УДК 519.86
DOI https://doi.org/10.24144/2616-7700.2026.49(2).292-299

Г. Г. Цегелик[1], М. І. Глебена[2], М. Г. Цегелик[3]

[1] Львівський національний університет імені Івана Франка,
професор кафедри математичного моделювання соціально-економічних процесів,
доктор фізико-математичних наук
hryhoriy.tsehelyk@gmail.com
ORCID: https://orcid.org/0000-0002-4934-3181

[2] ДВНЗ «Ужгородський національний університет»,
завідувач кафедри системного аналізу та теорії оптимізації,
кандидат фізико-математичних наук, доцент
myroslava.hlebena@uzhnu.edu.ua
ORCID: https://orcid.org/0000-0003-1100-515X

[3] ПП «Бінар» м. Львів, Україна,
директор
M.Tsehelyk@gmail.com
ORCID: https://orcid.org/0009-0004-9413-0769

НАБЛИЖЕНИЙ МЕТОД РОЗВ'ЯЗУВАННЯ ЗАДАЧІ БУЛЕВОГО ПРОГРАМУВАННЯ. ОЗНАКА ОПТИМАЛЬНОСТІ НАБЛИЖЕНОГО РОЗВ'ЯЗКУ

У статті розглядається математична модель задачі булевого програмування для випадків, коли всі коефіцієнти цільової функції та обмежень є додатними величинами. Оскільки точні класичні алгоритми вимагають значного обсягу роботи, для розв'язання задачі пропонується ітеративний наближений метод. Алгоритм поетапно формує часткові рішення з одночасною оцінкою цільової функції та відсіканням безперспективних варіантів, що дає змогу зменшити кількість необхідних обчислень. Наближений розв'язок не завжди є оптимальним, тому в статті наводиться достатня умова, при виконанні якої його можна оптимізувати.

Ключові слова: булеве програмування, математична модель, алгоритм наближеного методу, достатня умова оптимальності.

1. Вступ. Булеве програмування є важливим класом задач з дискретного програмування [1; 2, с. 215; 3, с. 205], до якого належать численні задачі дослідження операцій, такі як: розміщення виробництва, про призначення, фінансування інвестиційних проектів, фінансування видів діяльності підприємства та інші. Для розв'язування можна використовувати як класичні методи, такі як метод гілок та меж, так і спеціалізовані, наприклад, адитивний алгоритм Балаша [4] та його модифікацію [5], які дають можливість будувати двійкове дерево рішень з ефективним відсіканням нерентабельних варіантів. Оскільки дані алгоритми вимагають великого обсягу роботи, то нами в роботі [6] запропоновано алгоритм відшукання наближеного розв'язку задачі. В статті пропонується достатня умова оптимальності наближеного розв'язку.

Розглянемо математичну **модель задачі булевого програмування**

$$L = \sum_{i=1}^{m} p_i x_i \to \max$$

за умов:

$$\sum_{i=1}^{m} a_{j,i} x_i \leq b_j, \quad j = 1, 2, \ldots, n;$$

$$x_i = \begin{cases} 0 \\ 1 \end{cases}, \quad i = 1, 2, \ldots, m,$$

де m — кількість операцій; n — кількість різних ресурсів, що використовуються для виконання операцій; $a_{j,i}$ — кількість одиниць j-го ресурсу, необхідних для виконання i-ої операції; p_i — прибуток від виконання i-тої операції; b_j — запас j-го ресурсу;

$$x_i = \begin{cases} 1, & \text{якщо виконується } i\text{-та операція;} \\ 0, & \text{в протилежному випадку.} \end{cases}$$

2. Алгоритм наближеного методу. Нехай $M = \varnothing$.

Перший крок:

Знаходимо

$$\max_{1 \leq i \leq m} p_i.$$

Нехай

$$\max_{1 \leq i \leq m} p_i = p_{i_1}.$$

Покладаємо

$x_{i_1} = 1$, $S_{j,i_1} = a_{j,i_1}$ для всіх $j = 1, 2, \ldots, n$;

i_1 відносимо до множини M.

