

3.2 Наближені та точні алгоритми рішення задачі цілочисельного лінійного програмування з булевими змінними на основі рангового підходу

На основі висунутих стратегій $\{L_w\}$ і правил $\{K_w\}$ розробимо алгоритми рішення задачі цілочисельного лінійного програмування (ЦЛП) з булевими змінними (БЗ). У результаті були одержані одноетапні і багатоетапні алгоритми, класифікація яких зображені на рис. 1 [220-231]

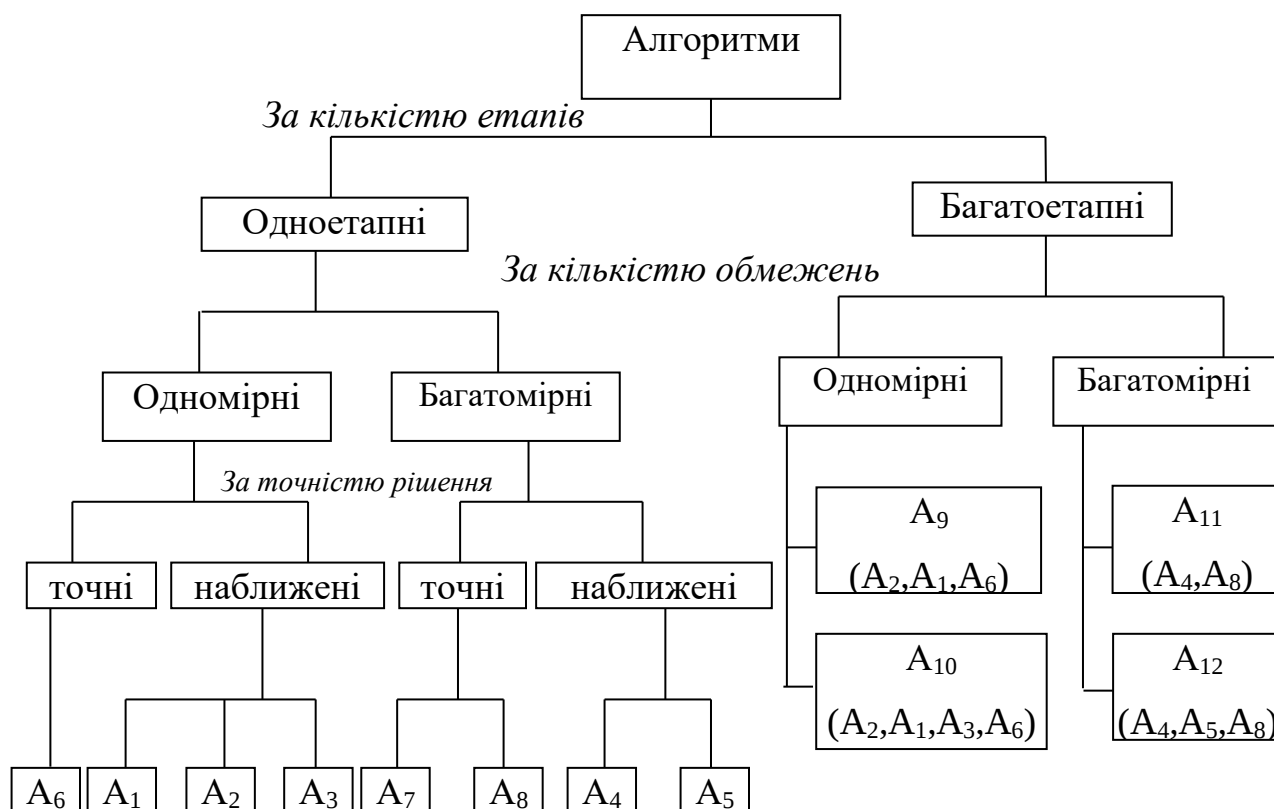


Рисунок 1. Класифікація алгоритмів на основі узагальненої
процедри A_n

Основними кваліфікаційними ознаками є кількість етапів і точність етапів рішення задачі. За першою ознакою алгоритми діляться на одноетапні і багатоетапні. Дане розділення означає, що початкове завдання може розв'язуватися один (одноетапні) або кілька разів (багатоетапні) різними алгоритмами, перелік яких поданий у дужках. Друга ознака визначає погрішність одержуваного допустимого рішення задачі ЦЛП з БП за вагою функціонала від

оптимального рішення. Якщо алгоритм принципово не може дати точного рішення за вагою функціонала, то такий алгоритм називатимемо наближеним.

