УДК 519.161

А. І. Лєтошко, Д. Д. Плечистий

ТОЧНИЙ ЕФЕКТИВНИЙ АЛГОРИТМ РОЗВ'ЯЗАННЯ ОБМЕЖЕНОГО ВАРІАНТА ЗАДАЧІ КОМІВОЯЖЕРА

У статті запропоновано точний поліноміальний алгоритм розв'язання частинного випадку задачі комівояжера, отриманого із задачі побудови конвеєрного розкладу для безпосередньо виконуваних робіт, який може бути застосованим для підвищення ефективності функціонування інтегрально-вимірювальних систем.

Постановка проблеми. Ефективність функціонування інтегрально-вимірювальних систем (ІВС) може бути підвищена за рахунок оптимізації режимів їх окремих ланок при одночасному узгодженні сусідніх частин [1]. Одним з найбільш важливих показників якості роботи ІВС є швидкодія, яка забезпечується не лише шляхом покращення технічних характеристик функціональних ланок, але й за допомогою методів організації їх узгодженої роботи. ІВС зручно описати у вигляді трьох компонент [1]. Складовими першої компоненти є поле джерел інформації, комутатор з пристроєм керування та аналогово-цифровий перетворювач, що йдуть один за одним. До другої належать запам'ятовуючий пристрій (ЗП) та канал зв'язку, що йде за ним. Третьою компонентою є обчислювальний пристрій (ОП).

Завдання підвищення ефективності функціонування ІВС може бути сформульованим у вигляді задачі побудови конвеєрного розкладу для безперервно виконуваних робіт.

Огляд останніх досліджень і публікацій. У роботі [1] розглянуто проблему підвищення ефективності функціонування ІВС за рахунок узгодження їх сусідніх частин. У роботі [2] описано процедуру переходу від задачі побудови конвеєрного розкладу для безперервно виконуваних робіт до задачі комівояжера (ЗК). У монографії [3] досліджено задачі оптимізації послідовно-паралельних процесів, побудови розкладів для конвеєрних систем, проаналізовано їх складність, запропоновано точні та наближені алгоритми їх розв'язання. У монографіях [4–5] розглянуто вузлові питання проблеми комівояжера, сформульовано та узагальнено ряд частинних випадків та узагальнень ЗК, запропоновано точні та наближені методи їх розв'язання.

Формулювання завдання дослідження. Метою статті є розробка точного поліноміального методу розв'язання одного частинного випадку ЗК, який виникає в задачі підвищення ефективності функціонування ІВС.

Виклад основного матеріалу. Нехай у конвеєрній системі із m машин роботу i, $i=\overline{1,n}$, виконують без переривань. Позначимо t_{ij} час виконання етапу (стадії) j роботи i, $j=\overline{1,m}$. Будемо вважати, якщо $t_{ij}=0$, то робота i надходить на машину j, але тривалість її перебування на цій машині є занадто малою, тому нею можна знехтувати. У конвеєрній системі, що виконує n робіт на m машинах, розклад π знаходимо з матриці $\left[t_{ij}\right]_{n\times m}$ та визначаємо заданням матриці моментів початку $\left[t_{ij}^{o}\right]_{n\times m}$ або матриці моментів закінчення етапів робіт $\left[\bar{t}_{ij}\right]_{n\times m}$. При цьому момент закінчення всіх робіт конвеєрного розкладу π дорівнює

$$\bar{t}_{max}(\pi) = max\left\{\bar{t}_{ij} \mid i=\overline{1,n},\ j=\overline{1,m}\right\}. \qquad (1)$$

Потрібно побудувати оптимальний за швидкодією розклад π^{*}, тобто такий, що надасть найменше значення $\bar{t}_{max}(\pi)$.

Відомо, що для безперервно виконуваних робіт i, які надходять на кожну машину j, розклад мінімальної довжини π^{*} слід шукати в класі переставних розкладів $\pi = (\pi[1],...,\pi[k],...,\pi[n])$, де $\pi[k]$ –номер k-ї роботи в послідовності π [2]. Довжину розкладу π визначаємо як

$$\bar{t}_{max}(\pi) = \bar{t}_{\pi[n]m} = t_{\pi[n]1}^{0} + \sum_{j=1}^{m} t_{\pi[n]j}\ .$$

На рис. 1 наведено дві безперервно виконувані роботи, що займають позиції $k-1$ та k у розкладі π.

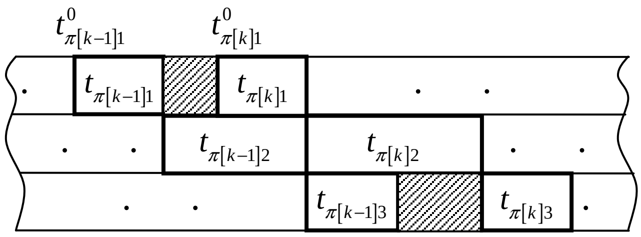

Рис. 1. Дві безперервно виконувані роботи

Приймемо $t_{\pi[1]1} = 0$ та оберемо довільне k, $2 \le k \le n$. З рис. 1 видно, що $t_{\pi[k]1}^{0} \ge t_{\pi[k-1]1}^{0} + t_{\pi[k-1]1}$, $t_{\pi[k]1}^{0} + \sum_{j=1}^{l-1} t_{\pi[k]j} \ge t_{\pi[k-1]1}^{0} + \sum_{j=1}^{l} t_{\pi[k-1]j}$, $l=\overline{2,m}$, звідки

$$t_{\pi[k]1}^{0} - t_{\pi[k-1]1}^{0} \ge \max_{1\le l\le m}\left(\sum_{j=1}^{l} t_{\pi[k-1]j} - \sum_{j=1}^{l-1} t_{\pi[k]j}\right).$$

