



СИСТЕМНИЙ АНАЛІЗ

УДК 519.85

П.І. СТЕЦЮК

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

В.О. СТОВБА

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

С.С. ТРЕГУБЕНКО

Центральний науково-дослідний інститут Збройних Сил України, Київ, Україна,
e-mail: info-cvni@ukr.net.

О.М. ХОМ'ЯК

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

МОДИФІКАЦІЇ ДВОЕТАПНОЇ ТРАНСПОРТНОЇ ЗАДАЧІ ТА ЇХНІ ЗАСТОСУВАННЯ¹

Анотація. Досліджено математичні моделі лінійного та Булевого лінійного програмування для збалансованої двоетапної транспортної задачі та двох її модифікацій. Перша модифікація враховує верхні межі пропускних спроможностей проміжних пунктів, а друга модифікація дає змогу вибирати фіксовану кількість проміжних пунктів, меншу, ніж їхня загальна кількість. Для всіх випадків обґрунтовано необхідні та достатні умови сумісності систем лінійних обмежень відповідних задач математичного програмування. Розглянуто приклади застосування модифікацій для оптимального розбиття множини, задачі про p -медіану графу та оптимального розміщення заданої кількості складів. Наведено результати обчислювальних експериментів з використанням солверів Gurobi та CPLEX та мови моделювання AMPL.

Ключові слова: двоетапна транспортна задача, задача лінійного програмування, задача Булевого лінійного програмування, AMPL, Gurobi, CPLEX, оптимальне розбиття множини, p -медіана графу, метод k -середніх, матеріально-технічне забезпечення.

ВСТУП

У роботах [1, 2] досліджено двоетапні неперервно-дискретні задачі оптимального розбиття-розподілу із заданими положеннями центрів підмножин та з пошуком оптимальних координат розташування центрів підмножин. Ці задачі розв'язуються двома етапами і полягають у визначенні зон збору неперервно розподіленого ресурсу (сировини) підприємствами і обсягів перевезень переробленого продукту від підприємств (перший етап) до споживачів (другий етап) з метою мінімізації сумарних витрат на транспортування ресурсу від постачальників через пункти переробки (збору, зберігання) до споживачів.

Окремим випадком таких задач є класична двоетапна транспортна задача [3, 4], де вантаж (продукцію) перевозять постачальники до споживачів тільки через проміжні пункти.

¹Робота підтримана CRDF Global (грант G-202102-68020).

© П.І. Стецюк, В.О. Стовба, С.С. Трегубенко, О.М. Хом'як, 2022

Розроблення моделей двоетапних (багатоетапних) транспортних задач є важливим напрямком досліджень, які є основою для моделювання різних процесів транспортування та зберігання виробленої продукції. Використовуються як збалансовані задачі, в яких обсяг продукції, що перевозять постачальники, дорівнює обсягу продукції, необхідної споживачам, так і незбалансовані задачі, в яких ця умова не виконується. Актуальним є визначення необхідних та достатніх умов сумісності систем обмежень для таких задач, що дає змогу перевіряти існування розв'язку задачі до початку її розв'язання.

Необхідні та достатні умови існування розв'язків збалансованої та незбалансованої двоетапної транспортної задачі та їхні модифікації розглянуто в [5–7]. У роботі [5] наведено обґрунтування необхідних умов несумісності системи лінійних обмежень для збалансованої двоетапної транспортної задачі. У статті [6] обґрунтовано необхідні умови несумісності системи лінійних обмежень для модифікації цієї задачі, в якій враховано верхні межі пропускних спроможностей проміжних пунктів, а також можливість вибору певної кількості проміжних пунктів. Стаття [7] містить обґрунтування необхідних та достатніх умов сумісності систем обмежень для незбалансованої двоетапної транспортної задачі та двох її модифікацій. У цій роботі, яка є продовженням роботи [8], розглянуто збалансовану двоетапну транспортну задачу та дві її модифікації, а також обґрунтовано необхідні та достатні умови існування їхніх розв'язків.

Стаття має таку структуру. У розд. 1 описано збалансовану двоетапну транспортну задачу та наведено обґрунтування необхідних та достатніх умов сумісності системи лінійних обмежень цієї задачі.

У розд. 2 наведено модифікацію збалансованої двоетапної транспортної задачі, в якій існують верхні межі пропускних спроможностей проміжних пунктів. Умотивовано необхідні та достатні умови сумісності системи лінійних обмежень задачі.

У розд. 3 досліджено модифікацію збалансованої двоетапної транспортної задачі, в якій крім верхніх меж пропускних спроможностей проміжних пунктів враховано можливість вибору фіксованої кількості цих пунктів. Наведено обґрунтування необхідних та достатніх умов існування розв'язку задачі.

У розд. 4 описано застосування збалансованої двоетапної транспортної задачі та її модифікації, де задано верхні межі пропускних спроможностей проміжних пунктів, до модельної задачі оптимального розбиття множини, дослідженої в роботі [1]. Наведено графічні результати розбиття множин для двох задач з різними обсягами ресурсів у пунктах другого етапу та розбиття множини постачальників з урахуванням меж пропускних спроможностей проміжних пунктів.

У розд. 5 проілюстровано подібність другої модифікації двоетапної транспортної задачі із задачею про p -медіану графу [9, 10] та з методом k -середніх [11, 12] для задачі кластеризації. Наведено графічні результати розв'язання цієї модифікації як задачі про p -медіану з різною кількістю споживачів та двома варіантами відстаней між точками.

У розд. 6 описано використання другої модифікації двоетапної транспортної задачі для раціонального розміщення заданої кількості складів з урахуванням визначеного положення постачальників та отримувачів матеріально-технічних засобів на території, де вони виконують свої завдання [6]. Наведено графічні та числові результати розв'язання задачі для порівняння солверів Gurobi та CPLEX.

Результати обчислювальних експериментів, наведених у розд. 4–6, отримано за допомогою солверів Gurobi [13] та CPLEX [14] з NEOS-сервера [15] та мови моделювання AMPL [16].

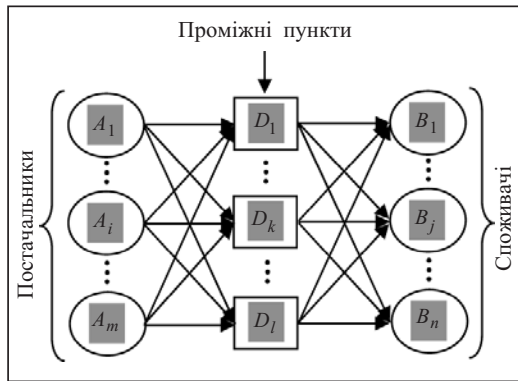


Рис. 1. Схема перевезень постачальники — проміжні пункти — споживачі

1. ДВОЕТАПНА ТРАНСПОРТНА ЗАДАЧА

Нехай є m пунктів постачання A_1, \dots, A_m , в яких знаходиться a_1, \dots, a_m одиниць продукції. Частину продукції потрібно перевезти до n споживачів B_1, \dots, B_n , задовольнивши їхні потреби b_1, \dots, b_n . Для транспортування продукції від постачальників до споживачів задіяно l проміжних пунктів D_1, \dots, D_l . Потрібно скласти оптимальний план транспортування продукції, де c_{ik} — витрати на перевезення одиниці продукції від

постачальника A_i до проміжного пункту D_k , а c_{kj} — витрати на перевезення одиниці продукції від проміжного пункту D_k до споживача B_j .

Схему функціонування перевезень постачальники — проміжні пункти — споживачі наведено на рис. 1.

Нехай $x = \{x_{ik}\}_{i=1, \dots, m}^{k=1, \dots, l}$, де x_{ik} — кількість одиниць продукції, яку перевозять від постачальника A_i до пункту D_k ; $y = \{y_{kj}\}_{k=1, \dots, l}^{j=1, \dots, n}$, де y_{kj} — кількість продукції, яку транспортують від пункту D_k до споживача B_j .