АЛГОРИТМ А₁. Розгляд наближених алгоритмів почнемо з одновимірної задачі ЦЛП з БП. Найбільш простим є алгоритм А₁, що побудований на основі процедури А₀ і реалізовує стратегії відсікання L₁ і L₃. Його покроковий опис має наступний вигляд.

КРОК 1. З вершини s будується множина шляхів m_{sj}^{r-1} , $j = \overline{(1, n)}$ першого рангу r , що задовольняє властивості, і визначаються в множини m_{sj}^{r-1} шляху максимальної довжини $\left\{ \mu_{sj}^{*r} \right\}$ за вагою функціонала c_j . Для кожної вершини j визначається вага γ_j за правилом:

$$\gamma_j = c_{j+1} + c_{j+2} + \dots + c_n, \quad \gamma_n = 0; \quad j = \overline{(1, n-1)}, \quad (1)$$

КРОК 2. Виключаються шляхи $\{\mu_{sp}^r\}$, $p = \overline{(r, n)}$ у множині m_{sj}^r поточного рангу r , довжині якої $d_c(\mu_{sp}^r)$ задовольняють нерівності:

$$d_c(\mu_{sp}^r) + \gamma_p < \max_{\{c_j\}} \left\{ d_c \left(\mu_{sp}^{*r} \right) \right\}, \quad (2)$$

де $d_c(\mu_{sp}^r)$ – довжина шляху μ_{sp}^r до вершини p рангу r за вагою c_j ; дозволяє виключити цей шлях з подальшого аналізу, як безперспективний, якщо умова виконана

КРОК 3. Формується множина шляхів $m_{sp}^{r=r+1}$, $p = \overline{(1, n)}$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу на основі рекурентного співвідношенн:

$$\mu_{sp}^{r=r+1} = \max_{c_j} \left\{ \mu_{sj}^r \cup (j, p) \right\}; \quad p = \overline{(r+1, n)}; \quad j = \overline{(r, n)}, \quad (3)$$

У освічених множинах $m_{sp}^{r=r+1}$ виділяємо щонайдовші шляхи $\left\{ \begin{matrix} * \\ \mu_{sp} \end{matrix} \right\}^{r=r+1}$. Якщо

опиниться декілька шляхів максимальної довжини, то серед них вибирається шлях з меншим значенням довжини за вагою обмежень a_{1j} .

КРОК 4. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 5; якщо ні, то перевіряємо $r = (n - 1)$. У разі виконання рівності переходимо до кроку 5, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 5. Виділяємо в множині шлях максимальної довжини і алгоритм A_1 закінчує роботу.

Отримання наближеного рішення обумовлене застосуванням стратегії вибору L_1 . Приклад 1, наведений в попередньому розділі ілюструє роботу алгоритму A_1 .

Приклад 1. Потрібно максимізувати функціонал

$$f(\bar{x}) = 20x_1 + 18x_2 + 18x_3 + 15x_4 + 10x_5 + 4x_6 + 1x_7$$

при обмеженні

$$7x_1 + 2x_2 + 5x_3 + 2x_4 + 3x_5 + 1x_6 + 4x_7 \leq 12$$

у відповідності зі стратегією L_1 і L_3 .

