

Zbigniew TARAPATA*

MODELE HARMONOGRAMOWANIA ZSYNCHRONIZOWANEGO PRZEMIESZCZANIA WIELU OBIEKTÓW

W pracy przedstawiono problem wyznaczania harmonogramu zsynchronizowanego przemieszczania wielu obiektów. Omówiono szereg modeli harmonogramowania przemieszczania. Zdefiniowano dwie grupy kryteriów, istotnych z punktu widzenia oceny harmonogramu: kryteria związane z szybkością przemieszczania obiektów oraz z „równoległością” ich przemieszczania. Skoncentrowano się na sformułowaniu nieliniowego zadania harmonogramowania przemieszczania obiektów. Przedstawiono również dwa równorzędne sformułowania problemu w postaci dwukryterialnych zadań programowania matematycznego. Wykazano, że macierz współczynników ograniczeń w tych zadaniach jest całkowicie unimodularna, co umożliwi zastosowanie efektywnych algorytmów rozwiązywania zadań programowania liniowego, przy poszukiwaniu np. leksykograficznego rozwiązania problemu dwukryterialnego. Omówiono podobieństwa i różnice między sformułowanym problemem harmonogramowania, a klasycznym problemem szeregowania zadań przed liniami krytycznymi w celu minimalizacji maksymalnego opóźnienia zadań. Zdefiniowano szereg rozszerzeń omawianego problemu.

Słowa kluczowe: *harmonogramowanie i synchronizacja przemieszczania, drogi najkrótsze, drogi rozłączne, wielokryterialne problemy dróg najkrótszych*

1. Wprowadzenie

Harmonogramowanie to jedno z najważniejszych zastosowań badań operacyjnych w przemyśle, transporcie, teleinformatyce, sieciach komputerowych, wojsku [1], [2]. Jednym z problemów jest harmonogramowanie przemieszczania obiektów, wykorzystywane w zagadnieniach: routingu w sieciach komputerowych, planowania przemieszczania mobilnych robotów, przetwarzania zadań w systemach równoległych

* Zakład Badań Operacyjnych i Wspomagania Decyzji, Instytut Systemów Informatycznych, Wydział Cybernetyki, Wojskowa Akademia Techniczna, ul. Kaliskiego 2, 00-908 Warszawa, e-mail: zbigniew.tarapata@wat.edu.pl

(rozproszonych). Szczególnym typem wspomnianego problemu jest harmonogramowanie równoczesnego (synchronicznego) przemieszczania wielu obiektów. Problem ten wykorzystywany jest między innymi w symulatorach walki, komputerowych grach symulacyjnych (w podsystemie planowania i symulacji przemieszczania obiektów (jednostek) [6], [7], [8], [9], [12], [13]), ale również przy synchronizacji pracy wielu przemieszczających się agentów (obiektów), jak np. w problemach sterowania ramionami wielu niezależnych robotów. W każdym z tych systemów harmonogramowanie przemieszczania ma wpływ na dokładność odwzorowania (adekwatność) modeli, efektywność, użyteczność rozwiązań i inne charakterystyki wspomnianych systemów. Szczególnie interesujące mogą być takie rozwiązania, które pozwalają osiągnąć pewne cele (jeden lub więcej), na przykład: osiągnięcie punktów docelowych przez wszystkie obiekty w ściśle określonym czasie, równoczesne przybycie wszystkich obiektów do punktów pośrednich, spełnienie pewnych dodatkowych ograniczeń (np. brak zatrzymywania się w punktach pośrednich).

Jednym ze wspomnianych zastosowań omawianego problemu jest planowanie i synchronizacja przemieszczania obiektów w komputerowych grach symulacyjnych [12], [13]. W grach typu człowiek–komputer wykorzystywana jest technika generowania przez komputer zachowania drugiej strony (tzw. systemy CGF, ang. *Computer Generated Forces*) [8], [13] lub SAF – ang. *Semi-Automated Forces* [7]). Niezależnie od rodzaju działań, jednostki każdego z graczy przemieszczane są jako grupy obiektów (jednostek niższego szczebla, pojazdów). Zachowanie odpowiedniego ugrupowania w czasie działań jest bardzo istotne z punktu widzenia realizacji celu działania. Dla przykładu: każdy obiekt podlegający przemieszczaniu (podczas ataku, przegrupowania itp.), będący elementem grupy obiektów (jednostek, zgrupowań, kolumn transportowych), musi „trzymać” ugrupowanie, tzn. musi przemieszczać się biorąc pod uwagę położenie innych obiektów grupy zgodnie z pewnym wzorcem ugrupowania.

W rozdziale 2 niniejszego artykułu podano podstawowe oznaczenia i definicje stosowane w pracy. Rozdział 3 zawiera sformułowanie problemu harmonogramowania zsynchronizowanego przemieszczania wielu obiektów wraz z komentarzem. W rozdziale 4 opisano pewne rozszerzenia problemu, zdefiniowanego w rozdziale 3. Rozdział 5 zawiera krótkie wprowadzenie do metod rozwiązywania sformułowanych w rozdziałach 3 i 4 problemów. Pracę podsumowują wnioski.

2. Definicje i oznaczenia

Zakładamy, że struktura środowiska przemieszczania (sieci dróg lub terenu podzielonego na heksagony, kwadraty itd. [8], [9], [12]) reprezentowana jest przez graf Berge’a G , przy czym $G = \langle V_G, A_G \rangle$, $V = |V_G|$, V_G jest zbiorem wierzchołków grafu (skrzyżowań lub kwadratów (ich środków) terenu), A_G – zbiorem łuków grafu,