Другий крок:

Знаходимо

$$\max_{i \notin M} p_i.$$

Нехай

$$\max_{i \notin M} p_i = p_{i_2}.$$

Визначаємо

$$x_{i_2} = \begin{cases} 1, \text{ якщо } \ S_{j,i_1} + a_{j,i_2} \leq b_j \ \text{ для всіх } \ j = 1, 2, \ldots, n; \\ 0, \text{ в протилежному випадку.} \end{cases}$$

Покладаємо

$$S_{j,i_2} = \begin{cases} S_{j,i_1} + a_{j,i_2}, & \text{якщо } x_{i_2} = 1; \\ S_{j,i_1}, & \text{якщо } x_{i_2} = 0. \end{cases}$$

i_2 відносимо до множини M.

Третій крок:

Знаходимо

$$\max_{i \notin M} p_i.$$

Нехай

$$\max_{i \notin M} p_i = p_{i_3}.$$

Визначаємо

$$x_{i_3} = \begin{cases} 1, \text{ якщо } \ S_{j,i_2} + a_{j,i_3} \le b_j \ \text{ для всіх } j = 1, 2, \dots, n; \\ 0, \text{ в протилежному випадку.} \end{cases}$$

Покладаємо

$$S_{j,i_3} = \begin{cases} S_{j,i_2} + a_{j,i_3}, \quad \text{якщо } x_{i_3} = 1; \\ S_{j,i_2}, \quad \text{якщо } x_{i_3} = 0. \end{cases}$$

i_3 відносимо до множини M.

І так далі …

Припустимо, що на k-му кроці $(k = 4, 5, \dots, m-1)$, $M = \{i_1, i_2, \dots, i_{k-1}\}$ та $\max\limits_{i \notin M} p_i = p_{i_k}$.

Визначаємо

$$x_{i_k} = \begin{cases} 1, \text{ якщо } \ S_{j,i_{k-1}} + a_{j,i_k} \le b_j \ \text{ для всіх } \ j = 1, 2, \dots, n; \\ 0, \text{ в протилежному випадку} \end{cases}$$

і покладаємо

$$S_{j,i_k} = \begin{cases} S_{j,i_{k-1}} + a_{j,i_k}, \text{ якщо } x_{i_k} = 1; \\ S_{j,i_{k-1}}, \text{ якщо } x_{i_k} = 0, \end{cases}$$

i_k відносимо до множини M.

Тоді на $(k+1)$-му кроці визначаємо

$$x_{i_{k+1}} = \begin{cases} 1, \text{ якщо } \ S_{j,i_k} + a_{j,i_{k+1}} \le b_j \text{ для всіх } \ j = 1, 2, \dots, n; \\ 0, \text{ в протилежному випадку} \end{cases}$$

і покладаємо

$$S_{j,i_{k+1}} = \begin{cases} S_{j,i_k} + a_{j,i_{k+1}}, \text{ якщо } x_{i_{k+1}} = 1; \\ S_{j,i_k}, \text{ якщо } x_{i_{k+1}} = 0. \end{cases}$$

i_{k+1} відносимо до множини M.

При $k+1 = m$ одержуємо

$$x_{i_m} = \begin{cases} 1, \text{ якщо } \ S_{j,i_{m-1}} + a_{j,i_m} \le b_j \ \text{ для всіх } \ j = 1, 2, \dots, n; \\ 0, \text{ в протилежному випадку.} \end{cases}$$

Через m кроків буде знайдено наближене значення розв'язку та значення цільової функції для нього.

Приклад 1. *Розглянемо задачу оптимального фінансування інвестиційних проектів, де цільова функція має вигляд:*

$$L = 14x_1 + 22x_2 + 13x_3 + 19x_4 + 10x_5 + 17x_6 + 25x_7 + 18x_8 \to \max$$

за умов:

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 + 4x_4 + 3x_5 + 6x_6 + 7x_7 + 2x_8 \le 25, \\ 2x_1 + 3x_2 + 4x_3 + 4x_4 + 2x_5 + 5x_6 + 6x_7 + 4x_8 \le 20; \end{cases}$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, 8.$$

Нехай $M = \varnothing$.