Рішення. Ілюстрація рішення приводиться на рис. 2. Тут і далі підкреслення шляху означає те, що тільки на основі цього шляху відбувається побудова множин наступного рангу, інакше шлях виключається з подальшого аналізу. Ліворуч у вигляді стовпця проставлене значення γ_j , $j = (\overline{1,7})$, обумовлене для кожної вершини графа ДД. Так, наприклад $\gamma_3 = c_4 + c_5 + c_6 + c_7 = 15 + 10 + 4 + 1 = 30$.

Відмінною рисою від приклада 1 є наявність порожніх множин m_{sj}^r . Це обумовлено застосуванням відсікань за умови (2). Наприклад, побудуємо шлях

$\mu_{s15}^{r=2}$. До цього часу $\max \left\{ d_c \left(\begin{matrix} * \\ \mu_{s12} \end{matrix} \right)^{r=2} \right\} = 38$, а $d_c(\mu_{s12}^{r=2}) = 30$. Тоді $d_c(\mu_{s15}^{r=2}) + \gamma_5 = 35$. Отже,

на основі шляху $\mu_{s15}^{r=2}$, неможливо побудувати шляхи, що перевищують за вагою функціонала значення вже існуючого локального екстремуму: шляхи $\mu_{s12}^{* r=2}$.

Як видно із приклада 1, спільне застосування стратегій L_1 і L_3 дозволяє в 3 рази зменшити число векторів, що будуються в множинах m_{sj}^r , у порівнянні із застосуванням однієї стратегії L_1 .

γ_j							
66	<u>s1(20,7)</u> 1						
48	<u>s2(18,2)*</u> 2	<u>s12(38,9)</u> 2					
30	<u>s3(18,5)</u> 3	<u>s13(38,12)</u> <u>s23(36,7)*</u> 3					
15	<u>S4(15,2)</u> 4	<u>s14(35,9)</u> <u>s24(33,4)</u> <u>s34(33,7)</u> 4	<u>s124(53,11)</u> <u>s234(51,9)*</u> 4				
5			<u>s125(48,12)</u> 5	<u>s2345(61,12)*</u> 5			
1							
0							
	r = 1	r = 2	r = 3	r = 4	r = 5	r = 6	r = 7

$$* r=4$$

$$\mu_{s2345} \rightarrow \vec{x}^{onm} = (0111100).$$

$$f(\vec{x}^{onm}) = 61$$

Рисунок 2. Ілюстрація роботи стратегії L_1 і L_3

АЛГОРИТМ A_2 . Іншою стратегією вибору шляхів є стратегія L_4 , яка за допомогою алгоритму A_2 дозволяє одержати шляхи максимально можливого рангу r в графі DA . Її можна реалізувати, якщо в D визначати найкоротші шляхи за вагою обмежень на основі процедури A_0 між вершиною s і всією рештою вершин графа D , використовуючи при цьому рекурентне співвідношення (2.15), і, природно, відсікання за правилом L_3 :

$$\mu_{sp}^{r=r+1} = \min_{a_{ij}} \{ \mu_{sj}^r \cup (j, p) \}; \quad p = (\overline{r+1}, n); \quad j = (\overline{r}, n), \quad i = 1. \quad (4)$$

Покроковий опис алгоритму A_2 мають наступний вигляд.

КРОК 1. З вершини s будується множина шляхів $m_{sj}^{r=1}, j = (\overline{1, n})$ першого рангу r , що задовольняє властивості, і визначаються в множини $m_{sj}^{r=1}$ шляху максимальної довжини $\left\{ \mu_{sj}^{*r} \right\}$ за вагою функціонала c_j . Для кожної вершини j визначається вага γ_j за правилом (1).

КРОК 2. Виключаються шляхи в множини m_{sj}^r поточного рангу r , довжини якої $d_c(\mu_{sp}^r)$ задовольняють нерівності (2).

КРОК 3. Формується множина шляхів $m_{sp}^{r=r+1}, p = (\overline{1, n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу, які мають мінімальне значення в m_{sj}^r довжини за вагою обмежень, на основі рекурентного співвідношення (4). У освічених множинах $m_{sp}^{r=r+1}$ виділяємо щонайдовші шляхи $\left\{ \mu_{sp}^{*r=r+1} \right\}$.