$A_G \subset V_G \times V_G$, $A = |A_G|$. Zakładamy, że dla każdego łuku grafu G dysponujemy wartością $d_{n,n'}$ funkcji d , która opisuje odległość terenową między wierzchołkami n oraz n' . Dysponujemy K obiektami (kolumnami pojazdów, pojazdami, zadaniami w sieci komputerowej itp.), które chcemy przemieścić z wektora $s = (s_1, s_2, \dots, s_K)$ wierzchołków początkowych do wektora $t = (t_1, t_2, \dots, t_K)$ wierzchołków końcowych w G . W dalszych rozważaniach przyjmujemy następujące oznaczenia:

$$I_k(s_k, t_k) = I_k = (i^0(k) = s_k, i^1(k), \dots, i^r(k), \dots, i^{R_k}(k) = t_k), \quad (1)$$

$$T_k(I_k) = T_k = (\tau^0(k), \tau^1(k), \dots, \tau^r(k), \dots, \tau^{R_k}(k)), \quad (2)$$

$$V_k(I_k) = V_k = (v_{i^0(k), i^1(k)}, v_{i^1(k), i^2(k)}, \dots, v_{i^{R_k-1}(k), i^{R_k}(k)}), \quad (3)$$

gdzie: I_k – wektor wierzchołków opisujących drogę dla k -tego obiektu, $\forall_{m \in \{1, \dots, R_k\}} (i^{m-1}(k), i^m(k)) \in A_G$, $i^r(k)$ – r -ty wierzchołek na drodze k -tego obiektu, T_k – wektor chwil osiągnięcia wierzchołków należących do drogi dla k -tego obiektu; $\tau^r(k)$ – chwila osiągnięcia wierzchołka $i^r(k)$ przez czoło k -tego obiektu, $\forall_{k=1, K} \forall_{r=0, R_k-1} \tau^{r+1}(k)$

$\geq \tau^r(k) \geq 0$, przy czym przyjmujemy, że $\forall_{k=1, K} \tau^0(k) \geq 0$ oraz

$$\tau^r(k) = \tau^0(k) + \sum_{j \in \{0, \dots, r-1\}} \frac{d_{i^j(k), i^{j+1}(k)}}{v_{i^j(k), i^{j+1}(k)}}, \quad r > 0, \quad (4)$$

V_k – wektor prędkości k -tego obiektu na łukach jego drogi; $v_{i^r(k), i^{r+1}(k)}$ – prędkość k -tego obiektu na łuku $(i^r(k), i^{r+1}(k))$ między wierzchołkami $i^r(k)$ oraz $i^{r+1}(k)$ drogi; $R_k + 1$ – liczba wierzchołków drogi dla k -tego obiektu. Niech $\Pi(s, t)$ opisuje zbiór wektorów $I(s, t) = (I_1, I_2, \dots, I_K)$ dróg z $s = (s_1, s_2, \dots, s_K)$ do $t = (t_1, t_2, \dots, t_K)$. Zdefiniujmy τ^* jako najpóźniejszy moment dotarcia do wierzchołka docelowego któregoś z obiektów:

$$\tau^* = \min_{I(s, t) \in \Pi(s, t)} \max_{k \in \{1, \dots, K\}} \tau^{R_k}(k). \quad (5)$$

Oznaczmy ponadto przez IP_k następujący wektor punktów wyrównania:

$$IP_k = (i_1(k), i_2(k), \dots, i_p(k), \dots, i_{p_k}(k)), \quad (6)$$

gdzie $i_p(k)$ – p -ta składowa wektora IP_k opisująca wierzchołek (punkt) wyrównania, w którym musimy wyrównać czoło k -tego obiektu w odniesieniu do czoł pozostałych obiektów, spełniająca warunki:

$$\begin{aligned} \forall_{p \in \{1, \dots, P_k\}} \exists_{r \in \{1, \dots, R_k\}} i_p(k) = i^r(k), \\ r_p(k) = r \in \{1, \dots, R_k\} \Leftrightarrow i_p(k) = i^r(k). \end{aligned} \quad (7)$$

Warunek (7) zapewnia, że dla k -tego obiektu droga I_k musi przechodzić przez wierzchołki należące do IP_k . Oznaczmy przez analogię do IP_k :

$$TP_k = (\tau_1(k), \tau_2(k), \dots, \tau_p(k), \dots, \tau_{P_k}(k)) \quad (8)$$

jako wektor chwil osiągnięcia przez k -ty obiekt punktów wyrównania należących do IP_k , $\tau_p(k)$ oznacza chwilę osiągnięcia p -tego punktu wyrównania przez k -ty obiekt,

$$\tau_p(k) = \tau^0(k) + \sum_{r \in \{0, \dots, r_p(k)-1\}} \frac{d_{i^r(k), i^{r+1}(k)}}{v_{i^r(k), i^{r+1}(k)}} \quad (9)$$

oraz przyjmujemy, że dla każdego $k = 1, \dots, K$ zachodzi: $\tau_0(k) = \tau^0(k)$.

Zauważmy, że $\tau_p(k)$ możemy również liczyć w sposób dynamiczny następująco:

$$\tau_p(k) = \tau_{p-1}(k) + \sum_{r \in \{r_{p-1}(k), \dots, r_p(k)-1\}} \frac{d_{i^r(k), i^{r+1}(k)}}{v_{i^r(k), i^{r+1}(k)}}, \quad p \geq 1, \quad (10)$$

przy czym $\tau_0(k) = \tau^0(k)$. Przyjmijemy również, że τ_p^{\max} oznacza chwilę dotarcia do p -tego punktu wyrównania najwolniejszego obiektu, tzn.

$$\tau_p^{\max} = \max_{k \in \{1, \dots, K\}} \tau_p(k). \quad (11)$$

Dodatkowo zakładamy, że $P_1 = P_2 = \dots = P_K = N$, tzn. dla każdego obiektu dysponujemy taką samą liczbą punktów wyrównania.

3. Modele harmonogramowania przemieszczania obiektów

3.1. Opis i sformułowanie problemu

Problem harmonogramowania zsynchronizowanego przemieszczania K obiektów, którym będziemy się zajmować możemy zdefiniować następująco: wyznaczyć dla każdego obiektu $k \in \{1, \dots, K\}$ drogę I_k przechodzącą przez punkty wyrównania ze zbioru IP_k oraz dla każdego odcinka (łuku) $(i^r(k), i^{r+1}(k))$, $r \in \{0, \dots, R_{k-1}\}$ drogi taką prędkość $0 < v_{i^r(k), i^{r+1}(k)} \leq v^{\max}(k)$, gdzie $v^{\max}(k)$ oznacza maksymalną prędkość

k -tego obiektu wynikającą z jego możliwości technicznych, że spełnione są pewne cele (jeden lub więcej). Generalnie cele te możemy podzielić na dwa typy:

- pierwszego typu (C.1), definiujące miary oceny szybkości przemieszczania:

$$\tau^{\max} = \max_{k \in \{1, \dots, K\}} \tau^{R_k}(k) \rightarrow \min; \quad (\text{C.1.1})$$

$$\sum_{k=1} \tau^{R_k}(k) \rightarrow \min; \quad (\text{C.1.2})$$

- drugiego typu (C.2), definiujące miary oceny „równoległości” przemieszczania:

$$\sum_{p=1}^{P_k} \sum_{k=1}^K \tau_p^{\max} - \tau_p(k) \rightarrow \min; \quad (\text{C.2.1})$$

$$\min_{p \in \{1, \dots, N\}} \max_{k \in \{1, \dots, K\}} (\tau_p^{\max} - \tau_p(k)); \quad (\text{C.2.2})$$

$$\sum_{p=1}^{P_k} \sum_{k=1}^K |\tau_p^{\text{avg}} - \tau_p(k)| \rightarrow \min; \quad (\text{C.2.3})$$

gdzie $\tau_p^{\text{avg}} = \frac{1}{K} \sum_{k=1}^K \tau_p(k)$.

Założenia dodatkowe, które mogą być brane pod uwagę przy sformułowaniu powyższego problemu harmonogramowania są następujące: drogi dla K obiektów muszą być rozłączne bądź nie, muszą przechodzić przez wskazane wcześniej punkty wyrównania lub punkty te wyliczane są dynamicznie w trakcie przemieszczania, każdy k -ty obiekt może planować przemieszczanie tylko wewnątrz pewnych podobszarów itp. W pierwszym przypadku, gdy składowe wektorów s i t są różne (czyli dysponujemy K parami różnych wierzchołków początkowych i końcowych) mamy do czynienia z NP-trudnym problemem poszukiwania K dróg rozłącznych i możemy go rozwiązać używając pewnych algorytmów przybliżonych [10], [11]. Jeżeli natomiast komponenty wektorów s i t są identyczne (czyli K obiektów startuje z tego samego wierzchołka początkowego i kończy w tym samym wierzchołku końcowym, przy czym $s \neq t$), wówczas możemy skorzystać z wniosków wynikających z twierdzeń Halla i Mengersa dotyczących dróg wierzchołkowo lub łukowo-rozłącznych i zastosować algorytm wyznaczania przepływu zaspokajającego, o minimalnym koszcie w pewnej sieci zastępczej [10]. W drugim i trzecim przypadku można zastosować jeden ze znanych algorytmów wyznaczania dróg przechodzących przez wskazane wierzchołki sieci [5]. Pewien specyficzny przypadek problemu rozłącznych dróg to sytuacja, kiedy każdy obiekt ma narzucony z góry swój „pas” terenu, w którym się przemieszcza, pasy są parami rozłączne i generując na bazie takich pasów podgrafy grafu G , a następnie poszukując w każdym z nich najkrótszej drogi, wyznaczamy faktycznie rozłączne drogi dla obiektów [12].

Jednym ze sposobów rozwiązania uogólnionego problemu harmonogramowania zsynchronizowanego przemieszczania K obiektów może być podejście dwuetapowe: najpierw wyznaczane są drogi najkrótsze I_k dla każdego k -tego obiektu (przy założeniu, że obiekty poruszają się w sieci z maksymalną prędkością $v^{\max}(k)$), a następnie rozwiązywany jest problem wyznaczenia takich prędkości $v_{i^r(k),i^{r+1}(k)}$, $r = \overline{0, R_k - 1}$, $k = \overline{1, K}$, że

$$\sum_{p=1}^{P_k} \sum_{k=1}^K \tau_p^{\max} - \tau_p(k) \rightarrow \min, \quad (12)$$

przy ograniczeniach:

$$v_{i^r(k),i^{r+1}(k)} \leq v^{\max}(k), \quad r = \overline{0, R_k - 1}, \quad k = \overline{1, K}, \quad (13)$$

$$v_{i^r(k),i^{r+1}(k)} > 0, \quad r = \overline{0, R_k - 1}, \quad k = \overline{1, K}, \quad (14)$$

gdzie $v^{\max}(k)$ oznacza maksymalną prędkość k -tego obiektu, wynikającą z jego możliwości technicznych. Sformułowany problem polega więc na takim poprawieniu (zmniejszeniu) prędkości odcinkowych dla każdego obiektu, aby uzyskać efekt „równoległości” przemieszczania obiektów, mierzony np. za pomocą wartości funkcji (12). Przyjęcie ograniczenia (14) powoduje, że zabronione jest zatrzymywanie się na każdym łuku (odcinku drogi). Jeżeli (14) zapiszemy w postaci nieostrej nierówności, to dopuścimy zatrzymywanie się na tych odcinkach. Biorąc pod uwagę (9) i (11), funkcję celu (12) możemy zdefiniować następująco:

$$\sum_{p=1}^N \sum_{k=1}^K \left(\max_{j \in \{1, \dots, K\}} \left(\tau^0(j) + \sum_{\substack{r \in \{0, \dots, R_j - 1\} \\ r \leq r_p(j)}} \frac{d_{i^r(j),i^{r+1}(j)}}{v_{i^r(j),i^{r+1}(j)}} \right) - \left(\tau^0(k) + \sum_{\substack{r \in \{0, \dots, R_k - 1\} \\ r \leq r_p(k)}} \frac{d_{i^r(k),i^{r+1}(k)}}{v_{i^r(k),i^{r+1}(k)}} \right) \right) \rightarrow \min \quad (15)$$

Opisane podejście dwuetapowe odpowiada sytuacji poszukiwania rozwiązania leksykograficznego dla problemu dwukryterialnego z następującą kolejnością ważności kryteriów: C.1.2, C.2.1, czyli (12). Najpierw będziemy poszukiwać dróg najkrótszych dla K obiektów w zbiorze dróg dopuszczalnych przy założeniu, że obiekty poruszają się z maksymalnymi dopuszczalnymi prędkościami (wówczas minimalizujemy sumę czasów przemieszczania wszystkich obiektów), a następnie – mając drogi najkrótsze dla K obiektów – rozwiązujemy zadanie (12)–(14).