Перший крок:

Шукаємо $\max\limits_{1 \le i \le m} p_i$.

Нехай

$$\max_{1 \le i \le m} p_i = p_{i_1} = 25, \quad i_1 = 7.$$

Покладаємо

$$x_{i_1} = 1, \quad S_{i_1 j} = a_{i_1 j} = a_{7j} = \begin{cases} 7 \\ 6 \end{cases}, \quad i_1 = 7 \text{ відносимо до множини } M,$$

$$M = \{7\}.$$

Другий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_2} = 22, \quad i_2 = 2.$$

Оскільки $7 + 5 = 12 \le 25$ і $6 + 3 = 9 \le 20$, то $x_{i_2} = 1$.

Покладаємо

$$S_{i_2 j} = \begin{cases} 12, \\ 9. \end{cases}$$

$$i_2 \text{ відносимо до множини } M,$$

$$M = \{7, 2\}.$$

Третій крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_3} = 19, \quad i_3 = 4.$$

Оскільки $12 + 4 = 16 \le 25$ і $9 + 4 = 13 \le 20$, то $x_{i_3} = 1$.

Покладаємо

$$S_{i_3 j} = \begin{cases} 16, \\ 13. \end{cases}$$

$$i_3 \text{ відносимо до множини } M,$$

$$M = \{7, 2, 4\}.$$

Четвертий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_4} = 18, \quad i_4 = 8.$$

Оскільки $16 + 2 = 18 \le 25$ і $13 + 4 = 17 \le 20$, то $x_{i_4} = 1$.

Покладаємо

$$S_{i_4 j} = \begin{cases} 18, \\ 17. \end{cases}$$

$$i_4 \text{ відносимо до множини } M,$$

$$M = \{7, 2, 4, 8\}.$$

П’ятий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_5} = 17, \quad i_5 = 6.$$

Оскільки $18 + 6 = 24 \le 25$ і $17 + 5 = 22 > 20$, то $x_{i_5} = 0$.

Покладаємо

$$S_{i_5 j} = \begin{cases} 18, \\ 17. \end{cases}$$

$$i_5 \text{ відносимо до множини } M,$$

$$M = \{7, 2, 4, 8, 6\}.$$

Шостий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_6} = 14, \quad i_6 = 1.$$

Оскільки $18 + 3 = 21 \le 25$ і $17 + 2 = 19 \le 20$, то $x_{i_6} = 1$.

Покладаємо

$$S_{i_6 j} = \begin{cases} 21, \\ 19. \end{cases}$$

$$i_6 \text{ відносимо до множини } M,$$

$$M = \{7, 2, 4, 8, 6, 1\}.$$

Сьомий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_7} = 13, \quad i_7 = 3.$$

Оскільки $21 + 2 = 23 \le 25$ і $19 + 4 = 23 > 20$, то $x_{i_7} = 0$.

Покладаємо

$$S_{i_7 j} = \begin{cases} 21, \\ 19. \end{cases}$$

$$i_7 \text{ відносимо до множини } M,$$

$$M = \{7, 2, 4, 8, 6, 1, 3\}.$$

Восьмий крок:

Шукаємо $\max\limits_{i \notin M} p_i$.

Нехай

$$\max_{i \notin M} p_i = p_{i_8} = 10, \quad i_8 = 5.$$

Оскільки $21 + 3 = 24 \le 25$ і $19 + 2 = 21 > 20$, то $x_{i_8} = 0$.

Отже,

$$M = \{7, 2, 4, 8, 6, 1, 3, 5\},$$

$$X = (1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 1,\ 1),$$

$$L = 14 + 22 + 19 + 25 + 18 = 98.$$

Для оптимізації наближеного розв'язку можна використати наступну теорему.