Класична двоетапна транспортна задача, яка визначає найоптимальніший план перевезення продукції від постачальників до споживачів через проміжні пункти, має такий вигляд:

знайти

$$f_{xy}^* = f_{xy}(x^*, y^*) = \min_{x, y} \left\{ f(x, y) = \sum_{i=1}^m \sum_{k=1}^l c_{ik} x_{ik} + \sum_{k=1}^l \sum_{j=1}^n c_{kj} y_{kj} \right\} \quad (1)$$

за обмежень

$$\sum_{k=1}^l x_{ik} = a_i, \quad i = \overline{1, m}, \quad (2)$$

$$\sum_{k=1}^l y_{kj} = b_j, \quad j = \overline{1, n}, \quad (3)$$

$$\sum_{i=1}^m x_{ik} - \sum_{j=1}^n y_{kj} = 0, \quad k = \overline{1, l}, \quad (4)$$

$$x_{ik} \geq 0, \quad y_{kj} \geq 0, \quad i = \overline{1, m}, \quad k = \overline{1, l}, \quad j = \overline{1, n}. \quad (5)$$

У подальшому вважатимемо, що $m \geq 2, n \geq 2, l \geq 2$ і крім того $a_i > 0, b_j > 0, i = \overline{1, m}, j = \overline{1, n}$, оскільки в протилежному випадку задача занадто спрощується і, як впливає з (2)–(5), частина змінних $x_{ik}, y_{kj}, i = \overline{1, m}, k = \overline{1, l}$, набуває нульових значень та може бути вилучена з розгляду.

Задача (1)–(5) є задачею лінійного програмування (ЛП-задача), яка містить $(m+n) \times l$ неперервних змінних x та y , а також $m+n+l$ лінійних обмежень. Цільова функція $f(x, y)$ задає сумарні витрати на транспортування продукції від постачальників до споживачів. Обмеження (2) означають транспортування a_1, \dots, a_m одиниць продукції із пунктів постачання до проміжних пунктів, а об-

меження (3) — що споживачам потрібно доставити необхідні обсяги b_1, \dots, b_n одиниць продукції з проміжних пунктів. Обмеження (4) задають умови на те, щоб уся продукція, яка надходить від постачальників до кожного проміжного пункту, має бути обов'язково відправлена споживачам.

Задача (1)–(5) належить до збалансованих транспортних задач, тобто всю продукцію постачальників потрібно доставити споживачам, не залишаючи нічого в проміжних пунктах. Це визначає умови сумісності системи обмежень (2)–(5) — системи лінійних рівностей. Справедливим є таке твердження.

Твердження 1. Система обмежень (2)–(5) є сумісною тоді і тільки тоді, коли виконується умова

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (6)$$

Доведення. Необхідність. Нехай існують невід'ємні $\bar{x} = \{\bar{x}_{ik}, i = \overline{1, m}, k = \overline{1, l}\}$ та $\bar{y} = \{\bar{y}_{kj}, k = \overline{1, l}, j = \overline{1, n}\}$, які задовольняють систему обмежень (2)–(4), тобто для $\bar{x} \geq 0$ та $\bar{y} \geq 0$ виконуються такі рівності:

$$\sum_{k=1}^l \bar{x}_{ik} = a_i, \quad i = \overline{1, m}, \quad (7)$$

$$\sum_{k=1}^l \bar{y}_{kj} = b_j, \quad j = \overline{1, n}, \quad (8)$$

$$\sum_{i=1}^m \bar{x}_{ik} - \sum_{j=1}^n \bar{y}_{kj} = 0, \quad k = \overline{1, l}. \quad (9)$$

Підсумувавши рівності (7)–(9) за індексами $i = \overline{1, m}$, $j = \overline{1, n}$ та $k = \overline{1, l}$ відповідно, отримуємо

$$\sum_{i=1}^m \sum_{k=1}^l \bar{x}_{ik} = \sum_{i=1}^m a_i, \quad (10)$$

$$\sum_{j=1}^n \sum_{k=1}^l \bar{y}_{kj} = \sum_{j=1}^n b_j, \quad (11)$$

$$\sum_{k=1}^l \sum_{i=1}^m \bar{x}_{ik} - \sum_{k=1}^l \sum_{j=1}^n \bar{y}_{kj} = 0. \quad (12)$$

Використовуючи рівності (10)–(12) та змінюючи порядок індексів, одержимо такий ланцюжок рівностей:

$$\sum_{i=1}^m a_i = \sum_{k=1}^l \sum_{i=1}^m \bar{x}_{ik} = \sum_{k=1}^l \sum_{j=1}^n \bar{y}_{kj} = \sum_{j=1}^n b_j. \quad (13)$$

Звідси випливає, що $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$.

Достатність. Нехай виконується рівність $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. Визначимо величини \tilde{x}_{ik} та \tilde{y}_{kj} , $i = \overline{1, m}$, $k = \overline{1, l}$, $j = \overline{1, n}$:

$$\tilde{x}_{ik} = \frac{a_i}{l}, \quad \tilde{y}_{kj} = \frac{b_j}{l}, \quad k = \overline{1, l}. \quad (14)$$

Покажемо, що ці величини задовольняють обмеження задачі (1)–(5). Дійсно,

- 1) $\sum_{k=1}^l \tilde{x}_{ik} = \frac{a_i}{l} \cdot l = a_i, \quad i = \overline{1, m};$
- 2) $\sum_{k=1}^l \tilde{y}_{kj} = \frac{b_j}{l} \cdot l = b_j, \quad j = \overline{1, n};$
- 3) $\sum_{i=1}^m \tilde{x}_{ik} - \sum_{j=1}^n \tilde{y}_{kj} = \frac{1}{l} \sum_{i=1}^m a_i - \frac{1}{l} \sum_{j=1}^n b_j = 0, \quad k = \overline{1, l};$
- 4) $\tilde{x}_{ik} \geq 0, \quad \tilde{y}_{kj} \geq 0,$ оскільки $a_i > 0, \quad b_j > 0.$

Отже, система обмежень (2)–(5) має допустиму точку (14).

Зазначимо, що за умови (6) обмеження (2)–(5) є лінійно залежними та одну з рівностей (2), (3), причому довільну, можна вилучити.

Зауважимо, що твердження 1 можна довести й іншим способом, а саме: зафіксувати один або декілька проміжних пунктів, через які продукція постачатиметься споживачам, а обсяги постачань через інші пункти мають дорівнювати нулю. Доведення в такий спосіб аналогічного твердження для незбалансованої двоетапної транспортної задачі наведено в роботі [7].

2. ПЕРША МОДИФІКАЦІЯ ЗАДАЧІ (1)–(5)

Перша модифікація враховує верхні межі пропускних спроможностей проміжних пунктів, де $d_1^{up}, \dots, d_l^{up}$ — максимальні спроможності пунктів D_1, \dots, D_l . Двоетапна транспортна задача з обмеженнями на пропускні спроможності проміжних пунктів має такий вигляд:

знайти

$$F_{xy}^* = F_{xy}(x^*, y^*) = \min_{x, y} \left\{ F(x, y) = \sum_{i=1}^m \sum_{k=1}^l c_{ik} x_{ik} + \sum_{k=1}^l \sum_{j=1}^n c_{kj} y_{kj} \right\} \quad (15)$$

за таких обмежень:

$$\sum_{k=1}^l x_{ik} = a_i, \quad i = \overline{1, m}, \quad (16)$$

$$\sum_{k=1}^l y_{kj} = b_j, \quad j = \overline{1, n}, \quad (17)$$

$$\sum_{i=1}^m x_{ik} - \sum_{j=1}^n y_{kj} = 0, \quad k = \overline{1, l}, \quad (18)$$

$$\sum_{i=1}^m x_{ik} \leq d_k^{up}, \quad k = \overline{1, l}, \quad (19)$$

$$x_{ik} \geq 0, \quad y_{kj} \geq 0, \quad i = \overline{1, m}, \quad k = \overline{1, l}, \quad j = \overline{1, n}. \quad (20)$$

Задача (15)–(20) є задачею лінійного програмування та містить $(m+n) \times l$ змінних x та y , а також $m+n+2l$ лінійних обмежень, які складають $m+n+l$ рівностей та l нерівностей. Тут цільова функція $F(x, y)$ та обмеження (16)–(20) мають такий самий зміст, як і цільова функція та обмеження задачі (1)–(5). Обмеження (19) задають верхні межі пропускних спроможностей проміжних пунктів. Їх також можна записати у вигляді

$$\sum_{j=1}^n y_{kj} \leq d_k^{up}, \quad k = \overline{1, l}. \quad (21)$$

Задача (15)–(20) належить до збалансованих транспортних задач, і для неї справедливим є таке твердження.