Zauważmy, że problem (12)–(14) jest podobny do problemu szeregowania zadań na równoległych procesorach [1], [2]. *Podobieństwa*: (a) związek z problemem szeregowania zadań przed liniami krytycznymi w celu minimalizacji sumy maksymalnych

opóźnień w punktach (wierzchołkach) wyrównania; p -ta linia krytyczna jest tworzona przez wierzchołki $i_p(1), i_p(2), \dots, i_p(K)$; (b) jako zadania rozpatrujemy odcinki dróg (łuki $(i^r(k), i^{r+1}(k))$); (c) jako procesory rozpatrujemy przemieszczane obiekty (K); (d) zadania są niepodzielne i zależne (zależność jest definiowana przez łuki $\forall_{m \in \{1, \dots, R_k\}}$ $(i^{m-1}(k), i^m(k)) \in A_G$, należące do drogi dla każdego obiektu). *Różnice*, które powodują, że problem zasadniczo odbiega od wspomnianego problemu szeregowania zadań są następujące: (a) zadania są już przydzielone do konkretnych procesorów (nie mamy na to wpływu), (b) decydujemy o opóźnieniach pracy procesorów w celu wydłużenia czasów realizacji zadań.

3.2. Harmonogramowanie przemieszczania jako dwukryterialne zadanie optymalizacji

Rozpatrywany w rozdziale 3.1 problem harmonogramowania możemy również sformułować w postaci dwukryterialnego zadania programowania matematycznego. Przyjmujemy następujące oznaczenia: $\mathbf{D} = [d_{ij}]_{V \times V}$ – macierz odległości (terenowych) między sąsiednimi wierzchołkami i oraz j ($d_{ij} = +\infty$ jeśli z i do j brak przejścia); $\mathbf{A} = [a_{ink}]_{V \times M \times K}$ – macierz wierzchołków startowych oraz docelowych poprzez punkty pośrednie dla każdego z obiektów (droga dla każdego obiektu dzielona jest na $M = N + 1$ odcinków dróg od jednego punktu pośredniego do drugiego, przy czym pierwszym odcinkiem drogi będzie droga od wierzchołka początkowego do pierwszego pośredniego, a ostatnim (M -tym) – droga od ostatniego wierzchołka pośredniego do końcowego): $a_{ink} = 1$ jeśli i -ty wierzchołek jest n -tym wierzchołkiem początkowym dla k -tego obiektu; $a_{ink} = -1$ jeśli i -ty wierzchołek jest n -tym wierzchołkiem końcowym dla k -tego obiektu; $a_{ink} = 0$ w przeciwnym przypadku; dodatkowo muszą być spełnione następujące warunki: $a_{i1k} = 1 \Leftrightarrow i = s_k$ (wierzchołkiem początkowym pierwszego odcinka drogi dla k -tego obiektu jest s_k), $a_{i1k} = -1 \Leftrightarrow i = i_1(k)$ (wierzchołkiem końcowym pierwszego odcinka drogi dla k -tego obiektu jest pierwszy punkt pośredni $i_1(k)$ tego obiektu), $a_{iMk} = 1 \Leftrightarrow i = i_N(k)$ (wierzchołkiem początkowym ostatniego odcinka drogi dla k -tego obiektu jest ostatni punkt pośredni $i_N(k)$ tego obiektu), $a_{iMk} = -1 \Leftrightarrow i = t_k$, (wierzchołkiem końcowym ostatniego odcinka drogi dla k -tego obiektu jest t_k), $\forall_{n \in \{1, \dots, N\}} a_{ink} = -1 \Rightarrow a_{i(n+1)k} = 1$ (tzn. wierzchołek końcowy n -tego odcinka drogi jest jednocześnie wierzchołkiem początkowym $(n+1)$ -szego odcinka drogi); $\mathbf{H} = [h_{ik}]_{V \times K}$ jest macierzą wierzchołków (generujących podgrafy grafu G), które mogą być brane pod uwagę przy wyznaczaniu drogi dla obiektów: $h_{ik} = 1$, jeśli i -ty wierzchołek może być brany pod uwagę przy wyznaczaniu drogi dla k -tego obiektu, $h_{ik} = 0$ w przeciwnym przypadku (w szczególności $i = s_k \Rightarrow h_{ik} = 1$, $i = t_k \Rightarrow h_{ik} = 1$); $\mathbf{X} = [x_{ijnk}]_{V \times V \times M \times K}$

– macierz zmiennych decyzyjnych dotyczących wyboru dróg, $x_{ijnk} = 1$, jeżeli łuk z i do j dla k -tego obiektu należy do n -tego odcinka drogi, $x_{ijnk} = 0$ w przeciwnym przypadku; $\mathbf{V} = [v_{ijk}]_{V \times V \times K}$ – macierz zmiennych decyzyjnych dotyczących prędkości na odcinkach dróg dla każdego obiektu. Sformułowanie problemu (PMI) jest następujące:

$$\sum_{i=1}^V \sum_{j=1}^V \sum_{k=1}^K \sum_{n=1}^M \frac{d_{ij}}{v_{ijk}} \cdot x_{ijnk} \rightarrow \min \quad (16)$$

$$\sum_{k=1}^K \sum_{n=1}^{M-1} \left(\begin{array}{l} \max_{l \in \{1, \dots, K\}} \left(\tau^0(l) + \sum_{p=1}^V \sum_{q=1}^V \frac{d_{pq}}{v_{pql}} \cdot x_{pqnl} \right) + \\ - \left(\tau^0(k) + \sum_{p=1}^V \sum_{q=1}^V \frac{d_{pq}}{v_{pqk}} \cdot x_{pqnk} \right) \end{array} \right) \rightarrow \min \quad (17)$$

przy ograniczeniach:

$$\sum_{j=1}^V x_{ijnk} - \sum_{j=1}^V x_{jink} = a_{ink}, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (18)$$

$$\sum_{j=1}^V x_{ijnk} \leq 1, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (19)$$

$$\sum_{j=1}^V x_{jink} \leq 1, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (20)$$

$$\sum_{j=1}^V x_{ijnk} \leq h_{ik}, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (21)$$

$$\sum_{n=1}^M \sum_{k=1}^K x_{ijnk} \leq 1, \quad i, j = \overline{1, V}, \quad (22)$$