Момент часу $t_{\pi[k]1}^{o}$ оберемо таким, щоб остання нерівність перетворилася в рівність, тоді

$$t_{\pi[1]1}^{0} = 0,\ t_{\pi[k]1}^{0} - t_{\pi[k-1]1}^{0} = \max_{1\le l\le m}\left(\sum_{j=1}^{l} t_{\pi[k-1]j} - \sum_{h-1}^{l-1} t_{\pi[k]j}\right),\ k=\overline{2,n},$$

$$t^0_{\pi[n]1} = \sum_{k=1}^{n-1} \max_{1\le l\le m}\left(\sum_{j=1}^{l} t_{\pi[k-1]j} - \sum_{j=1}^{l-1} t_{\pi[k]j}\right),$$

$$\bar{t}_{\pi[n]m} = \bar{t}_{max}(\pi) = \sum_{k=1}^{n-1} \max_{1\le l\le m}\left(\sum_{j=1}^{l} t_{\pi[k-1]j} - \sum_{j=1}^{l-1} t_{\pi[k]j}\right) + \sum_{j=1}^{m} t_{\pi[n]j} \, . \tag{2}$$

З урахуванням висновків з [2] задача мінімізації (1) формулюється як ЗК з $n+1$ містами, які позначено числами 0, 1, …, n, та матрицею вартостей $\left[c_{pq}\right]_{n+1}$, де $c_{pp} = \infty$, $c_{pq} \ge 0$, $p = \overline{0,n}$, $q = \overline{0,n}$, $p \ne q$. Для такого варіанта ЗК потрібно знайти перестановку $\pi_0^* = \left(0, \pi[1]^*, ..., \pi[n]^*\right)$, що мінімізує на множині всіх $(n-1)!$ перестановок $\pi_0 = (0, \pi[1], ..., \pi[n])$ функціонал

$$C(\pi_0) = c_{0\pi[1]} + \sum_{k=1}^{n-1} c_{\pi[k]\pi[k+1]} + c_{\pi[n]0} \, , \tag{3}$$

де $c_{0p} = 0$, $c_{p0} = \sum_{j=1}^{m} t_{pj}$, $c_{pq} = \max_{1\le l\le m}\left(\sum_{j=1}^{l} t_{pj} - \sum_{j=1}^{l-1} t_{qj}\right)$, $p = \overline{1,n}$, $q = \overline{1,n}$, $p \ne q$, $c_{00} = c_{pp} = \infty$.

Розв’язання ЗК у наведеній постановці дає оптимальний розклад $\pi^* = \left(\pi[1]^*, ..., \pi[n]^*\right)$ з n безперервно виконуваних робіт у конвеєрній системі, що містить m машин.

За відсутності обмежень на значення тривалостей етапів робіт у матриці $\left[t_{ij}\right]_{n\times m}$, $m > 2$, матриця вартостей $\left[c_{pq}\right]_{n+1}$, утворена з неї, не містить властивостей, які дозволили б зменшити трудомісткість знаходження π^* або отримати за поліноміальний час наближений розв’язок з гарантованою, аналітично вираженою оцінкою похибки. У такому разі для роботи з матрицями вартості реальних розмірів практичними алгоритмами залишаються лише швидкодіючі евристичні конструкції з експериментально підтвердженими межами відхилення від оптимуму.

Зрозуміло, що задача мінімізації (1) у загальному випадку є NP-повною в сильному розумінні. Разом з тим, за певних припущень щодо значень параметрів матриці $\left[t_{ij}\right]_{n\times m}$ вона може бути розв’язана за поліноміальну кількість операцій.

Відомо, що при $m = 2$ перестановку π^*, яка надає мінімум (1), знаходять за час $O(n \log_2 n)$ [3]. У цьому разі

$$\bar{t}_{max}(\pi) = \sum_{k=1}^{n-1} \max\left\{b_{\pi[k]} - a_{\pi[k+1]}, 0\right\} + \sum_{k=1}^{n} a_{\pi[k]} + b_{\pi[k]} \, , \tag{3}$$

де $a_i = t_{i1}$, $b_i = t_{i2}$, $i = \overline{1,n}$.

Тоді в ЗК, яка відповідає задачі побудови оптимального за швидкодією розкладу для n робіт, кожну з яких виконують без переривань, у конвеєрній системі з двох машин, отримаємо такі значення матриці вартостей $\left[c_{pq}\right]_{n+1}$:

$$c_{0p}=0,\; c_{p0}=b_p,\; c_{pq}=max\left\{b_p-a_q,0\right\},\; p=\overline{1,n},\; q=\overline{1,n},\; p\neq q.$$

Задача знаходження мінімуму (1) при $m=3$ та за відсутності обмежень на невід'ємні значення $t_{i1}=a_i$, $t_{i2}=b_i$, $t_{i3}=c_i$, $i=\overline{1,n}$, є NP-складною в сильному розумінні [2]. Покажемо, що вона ефективно розв'язувана, якщо $b_i=b$ для всіх $i=\overline{1,n}$.