Твердження 2. Система обмежень (16)–(20) є сумісною тоді і тільки тоді, коли виконується умова

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \leq \sum_{k=1}^l d_k^{up}. \quad (22)$$

Доведення. Необхідність. Справедливість рівності $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ випливає з доведення необхідності в твердженні 1. Покажемо справедливість нерівності $\sum_{j=1}^n b_j \leq \sum_{k=1}^l d_k^{up}$. Підсумувавши рівність (16) за індексом $i = \overline{1, m}$ та рівність (19)

за індексом $k = \overline{1, l}$, з урахуванням рівності $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ отримуємо

$$\sum_{j=1}^n b_j = \sum_{i=1}^m a_i = \sum_{i=1}^m \sum_{k=1}^l \bar{x}_{ik} \leq \sum_{k=1}^l d_k^{up},$$

звідси випливає справедливість нерівності $\sum_{j=1}^n b_j \leq \sum_{k=1}^l d_k^{up}$.

Достатність. Нехай виконується умова (22). Визначимо індекс k^* у такий спосіб:

$$\sum_{k=1}^{k^*} d_k^{up} \leq \sum_{j=1}^n b_j, \quad \sum_{k=1}^{k^*+1} d_k^{up} > \sum_{j=1}^n b_j. \quad (23)$$

Розглянемо величини \tilde{x}_{ik} та \tilde{y}_{kj} , $i = \overline{1, m}$, $k = \overline{1, l}$, $j = \overline{1, n}$, і обчислимо їх:

$$\tilde{x}_{ik} = \begin{cases} a_i t, & k \leq k^*, \\ a_i t^*, & k = k^* + 1, \\ 0 & \text{інакше,} \end{cases} \quad \tilde{y}_{kj} = \begin{cases} b_j t, & k \leq k^*, \\ b_j t^*, & k = k^* + 1, \\ 0 & \text{інакше,} \end{cases} \quad (24)$$

де $t = d_k^{up} / \sum_{i=1}^m a_i$, $t^* = \left(\sum_{i=1}^m a_i - \sum_{k=1}^{k^*} d_k^{up} \right) / \sum_{i=1}^m a_i$. Покажемо, що величини \tilde{x}_{ik} та \tilde{y}_{kj} задовольняють обмеження (16)–(20):

$$1) \sum_{k=1}^l \tilde{x}_{ik} = \sum_{k=1}^{k^*} a_i t + a_i t^* = a_i \left(\sum_{k=1}^{k^*} t + t^* \right) = a_i \left(\frac{\sum_{k=1}^{k^*} d_k^{up} + \sum_{i=1}^m a_i - \sum_{k=1}^{k^*} d_k^{up}}{\sum_{i=1}^m a_i} \right) = a_i,$$

$i = \overline{1, m}$;

$$2) \sum_{k=1}^l \tilde{y}_{kj} = \sum_{k=1}^{k^*} b_j t + b_j t^* = b_j \left(\sum_{k=1}^{k^*} t + t^* \right) = b_j, \quad j = \overline{1, n};$$

3) для $k \leq k^*$ маємо

$$\sum_{i=1}^m \tilde{x}_{ik} - \sum_{j=1}^n \tilde{y}_{kj} = t \sum_{i=1}^m a_i - t \sum_{j=1}^n b_j = t \left(\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \right) = 0,$$

а для $k = k^* + 1$ маємо

$$\sum_{i=1}^m \tilde{x}_{i,k^*+1} - \sum_{j=1}^n \tilde{y}_{k^*+1,j} = t^* \sum_{i=1}^m a_i - t^* \sum_{j=1}^n b_j = t^* \left(\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \right) = 0;$$

4) для $k \leq k^*$ маємо

$$\sum_{i=1}^m \tilde{x}_{ik} = \sum_{i=1}^m a_i t = \frac{d_k^{up}}{\sum_{i=1}^m a_i} \sum_{i=1}^m a_i = d_k^{up},$$

а для $k = k^* + 1$

$$\sum_{i=1}^m \tilde{x}_{i,k^*+1} = \sum_{i=1}^m a_i t^* = \frac{\sum_{i=1}^m a_i - \sum_{k=1}^{k^*} d_k^{up}}{\sum_{i=1}^m a_i} \sum_{i=1}^m a_i = \sum_{i=1}^m a_i - \sum_{k=1}^{k^*} d_k^{up} \leq d_{k^*+1}^{up};$$

5) невід'ємність величин \tilde{x}_{ik} та \tilde{y}_{kj} впливає з їхньої побудови.

Отже, система обмежень (16)–(20) має допустиму точку (24).

Для одних і тих самих даних оптимальне значення функції F_{xy}^* задачі (15)–(20)

більше або дорівнює оптимальному значенню функції f_{xy}^* задачі (1)–(5), причому рівність досягається тоді і тільки тоді, коли жодне з обмежень на верхні межі (19) не є активним.

3. ДРУГА МОДИФІКАЦІЯ ЗАДАЧІ (1)–(5)

Друга модифікація двоетапної транспортної задачі враховує можливість вибору d — фіксованої кількості проміжних пунктів, яка є меншою за їхню загальну кількість l . Нехай $z = \{z_k\}_{k=\overline{1,l}}$ — Булеві змінні, де z_k дорівнює одиниці, якщо проміжний пункт D_k використовується, та нулю — в протилежному випадку. Двоетапна транспортна задача із заданою кількістю проміжних пунктів d ($1 < d < l$) формулюється так:

знайти

$$F_{xyz}^* = F_{xyz}(x^*, y^*) = \min_{x,y,z} \left\{ F(x, y) = \sum_{i=1}^m \sum_{k=1}^l c_{ik} x_{ik} + \sum_{k=1}^l \sum_{j=1}^n c_{kj} y_{kj} \right\} \quad (25)$$

за обмежень

$$\sum_{k=1}^l x_{ik} = a_i, \quad i = \overline{1, m}, \quad (26)$$

$$\sum_{k=1}^l y_{kj} = b_j, \quad j = \overline{1, n}, \quad (27)$$

$$\sum_{i=1}^m x_{ik} - \sum_{j=1}^n y_{kj} = 0, \quad k = \overline{1, l}, \quad (28)$$

$$\sum_{i=1}^m x_{ik} \leq d_k^{up} z_k, \quad k = \overline{1, l}, \quad (29)$$

$$\sum_{k=1}^l z_k = d, \quad (30)$$

$$x_{ik} \geq 0, \quad y_{kj} \geq 0, \quad z_k = 0 \vee 1, \quad i = \overline{1, m}, \quad k = \overline{1, l}, \quad j = \overline{1, n}. \quad (31)$$

Задача (25)–(31) є задачею Булевого лінійного програмування: містить $(m+n) \times l$ неперервних змінних x і y та l Булевих змінних z . Задача містить $m+n+2l+1$ обмежень. Тут цільова функція $F(x, y)$ має такий самий зміст, як і в задачах (1)–(5) та (15)–(20), але це стосується обмеженої кількості проміжних пунктів. Обмеження (30) визначають, що задіяно рівно d проміжних пунктів, а двобічні обмеження (29) задають для них верхні межі на пропускну спроможність. Зауважимо, що обмеження (29) можна записати у вигляді

$$\sum_{j=1}^n y_{kj} \leq d_k^{up} z_k, \quad k = \overline{1, l}. \quad (32)$$

Задача (25)–(31) належить до збалансованих транспортних задач, для яких справедливим є таке твердження.

Твердження 3. Система обмежень (26)–(31) є сумісною тоді і тільки тоді, коли виконується умова

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \leq \sum_{k=1}^d d_k^{up}, \quad (33)$$

де $d_1^{up} \geq d_2^{up} \geq \dots \geq d_l^{up}$.

Доведення твердження 3 аналогічне доведенню твердження 2 для $l = d$. Зауважимо, що для $l = d$ задача (25)–(31) є еквівалентною задачі (15)–(20). Якщо $l = d$ та верхні межі $d_k^{up} \geq \sum_{i=1}^m a_i$, $k = \overline{1, l}$, то задача (25)–(31) є еквівалентною задачі (1)–(5).