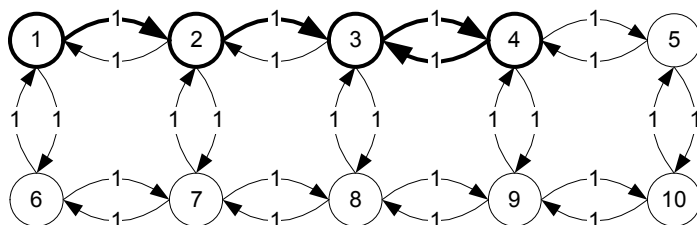
$$v_{ijk} \leq v^{\max}(k), \quad i, j = \overline{1, V}, \quad k = \overline{1, K}, \quad (23)$$

$$v_{ijnk} \geq 0, \quad v_{ijk} > 0, \quad i, j = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}. \quad (24)$$

Zwróćmy uwagę, że (9) możemy równoważnie zapisać następująco:

$$\tau_p(k) = \tau^0(k) \sum_{i=1}^V \sum_{j=1}^V \frac{d_{ij}}{v_{ijk}} \cdot x_{ijpk}, \quad \text{stad formuła C.2.1 przybiera postać (17). Z kolei formuła C.1.2 przybiera postać (16). Warunek (18) jest klasycznym warunkiem bilansu}$$

dla każdego wierzchołka, który musi być spełniony, aby ciąg zmiennych decyzyjnych x_{ijnk} o wartościach równych 1 reprezentował drogę dla k -tego obiektu na n -tym odcinku tej drogi (suma łuków „wychodzących” z każdego wierzchołka oraz „wchodzących” do niego, które zostały wybrane do drogi musi być taka sama (oprócz wierzchołka początkowego i końcowego dla każdego n -tego odcinka drogi)). Warunki (19)–(20) gwarantują, że dla każdego wierzchołka grafu, dla k -tego obiektu, nie więcej jak jeden łuk „wychodzący” z („wchodzący” do) tego wierzchołka będzie należał do jakiegokolwiek odcinka drogi. Zapobiega to sytuacji, którą można zobrazować jak w przykładzie na rysunku 1. Droga najkrótsza (przy braku ograniczeń (19)–(20)) z wierzchołka 1 do 3, która ma przechodzić przez wierzchołki pośrednie 2 i 4 (o postaci: 1-2-3-4-3, złożona z trzech odcinków: 1-2, 2-3-4, 4-3) jest nieprosta w tym przykładzie, czyli niedopuszczalna z praktycznego punktu widzenia (przechodzimy przez wierzchołek 3 po to tylko, aby osiągnąć pośredni wierzchołek 4, a następnie i tak wracamy do wierzchołka 3 jako końcowego). Drogą optymalną jest natomiast droga: 1-2-7-8-9-4-3 (składająca się z następujących trzech odcinków: 1-2, 2-7-8-9-4, 4-3), którą uzyskamy dokładając ograniczenia (19)–(20).



Rys. 1. Przykład drogi optymalnej z wierzchołka 1 do 3, która ma przechodzić przez wierzchołki 2 i 4: brak ograniczeń (19) i (20) może doprowadzić do uzyskania, jak na rysunku, drogi nieprostej (o powtarzających się wierzchołkach) postaci: 1-2-3-4-3

Warunek (21) gwarantuje, że n -ty odcinek drogi dla każdego k -tego obiektu będzie przechodził wyłącznie przez wierzchołki należące do dopuszczalnego podzbioru zbioru wierzchołków (generującego podgraf) dla tego obiektu. Warunek (22) zapewnia, że dla każdej pary (i, j) wierzchołków grafu (tworzącej łuk grafu G) para ta będzie należeć do drogi dla co najwyżej jednego obiektu, na co najwyżej jednym jego odcinku (spełnienie warunku rozłączności dróg). Warunek (23) dodaje ograniczenia na dopuszczalne prędkości na łuku (i, j) dla każdego obiektu. Zwróćmy uwagę, że $x_{ijnk} \geq 0$, a nie $x_{ijnk} \in \{0, 1\}$. W dowodzie twierdzenia 1 pokazano, że macierz współczynników ograniczeń dla pewnego zadania $PM2$, równoważnego zadaniu $PM1$, jest całkowicie unimodularna, a ponieważ prawe strony ograniczeń są całkowitoliczbowe, z tych dwóch własności wynika więc, że każde rozwiązanie bazowe będzie całkowitoliczbowe. Dodatkowo, charakter ograniczeń gwarantuje, że wartości zmiennych $x_{ijnk} \in \{0, 1\}$. Podobną tezę można udowodnić dla zadania $PM1$.

Zauważmy, że jeżeli $N = V - 2K$, tzn. zbiór wierzchołków pośrednich tworzą wszystkie wierzchołki grafu z pominięciem wierzchołków początkowych i końcowych (przy jednoczesnym założeniu, że wierzchołki początkowe i końcowe nie mogą się powtarzać), to drogi, które wyznaczamy są drogami Hamiltona, a problem staje się specyficznym problemem komiwojażera. Oczywiście wówczas warunek (22) o rozłączności dróg nie ma sensu, gdyż powoduje, że zbiór rozwiązań dopuszczalnych dla $K > 1$ będzie zawsze pusty.

W sformułowanym zadaniu występuje $V^2K(M + 1)$ zmiennych decyzyjnych oraz $4VMK + V^2(K + 1)$ ograniczeń (plus grupa ograniczeń (24), dotycząca nieujemności zmiennych decyzyjnych). Dodatkowym utrudnieniem jest fakt, że sformułowane zadanie jest zadaniem programowania ilorazowego (zmienna v_{ijk} występująca w mianowniku w definicji funkcji celu).

