

AMPL-РЕАЛІЗАЦІЯ ДВОЕТАПНОЇ ТРАНСПОРТНОЇ ЗАДАЧІ

Стецюк^{1,2} П.І., Мазютинець² Г.В., Мілешовський² Б.І.
gabbimaz@gmail.com

¹Інститут кібернетики ім. В.М. Глушкова НАН України

²Ужгородський національний університет

Вступ. При формулюванні класичної двоетапної транспортної задачі [1] мається на увазі, що вантаж перевозиться від постачальників до споживачів тільки через проміжні пункти. Схема функціонування перевезень вантажу наведена на рис. 1. В якості проміжних пунктів можуть виступати посередницькі фірми та різного роду сховища (склади).

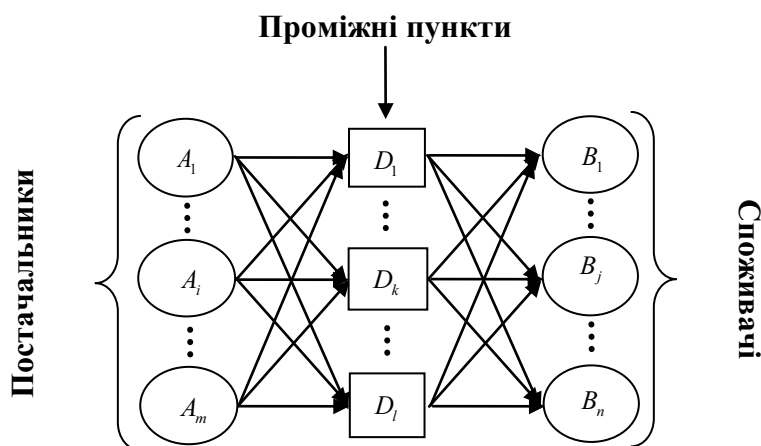


Рисунок 1. Система «постачальники – проміжні пункти – споживачі» у двоетапній транспортній задачі

Для розв'язання двоетапної транспортної задачі застосовують різного роду спеціалізовані алгоритми [1,2]. У статті розглянемо універсальний спосіб розв'язання двоетапної транспортної задачі з використанням сучасного програмного забезпечення. Цей спосіб використовує опис задачі на мові моделювання AMPL [3] та залежить тільки від того, яка із відомих програм буде використовуватися для розв'язання задачі лінійного програмування.

Матеріал роботи викладено в такому порядку. У розділі 1 описано формулювання двоетапної транспортної задачі та наведено її властивості. У розділі 2 наведено опис задачі на мові моделювання AMPL [3] та результати розрахунків для тестового прикладу.

1. Двоетапна транспортна задача та її властивості. Нехай в m пунктах постачання $A_1, \dots, A_m \in a_1, \dots, a_m$ одиниць продукції, яку потрібно перевезти до n споживачів B_1, \dots, B_n , задовольнивши їх потреби b_1, \dots, b_n . Для транспортування продукції від постачальників до споживачів можна задіяти l проміжних пунктів D_1, \dots, D_l . Витрати на перевезення одиниці продукції з пункту постачання A_i до проміжного пункту D_k позначимо c_{ik} , з проміжного пункту D_k до споживача B_j – c_{kj} . Кількість продукції, яка перевозиться від постачальника A_i до проміжного пункту D_k , позначимо x_{ik} , кількість продукції від проміжного пункту до споживача – y_{kj} .

Двоетапна транспортна задача має такий вигляд: знайти

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

за обмежень

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

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

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

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

Задача (1)–(5) є задачею лінійного програмування, яка містить $m \times l + l \times n$ змінних x_{ik} , y_{kj} та $m + n + l$ обмежень загального виду, не враховуючи обмежень (5) – умов на невід’ємність усіх змінних. Цільова функція (1) задає сумарні витрати на транспортування продукції від постачальників до споживачів через проміжні пункти. Обмеження (2) означають транспортування усієї продукції a_1, \dots, a_m із пунктів постачання до проміжних пунктів, а обмеження (3) – що споживачам потрібно доставити необхідну продукцію b_1, \dots, b_n з проміжних пунктів. Обмеження (4) задають умови на те, щоб вся продукція яка приходить від постачальників до кожного проміжного пункту, була обов’язково відправлена споживачам. Це визначає

умови на сумісність системи обмежень (2)–(5) – системи лінійних рівностей та лінійних нерівностей. Звідси випливає наступне твердження.

Лема 1. Система обмежень (2)–(5) є несумісною, якщо $\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$.