Булевий аналог такої задачі розглянуто в роботі [6]. Він має такий вигляд:

$$f_{xyz}^* = f_{xyz}(x^*, y^*) = \min_{x, y, z} \left\{ f(x, y) = \sum_{i=1}^m \sum_{k=1}^l c_{ik} x_{ik} + \sum_{k=1}^l \sum_{j=1}^n c_{kj} y_{kj} \right\} \quad (34)$$

за обмежень

$$\sum_{k=1}^l x_{ik} = a_i, \quad i = \overline{1, m}, \quad (35)$$

$$\sum_{k=1}^l y_{kj} = b_j, \quad j = \overline{1, n}, \quad (36)$$

$$\sum_{i=1}^m x_{ik} - \sum_{j=1}^n y_{kj} = 0, \quad k = \overline{1, l}, \quad (37)$$

$$\sum_{i=1}^m x_{ik} \leq z_k \sum_{i=1}^m a_i, \quad k = \overline{1, l}, \quad (38)$$

$$\sum_{k=1}^l z_k = d, \quad (39)$$

$$x_{ik} \geq 0, \quad y_{kj} \geq 0, \quad z_k = 0 \vee 1, \quad i = \overline{1, m}, \quad k = \overline{1, l}, \quad j = \overline{1, n}. \quad (40)$$

Твердження 4. Система обмежень (35)–(40) є сумісною тоді і тільки тоді, коли виконується умова

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (41)$$

Задача (34)–(40) не зміниться, якщо у ній вилучити довільне обмеження з обмежень (35), (36). Для цього замість (35) достатньо розглянути обмеження

$$\sum_{k=1}^l x_{ik} = a_i, \quad i = \overline{1, m-1}, \quad (42)$$

а замість (36) розглянути обмеження

$$\sum_{k=1}^l y_{kj} = b_j, \quad j = \overline{1, n-1}. \quad (43)$$

Для обмеження (42) вилучають обмеження для останнього постачальника, а для обмеження (43) вилучають обмеження для останнього споживача.

4. ЗАСТОСУВАННЯ ЗАДАЧ (1)–(5), (15)–(20) ТА ОПТИМАЛЬНЕ РОЗБИТТЯ МНОЖИН

Розглянемо застосування задач (1)–(5) та (15)–(20) до модельної задачі оптимального розбиття множини, описаної в роботі [1]. Нехай деякий постачальник однорідного ресурсу (сировини), неперервно розподілений зі щільністю $\rho(x) = 1$ в області $\Omega = \{x = (x^{(1)}, x^{(2)}): 0 \leq x^{(1)} \leq 1, 0 \leq x^{(2)} \leq 1\}$, постачає його в п'ять пунктів (першого етапу) для первинного перероблення чи зберігання. Задано координати $\tau_i^I = (\tau_i^{I(1)}, \tau_i^{I(2)})$, $i = \overline{1, 5}$, розташування цих пунктів: $\tau_1^I = (0.2; 0.2)$, $\tau_2^I = (0.3; 0.5)$, $\tau_3^I = (0.8; 0.3)$, $\tau_4^I = (0.6; 0.8)$, $\tau_5^I = (0.6; 0.1)$. Задано також координати $\tau_j^{II} = (\tau_j^{II(1)}, \tau_j^{II(2)})$, $i = \overline{1, 3}$, пунктів (другого етапу) споживання ресурсу, що був перероблений (зберігався) в пунктах першого етапу: $\tau_1^{II} = (0.2; 0.8)$, $\tau_2^{II} = (0.6; 0.4)$, $\tau_3^{II} = (0.8; 0.7)$.

Вартість транспортування одиниці ресурсу від постачальника з координатами $x = (x^{(1)}, x^{(2)})$ до пункту першого етапу з координатами $\tau_i^I = (\tau_i^{I(1)}, \tau_i^{I(2)})$ задана у вигляді $c_i^I(x^{(1)}, x^{(2)}, \tau_i^I) = \sqrt{(x^{(1)} - \tau_i^{I(1)})^2 + (x^{(2)} - \tau_i^{I(2)})^2}$, $i = \overline{1, 5}$.

Затрати на транспортування одиниці продукції з i -го пункту першого етапу $\tau_i^I = (\tau_i^{I(1)}, \tau_i^{I(2)})$ до пункту другого етапу $\tau_j^{II} = (\tau_j^{II(1)}, \tau_j^{II(2)})$ задано у вигляді $c_{ij}^{II}(\tau_i^I, \tau_j^{II}) = \sqrt{(\tau_i^{I(1)} - \tau_j^{II(1)})^2 + (\tau_i^{I(2)} - \tau_j^{II(2)})^2}$, $i = \overline{1, 5}$, $j = \overline{1, 3}$.

Потрібно розбити множину Ω постачальників ресурсу на сфери їхнього обслуговування в п'яти пунктах першого етапу, тобто на підмножини $\Omega_1, \dots, \Omega_5$,

$\bigcup_{i=1}^5 \Omega_i = \Omega$, $i = \overline{1, 5}$, так, щоб мінімізувати сумарну вартість транспортування ре-

сурсу від постачальників до пунктів первинного перероблення (першого етапу) і сумарну вартість доставки переробленого ресурсу до пунктів кінцевого споживання (другого етапу) з планом перевезень $\nu = \{\nu_{ij}\}$ та потребами $b^{II} = \{b_j^{II}\}$:

$$F(\Omega_1, \dots, \Omega_5) = \sum_{i=1}^5 \int_{\Omega_i} c_i^I(x, \tau_i^I) \rho(x) dx + \sum_{i=1}^5 \sum_{j=1}^3 c_{ij}^{II}(\tau_i^I, \tau_j^{II}) \nu_{ij} \rightarrow \min,$$

і до того ж весь перероблений продукт зі всіх пунктів першого етапу необхідно транспортувати до пунктів другого етапу $\sum_{i=1}^5 \int_{\Omega_i} \rho(x) dx = \sum_{j=1}^3 b_j^{II}$.

Дискретний аналог цієї задачі за додаткової умови, що обсяги однорідних ресурсів у пунктах кінцевого споживання вважаються заданими, легко сформулювати за допомогою задач (1)–(5), де постачальниками є m ділянок з дискретного розбиття області Ω , споживачами є три пункти другого етапу, а проміжними пунктами є п'ять пунктів першого етапу. Вартість транспортування одиниці продукції від постачальників до проміжних пунктів та від проміжних пунктів до

споживачів визначається як Евклідова відстань між точками, що їм відповідають. Якщо обсяги ресурсу постачальників вибрати пропорційними площам ділянок, а обсяги ресурсів у пунктах кінцевого споживання вважати заданими, то розв'язок задачі (1)–(5) відповідатиме одному з варіантів оптимального розбиття множини Ω постачальників ресурсу з точністю до вибраної дискретизації ділянок.

Нами було розглянуто рівномірне розбиття області Ω на квадратні ділянки, кількість яких визначає розмірність задачі (1)–(5) та час, витрачений на її виконання. У разі розбиття області 100×100 маємо 10000 постачальників, 50015 змінних та 10008 лінійних обмежень-рівностей у відповідній задачі (1)–(5), на розв'язання задачі витрачається близько 30 с. У разі розбиття області 500×500 маємо 250000 постачальників, 1250015 змінних та 250008 обмежень, на розв'язання задачі витрачається близько 80 с. У разі розбиття 1000×1000 маємо 1000000 постачальників, задача (1)–(5) містить 5000015 змінних та 1000008 лінійних обмежень-рівностей, на розв'язання задачі витрачається близько 5 хв.

На рис. 2 наведено оптимальне розбиття множини Ω для двох варіантів задачі (1)–(5), які відрізняються обсягами ресурсів в пунктах другого етапу: у першій задачі споживачам потрібно отримати однакові обсяги ресурсу (рис. 2, а), для другої задачі розподіли обсягів ресурсу для кожного із споживачів складають 23 %, 43 %, 34 % (рис. 2, б). Середній час, витрачений солвером CPLEX 20.1.0.0 з NEOS-сервера на розв'язання першої задачі у разі розбиття 1000×1000 , де кожній ділянці відповідає одна одиниця ресурсу, становить 207 с. Для цього п'ять разів було запущено солвер та обчислено середнє арифметичне часу його виконання, максимальний час — 213 с, мінімальний час — 203 с. За однакового розподілу ресурсу оптимальне значення цільової функції дорівнює 485323 умовних одиниць (у.о.), що на 1.6 % більше за оптимальне значення цільової функції для заданих розподілів обсягів ресурсу споживачів, що дорівнює 477738 у.о. Середній час, витрачений солвером CPLEX на розв'язання другої задачі, становить 296 с (максимальний час — 357 с, мінімальний час — 225 с).