Zadanie (16)–(24) może zostać zapisane w równoważnej, prostszej do rozwiązania postaci. Niektóre z wcześniejszych oznaczeń będą miały inną interpretację, dodamy też nowe oznaczenia, reszta oznaczeń pozostaje bez zmian. I tak: $\mathbf{B} = [b_{ij}]_{V \times A}$ oznacza binarną macierz incydencji grafu G : $b_{ij} = 1$ jeśli j -ty łuk rozpoczyna się w i -tym wierzchołku, $b_{ij} = -1$ jeśli j -ty łuk kończy się w i -tym wierzchołku, $b_{ij} = 0$ w przeciwnym przypadku; przypomnijmy, że z własności macierzy incydencji grafu wynika, że w dowolnej kolumnie macierzy znajdują się albo same zera, albo dokładnie dwa elementy niezerowe, z których jeden jest równy $+1$, a drugi -1 ; $\mathbf{D} = [d_j]_{1 \times A}$ – wektor długości (terenowych) łuków grafu G ; $\mathbf{X} = [x_{jnk}]_{A \times M \times K}$ – macierz zmiennych decyzyjnych dotyczących wyboru dróg, $x_{jnk} = 1$ jeżeli j -ty łuk dla k -tego obiektu należy do n -tego odcinka drogi, $x_{jnk} = 0$ w przeciwnym przypadku; $\mathbf{V} = [v_{jk}]_{A \times K}$ – macierz zmiennych decyzyjnych dotyczących prędkości na odcinkach dróg (łukach) dla każdego obiektu. Zadanie (16)–(24) po modyfikacji ma postać (PM2):

$$\sum_{j=1}^A \sum_{k=1}^K \sum_{n=1}^M \frac{d_j}{v_{jnk}} \cdot x_{jnk} \rightarrow \min, \quad (25)$$

$$\sum_{k=1}^K \sum_{n=1}^{M-1} \left(\begin{array}{l} \max_{l \in \{1, \dots, K\}} \left(\tau^0(l) + \sum_{p=1}^A \frac{d_p}{v_{pl}} \cdot x_{pnl} \right) + \\ - \left(\tau^0(k) + \sum_{p=1}^A \frac{d_p}{v_{pk}} \cdot x_{pnk} \right) \end{array} \right) \rightarrow \min, \quad (26)$$

przy ograniczeniach:

$$\sum_{j=1}^A b_{ij} x_{jnk} = a_{ink}, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (27)$$

$$\sum_{j=1}^A b_{ij} x_{jnk} \leq 1, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (28)$$

$$\sum_{j=1}^A b_{ij} x_{jnk} \geq -1, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (29)$$

$$\sum_{j=1}^A b_{ij} x_{jnk} \leq h_{ik}, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (30)$$

$$\sum_{j=1}^A b_{ij} x_{jnk} \geq -1, \quad i = \overline{1, V}, \quad n = \overline{1, M}, \quad k = \overline{1, K}, \quad (31)$$

$$\sum_{n=1}^M \sum_{k=1}^K x_{jnk} \leq 1, \quad j = \overline{1, A}, \quad (32)$$

$$v_{jk} \leq v^{\max}(k), \quad j = \overline{1, A}, \quad k = \overline{1, K}, \quad (33)$$

$$x_{jnk} \geq 0, \quad v_{jk} > 0, \quad j = \overline{1, A}, \quad n = \overline{1, M}, \quad k = \overline{1, K}. \quad (34)$$

Odpowiednikiem warunku (18) jest warunek (27) (zapiszemy to: (18)→(27)). Pozostałe warunki możemy opisać następująco: (19)→(28), (20)→(29), (21)→((30)–(31)), (22)→(32), (23)→(33), (24)→(34).

Zauważmy, że sformułowane zadania *PM1* oraz *PM2* są zadaniami programowania ilorazowego (zmienna decyzyjna występująca w mianowniku funkcji celu). Podstawowa różnica między sformułowaniami zadań *PM1* oraz *PM2* jest związana z liczbą zmiennych decyzyjnych i ograniczeń: w pierwszym sformułowaniu mamy $V^2K(M+1)$ zmiennych oraz $4VMKV^2K(K+1)$ ograniczeń, a w drugim $AK(M+1)$ zmiennych oraz $5VMK + A(K+1)$ ograniczeń. Gdy G jest grafem rzadkim, tzn. $A \ll V^2$, wówczas zadanie *PM2* jest dużo łatwiejsze (w sensie złożoności obliczeniowej) do rozwiązania.

Zwróćmy uwagę, że nie ma potrzeby pisania warunków $x_{jnk} \in \{0, 1\}$ (które powodowałyby, że mielibyśmy do czynienia z binarnym zadaniem programowania ilorazowego), chociaż zmienne x_{jnk} będą przyjmowały wyłącznie wartości 0 i 1. Mówi o tym następujące twierdzenie.

Twierdzenie 1

Niech \mathbf{W} oznacza macierz współczynników ograniczeń w zadaniu *PM2* ze względu na zmienne x_{jnk} , \mathbf{Y} oznacza dowolną podmacierz kwadratową macierzy \mathbf{W} , $\det \mathbf{Y}$ oznacza wyznacznik macierzy \mathbf{Y} . Wówczas macierz \mathbf{W} jest całkowicie unimodularna, tzn. $\forall \mathbf{Y} \det \mathbf{Y} \in \{0, \pm 1\}$.

Dowód

Przedstawimy szkic dowodu indukcyjnego względem $k \in \{1, \dots, K\}$ oraz $n \in \{1, \dots, M\}$. Weźmy $K = 1$ i $M = 1$. W tym przypadku macierz \mathbf{W} (zgodnie z założeniami twierdzenia interesują nas wyłącznie ograniczenia (27)–(32)) będzie miała postać kolumnową, którą dla wygody zapiszemy w postaci transponowanej: $\mathbf{W}^T = (\mathbf{B}^1 \ \mathbf{B}^2 \ \mathbf{B}^3 \ \mathbf{B}^4 \ \mathbf{B}^5 \ \mathbf{J})$, gdzie $\mathbf{B}^i = \mathbf{B}$, $i = 1, \dots, 5$ są macierzami współczynników ograniczeń (27)–(31), \mathbf{J} jest macierzą współczynników ograniczeń dla ograniczenia (32) (\mathbf{J} jest macierzą jednostkową). Z definicji całkowitej unimodularności macierzy oraz z twierdzenia dowodzącego, że macierz \mathbf{B} incydencji dowolnego diagrafu jest całkowicie unimodularna (patrz np. [10], rozdz. 13.2) wynika, że każda z podmacierzy kwadratowych \mathbf{B}' macierzy \mathbf{B} jest całkowicie unimodularna, tzn. $\forall \mathbf{B}' \det \mathbf{B}' \in \{0, \pm 1\}$. Przyjmijmy na potrzeby ilustracji, że

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}.$$

Rozpatrzmy trzy przypadki.