Нехай $X = (x_1, x_2, \ldots, x_m)$ наближений розв'язок задачі булевого програмування і $M = (i_1, i_2, \ldots, i_k)$, де $i_s = 1$, $s = 1, 2, \ldots, k$. Покладемо $N = \{1, 2, \ldots, m\} \setminus M$.

Теорема 1. *Для того, щоб наближений розв'язок був оптимальним досить щоб не існувало таких двох індексів r і $l \in N$ та індексу $i_s \in M$ для яких виконуються умови*

$$a_{j,i_s} - 1 \le a_{j,r} + a_{j,l} \le a_{j,i_s}; \quad j = 1, 2, \ldots, \tag{1}$$

та

$$p_r + p_l > p_{i_s}. \tag{2}$$

Доведення. Припустимо, що існують такі два індекси r і $l \in N$ та індекс $i_s \in M$, і виконуються умови (1), (2). Оскільки виконується умова (1), то замість a_{j,i_s} можна використати $a_{j,r} + a_{j,l}$, при цьому умови задачі будуть виконуватися.

Згідно умови (2) замість p_{i_s} у вираз для значення цільової функції можна підставити $p_r + p_l$. В результаті значення цільової функції збільшиться. Це означає, що наближений розв'язок не є оптимальним.

Найбільш ефективним використанням теореми є у випадку коли кількість обмежень у булевій задачі є не більша за два. В даному випадку можна побудувати алгоритм оптимізації наближеного розв'язку. Наближений розв'язок в прикладі 1 виявився оптимальним.

Приклад 2. *Нехай задача булевого програмування має вигляд:*

$$L = 15x_1 + 22x_2 + 13x_3 + 19x_4 + 14x_5 + 17x_6 + 25x_7 + 18x_8 \to \max$$

за умов:

$$\begin{cases} 4x_1 + 4x_2 + 2x_3 + 4x_4 + 3x_5 + 6x_6 + 5x_7 + 2x_8 \le 25, \\ 3x_1 + 2x_2 + 3x_3 + 3x_4 + 3x_5 + 7x_6 + 2x_7 + 3x_8 \le 20; \end{cases}$$

$$x_i \in \{0, 1\}, \quad i = 1, 2, \ldots, 8.$$

Використовуючи алгоритм наближеного методу знайдено наближений розв'язок $X = (1,\ 1,\ 0,\ 1,\ 0,\ 1,\ 1,\ 1)$, для якого $L = 116$, він не є оптимальним, однак його можна оптимізувати.

Оскільки $a_{j,3} + a_{j,5} = \begin{cases} 5, \\ 6, \end{cases} < a_{j,6} = \begin{cases} 6, \\ 7, \end{cases}$ та $p_3 + p_5 > p_6$, то $x_3 = 1$, $x_5 = 1$, $x_6 = 0$. Тому $X_{opt} = (1,\ 1,\ 1,\ 1,\ 1, 0,\ 1,\ 1)$, для якого L_{opt}=126.

Приклад 3.

$$L = 20x_1 + 10x_2 + 40x_3 + 30x_4 + 35x_5 \to \max$$

за умов:

$$\begin{cases} 3x_1 + 2x_2 + 8x_3 + 3x_4 + 5x_5 \leq 14, \\ 2x_1 + x_2 + 9x_3 + 4x_4 + 6x_5 \leq 18, \end{cases}$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, 5.$$

Використовуючи алгоритм наближеного методу одержимо наближений розв'язок $X = (0,\ 0,\ 1,\ 0,\ 1)$, для якого $L = 75$, він не є оптимальним, будемо його оптимізувати.

Оскільки $a_{j,2} + a_{j,4} = \begin{cases} 5, \\ 5, \end{cases} < a_{j,5} = \begin{cases} 5, \\ 6, \end{cases}$ та $10 + 30 > 35$, то $x_2 = 1$, $x_4 = 1$, $x_5 = 0$. Тому покращений наближений розв'язок $X = (0,\ 1,\ 1,\ 1,\ 0)$, для якого $L = 80$.