Лема 1. Щоб отримати оптимальний розв'язок задачі мінімізації (1) при $m=3$ та $b_i=b$, $i=\overline{1,n}$, достатньо знайти перестановку π^*, якій відповідає найменше значення (3) для $b_i=max(b,c_i)$, $i=\overline{1,n}$.

Доведення. З (2) при $m=3$ отримаємо

$$\bar{t}_{max}(\pi)=\sum_{k=1}^{n-1}max\left(a_{\pi[k]},a_{\pi[k]}+b-a_{\pi[k+1]},a_{\pi[k]}+b+c_{\pi[k]}-b-a_{\pi[k+1]}\right)+a_{\pi[n]}+b+c_{\pi[n]}=$$

$$=\sum_{k=1}^{n-1}max\left(0,b-a_{\pi[k+1]},c_{\pi[k]}-a_{\pi[k+1]}\right)+\sum_{k=1}^{n}a_{\pi[k]}+b+c_{\pi[k]}.$$ Нехай $b_{\pi[k]}=max\left(b,c_{\pi[k]}\right),k=\overline{1,n},$ тоді

$$\bar{t}_{max}(\pi)=\sum_{k=1}^{n-1}max\left(0,b_{\pi[k]}-c_{\pi[k+1]}\right)+\sum_{k=1}^{n}a_{\pi[k]}+b+c_{\pi[n]}.$$

Якщо $c_{\pi[k]}>b$, $k\in\{1,2,...,n\}$, то вираз для $\bar{t}_{max}(\pi)$ збігається з формулою (3) з точністю до константи: $\bar{t}_{max}(\pi)=\sum_{k=1}^{n-1}max\left(0,b_{\pi[k]}-a_{\pi[k+1]}\right)+\sum_{k=1}^{n}a_k+b+b_{\pi[n]}$.

Якщо $c_{\pi[k]}\le b$, то воно не залежить від параметра $b_{\pi[n]}$:

$$\bar{t}_{max}(\pi)=\sum_{k=1}^{n-1}max\left(0,b_{\pi[k]}-a_{\pi[k+1]}\right)+\sum_{k=1}^{n}a_k+2b$$
.

Наслідок 1. Нехай у задачі мінімізації (1) $t_{i2}=t_{i3}=...=t_{im-1}=b$, $i=\overline{1,n}$, $n>3$. Тоді її оптимальним розв'язком є перестановка π^*, що надає мінімум (3) при $b_i=max(b,c_i)$.

Викладені міркування можуть дати практичні результати, наприклад, на стадії розробки ІВС, тому опишемо не зовсім очевидну, на перший погляд, можливість використання одного з варіантів задачі мінімізації (1) для підвищення швидкодії ІВС.

Перша компонента ІВС незалежно від принципу дії комутатора, кількості джерел, груп, що обслуговуються та швидкості опитування, є частиною, яка виробляє послідовність обсягів даних, що надходять до ЗП через визначені (у загальному випадку нерівні) інтервали часу.

Нехай система виконує виміри за допомогою n пристроїв i, що опитуються комутатором, та процес її функціонування починається в момент часу 0. Опитування джерел інформації виконується впродовж визначеного інтервалу часу. Комутатор з інтервалом заданої тривалості Δ_i підключає джерело інформації i, $i=\overline{1,n},$ до наступних функціональних ланок ІВС. Найбільш часто опитування виконується за жорсткою або циклічною схемою, згідно з якою порядок підключення джерел упродовж всього проміжку часу роботи системи залишається незмінним. Будемо називати повним циклом послідовність з n підключень комутатора до n різних джерел інформації.

Перетворення починається одразу ж після підключення комутатора до чергового об'єкта i, $i=\overline{1,n}$. Перетворювач у кожний момент може виконувати дії не більш ніж над обсягом одного i.

Нехай час перетворення i-го обсягу інформації дорівнює t_{i1}, $i=\overline{1,n}$. Якщо підключення комутаторів виконуються з інтервалами Δ_i, то $t_{i1} \le \Delta_i$. Припустимо, що система виконує один повний цикл перетворень.

Позначимо t_{i2} час введення в ЗП i-ої сукупності даних. Для ряду ІВС ЗП являє собою буферний пристрій з обмеженим обсягом пам'яті, що дозволяє зберігати результати тільки за одним опитуваним пристроєм. Якщо ЗП здатен містити дані, отримані лише з одного джерела інформації, то в цьому випадку повинна виконуватися умова

$$\bar{t}_{(i-1)2} \le \bar{t}_{i1},\ i=\overline{2,n},$$

де $\bar{t}_{i1}$ – момент закінчення дій з перетворення даних, отриманих у результаті послідовного опитування комутатора з джерела i, $\bar{t}_{(i-1)2}$ – момент закінчення введення до ЗП $(i-1)$-го обсягу даних.

При обсязі пам'яті, що забезпечує одночасне зберігання даних для всього циклу опитування, наведена нерівність не виконується. Її невиконання означає розширення можливостей для збільшення швидкодії системи за рахунок зростання обсягу ЗП, а в кінцевому підсумку – ускладнення конфігурації ІВС.