Przypadek 1. \mathbf{Y} jest podmacierzą, którejś z macierzy \mathbf{B}^i w \mathbf{W} , tzn. $\mathbf{Y} = \mathbf{B}'$. Wówczas $\det \mathbf{Y} \in \{0, \pm 1\}$.

Przypadek 2. Podmacierz \mathbf{Y} jest zbudowana na bazie kilku podmacierzy (niekoniecznie kwadratowych) spośród macierzy \mathbf{B}^i , $i = 1, \dots, 5$, np.

$$\mathbf{Y} = \begin{pmatrix} \left(\begin{matrix} b_{21}^1 & b_{22}^1 & b_{23}^1 & b_{24}^1 \\ b_{31}^1 & b_{32}^1 & b_{33}^1 & b_{34}^1 \end{matrix} \right) \\ \left(\begin{matrix} b_{11}^2 & b_{12}^2 & b_{13}^2 & b_{14}^2 \\ b_{21}^2 & b_{22}^2 & b_{23}^2 & b_{24}^2 \end{matrix} \right) \end{pmatrix},$$

gdzie b_{jk}^i oznacza element w j -tym wierszu i k -tej kolumnie macierzy \mathbf{B}^i . Możemy mieć dwie sytuacje:

1°. w macierzy \mathbf{Y} powtarzają się podwiersze macierzy \mathbf{B} o tych samych numerach (np. dla \mathbf{Y} jak wyżej, wiersz 1 i wiersz 4, macierz \mathbf{B} zawiera 3 wiersze) i wówczas z własności wyznacznika wynika, że $\det \mathbf{Y} = 0$ lub

2°. w macierzy \mathbf{Y} nie powtarzają się podwiersze macierzy \mathbf{B} o tych samych numerach, np.

$$\mathbf{Y} = \begin{pmatrix} \left(\begin{matrix} b_{21}^1 & b_{22}^1 & b_{23}^1 \\ b_{31}^1 & b_{32}^1 & b_{33}^1 \end{matrix} \right) \\ \left(\begin{matrix} b_{11}^2 & b_{12}^2 & b_{13}^2 \end{matrix} \right) \end{pmatrix}.$$

Jeżeli liczba wierszy macierzy \mathbf{Y} będzie w tej sytuacji równa liczbie wierszy macierzy \mathbf{B} , to \mathbf{Y} będzie podmacierzą macierzy \mathbf{B} i mamy Przypadek 1 (z dokładnością do numeracji wierszy, ale pamiętamy, że przy zamianie miejscami wierszy macierzy wartość wyznacznika jedynie zmieni się na przeciwną (z +1 na -1 lub odwrotnie, lub będzie równa 0)).

Jeżeli liczba wierszy macierzy \mathbf{Y} będzie różna od liczby wierszy macierzy \mathbf{B} , np.

$$\mathbf{Y} = \begin{pmatrix} (b_{32}^1 & b_{33}^1) \\ (b_{12}^2 & b_{13}^2) \end{pmatrix},$$

to zawsze możemy tak poprzestawiać wiersze macierzy \mathbf{B} , aby w macierzy \mathbf{Y} tworzyły podmacierz macierzy \mathbf{B} i doprowadzamy do przypadku 1, na przykład zamieniamy miejscami w macierzy \mathbf{B} wiersz 1 z 2 i wówczas

$$\mathbf{Y} = \begin{pmatrix} (b_{32}^1 & b_{33}^1) \\ (b_{22}^2 & b_{23}^2) \end{pmatrix}.$$

Jeżeli teraz w macierzy \mathbf{Y} zamienimy miejscami te dwa wiersze, otrzymamy podmacierz macierzy \mathbf{B} .

Przypadek 3. Podmacierz \mathbf{Y} jest zbudowana na bazie kilku podmacierzy (niekoniecznie kwadratowych) spośród macierzy \mathbf{B}^i , $i = 1, \dots, 5$ oraz na bazie podmacierzy macierzy \mathbf{J} . Pominiemy sytuację, gdy w macierzy \mathbf{Y} powtarzają się podwiersze macierzy \mathbf{B} o tych samych numerach, bo wówczas, jak to pokazaliśmy wcześniej, $\det \mathbf{Y} = 0$ (przypadek 2). Rozpatrujemy zatem sytuację, gdy macierz \mathbf{Y} zbudowana jest z podwierszy macierzy \mathbf{B}^5 i (lub) \mathbf{J} , np.

$$\mathbf{Y} = \begin{pmatrix} b_{21}^5 & b_{22}^5 & b_{23}^5 & b_{24}^5 \\ b_{31}^5 & b_{32}^5 & b_{33}^5 & b_{34}^5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{lub} \quad \mathbf{Y} = \begin{pmatrix} b_{31}^5 & b_{32}^5 & b_{33}^5 & b_{34}^5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Macierz \mathbf{Y} można więc zapisać następująco:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{B}' & \vdots & \mathbf{B}'' \\ \cdots & \ddots & \cdots \\ \mathbf{J}' & \vdots & \mathbf{J}'' \end{pmatrix},$$

gdzie \mathbf{J}' jest podmacierzą jednostkową macierzy \mathbf{J} , \mathbf{J}'' jest macierzą zerową, a \mathbf{B}' i \mathbf{B}'' są podmacierzami macierzy \mathbf{B}^5 . Oznaczmy przez $w \leq \min\{V, A\}$ liczbę wierszy (kolumn) macierzy \mathbf{Y} oraz przez u liczbę wierszy macierzy \mathbf{B}^5 , z których podwiersze „wchodzą” do \mathbf{Y} . Wówczas wymiary poszczególnych macierzy są następujące:

$\mathbf{J}' - (w-u) \times (w-u)$, $\mathbf{J}'' - (w-u) \times u$, $\mathbf{B}' - u \times (w-u)$, $\mathbf{B}'' - u \times u$. Ponieważ \mathbf{J}' jest macierzą jednostkową, więc $\det \mathbf{J} = 1$, \mathbf{J}'' jest macierzą zerową, $\det(\mathbf{B}' \cdot \mathbf{J}'') = 0$, a z wcześniejszych rozważań wynika, że $\det \mathbf{B}'' \in \{0, \pm 1\}$, czyli $\det(\mathbf{B}'' \cdot \mathbf{J}') \in \{0, \pm 1\}$. W związku z powyższym $\det \mathbf{Y} \in \{0, \pm 1\}$.