Застосування задачі (15)–(20) до розглядуваного модельного прикладу дає змогу визначати верхні межі обсягів ресурсів для кожного пункту першого етапу (проміжні пункти). До відповідної задачі лінійного програмування (1)–(5) додаються п'ять обмежень у вигляді лінійних нерівностей. Так, для розбиття множини 1000×1000 маємо 1000000 постачальників, задача (15)–(20) містить

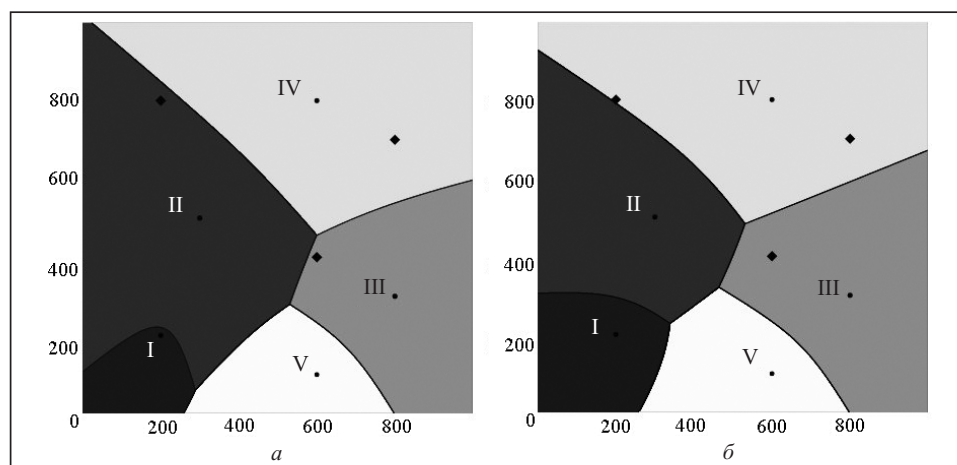


Рис. 2. Оптимальне розбиття множини Ω (перший етап) для двох варіантів задачі (1)–(5): частки ресурсу споживачів 1/3, 1/3, 1/3 (а); ресурс споживачів 23 %, 43 %, 34 % (б)

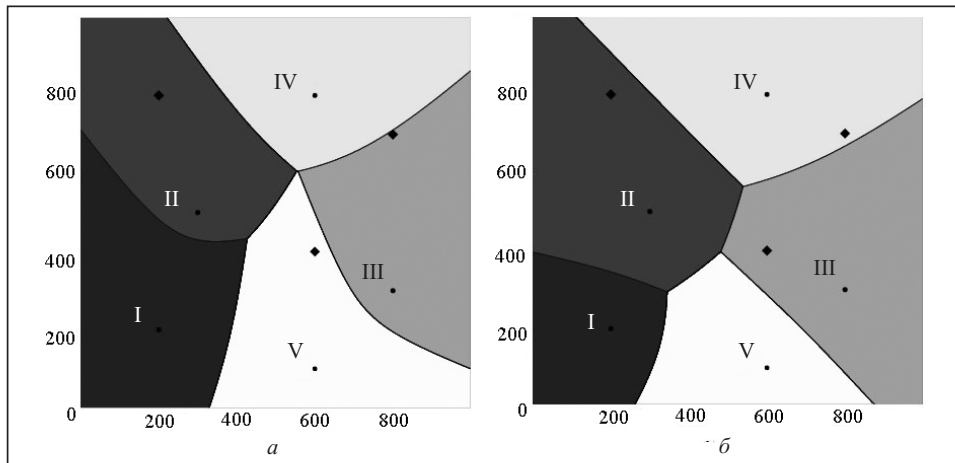


Рис. 3. Оптимальні розв'язки для першого етапу в двох варіантах задачі (15)–(20): d_1^{up} не перевищують $1/5$ всього ресурсу (а) та $1/4$ всього ресурсу (б)

5000015 змінних та 1000008 лінійних обмежень-рівностей та п'ять обмежень-нерівностей. На рис. 3 наведено два варіанти оптимального розбиття множини постачальників з урахуванням верхніх меж на пропускні спроможності проміжних пунктів із заданими розподілами обсягів ресурсу споживачів (23 %, 43 %, 34 %). Рис. 3, а відповідає варіанту, коли проміжні спроможності не перевищують $1/5$ всього обсягу ресурсу, а рис. 3, б — варіанту, коли не перевищують $1/4$ обсягу ресурсу. До того ж оптимальне значення F_{xy}^* для першої задачі є більшим, ніж F_{xy}^* для другої задачі на 47848 умовних одиниць, що складає 8,7%. З рис. 3, б випливає, що другий варіант є ближчим до оптимального розбиття на рис. 2, б, ніж перший варіант на рис. 3, а. Вартості перевезення від постачальників до проміжних пунктів та від проміжних пунктів до споживачів порівняно з розбиттям на рис. 2, б збільшились на 5 % для другої задачі (рис. 3, б) та на 13 % — для першої задачі (рис. 3, а). Середній час, витрачений солвером CPLEX на розв'язання цих задач, становить 300 с.

5. ЗАДАЧА (34)–(40) ТА ЗАДАЧА ПРО p -МЕДІАНУ ГРАФУ

Наведемо результати, що ілюструють подібність задачі (34)–(40) з задачею про p -медіану графу [9, 10] та з методом k -середніх для задачі кластеризації [11, 12]. Нехай для задачі (34)–(40) потрібно вибрати вершини проміжних пунктів так, щоб суми відстаней від них до вершин постачальників та від вершин проміжних пунктів до вершин споживачів були мінімальними. Постачальників, проміжні пункти та споживачів у задачі (34)–(40) задано координатами в \mathbb{R}^2 і їх отожднюють з вершинами деякого графу (рис. 4 та рис. 5), де обсяги ресурсу постачальників дорівнюють одиниці. Якщо змінні x_{ik} зробити Булевими, то кожен постачальник однозначно буде зв'язаний тільки з одним проміжним пунктом. Якщо крім цього покласти $c_{kj} = 0$, то задача (34)–(40) стає еквівалентною задачі про p -медіану графу.

На рис. 4 та рис. 5 наведено графічні результати розв'язання задачі (34)–(40) з використанням набору даних *kroA100.tsp* з відомої бібліотеки TSPLIB [17]. Відповідні задачі розв'язувались за допомогою солверів Gurobi та CPLEX.

Рис. 4, а та рис. 4, б ілюструють розв'язок задачі (34)–(40), коли як відстань між точками використовується Евклідова відстань, а рис. 4, в та рис. 4, г —