Припустимо, що відома тривалість t_{i3} обробки ОП i-го об'єкта даних. Обмеження на обсяг ЗП визначають співвідношення

$$\bar{t}_{(i-1)3} \le \bar{t}_{i2},\ i=\overline{2,n},$$

де $\bar{t}_{(i-1)3}$ – момент завершення дій, виконаних ОП над $(i-1)$-м обсягом даних.

У результаті побудови послідовності опитування джерел інформації $\pi^* = \left(\pi[1]^*,...,\pi^*[i],...\pi[n]^*\right)$, що мінімізує $\bar{t}_{max}(\pi)$, можна визначити моменти $t^0_{\pi[i]^*1}$ та інтервали підключення комутатора $\Delta_{\pi[i]^*}$ до кожного вимірюваного об'єкта i, які пов'язані такими співвідношеннями:

$$\Delta_{\pi^*[i-1]} = t^0_{\pi[i]^*1} - t^0_{\pi[i-1]^*1},\ i=\overline{2,n}, \Delta_{\pi^*[n]} = t_{\pi[n]^*1}.$$

Таким чином, $t_{max}\left(\pi^*\right) = \sum_{i=2}^{n} \Delta_{\pi[i-1]^*1} + t_{\pi[n]^*1} + t_{\pi[n]^*2} + t_{\pi[n]^*3}$.

Метод побудови локальних оптимальних послідовностей можна успішно застосовувати при розробці алгоритмів ефективно розв'язуваних підзадач ЗК. Розглянемо його особливості для побудови оптимального за швидкодією розкладу задачі (1), у якій вихідна матриця тривалостей $\left[t_{ij}\right]_{n\times m}$ є упорядкованою за рядками, тобто для будь-якої пари робіт i та k справедливо $t_{ij} \ge t_{kj}$ або $t_{ij} \le t_{kj}$, $j=\overline{1,m}$. У [2] показано, що введені обмеження на тривалості стадій дозволяють мінімізувати (1) за поліноміальний час. Спочатку обговоримо властивості цього частинного випадку задачі знаходження мінімуму (1), а потім на їх основі опишемо алгоритм, що прискорює процес утворення розкладу π^* мінімальної довжини при гранично малому обсязі пам'яті, що використовується.

Будемо вважати, що в матриці, впорядкованій за рядками, $t_{ij} \le t_{kj}$ та $i<k$, $j=\overline{1,m}$. Якщо у вихідній матриці, упорядкованій за рядками, для деякої пари робіт i та k $t_{ij} \ge t_{kj}$, $i<k$, $j=\overline{1,m}$, то подамо рядки i в порядку неспадання значень t_{ij}, $i=\overline{1,n}$, $j=\overline{1,m}$.

З [2] відомо, що оптимальний розв'язок задачі (1) для випадку, коли $\left[t_{ij}\right]_{n\times m}$ упорядковано за рядками, достатньо шукати в класі пірамідальних перестановок $p=\left(p[1],\ldots,p[n]\right)$. Пірамідальною перестановкою називають послідовність з n об'єктів, пронумерованих числами $1,2,\ldots,n$, вигляду $\left(p[1],\ldots,p[l-1],p[l],p[l+1],\ldots,p[n]\right)$, де $p[l]=n$, $p[1]<\ldots<p[l-1]$, $p[l+1]>\ldots>p[n]$.

Зазначимо, що пірамідальна перестановка з n елементів множини $\{1,2,\ldots,n\}$ породжує дві пірамідальні перестановки з $n+1$ елементів множини $\{1,2,\ldots,n,n+1\}$ у результаті вставки елемента $n+1$ безпосередньо перед та після елемента n. Звідси випливає простий спосіб утворення всіх перестановок $p=\left(p[1],\ldots,p[n]\right)$, який полягає в такому. Нехай P_p – множина всіх пірамідальних перестановок $p=\left(p[1],\ldots,p[n]\right)$. Подамо клас P_p як розбиття на два підкласи P_{p1} та P_{p2}. Підклас P_{p1} містить всі послідовності з n елементів, що починаються з елемента 1, а до P_{p2} включено всі перестановки, які закінчуються цим елементом. Тоді при $n=2$ $P_{p1}=\{(1,\ 2)\}$, $P_{p2}=\{(2,\ 1)\}$. При $n=3$ отримаємо $P_{p1}=\{(1,\ 3,\ 2),(1,\ 2,\ 3)\}$, $P_{p2}=\{(3,\ 2,\ 1),(2,\ 3,\ 1)\}$. Таким чином можна отримати для будь-якого n всі перестановки множин P_{p1} та P_{p2}. Зрозуміло, що $\left|P_{p1}\right|=\left|P_{p2}\right|=2^{n-2}$, $\left|P_p\right|=2^{n-1}$.

Нескладно побачити, що пірамідальна перестановка $p=\left(p[1],\ldots,p[n]\right)$ є циклічною та $P_p \subset P_\tau$, де P_τ – множина всіх циклічних перестановок з n елементів множини $\{1,2,\ldots,n\}$, $\left|P_\tau\right|=(n-1)\,!$. У пірамідальній перестановці $p=\left(p[1],\ \ldots,\ p[l-1],\ \ p[l],\ p[l+1],\ \ldots,\ p[n]\right)$, де $p[l]=n$, деякий один з елементів $p[l-1],p[l+1]$ дорівнює $n-1$.