Оскільки $a_{j,1} + a_{j,5} = \begin{cases} 8, \\ 8, \end{cases} \quad a_{j,3} = \begin{cases} 8, \\ 9, \end{cases}$ та $20+35 > 40$, то $x_1 = 1$, $x_5 = 1$, $x_3 = 0$. Отримали оптимальний розв'язок $X_{opt} = (1,\ 1,\ 0,\ 1,\ 1)$, для якого $L_{opt} = 95$.

3. Висновки. У статті розглянуто метод наближеного розв'язування задачі булевого програмування в випадку коли коефіцієнти цільової функції та обмежень є додатними. Приводиться достатня умова при виконанні якої наближений розв'язок буде оптимальним. Наведені приклади підтверджують ефективність теореми у випадку коли кількість обмежень не більше два.

Конфлікт інтересів

Глебена Мирослава Іванівна, членкиня редакційної колегії, є авторкою цієї статті та не брала участі в редакційному розгляді й ухваленні рішення щодо рукопису. Опрацювання рукопису здійснювалося незалежним редактором. Інші редактори заявляють про відсутність конфлікту інтересів.

Фінансування

Дослідження здійснено в рамках кафедральної науково-дослідної роботи «Моделі і методи системного аналізу в міждисциплінарних дослідженнях» (державний обліковий номер 0125U003246).

Доступність даних

Усі дані доступні в цифровій або графічній формі в основному тексті рукопису.

Використання штучного інтелекту

Автори підтверджують, що при створенні даної роботи вони не використовували технології штучного інтелекту.

Внесок авторів

Г. Г. Цегелик: концептуалізація, курація даних, формальний аналіз М. І. Глебена: курація даних, формальний аналіз, методологія, написання — рецензування та редагування. М. Г. Цегелик: формальний аналіз, написання — рецензування та редагування.

Список використаної літератури

1. Hnatenko, I. M., Liashenko, V. I., & Manhasarian, A. V. (2013). *Boolean programming: theory and practice.* KNEU [in Ukrainian].
2. Zaichenko, Yu. P. (2005). *Operations research.* Vydavnychyi tsentr "Akademiia" [in Ukrainian].
3. Katrenko, A. V. (2004). *Operations research.* Mahnoliia Plius [in Ukrainian].
4. Balas, E. (1965). An additive algorithm for solving linear programs with zero-one variables. *Operations Research,* 13(4), 517–546.
5. Tsehelyk, H. H., Seno, P. S., Hlebena, M. I., & Tsehelyk, M. H. (2026). An approach to solving boolean programming problems based on structural interpretation. *Scientific Bulletin of Uzhhorod University. Series of Mathematics and Informatics,* 48(1), 232–242. https://doi.org/10.24144/2616-7700.2026.48(1).232-242 [in Ukrainian].
6. Tsehelyk, H. H., & Tsehelyk, M. H. (2026). Approximate method for solving boolean programming problems. In *International Scientific and Practical Conference "Current problems of science, education and technologies: from theoretical foundations to practical solutions of the XXI century".* (pp. 120–124). [in Ukrainian].

Tsehelyk H. H., Hlebena M. I., Tsehelyk M. H. An approximate method for solving a boolean programming problem. An optimality criterion for the approximate solution.

The article considers a mathematical model of a Boolean programming problem for cases where all coefficients of the objective function and constraints are positive. Since exact classical algorithms require a significant amount of computational effort, an iterative approximate method is proposed for solving the problem. The algorithm incrementally constructs partial solutions while simultaneously evaluating the objective function and pruning non-promising alternatives, which reduces the number of required computations. The approximate solution is not always optimal; therefore, the paper provides a sufficient condition under which it can be considered optimal.

Keywords: Boolean programming, mathematical model, approximate algorithm, sufficient optimality condition.

Отримано: 02.04.2026 Прийнято: 18.04.2026 Опубліковано: 30.04.2026