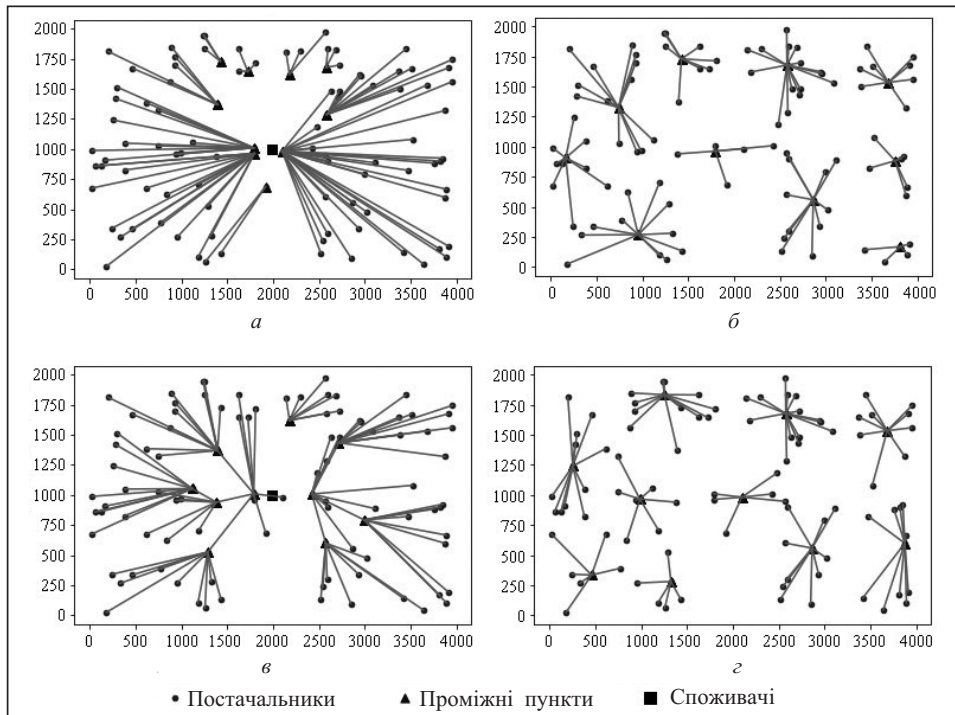


Рис. 4. Розв'язок модельного прикладу задачі (34)–(40) для одного споживача

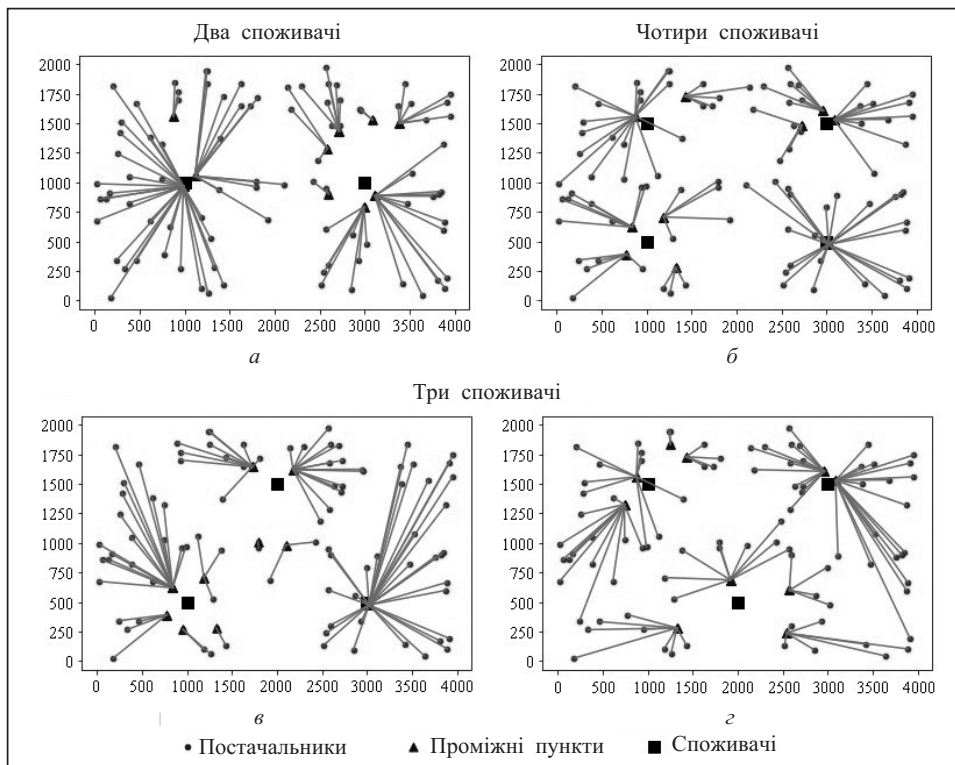


Рис. 5. Розв'язок модельного прикладу задачі (34)–(40) для двох (а), трьох (в, г) та чотирьох (б) споживачів

квадрат Евклідової відстані. Для рис. 4, б та рис. 4, г величини c_{kj} покладають рівними нулю, унаслідок цього задача (34)–(40) стає еквівалентною задачі про

p -медіану, для якої Булеві змінні замінюють на неперервні, а також еквівалентною методу k -середніх за умови, що як середні вибирається одна з вершин відповідно. Якщо величини c_{kj} є ненульовими (рис. 4, а та рис. 4, в), то задача має два етапи обчислення: вибір медіанних вершин та прив'язка цих вершин до споживачів. На рис. 4 наведено результати розв'язання задачі для одного споживача. На рис. 5 наведено результати для двох, трьох та чотирьох споживачів відповідно; як відстань між точками використано Евклідову відстань.

Зазначимо, що задаючи верхні межі на пропускні спроможності проміжних пунктів, ми, фактично, задаємо обмеження на кількість вершин, які відповідають вершинам постачальників.

6. ВИКОРИСТАННЯ ЗАДАЧІ (34)–(40) ДЛЯ РАЦІОНАЛЬНОГО РОЗМІЩЕННЯ ЗАДАНОЇ КІЛЬКОСТІ СКЛАДІВ

Задачу (34)–(40) можна використовувати для раціонального розміщення заданої кількості складів з урахуванням визначеного положення постачальників та отримувачів матеріально-технічних засобів на території, де вони виконують свої завдання [6, 18]. Розглянемо модельний приклад такої задачі, де постачальники, проміжні пункти та споживачі відповідатимуть координатам в \mathbb{R}^2 . Нехай є 100 постачальників (пункти A з координатами (x_a, y_a)), 100 проміжних пунктів (пункти D з координатами (x_d, y_d)) та 25 споживачів (пункти B з координатами (x_b, y_b)). Координати постачальників, проміжних пунктів та споживачів згенеровано випадково за допомогою стандартного рівномірного розподілу $U(0, 1)$: $x_a \sim 50 * U(0, 1)$, $y_a \sim 50 * U(0, 1)$; $x_d \sim 100 + 50 * U(0, 1)$, $y_d \sim 50 * U(0, 1)$; $x_b \sim 200 + 50 * U(0, 1)$, $y_b \sim 50 * U(0, 1)$. Обсяги ресурсу в пунктах A та в пунктах B обчислено з використанням матриці $m \times n$, де цілі коефіцієнти генерувались випадковим чином за допомогою стандартного рівномірного розподілу $\text{round}(U(0.1, 2))$. Ресурс постачальників дорівнював сумі елементів у рядку матриці, а ресурс споживачів — сумі елементів у стовпчику матриці. Для $d = 10$ (вибрано 10 складів із 100 можливих) результат розв'язання задачі (34)–(40) наведено на рис. 6.

Другу частину обчислювальних експериментів складають дослідження затрат солверів CPLEX та Gurobi для тестових задач з такими параметрами: 80 постачальників, 80 споживачів та 80 проміжних пунктів. Ці параметри задачі вибрано так, щоб гарантувати отримання результату упродовж восьми годин — максимальний ліміт часу, встановлений на NEOS-сервері.

Розглядалися три еквівалентні задачі. Перша з них — це задача (34)–(40)