Доведення. Доведення проведемо методом від супротивного. Приймемо, що існують такі $\bar{x} = \{\bar{x}_{ik}, i=1, \dots, m, k=1, \dots, l\}$ та $\bar{y} = \{\bar{y}_{kj}, k=1, \dots, l, j=1, \dots, n\}$, які задовільняють систему обмежень (2)–(5).

Якщо всі рівності із (2) просумувати за індексом $i=1, \dots, m$, а всі рівності із (3) просумувати за індексом $j=1, \dots, n$, то отримаємо такі рівності

$$\sum_{i=1}^m \sum_{k=1}^l \bar{x}_{ik} = \sum_{i=1}^m a_i, \quad (6)$$

$$\sum_{j=1}^n \sum_{k=1}^l \bar{y}_{jk} = \sum_{j=1}^n b_j. \quad (7)$$

Віднявши від рівності (6) рівність (7) та змінивши порядок індексів в сумах, отримаємо наступну рівність

$$\sum_{k=1}^l \sum_{i=1}^m \bar{x}_{ik} - \sum_{k=1}^l \sum_{j=1}^n \bar{y}_{jk} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j. \quad (8)$$

Якщо всі рівності із (4) просумувати за індексом $k=1, \dots, l$, то отримаємо таку рівність

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

За умови $\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$ із рівності (8) випливає $\sum_{k=1}^l \sum_{i=1}^m \bar{x}_{ik} - \sum_{k=1}^l \sum_{j=1}^n \bar{y}_{jk} \neq 0$, що протирічить рівності (9). Це протиріччя доводить, що вектори \bar{x} та \bar{y} не можуть бути допустимими для обмежень (2)–(4), а значить при виконанні умови $\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$ система обмежень (2)–(5) є несумісною. **Лема доведена.**

2. AMPL-реалізація задачі (1)–(5). AMPL-код для опису двоетапної транспортної задачі (1)–(5) має наступний вигляд:

```

param m>=2; #Кількість постачальників
param l>=1; #Кількість проміжних пунктів
param n>=2; #Кількість споживачів
#Вартості перевезення одиниці продукції:
param cik{i in 1..m, k in 1..l} >= 0; #від A до D
param ckj{k in 1..l, j in 1..n} >= 0; #від D до B
#Інші дані
param a{i in 1..m} >= 0; #Продукція в A
param b{j in 1..n} >= 0; #Потреби B
check: sum{i in 1..m} a[i] = sum{j in 1..n} b[j]; # лема 1
#Невідомі (продукція, яку потрібно перевезти)
var x{i in 1..m, k in 1..l} >=0; #від A до D
var y{k in 1..l, j in 1..n} >=0; #від D до B
minimize f_opt: #Мінімізувати витрати на перевезення продукції:
sum{i in 1..m, k in 1..l} cik[i,k]*x[i,k]+ #від A до D
sum{k in 1..l, j in 1..n} ckj[k,j]*y[k,j]; #від D до B
subject to #за обмежень
con2 {i in 1..m}: #перевезення продукції з A до D
    sum{k in 1..l} x[i,k] = a[i];
con3 {j in 1..n}: #задоволення потреб B з D
    sum{k in 1..l} y[k,j] = b[j];
con4 {k in 1..l}: #всю продукцію потрібно доставити в B
    sum{i in 1..m} x[i,k] - sum{j in 1..n} y[k,j] = 0;

```

Тут оператор **param** використано для опису розмірів та даних задачі; оператор **var** – для опису змінних задачі, де враховано їх невід’ємність; оператор **check** – для перевірки сумісності обмежень (2)–(5), яка визначається лемою 1 та вимагає, щоб уся продукція постачальників була відправлена споживачам.

Якщо цей AMPL-код доповнити необхідними даними для конкретної задачі, то його можна використовувати для розв’язання двоетапних транспортних задач за допомогою стандартного програмного забезпечення для розв’язання задач лінійного програмування. Це можна зробити як за допомогою тих програм NEOS-сервера [4] із розділу «Linear Programming», для яких підтримуються вхідні формати даних на AMPL, так і за допомогою комерційних або з вільним доступом версій AMPL*.