КРОК 4. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 5; якщо ні, то перевіряємо $r = (n - 1)$. У разі виконання рівності переходимо до кроку 5, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 5. Виділяємо в множині шлях максимальної довжини і номер останньої вершини w , що належить цьому шляху, після чого алгоритм A_2 закінчує роботу.

АЛГОРИТМ A_3 . Виходячи з поняття коридору і відповідно до стратегії L_7 пропонується наступний наближений алгоритм A_3 рішення задачі ЦЛП х БП який полягає в тому, щоб з множини m_{sj}^r рангу r в множину $m_{sj}^{r=r+1}, j = (\overline{r+1, n})$ наступного рангу вибиратимуться не більше двох шляхів, які задовольняють властивості v .

Першим є шлях з максимальною вагою за функціоналом c_j (відносно рекурентного співвідношення (3)), а другим – шлях з найменшою вагою за

обмеженням $\{a_{1j}\}$ (відносно рекурентного співвідношення (3)), що відповідним чином змінить крок 3 алгоритми A_1 . У іншому покроковий опис алгоритму A_3 повністю співпадає з описом алгоритму A_1 . Втрата оптимального рішення обумовлена його попаданням всередину коридору. Тому, як видно з прикладу 2, даний алгоритм A_3 є наближеним.

Приклад 2. Потрібно максимізувати функціонал

$$f(\bar{x}) = 20x_1 + 19x_2 + 17x_3 + 13x_4 + 12x_5 + 10x_6 + 7x_7 + 5x_8 + 2x_9$$

при обмеженні

$$9x_1 + 8x_2 + 8x_3 + 4x_4 + 3x_5 + 2x_6 + 1x_7 + 5x_8 + 1x_9 \leq 30$$

алгоритмом A_3 (рис. 3).

Рішення. Порядок побудови шляхів в графі D описаний в попередньому розділі.

З кожної множини m_{sj}^r вибирається тільки два шляхи: максимальний за вагою функціонала і мінімальний за вагою обмеження (ці шляхи підкреслені), а інші – ігноруються. З цієї причини в множині $m_{s5}^{r=5}$ виключився з подальшого розгляду шлях, що належить оптимальному, оскільки домінували за алгоритмом A_3 шляхи $\mu_{s125}^{r=3}$ і $\mu_{s245}^{r=3}$. Отже і результатом рішення вийшов шлях, відповідний вектору, що і дало відносну погрішність $f = (64 - 63) / 64 = 1,5\%$.

АЛГОРИТМ A_4 . Для багатовимірної задачі ЦЛП з БП аналогічно також можна побудувати ефективні наближені алгоритми. Таким є алгоритм A_4 , заснований на стратегії L_{10} , який з множини поточного рангу будує в множині наступного рангу $r = r + 1$ тільки шлях з максимальною довжиною за вагою функціонала $\{c_j\}$.

TECHNICAL AND AGRICULTURAL SCIENCES IN MODERN REALITIES: PROBLEMS, PROSPECTS AND SOLUTIONS

γ_j
85
66
49
36
24
14
7
2
0

$* r=6$
 $\mu_{s_{145679}} \rightarrow \bar{x}^{om} = (100111101);$
 $f(\bar{x}^{om}) = 64.$

1						
2	2					
3	3	3				
4	4	4	4			
5	5	5	5	5		
6	6	6	6	6	6	
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9

$r = 1$ $r = 2$ $r = 3$ $r = 4$ $r = 5$ $r = 6$ $r = 7$

Рисунок 3 Ілюстрація роботи алгоритму A_3

При цьому тимчасова складність алгоритму зростає в $2m$ рази, оскільки необхідно виконати для кожного вектора m операцій складання і m операцій порівняння, щоб переконатися, чи задовольняє вектор, відповідний цьому шляху, обмеженню, чи ні. Отже і покроковий опис для алгоритму A_4 співпадає з алгоритмом A_1 .