Сформулюємо задачу мінімізації (1) для матриці $\left[t_{ij}\right]_{n\times m}$, упорядкованої за рядками, як ЗК з $n+1$ містами $0,\ 1,\ 2,\ \ldots,\ n$ та з матрицею вартостей $\left[c_{sq}\right]_{n+1}$, де $c_{ss}=\infty$, $c_{sq}\ge 0$,

$s=\overline{0,n}$, $q=\overline{0,n}$, $s\neq q$. Для такого варіанта ЗК потрібно на множині всіх послідовностей $p_0=(0,p)$, $p=(p[1],...,p[n])$ знайти послідовність $p_0^*=\left(0,p^*\right)$, що надає мінімум функціоналу

$$C(p_0)=c_{0p[1]}+\sum_{k=1}^{n-1}c_{p[k]p[k+1]}+c_{p[n]0},$$

де $c_{0s}=0$, $c_{s0}=\sum_{j=1}^{m}t_{sj}$, $c_{sq}=\max\limits_{1\le l\le m}\left(\sum_{j=1}^{l}t_{sj}-\sum_{j=1}^{l-1}t_{qj}\right)$, $s=\overline{1,n}$, $q=\overline{1,n}$, $s\neq q$, $c_{00}=c_{ss}=\infty$.

Скористаймося відомими властивостями матриці $\left[c_{sq}\right]_{n+1}$, отриманої в результаті перетворення матриці $\left[t_{ij}\right]_{n\times m}$, упорядкованої за рядками [3]:

а) $c_{rk}\ge c_{ik}$, $n\ge r>i\ge 1$, $r\neq k$, $i\neq k$, $k=\overline{1,n}$;

б) $c_{ik}\ge c_{ij}$, $n\ge j>k\ge 1$, $i\neq k$, $i\neq j$, $i=\overline{1,n}$;

в) $c_{rk}-c_{ik}\ge c_{rj}-c_{ij}$, $n\ge j>k\ge 1$, $n\ge r>i\ge 1$.

З останньої властивості отримаємо такий результат.

Лема 2. Якщо $\left[t_{ij}\right]_{n\times m}$ – матриця, упорядкована за рядками, то у відповідній до неї матриці вартостей ЗК $[c_{cs}]_{n+1}$ виконується нерівність $c_{r0}-c_{i0}\ge c_{rk}-c_{ik}$, $n\ge r>i\ge 1$, $k=\overline{1,n}$.

Доведення. Із нерівності в) випливає, що для доказу твердження достатньо показати правдивість нерівності $c_{r0}-c_{i0}\ge c_{r1}-c_{i1}$, $n\ge r>i\ge 1$. Тоді $c_{r0}=\sum_{j=1}^{m}t_{rj}$, $c_{i0}=\sum_{j=1}^{m}t_{ij}$, $c_{r1}=\max\limits_{1\le l\le m}\left(\sum_{j=1}^{l}t_{rj}-\sum_{j=1}^{l-1}t_{1j}\right)$, $c_{i1}=\max\limits_{1\le l\le m}\left(\sum_{j=1}^{l}t_{ij}-\sum_{j=1}^{l-1}t_{1j}\right)$. Оскільки $t_{rj}\ge t_{ij}$ для всіх j, $j=\overline{1,m}$, то завжди виконується нерівність $\sum_{j=1}^{m}t_{rj}-\sum_{j=1}^{m}t_{ij}\ge\max\limits_{1\le l\le m}\left(\sum_{j=1}^{l}t_{rj}-\sum_{j=1}^{l}t_{ij}\right)$.

Зазначимо ще одну властивість матриці $\left[c_{sq}\right]_{n+1}$, отриману в результаті перетворення матриці $\left[t_{ij}\right]_{n\times m}$, упорядкованої за рядками.

Лема 3. Елементи матриці $\left[c_{sq}\right]_{n+1}$ пов'язані нерівністю

г) $c_{rk}-c_{rj}\ge c_{ik}-c_{ij}$, $n\ge j>k\ge 0$, $n\ge r>i\ge 0$, $r\neq k$, $i\neq k$, $r\neq j$, $i\neq j$.

Доведення. Нехай $c_{ik}\ge c_{rj}$. Тоді

$$c_{rk}-c_{rj}=\left(c_{rk}-c_{ik}\right)+\left(c_{ik}-c_{rj}\right)\ge c_{ik}-c_{ij}=\left(c_{ik}-c_{rj}\right)+\left(c_{rj}-c_{ij}\right).$$

Якщо $c_{ik}<c_{rj}$, то

$$c_{rk}-c_{rj}=\left(c_{rk}-c_{ik}\right)-\left(c_{rj}-c_{ik}\right)\ge c_{ik}-c_{ij}=\left(c_{rj}-c_{ij}\right)-\left(c_{rj}-c_{ik}\right).$$

Тепер маємо можливість описати точний алгоритм розв'язання обмеженої версії ЗК у постановці (2).

S0. $\left[c_{pq}\right]_{n+1}$ – матриця вартостей ЗК, у якій значення c_{pq}, $p=\overline{0,n}$, $q=\overline{0,n}$, пов'язані умовами а)–г).