* Безкоштовні версії програми знаходяться за посиланням: <http://ampl.com/try-ampl/download-a-free-demo/>

Тестовий приклад. Виробниче об'єднання складається з трьох філіалів: A_1, A_2, A_3 , які виготовляють однорідну продукцію в обсягах відповідно 1000, 1500 та 1200 одиниць на місяць. Ця продукція відправляється на два склади D_1 і D_2 , а потім – до п'яти споживачів B_1, B_2, \dots, B_5 , попит яких становить відповідно 900, 700, 1000, 500 і 600 одиниць. Вартості перевезень одиниці продукції (в умовних одиницях) від виробників на склади, а потім – зі складів до споживачів наведені в наступних таблицях:

$A \setminus D$	D_1	D_2
A_1	2	8
A_2	3	5
A_3	1	4

$D \setminus B$	B_1	B_2	B_3	B_4	B_5
D_1	1	3	8	5	4
D_2	2	4	5	3	1

Для цього прикладу опис вхідних даних задачі (1)–(5) реалізує AMPL-код:

```
data; #Блок вхідних даних, де задаємо:
param m := 3; #Кількість постачальників A
param l := 2; #Кількість проміжних пунктів D
param n := 5; #Кількість споживачів B
#Витрати на перевезення одиниці продукції:
param cik: 1 2 := #від A (↓) до D (→)
          1 2 8
          2 3 5
          3 1 4;
param ckj: 1 2 3 4 5 := #від D (↓) до B (→)
          1 1 3 8 5 4
          2 2 4 5 3 1;
param a:= #Продукція в A
1 1000 2 1500 3 1200; #A_1, A_2, A_3
param b:= #Потреби B
1 900 2 700 3 1000 #B_1, B_2, B_3
4 500 5 600; #B_4, B_5
solve; #Розв'язати задачу (1)–(5)
#Роздрукувати значення цільової функції та час розв'язання
display f_opt, _solve_time;
#Роздрукувати значення оптимальних змінних
display x; display y;
```

Розв'язок задачі (1) – (5) для тестового прикладу представлений на рис. 2. Для нього значення цільової функції дорівнює 22100 умовних одиниць, склад D_1 завантажений на 2200 одиниць, а склад D_2 – на 1500 одиниць.

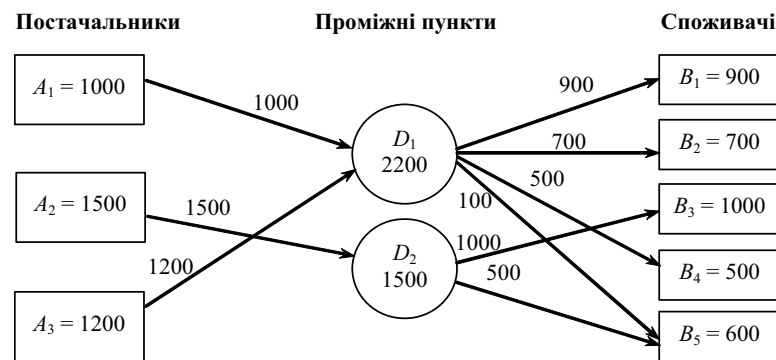


Рисунок 2. Оптимальний план перевезень продукції

Висновки. У даній роботі описано двоетапну транспортну задачу та наведено AMPL-код для її розв'язання за допомогою сучасного програмного забезпечення для задач лінійного програмування. Наведено демонстраційний приклад з результатами розрахунку. Розроблений AMPL-код планується використати при вивченні навчальної дисципліни «Мережеві задачі оптимізації» магістрантами спеціальностей 8.122 «Комп'ютерні науки та інформаційні технології» та 8.121 «Інженерія програмного забезпечення» ДВНЗ «Ужгородський національний університет» [5].

Модифікації двоетапної транспортної задачі (1)–(5) можуть бути використані студентами вищих навчальних закладів для розробки власних програмних проектів, пов'язаних з теорією та методами оптимізації. Так, наприклад, якщо обмеження (5) замінити на обмеження, які враховують нижні та верхні межі на кількість продукції, що перевозиться, то описаний метод не зміниться, а спеціалізовані алгоритми будуть вимагати значної корекції.

Бібліографічні посилання

1. Карагодова О.О. Дослідження операцій: Навч. посіб. / О.О. Карагодова, В.Р. Кігель, В.Д. Рожок – К.: Центр учбової літератури, 2007. – 256 с.
2. Наконечний С.І. Математичне програмування: Навч. посіб. / С.І. Наконечний, С.С. Савіна – К.: КНЕУ, 2003. – 452 с.
3. Fourer, R. AMPL, A Modeling Language for Mathematical Programming / R. Fourer, D. Gay, B. Kernighan. – Belmont: Duxbury Press, 2003. – 517 p.
4. NEOS Solver [Electronic resource]: <https://neos-server.org/neos/solvers/>.
5. Стецюк П.І. Мережні інформаційні технології: методичні рекомендації до вивчення курсу / П.І. Стецюк., О.В. Міца., В.І. Пецко. – Ужгород: Видавництво УжНУ «Говерла», 2014. – 65 с.