АЛГОРИТМ A_5 . Застосування стратегії L_{11} дозволило побудувати алгоритм A_5 , суть якого полягає у виборі з кожної множини m_{sj}^r поточного рангу r одного шляху, максимального за вагою функціонала і, у гіршому разі, m шляхів,

мінімальних за вагою обмежень, що задовольняють властивості v . Тоді, на відміну від алгоритму A_1 , покроковий опис алгоритму A_5 зміниться на:

КРОК 1. З вершини s будується множина шляхів $m_{sj}^{r=1}, j = (\overline{1, n})$ першого рангу r , що задовольняє властивості, і визначаються в множини $m_{sj}^{r=1}$ шляху максимальної довжини $\left\{ \mu_{sj}^r \right\}$ за вагою функціонала c_j . Для кожної вершини j визначається вага c_j за правилом (1).

КРОК 2. Виключаються шляхи в множини m_{sj}^r поточного рангу r , довжини якої $d_c(\mu_{sp}^r)$ задовольняють нерівності (2).

КРОК 3. Формується множина шляхів $m_{sp}^{r=r+1}, p = (\overline{1, n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу на основі рекурентних співвідношень. У освічених множинах $m_{sp}^{r=r+1}$ виділяємо щонайдовші шляхи $\left\{ \mu_{sp}^{r=r+1} \right\}$. Якщо виявиться декілька шляхів максимальної довжини, то серед них вибирається шлях з меншим значенням довжини за вагою обмеження $\{a_{ij}\}$. Якщо виявиться декілька шляхів з мінімальним значенням по i -му обмеженню, то вибирається той, у якого більше довжина за вагою функціонала.

КРОК 4. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 5, якщо ні, то перевіряємо $r = (n - 1)$. У разі виконання рівності переходимо до кроку 5, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 5. Виділяємо в множині шлях максимальної довжини і алгоритм A_5 закінчує роботу.

Цей алгоритм має одну невизначеність у тому випадку, коли в множині опиниться декілька векторів з максимальною довжиною за вагою функціонала і з різними значеннями за вагою обмежень, тобто, наприклад, за вагою першого обмеження більше другий вектор, за вагою другого - перший, за вагою третього – знову другий і т. д.

Тому, для уникнення цієї невизначеності при рівності довжин за вагою функціонала вибирається шлях з меншим значенням за вагою всіх обмежень. Якщо таке неможливе, то вибирається будь-який. Може статися, що відкинутий шлях належатиме оптимальному шляху. Ця обставина і є однією з причин отримання алгоритмом A_5 наближеного рішення.

Побудова точних алгоритмів ґрунтується на понятті коридору і використанні стратегій $L_6, L_8, L_9, L_{12}, L_{13}$ відсікання. Для точних алгоритмів аналітична оцінка тимчасової складності і величини апаратурних витрат становить трудність із-за NP-повноти розв'язуваної задачі. Основною причиною неможливості оцінки є випадкове число векторів усередині коридору, що виділяється. Природа випадковості кількості векторів усередині коридору визначається співвідношеннями між коефіцієнтами функціонала c_j і коефіцієнтами обмежень a_{ij} , а також приростами цих співвідношень між j -мі компонентами двійкового вектора $\bar{x} = x_1, \dots, x_n$ при заданому законі розподілу коефіцієнтів у функціоналі і обмеженнях.

АЛГОРИТМ A_6 . На основі стратегій L_6, L_8, L_9 побудуємо точний алгоритм A_6 рішення одновимірної задачі ЦЛП з БП, покроковий опис якого матиме наступний вигляд.