Побудувати цикли $\tau_3^1=(0,\ 1,\ 3,\ 2,\ 0)$, $\tau_3^2=(0,\ 1,\ 2,\ 3,\ 0)$, $\tau_3^3=(0,\ 3,\ 2,\ 1,\ 0)$, $\tau_3^4=$ $=(0,\ 2,\ 3,\ 1,\ 0)$, визначити вартості $C\left(\tau_3^1\right)=c_{13}+c_{32}+c_{20}$, $C\left(\tau_3^2\right)=c_{12}+c_{23}+c_{30}$, $C\left(\tau_3^3\right)=$ $=c_{32}+c_{21}+c_{10}$, $C\left(\tau_3^4\right)=c_{23}+c_{31}+c_{10}$; прийняти, що $l=3$, $m_l=4$.

S1. Якщо $l=n$, то кінець: вартість шуканого циклу τ^* дорівнює $C\left(\tau^*\right)=min\left\{C\left(\tau_l^{jl}\right) \mid j_l=\overline{1,m_l}\right\}$.

S2. Виконати розбиття множини $\left\{\tau_l^{jl} \mid j_l=\overline{1,m_l}\right\}$ на дві підмножини T_R та T_L так, щоб у кожному циклі множини T_R індекс $l-1$ йшов би після l, а в кожному циклі множини T_L він розташовувався б перед l; виключити з аналізу безперспективні цикли в T_R та T_L, звертаючись, відповідно, до випадків 1, 2 та 1, 3.

S3. $l=l+1$; з кожного циклу отриманої множини утворити два цикли $(\ldots,l,l-1,\ldots)$ та $(\ldots,l-1,l,\ldots)$, обчислити їх вартості $C\left(\tau_l^{jl}\right)$; перейти до кроку S1.

Теорема 1. Алгоритм, описаний послідовністю кроків S0–S3, за час $O\left(n^2\right)$ знаходить розв'язок, який надає мінімум функціоналу (2).

Доведення. Опишемо процес побудови циклу мінімальної вартості τ^* за допомогою дерева з n рівнів. Зіставимо єдину вершину першого рівня, що є коренем дерева, цикл $(0,\ 1,\ 0)$ та його вартість c_{10}. Другий рівень складається з двох вершин $\tau_2^1=(0,\ 1,\ 2,\ 0)$ та $\tau_2^2=(0,\ 2,\ 1,\ 0)$ з вартостями $C\left(\tau_2^1\right)=c_{12}+c_{20}$ та $C\left(\tau_2^2\right)=c_{21}+c_{10}$. З кожної вершини другого рівня виходить дві дуги, кінцями яких є чотири вершини третього рівня $\tau_3^1=(0,\ 1,\ 3,\ 2,\ 0)$, $\tau_3^2=(0,\ 1,\ 2,\ 3,\ 0)$, $\tau_3^3=(0,\ 3,\ 2,\ 1,\ 0)$, $\tau_3^4=(0,\ 2,\ 3,\ 1,\ 0)$, що визначають множину всіх пірамідальних перестановок розмірності 3. Вершина τ_2^1 породжує пару нащадків τ_3^1 та τ_3^2, а вершина τ_2^2 – пару нащадків τ_3^3 та τ_3^4. Кожній вершині третього рівня $\tau_3^{j_3}$, $j=\overline{1,4}$, приписано вагу, що дорівнює вартості відповідного циклу: $C\left(\tau_3^1\right)=C\left(\tau_2^1\right)+c_{13}+c_{32}-c_{12}$, $C\left(\tau_3^2\right)=C\left(\tau_2^1\right)+c_{23}+c_{30}-c_{20}$, $C\left(\tau_3^3\right)=C\left(\tau_2^2\right)+c_{03}+$ $+c_{32}-c_{02}$, $C\left(\tau_3^4\right)=C\left(\tau_2^2\right)+c_{23}+c_{31}-c_{21}$. З будь-якої вершини рівня l, $l=\overline{4,n}$, виходить не більше двох дуг.

Множину всіх циклів $\tau_n^{j_n}$, $j_n=\overline{1,m_n}$, що містить оптимальний розв'язок задачі, можна подати у вигляді поєднання неперетинних підмножин T_{32} та T_{23}. Підмножина T_{32} складається з усіх циклів, у яких індекс 2 йде за індексом 3, а підмножина T_{23} містить усі цикли, де індекс 2 йде перед індексом 3. Очевидно, значення $C\left(\tau_3^1\right)$ та $C\left(\tau_3^3\right)$ обмежують знизу мінімум функціонала (2), визначеного на підмножині T_{32}. Аналогічно вартості $C\left(\tau_3^2\right)$ та $C\left(\tau_3^4\right)$ є оцінками знизу мінімуму функціонала (2) на підмножині T_{23}.

При зіставленні циклів τ_3^1 та τ_3^3, які утворюють підмножину T_R, з неї у випадку $C\left(\tau_3^1\right) \le C\left(\tau_3^3\right)$ виключається частковий розв'язок τ_3^3, а значить, і всі допустимі розв'язки $\tau_n^{j_n} \in T_{32}$ з нижньою межею, не меншою $C\left(\tau_3^3\right)$. Тоді вершина τ_3^1 на рівні 4 породжує дві вершини $\tau_4^1 = (0,\ 1,\ 4,\ 3,\ 2,\ 0)$ та $\tau_4^2 = (0,\ 1,\ 3,\ 4,\ 2,\ 0)$, $\tau_4^1 \in T_R$, $\tau_4^2 \in T_L$.