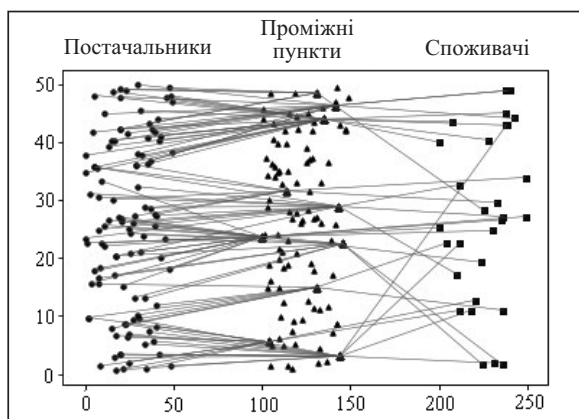


Рис. 6. Розв'язок задачі (34)–(40) для $m = 100$, $n = 100$, $l = 100$, $d = 10$

з лінійно залежними обмеженнями, що впливає з твердження 4. Із другої та третьої задач вилучено одне лінійно залежне обмеження (для другої задачі вилучено обмеження для останнього постачальника — обмеження (35) замінено на обмеження (42), а для третьої задачі не враховували обмеження для останнього споживача — обмеження (36) замінено на обмеження (43)). AMPL-код для цих трьох задач наведено на рис. 7.

```

param m = 80; #Кількість постачальників (пункти А)
param n = 80; #Кількість споживачів (пункти В)
param l = 80; #Всього проміжних пунктів (пункти D) #10000;
param d = 0; #Кількість пунктів D, які вибираються (l=1);

param Table1{i in 1..10, j in 1..6}; # інформація про розрахунки

# генеруємо координати постачальників, споживачів, проміжних пунктів
param xa{i in 1..m} = 50*Uniform(0,1);
param ya{i in 1..m} = 50*Uniform(0,1);
param xb{j in 1..n} = 200 + 50*Uniform(0,1);
param yb{j in 1..n} = 50*Uniform(0,1);
param xd{k in 1..l} = 100 + 50*Uniform(0,1);
param yd{k in 1..l} = 50*Uniform(0,1);

#Вартості перевезення одиниці продукції від А до В
param cik{i in 1..m, k in 1..l} = sqrt((xa[i]-xd[k])^2+(ya[i]-yd[k])^2)/100.;
param ckj{k in 1..l, j in 1..n} = sqrt((xd[k]-xb[j])^2+(yd[k]-yb[j])^2)/100.;

# Продукція в А та потреби в В
param x0{i in 1..m, j in 1..n} = round(Uniform(0,2));
param a{i in 1..m} = sum{j in 1..n} x0[i,j]; #Продукція в А
param b{j in 1..n} = sum{i in 1..m} x0[i,j]; #Потреби в В
check: sum{i in 1..m} a[i] = sum{j in 1..n} b[j]; # Твердження 4

#Невідомі (продукція, яку потрібно перевезти)
var x{i in 1..m, k in 1..l} = 0; #від А до D
var y{k in 1..l, j in 1..n} = 0; #від D до В
var z{k in 1..l} binary; # 1-задіяно D, 0-не задіяно D

minimize f_xyz: #Мінімізувати витрати на перевезення продукції:
sum{i in 1..m, k in 1..l} cik[i,k]*x[i,k] + #від А до D
sum{k in 1..l, j in 1..n} ckj[k,j]*y[k,j]; #від D до В
subject to #за обмежень на перевезення продукції з А до D
con35 {i in 1..m}: sum{k in 1..l} x[i,k] = a[i];
con42 {i in 1..(m-1)}: sum{k in 1..l} x[i,k] = a[i];
# на перевезення продукції з D до В
con36 {j in 1..n}: sum{k in 1..l} y[k,j] = b[j];
con43 {j in 1..(n-1)}: sum{k in 1..l} y[k,j] = b[j];
# не залишати продукцію в проміжних пунктах D
con37 {k in 1..l}: sum{i in 1..m} x[i,k] - sum{j in 1..n} y[k,j] = 0;
#вкл./викл. проміжний пункт
con38 {k in 1..l}: sum{i in 1..m} x[i,k][k]*sum{i in 1..m} a[i];
#вибрати рівно d проміжних пунктів
con39: sum{k in 1..l} z[k] = d;

problem dka: f_xyz, x, y, z, con35, con36, con37, con38, con39;
problem dkb: f_xyz, x, y, z, con42, con36, con37, con38, con39;
problem dkc: f_xyz, x, y, z, con35, con43, con37, con38, con39;

# перевіряємо мінімальні відстані між пунктами А, В, D
display min{i in 1..m, j in (i+1)..m} sqrt((xa[i]-xa[j])^2+(ya[i]-ya[j])^2);
display min{i in 1..n, j in (i+1)..n} sqrt((xb[i]-xb[j])^2+(yb[i]-yb[j])^2);
display min{i in 1..l, j in (i+1)..l} sqrt((xd[i]-xd[j])^2+(yd[i]-yd[j])^2);

# розв'язуємо три еквівалентні задачі для d=2,3,...,10
for{irun in 2..10} {
  let d := irun; display d; # вибираємо та друкуємо d
  solve dka; display f_xyz, _solve_time; # задача (34)-(40)
  let Table1[irun,1] := f_xyz; let Table1[irun,2] := _solve_time;
  solve dkb; display f_xyz, _solve_time;
  let Table1[irun,3] := f_xyz; let Table1[irun,4] := _solve_time;
  solve dkc; display f_xyz, _solve_time;
  let Table1[irun,5] := f_xyz; let Table1[irun,6] := _solve_time;
}
display Table1;

```

Рис. 7. AMPL-код для трьох задач вигляду (34)–(40) для $m = 80$, $n = 80$, $l = 80$

В табл. 1 та 2 наведено порівняння часу (в секундах), затраченого солверами CPLEX 20.1.0.0 та Gurobi 9.1.2 для розв'язання трьох розглянутих задач для значень $d \in \{2, \dots, 10\}$. З табл. 1 випливає, що CPLEX успішно розв'язує усі тестові приклади. Однак для $d = 3$ час розв'язання задач суттєво збільшується порівняно з розв'язанням задач для інших значень d .

Таблиця 1. Показники солвера CPLEX для задач, якщо $m = 80$, $n = 80$, $l = 80$

| d | Задача 1 | | Задача 2 | | Задача 3 | |
|-----|-------------|---------|-------------|---------|-------------|--------|
| | f_{xyz}^* | Час, с | f_{xyz}^* | Час, с | f_{xyz}^* | Час, с |
| 2 | 12898.2 | 113.94 | 12898.2 | 105.79 | 12898.2 | 108.65 |
| 3 | 12876.5 | 1340.94 | 12876.5 | 1219.05 | 12876.5 | 984.82 |
| 4 | 12869.3 | 164.96 | 12869.3 | 250.28 | 12869.3 | 663.05 |
| 5 | 12866.7 | 78.08 | 12866.7 | 80.36 | 12866.7 | 73.77 |
| 6 | 12864.8 | 58.90 | 12864.8 | 74.93 | 12864.8 | 74.54 |
| 7 | 12863.6 | 39.75 | 12863.6 | 33.54 | 12863.6 | 52.64 |
| 8 | 12863.0 | 32.68 | 12863.0 | 21.16 | 12863.0 | 16.65 |
| 9 | 12862.7 | 7.53 | 12862.7 | 2.46 | 12862.7 | 12.86 |
| 10 | 12862.5 | 1.13 | 12862.5 | 0.73 | 12862.5 | 0.72 |

Таблиця 2. Показники солвера Gurobi для трьох задач, якщо $m = 80$, $n = 80$, $l = 80$

| d | Задача 1 | | Задача 2 | | Задача 3 | |
|-----|-------------|--------|-------------|--------|-------------|--------|
| | f_{xyz}^* | Час, с | f_{xyz}^* | Час, с | f_{xyz}^* | Час, с |
| 2 | 12898.4 | 116.02 | 12898.2 | 131.19 | 12898.2 | 65.82 |
| 3 | 12877.1 | 74.80 | 12876.5 | 739.15 | 12876.5 | 92.76 |
| 4 | 12869.5 | 60.75 | 12869.5 | 86.69 | 12869.5 | 55.29 |
| 5 | 12866.6 | 36.22 | 12866.6 | 49.48 | 12866.6 | 25.95 |
| 6 | 12865.0 | 23.14 | 12864.7 | 25.43 | 12864.7 | 21.69 |
| 7 | 12864.1 | 22.77 | 12864.1 | 24.18 | 12864.1 | 13.60 |
| 8 | 12863.5 | 14.49 | 12863.3 | 16.39 | 12863.1 | 13.61 |
| 9 | 12863.0 | 11.70 | 12862.8 | 12.77 | 12862.8 | 15.02 |
| 10 | 12862.7 | 2.61 | 12862.7 | 2.35 | 12862.7 | 2.61 |

Таблиця 3. Показники солверів CPLEX та Gurobi для задач, якщо $m = 80$, $n = 80$, $l = 80$

| d | CPLEX 20.1.0.0 | | | | Gurobi 9.1.2 | | | |
|-----|----------------|--------|------------|----------|--------------|--------|------------|----------|
| | f_{xyz}^* | Час, с | N_{iter} | N_{br} | f_{xyz}^* | Час, с | N_{iter} | N_{br} |
| 2 | 12898.2 | 105.79 | 832860 | 4234 | 12898.2 | 65.82 | 119070 | 1079 |
| 3 | 12876.5 | 984.82 | 4715775 | 40967 | 12876.5 | 92.76 | 194169 | 1417 |
| 4 | 12869.3 | 164.96 | 556308 | 5345 | 12869.5 | 55.29 | 135660 | 1220 |
| 5 | 12866.7 | 73.77 | 289190 | 2300 | 12866.6 | 25.95 | 87503 | 1293 |
| 6 | 12864.8 | 58.90 | 277605 | 2352 | 12864.7 | 21.69 | 92711 | 1762 |
| 7 | 12863.6 | 33.54 | 158922 | 1329 | 12864.1 | 13.60 | 47348 | 1030 |
| 8 | 12863.0 | 16.65 | 60699 | 661 | 12863.1 | 13.61 | 60029 | 1106 |
| 9 | 12862.7 | 2.46 | 3189 | 0 | 12862.8 | 12.77 | 56007 | 1094 |
| 10 | 12862.5 | 0.72 | 1953 | 0 | 12862.7 | 2.35 | 4410 | 15 |

З табл. 2 бачимо, що солвер Gurobi не знайшов оптимального значення цільової функції для жодного тестового прикладу, крім прикладу для $d = 5$: для нього значення цільової функції знайдено кращим на 0.1, ніж за допомогою програми CPLEX. Скажімо, для $d = 2$ для задачі 1 мінімальне значення цільової функції, отримане за допомогою солвера CPLEX, є на 0.2 меншим, ніж мінімальне значення цільової функції, отримане за допомогою солвера Gurobi.

