі безумовної оптимізації.

Знайти вектор $\boldsymbol{x}=(x_1,\ldots,x_n)$, що мінімізує (максимізує) функцію

$$F(\boldsymbol{x})=F(x_1,\ldots,x_n).$$

Градієнтні методи безумовної оптимізації.

Для задачі безумовної мінімізації метод полягає в обчисленні послідовності наближень $\boldsymbol{x}[s]$ за правилом

$$\boldsymbol{x}[s+1] = \boldsymbol{x}[s] - \rho[s]\,\nabla F(\boldsymbol{x}[s]),$$

де $\nabla F(.)$ — градієнт функції $F(.)$, який задається співвідношенням

$$\nabla F(.) = \left(\frac{\partial}{\partial\ x_1} F(.)\ , \ldots ,\ \frac{\partial}{\partial\ x_n} F(.) \right),$$

$\rho[s] > 0$ — крок, величина якого визначається конкретним градієнтним методом. Початковий розв'язок $\boldsymbol{x}[0]$ обирається довільно. Ітерації припиняють, якщо на деякому кроці s виконується нерівність $\|\nabla F(\boldsymbol{x}[s])\| < \varepsilon$, де норма градієнта визначається формулою

$$\|\nabla F(.)\| = \sqrt{\left(\frac{\partial}{\partial\ x_1} F(.) \right)^2 + \ldots + \left(\frac{\partial}{\partial\ x_n} F(.) \right)^2},$$

а $\varepsilon > 0$ — деяка наперед задана величина, що визначає точність розв'язку.

Метод найшвидшого спуску.

Метод найшвидшого спуску являє собою градієнтний метод, в якому величина кроку $\rho[s]$ вибирається за правилом

$$F(\boldsymbol{x}[s] - \rho[s] \nabla F(\boldsymbol{x}[s])) = \min(F(\boldsymbol{x}[s] - \rho \nabla F(\boldsymbol{x}[s]))),$$

де мінімум береться по всіх $\rho > 0$.

При деяких умовах $\boldsymbol{x}[s]$ прямує до стаціонарної точки функції $F(\boldsymbol{x})$, якщо s прямує до нескінченності.

Програмне забезпечення.

Навчаючий модуль, за допомогою якого задача безумовної оптимізації розв'язується в діалозі з користувачем методом найшвидшого спуску, завантажується з розділу «НЕЛІНІЙНЕ ПРОГРАМУВАННЯ» головного меню пакета *ПЗ–МО*.

Завдання.

Розв'язати методом найшвидшого спуску задачі безумовної оптимізації, умови яких задаються модулем за допомогою команди «Дані» головного меню (задачі №1–№9).

Література

1. **Ю.М.Ермольев, И.И.Ляшко, В.С.Михалевич, В.И.Тюптя.** Математические методы исследования операций. Киев, «Вища школа», 1979.
2. **Ю.Д.Попов, В.І.Тюптя, В.І.Шевченко.** Методи оптимізації. Киев, Абрис, 1999.
3. **Ф.П.Васильев.** Численные методы решения экстремальных задач. Москва, «Наука», 1980.
4. **И.Л.Калихман.** Сборник задач по математическому программированию. Москва, «Высшая школа», 1975.
5. **В.Ф.Капустин.** Практические занятия по курсу математического программирования. Издательство Ленинградского университета, 1976.
6. **Ю.П.Зайченко, С.А.Шумилова.** Исследование операций, сборник задач. Киев, «Вища школа», 1990.

Зміст

Навчальне електронне видання

Попов Юрій Дмитрович

ТЮПТЯ Володимир Іванович

ШЕВЧЕНКО Віталій Іванович

Методичні рекомендації до виконання лабораторних робіт з методів оптимізації на персональних комп'ютерах