Порівняння циклів τ_3^2 та τ_3^4, які утворюють підмножину T_L, у випадку $C\left(\tau_3^4\right) \le C\left(\tau_3^2\right)$ дозволяє відкинути частковий розв'язок τ_3^2 та ті допустимі розв'язки в T_{23}, для яких нижня межа значень (2) не менше $C\left(\tau_3^2\right)$. Тоді при $l = 4$ з τ_3^4 утворюються дві вершини: $\tau_4^3 = (0,\ 2,\ 4,\ 3,\ 1,\ 0)$, $\tau_4^4 = (0,\ 2,\ 3,\ 4,\ 1,\ 0)$, $\tau_4^3 \in T_R$, $\tau_4^4 \in T_L$.

Якщо $C\left(\tau_3^3\right) < C\left(\tau_3^1\right)$, то в результаті порівняння вартостей циклів $(0,\ 1,\ 3,\ 4,\ 2,\ 0)$ та $(0,\ 3,\ 4,\ 2,\ 1,\ 0)$, що дорівнюють $C\left(\tau_3^1\right) + c_{34} + c_{42} - c_{32}$ та $C\left(\tau_3^3\right) + c_{34} + c_{42} - c_{32}$, виключається цикл $(0,\ 1,\ 3,\ 4,\ 2,\ 0)$ та, відповідно, усі допустимі розв'язки вигляду $(0,\ 1,\ 3,\ ...,\ n,\ ...,\ 2,\ 0)$ з підмножини T_{32}. На рівні $l = 4$ з вершин τ_3^1 та τ_3^3 утворюються три вершини: $\tau_4^1 = (0,\ 1,\ 4,\ 3,\ 2,\ 0)$, $\tau_4^2 = (0,\ 4,\ 3,\ 2,\ 1,\ 0)$, $\tau_4^3 = (0,\ 3\ ,4,\ 2,\ 1,\ 0)$, τ_4^1, $\tau_4^2 \in T_R$, $\tau_4^3 \in T_L$.

У випадку $C\left(\tau_3^2\right) < C\left(\tau_3^4\right)$ порівняння вартостей циклів $(0,\ 1,\ 2,\ 4,\ 3,\ 0)$ та $(0,\ 2,\ 4,\ 3,\ 1,\ 0)$, що відповідно дорівнюють $C\left(\tau_3^2\right) + c_{24} + c_{43} - c_{23}$ та $C\left(\tau_3^4\right) + c_{24} + c_{43} - c_{23}$, призводить до виключення часткового розв'язку $(0,\ 2,\ 4,\ 3,\ 1,\ 0)$ та всіх допустимих розв'язків вигляду $(0,\ 2,\ ...,\ n,\ ...,\ 3,\ 1,\ 0)$ з підмножини T_{23}. Тоді вершини τ_3^2 та τ_3^4 дерева розв'язків на рівні 4 утворюють три вершини $\tau_4^4 = (0,\ 1,\ 2,\ 4,\ 3,\ 0)$, $\tau_4^5 =$ $= (0,\ 1,\ 2,\ 3,\ 4,\ 0)$, $\tau_4^6 = (0,2,3,4,1,0)$, $\tau_4^4 \in T_R$, τ_4^5, $\tau_4^6 \in T_L$.

У загальному випадку в результаті порівняння циклів, отриманих на $l-1$-му рівні, $l = \overline{3, n-1}$, утворюється підмножина T_R часткових розв'язків вигляду $(...,\ l,\ l-1,\ ...)$ та підмножина T_L часткових розв'язків вигляду $(...,\ l-1,\ l,\ ...)$. На n-му рівні дерева множині $T_R \cup T_L$ відповідає підмножина множини $T_{32} \cup T_{23}$ допустимих розв'язків задачі, що містить шуканий розв'язок.

Оцінимо трудомісткість запропонованого алгоритму, яка залежить від кількості всіх циклів $\tau_l^{j_l}$, $l = \overline{2, n}$, $j_l = \overline{1, m_l}$, та буде потрібною в гіршому випадку для побудови оптимального розв'язку τ^*.

Зазначимо, що $m_l = 2$ при $l = 2$, $m_l = 4$ при $l = 3$. У гіршому випадку, коли $C\left(\tau_3^3\right) < C\left(\tau_3^1\right)$ та $C\left(\tau_3^2\right) < C\left(\tau_3^4\right)$, маємо $m_4 = 6$. Таким чином, $\left|T_R \cup T_L\right| \le 2l - 2$, $l = \overline{2, 4}$.

Покажемо, що на рівні l, $l=\overline{5,n}$, кількість часткових розв'язків, що виключають втрату оптимального розв'язку τ^*, обмежено величиною $2l-2$.