У табл. 3 наведено основні характеристики солверів CPLEX та Gurobi для задач 1, 2, 3: N_{iter} — кількість симплексних ітерацій (для CPLEX — MPI simplex iterations, для Gurobi — simplex iterations), N_{br} — кількість вершин розгалуження (для CPLEX — branch-and-bound nodes, для Gurobi — branch-and-cut nodes). У табл. 3 фігурують дані про ті задачі, для яких отримано найкраще мінімальне значення цільової функції. Якщо для двох задач мінімальні значення цільової функції були ті самі, то до табл. 3 записували дані про ту задачу, на розв'язання якої витрачалось менше часу відповідним солвером.

ВИСНОВКИ

У статті досліджено математичні моделі лінійного та Булевого лінійного програмування для збалансованої двоетапної транспортної задачі та двох її модифікацій. Перша модифікація враховує верхні межі на пропускні спроможності проміжних пунктів, друга враховує можливість вибору фіксованої кількості проміжних пунктів, меншої за їхню загальну кількість. Для всіх трьох випадків обґрунтовано необхідні та достатні умови існування розв'язку відповідних задач математичного програмування. Розглянуто три приклади застосування модифікацій збалансованої двоетапної транспортної задачі: задача оптимального розбиття множини, задача про p -медіану графу та задача оптимального розміщення заданої кількості складів. Наведено результати обчислювальних експериментів з використанням солверів Gurobi та CPLEX і мови моделювання AMPL.

Другу модифікацію двоетапної транспортної задачі можна використовувати для визначення відповідних місць розташування в Об'єднаній енергетичній системі (ОЕС) України накопичувачів електричної енергії та їхніх енергоємностей. Розроблений AMPL-код можна легко модифікувати та налаштувати на підготовку відповідних витрат на транспортування електричної енергії, враховуючи їхній вплив на потоки потужності контрольованими перетинами ОЕС України [19, 20]. Задачі (1)–(5), (15)–(20), (25)–(31) та (34)–(40) заплановано використовувати для узагальнення двоетапної транспортної моделі на два види диференційованої продукції, де кожному виду продукції відповідає власний тип енергогенерації [21, 22].

Розглянуті модифікації двоетапної транспортної задачі належать класу відомих виробничо-транспортних задач, блокова структура яких дає змогу будувати ефективні методи їхнього розв'язання на основі алгоритмів негладкої оптимізації у поєднанні з відомими схемами декомпозиції. Численні застосування субградієнтних алгоритмів негладкої оптимізації для таких задач описано в [23, 24]. На їхній основі розроблено програмне забезпечення для низки моделей оптимального планування, проєктування і керування.

СПИСОК ЛІТЕРАТУРИ

1. Киселева Е.М., Притоманова О.М., Ус С.А. Решение двухэтапной непрерывно-дискретной задачи оптимального разбиения-распределения с заданным положением центров подмножеств. *Кибернетика и системный анализ*. 2020. Т. 56, № 1. С. 3–15.
2. Kiseleva E., Prytomanova O., Hart L. Solving a two-stage continuous-discrete problem of optimal partitioning-allocation with the subsets centers placement. *Open Computer Science*. 2020. Vol. 10. P. 124–136.
3. Карагодова О.О., Кігель В.Р., Рожок В.Д. Дослідження операцій. Київ: Центр учбової літератури, 2007. 256 с.
4. Наконечний С.І., Савіна С.С. Математичне програмування. Київ: КНЕУ, 2003. 452 с.
5. Стецюк П.І., Ляшко В.І., Мазютинець Г.В. Двоетапна транспортна задача та її AMPL-реалізація. *Наукові записки НАУКМА. Комп'ютерні науки*. 2018. Т. 1. С. 14–20.

6. Стецюк П.І., Бисага О.П., Трегубенко С.С. Двоетапна транспортна задача з обмеженням на кількість проміжних пунктів. *Компьютерная математика*. 2018. № 2. С. 119–128.
7. Стецюк П., Стівба В., Хом'як О. Двоетапна транспортна задача та дві її модифікації. *Журнал обчислювальної та прикладної математики*. 2022. № 3 (136). С. 97–107.
8. Стецюк П.І., Трегубенко С.С. Про двоетапну транспортну задачу з заданою кількістю проміжних пунктів. *Матеріали Двадцятого Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування»* (Кропивницький, 13–14 квітня 2018 р.). С. 128–132.
9. Christofides N. *Graph theory: An algorithmic approach*. Cambridge: Academic Press, 1975. 400 p.
10. Jain A.K., Dubes R.C. *Algorithms for clustering data*. Prentice-Hall, 1988. 320 p.
11. Lloyd S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*. 1982. 28(2). P. 129–137.
12. Arthur D., Vassilvitskii S. Kmeans++: the advantage of careful seeding. *Proc. of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 07)*, SIAM, Philadelphia. 2007. P. 1027–1035.
13. Gurobi Optimization Inc.: Gurobi Optimizer Reference Manual. URL: <http://www.gurobi.com/documentation/>.
14. IBM: CPLEX User's Manual, Version 12 Release 8. URL: <https://www.ibm.com/support/knowledgecenter/SSSA5P12.8.0/ilog.odms.studio.help/pdf/usrcplex.pdf>.
15. NEOS Server: State-of-the-Art Solvers for Numerical Optimization. URL: <https://neos-server.org/neos/>.
16. Fourer R., Gay D.M., Kernighan B.W. *AMPL: A modeling language for mathematical programming*. Pacific Grove: Duxbury Press, 2002.
17. Бібліотека TSPLIB — A Traveling Salesman Problem Library. URL: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/>.
18. Романченко І.С., Хазанович О.І., Трегубенко С.С. Моделювання системи матеріально-технічного забезпечення. Львів: НАСВ ЗС України, 2015. 156 с.
19. Буткевич О.Ф., Юнєєва Н.Т., Гурєєва Т.М. До питання про розміщення накопичувачів енергії в ОЕС України. *Технічна електродинаміка*. 2019. № 6. С. 59–64.
20. Буткевич О.Ф., Юнєєва Н.Т., Гурєєва Т.М., Стецюк П.І. Задача розташування накопичувачів електроенергії в ОЕС України з урахуванням його впливу на потоки потужності контрольованими перетинами. *Технічна електродинаміка*. 2020. № 4. С. 46–50.
21. Горбачук В.М., Дунаєвський М.С., Сулейманов С.-Б. Аналіз ланцюгів вартості на основі нових технологій енергонакопичення. Удосконалення енергоустановок методами математичного і фізичного моделювання. Секція 3 (8–10 жовтня 2019 р., Харків). Харків: Ін-т проблем машинобудування ім. А.М. Підгорного НАН України, 2019. С. 20–21.
22. Горбачук В.М., Шулінок Г.О., Сирку А.А. До вимірювання загальної енергоефективності держав. *Теорія оптимальних рішень*. 2019. С. 109–115.
23. Шор Н.З. *Методы минимизации недифференцируемых функций и их приложения*. Київ: Наук. думка, 1979. 199 с.
24. Михалевич В.С., Трубин В.А., Шор Н.З. *Оптимизационные задачи производственно-транспортного планирования: Модели, методы, алгоритмы*. Москва: Наука, 1986. 264 с.

P.I. Stetsyuk, V.O. Stovba, S.S. Tregubenko, O.M. Khomiak

TWO-STAGE TRANSPORTATION PROBLEM MODIFICATIONS AND THEIR APPLICATIONS

Abstract. Mathematical models of linear and Boolean linear programming for balanced two-stage transportation problem and their two modifications are analyzed. The first modification takes into account upper limits on the capacity of intermediate points, the second modification takes into account the possibility of choosing a fixed number of intermediate points, less than their total number. For all the three cases, the necessary and sufficient conditions for the consistency of the linear constraints systems of the corresponding mathematical programming problems are justified. Examples of the application of the modifications for the optimal set partitioning, the p -median graph problem, and the rational arrangement of the given number of storages are considered. The results of computational experiments using Gurobi and CPLEX solvers and the AMPL modeling language are presented.

Keywords: two-stage transportation problem, linear programming problem, Boolean linear programming problem, AMPL, Gurobi, CPLEX, optimal set partition, p -median, k -means, material and technical support.

Надійшла до редакції 08.07.2022