КРОК 1. З вершини s будується множина шляхів $m_{sj}^{r=1}, j = \overline{(1, n)}$ першого рангу r , що задовольняє властивості, і визначаються в множині $m_{sj}^{r=1}$ шляху максимальної довжини $\left\{ \mu_{sj}^{*r} \right\}$ за вагою функціонала c_j . Для кожної вершини j формуються калібрувальні вектори $\bar{y}_j$ (5) і $\bar{z}_j$ (6):

$$y_{jk} = a_{jk} + y_{j(k-1)}; k = \overline{(1, n-j)}; y_{j0} = 0; y_{jn} = b_j; j = \overline{(1, n-1)}. \quad (5)$$

$$\hat{z}_{jk}^B = c_{j+k} + \hat{z}_{j(k-1)}^B; k = \overline{(1, n-j)}; \hat{z}_{j0}^B = 0; \hat{z}_{jn}^B = 0; j = \overline{(1, n-1)}. \quad (6)$$

КРОК 2. Виключаються шляхи в множині m_{sj}^r поточного рангу r , довжини якої $d_c(\mu_{sp}^r)$ задовольняють нерівності (1).

КРОК 3. Для кожного шляху $\{\mu_{sj}^r\}$, $j = (\overline{1, n})$ поточного рангу r визначається за правилом K_1 значення $\hat{r}_e$. Виключаються шляхи, довжини яких задовольняють нерівності:

$$d_c(\mu_{sp}^r) + \hat{z}_{p_{i_e}}^e(\mu_{sp}^r) < \max_{c_j} \left\{ d_c \left(\mu_{sp}^{\begin{smallmatrix} * \\ r \end{smallmatrix}} \right) \right\}. \quad (7)$$

КРОК 4. Формується множина шляхів $m_{sp}^{r=r+1}$, $p = (\overline{1, n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу на основі принципу оптимізації за напрямком з виділенням коридору по стратегії L_6 і виключенням векторів усередині коридору відповідно до стратегії L_8 . Шлях в множині $m_{sp}^{r=r+1}$ може бути сформований, якщо він задовольняє властивості. Якщо властивість v не виконується, то шлях виключається з подальшого аналізу.

У освічених множинах $m_{sp}^{r=r+1}$ виділяємо щонайдовші шляхи $\left\{ \mu_{sp}^{\begin{smallmatrix} * \\ r=r+1 \end{smallmatrix}} \right\}$.

КРОК 5. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 6, якщо ні, то перевіряємо $r = (n - 1)$. У разі виконання рівності переходимо до кроку 6, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 6. Виділяємо в множині шлях максимальної довжини і алгоритм A_6 закінчує роботу.

АЛГОРИТМ A_7 . Для рішення багатовимірної задачі ЦЛП з БЗрозглянемо два точних алгоритми A_7 і A_8 , заснованих на понятті m -мірного коридору і стратегій вибору L_{12} , L_{13} відповідно. Принципова відмінність алгоритмів A_7 і A_8 полягає в тому, що перший виділяє m -мірний коридор усередині множини поточного рангу r графа D , а другий виділяє m -мірний коридор безпосередньо на всьому ярусі графа $D\Delta$. Відсікання, засновані на стратегії L_9 і правилах K_2 , K_3 , K_4 , залишаються однаковими для обох алгоритмів. Покроковий опис алгоритму A_7 має наступний вигляд.

КРОК 1. З вершини s будується множина шляхів $m_{sj}^{r=1}, j = (\overline{1, n})$ першого рангу r , що задовольняє властивості, і визначаються в множині $m_{sj}^{r=1}$ шляху максимальної довжини $\left\{ \begin{smallmatrix} * & r \\ \mu_{sj} \end{smallmatrix} \right\}$ за вагою функціонала c_j . Для кожної вершини j формуються калібрувальні вектора $\bar{y}_j^i$ (5) і $\bar{z}_j$ (6).

КРОК 2. Для кожного шляху $\{\mu_{sj}^r\}, j = (\overline{r, n})$ поточного рангу r визначається за правилом K_1 значення r_B . Виключаються шляхи, довжини яких задовольняють нерівності (7).