Нехай $l=5$ та $\left|T_R \cup T_L\right|=6$ на рівні 4. Тоді $T_R=\left\{\tau_4^1,\ \tau_4^2,\ \tau_4^4\right\}$, $T_L=\left\{\tau_4^3,\ \tau_4^5,\ \tau_4^6\right\}$. З T_R отримаємо три цикли $\tau_5^1=(0,\ 1,\ 5,\ 4,\ 3,\ 2,\ 0)$, $\tau_5^3=(0,\ 5,\ 4,\ 3,\ 2,\ 1,\ 0)$, $\tau_5^5=(0,\ 1,\ 2,\ 5,\ 4,\ 3,\ 0)$ та три цикли $\tau_5^2=(0,\ 1,\ 4,\ 5,\ 3,\ 2,\ 0)$, $\tau_5^4=(0,\ 4,\ 5,\ 3,\ 2,\ 1,\ 0)$, $\tau_5^6=(0,\ 1,\ 2,\ 4,\ 5,\ 3,\ 0)$, серед яких завжди в результаті порівняння способом 1 обирається один цикл з найменшою вартістю. Аналогічно з T_L визначимо три цикли $\tau_5^8=(0,\ 3,\ 4,\ 5,\ 2,\ 1,\ 0)$, $\tau_5^{10}=(0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 0)$, $\tau_5^{12}=(0,\ 2,\ 3,\ 4,\ 5,\ 1,\ 0)$, а з трьох інших циклів $\tau_5^7==(0,\ 3,\ 5,\ 4,\ 2,\ 1,\ 0)$, $\tau_5^9=(0,\ 1,\ 2,\ 3,\ 5,\ 4,\ 0)$, $\tau_5^{11}=(0,\ 2,\ 3,\ 5,\ 4,\ 1,\ 0)$ надалі будемо розглядати той, якому відповідає $min\left\{C\left(\tau_5^7\right),C\left(\tau_5^9\right),C\left(\tau_5^{11}\right)\right\}$. Таким чином, у гіршому випадку, коли $C\left(\tau_5^3\right)<C\left(\tau_5^1\right)<C\left(\tau_5^5\right)$ і $C\left(\tau_5^{10}\right)<C\left(\tau_5^{12}\right)<C\left(\tau_5^8\right)$, кількість циклів множини $T_R \cup T_L$ не більше $2l-2=8$.

Припустимо, що на рівні $l-1$, $l=\overline{6,n}$, $m_{l-1}=\left|T_R\right|+\left|T_L\right|\le 2(l-1)-2$.

Тоді на рівні l у гіршому випадку отримаємо $\left|T_R\right|$ часткових розв'язків вигляду $(...,\ j,\ l,\ l-1,\ ...)$, $j\in\{0,\ 1,\ ...,\ l-3\}$, $\left|T_L\right|$ часткових розв'язків вигляду $(...,\ l-1,\ l,\ k,\ ...)$, $k\in\{0,\ 1,\ ...,\ l-3\}$, а також два цикли $(...,\ l-2,\ l,\ l-1,\ ...)$ та $(...,\ l-1,\ l,\ l-2,\ ...)$. Таким чином, $m_l=\left|T_R\right|+\left|T_L\right|+2\le 2l-2$.

Отже, кількість усіх побудованих вершин дерева розв'язання становить: $\sum_{l=1}^{n} m_l \le \sum_{l=1}^{n}(2l-2)=n(n-1)$, а складність алгоритму, відповідно, – $O\left(n^2\right)$.

Висновки. Запропоновано точний поліноміальний алгоритм розв'язання частинного випадку ЗК, отриманого із задачі побудови конвеєрного розкладу для безпосередньо виконуваних робіт, до якої зводиться задача підвищення ефективності функціонування ІВС. Оптимальний розв'язок задачі знайдено за допомогою розробленого алгоритму за час $O\left(n^2\right)$.

СПИСОК ЛІТЕРАТУРИ

1. Акимов А. А. Выбор последовательностей для опроса измерительных устройств / А. А. Акимов, А. В. Панишев // Электронное моделирование. – 1988. – № 5. – С. 95–96.
2. Танаев В. С. Теория расписаний. Многостадийные системы / В. С. Танаев, Ю. Н. Сотсков, В. А. Струсевич. – М. : Наука, 1989. – 328 с.
3. Панішев А. В. Вступ до теорії складності дискретних задач : монографія / А. В. Панішев, О. М. Данильченко, В. О. Скачков. – Житомир : ЖДТУ, 2004. – 236 с.
4. Панишев А. В. Модели и методы оптимизации в проблеме коммивояжера / А. В. Панишев, Д. Д. Плечистый. – Житомир : ЖГТУ, 2006. – 300 с.

5. Панишев А. В. Модели и методы оптимизации замкнутых маршрутов на транспортной сети : монография / А. В. Панишев, А. В. Морозов. – Житомир : ЖДТУ, 2014. – 316 с.

Подано 18.05.2015

А. И. Летошко, Д. Д. Плечистый

ТОЧНЫЙ ЭФФЕКТИВНЫЙ АЛГОРИТМ РЕШЕНИЯ ОГРАНИЧЕННОГО ВАРИАНТА ЗАДАЧИ КОММИВОЯЖЕРА

В статье предлагается точный полиномиальный алгоритм решения частного случая задачи коммивояжера, полученного из задачи построения конвейерного расписания для непосредственно выполняемых работ, который может быть применен для повышения эффективности функционирования интегрально-измерительных систем.

A. I. Letoshko, D. D. Plechystyy

EXACT EFFICIENT ALGORITHM FOR SOLVING LIMITED VERSION OF THE TRAVELING SALESMAN PROBLEM

The exact polynomial algorithm for solving a particular case of the Traveling Salesman Problem, resulting from the problem of constructing a conveyor schedules directly to work done, which can be used to enhance the operation of integrated measurement systems is proposed in the article.