КРОК 3. Формується множина шляхів $m_{sp}^{r=r+1}, p = (\overline{1, n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу на основі принципу оптимізації за напрямком з виділенням m -мірного коридору у вигляді підмножини $m_{sp}^{rk} \in m_{sp}^{r=r+1}$ відповідно до стратегії L_{12} і відсіканням неперспективних векторів за правилом K_4 . Шлях в множині $m_{sp}^{r=r+1}$ може бути сформований, якщо він задовольняє властивості. Якщо властивість v не виконується, то шлях виключається з подальшого аналізу. У освічених множинах m_{sp}^{rk} виділяємо щонайдовші шляхи $\left\{ \begin{smallmatrix} * & r=r+1 \\ \mu_{sp} \end{smallmatrix} \right\}$.

КРОК 4. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 5, якщо ні, то перевіряємо $r = (n - 1)$. У разі виконання рівності переходимо до кроку 5, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 5. Виділяємо в множині шлях максимальної довжини і алгоритм A_7 закінчує роботу.

АЛГОРИТМ A_8 . З метою усунення цього недоліку пропонується алгоритм A_8 , що здійснює виділення m -мірного коридору на всьому ярусі. Суть цього алгоритму полягає в тому, що якщо на ярусі в множинах, які лежать в графі ДД вище за деякий шлях μ_{sp}^{*r} рангу r існує шлях, котрий має не менше значення за

вагою функціонала і менше значення за вагою всіх обмежень, то такий шлях $m_{sp}^{r=r+1}$ не потрапляє в m -мірний коридор на ярусі. В цьому випадку покроковий опис A_8 прийме наступний вигляд.

КРОК 1. З вершини s будується множина шляхів $m_{sj}^{r=1}, j = (\overline{1, n})$ першого рангу r , що задовольняє властивості, і визначаються в множини $m_{sj}^{r=1}$ шляхи

максимальної довжини $\left\{ \begin{matrix} * & r \\ \mu_{sj} \end{matrix} \right\}$ за вагою функціонала c_j . Для кожної вершини j

формується калібрувальні вектори $\bar{y}_j$ (5) і $\bar{z}_j$ (6).

КРОК 2. Виключаються шляхи в множини m_{sj}^r поточного рангу r , довжини якої $d_c(\mu_{sp}^r)$ задовольняють нерівності (1).

КРОК 3. Формується множина шляхів $m_{sp}^{r=r+1}, p = (\overline{1, n})$ наступного рангу, що задовольняє властивості, на базі множини шляхів m_{sj}^r попереднього рангу на основі принципу оптимізації за напрямком. Шлях в множині $m_{sp}^{r=r+1}$ може бути сформований, якщо він задовольняє властивості. Якщо властивість v не виконується, то шлях виключається з подальшого аналізу. У освічених множинах $m_{sp}^{r=r+1}$ виділяємо локальні екстремуми $\left\{ \begin{matrix} * & r=r+1 \\ \mu_{sp} \end{matrix} \right\}$.

КРОК 4. Виконуємо процедуру виділення m -мірного коридору на всьому сформованому ранзі $r = r + 1$. Для цього послідовно перевіряються всі множини $m_{sp}^{r=r+1}, p = (\overline{r+1, n})$, відповідно до стратегії L_{13} і проводиться відсікання неперспективних векторів за правилом K_4 .

КРОК 5. Перевіряється, чи вся множина шляхів наступного $(r+1)$ -го рангу порожня. Якщо це так, то переходимо до кроку 6, якщо ні, то перевіряємо $r = (n-1)$. У разі виконання рівності також переходимо до кроку 6, інакше збільшуємо r на 1 і виконуємо крок 2.

КРОК 6. Виділяємо в множині, шлях максимальної довжини і алгоритм A_8 закінчує роботу.

Таким чином, на основі принципу оптимізації за напрямком у дискретному просторі станів, який визначається графом $D\Delta$ вдалося побудувати точні алгоритми для вирішення як одновимірного, так і багатовимірного завдання ЦЛП з БЗ. Покажемо, що для зниження загальної тимчасової складності, за рахунок звуження ширини коридору і вагою функціонала, доцільно використовувати багатоетапну фільтрацію.