Wykazaliśmy zatem, że dla przypadku, gdy $K = 1$ i $M = 1$ macierz współczynników ograniczeń w zadaniu *PM2* ze względu na zmienne x_{jnk} jest całkowicie unimodularna.

Rozpatrzmy obecnie przypadek, gdy $K = 1$ i $M = 2$ lub $K = 2$ i $M = 1$. Zwróćmy uwagę, że dla tego przypadku macierz \mathbf{W} (transponowana dla wygody zapisu) będzie miała postać:

$$\mathbf{W}^T = \begin{pmatrix} \mathbf{B}^1 & \mathbf{B}^2 & \mathbf{B}^3 & \mathbf{B}^4 & \mathbf{B}^5 & \mathbf{J} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}^1 & \mathbf{B}^2 & \mathbf{B}^3 & \mathbf{B}^4 & \mathbf{B}^5 & \mathbf{J} \end{pmatrix}.$$

Można ją zatem zapisać w postaci macierzy przekątniowej następująco:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & \vdots & \mathbf{0} \\ \cdots & \ddots & \cdots \\ \mathbf{0} & \vdots & \mathbf{W}_{11} \end{pmatrix},$$

gdzie $\mathbf{W}_{11}^T = (\mathbf{B}^1 \mathbf{B}^2 \mathbf{B}^3 \mathbf{B}^4 \mathbf{B}^5 \mathbf{J})$ oznacza transponowaną macierz \mathbf{W} dla $M = 1$, $K = 1$. Postać macierzy \mathbf{W} możemy uogólnić dla dowolnego k i m : jeżeli przez \mathbf{W}_{mk} oznaczymy macierz \mathbf{W} dla pewnego $m \in \{1, \dots, M\}$ i pewnego $k \in \{1, \dots, K\}$, to możemy zapisać, że

$$\mathbf{W}_{m,k+1} = \mathbf{W}_{m+1,k} = \begin{pmatrix} \mathbf{W}_{mk} & \vdots & \mathbf{0} \\ \cdots & \ddots & \cdots \\ \mathbf{0} & \vdots & \mathbf{W}_{mk} \end{pmatrix}.$$

Teraz już łatwo pokazać przez indukcję względem k i m , że macierz \mathbf{W} jest całkowicie unimodularna. Załóżmy, że $\mathbf{W}_{m,k}$ jest całkowicie unimodularna (np. wykazaliśmy, że jest całkowicie unimodularna dla $k = 1$, $m = 1$). Wykażemy, że macierz $\mathbf{W}_{m,k+1} = \mathbf{W}_{m+1,k}$ jest również całkowicie unimodularna. Oznaczmy, tak jak poprzednio, przez \mathbf{Y} dowolną podmacierz kwadratową macierzy $\mathbf{W}_{m,k+1} = \mathbf{W}_{m+1,k}$. Możliwe są cztery przypadki:

Przypadek 1. Macierz \mathbf{Y} jest podmacierzą macierzy $\mathbf{W}_{m,k}$. Wówczas z założenia indukcyjnego wynika, że $\det \mathbf{Y} \in \{0, \pm 1\}$.

Przypadek 2. Macierz \mathbf{Y} jest podmacierzą macierzy $\mathbf{0}$. Wówczas $\det \mathbf{Y} = 0$.

Przypadek 3. Macierz \mathbf{Y} jest podmacierzą macierzy $\mathbf{W}_{m,k}$ i $\mathbf{0}$. Wówczas w \mathbf{Y} występują albo zerowy wiersz, albo zerowa kolumna czyli $\det \mathbf{Y} = 0$.

Przypadek 4. Macierz \mathbf{Y} jest podmacierzą obejmującą elementy wszystkich czterech podmacierzy w $\mathbf{W}_{m,k+1} = \mathbf{W}_{m+1,k}$. Wówczas \mathbf{Y} możemy zapisać następująco:

$$\mathbf{Y} = \begin{pmatrix} \hat{\mathbf{W}}_{mk} & \vdots & \hat{\mathbf{0}} \\ \cdots & \ddots & \cdots \\ \hat{\mathbf{0}} & \vdots & \hat{\mathbf{W}}_{mk} \end{pmatrix},$$

gdzie $\hat{\mathbf{W}}_{mk}$ i $\hat{\mathbf{0}}$ są pewnymi podmacierzami odpowiednio macierzy $\mathbf{W}_{m,k}$ i $\mathbf{0}$. Ponieważ $\det \hat{\mathbf{W}}_{mk} \in \{0, \pm 1\}$ oraz $\det \hat{\mathbf{0}} = 0$, więc $\det \mathbf{Y} \in \{0, \pm 1\}$.

Wykazaliśmy zatem, że macierz $\mathbf{W}_{m,k+1} = \mathbf{W}_{m+1,k}$ jest całkowicie unimodularna dla dowolnego $m \in \{1, \dots, M\}$ i $k \in \{1, \dots, K\}$. \diamond

Wniosek z twierdzenia 1 jest oczywisty: ponieważ macierz współczynników ograniczeń (27)–(32) jest całkowicie unimodularna oraz prawe strony ograniczeń są całkowitoliczbowe, z tych dwóch własności wynika więc, że każde rozwiązanie bazowe zadania *PM2* będzie całkowitoliczbowe, więcej – charakter ograniczeń gwarantuje, że wartości zmiennych $x_{jmk} \in \{0, 1\}$. W podobny sposób można udowodnić całkowitą unimodularność macierzy współczynników ograniczeń zadania *PM1*. Wykazana w ten sposób własność zadania *PM2* przyda się również przy definiowaniu rozszerzeń omawianego problemu.

4. Rozszerzenia problemu harmonogramowania przemieszczania

Problem (12)–(14) możemy rozszerzyć następująco:

- dodając ograniczenie

$$\tau^0(k) + \sum_{r \in \{0, \dots, R_k - 1\}} \frac{d_{i^r(k), i^{r+1}(k)}}{v_{i^r(k), i^{r+1}(k)}} \leq T^{\max}, \quad k = \overline{1, K}, \quad (35)$$

poszukujemy takiego harmonogramu przemieszczania, aby moment osiągnięcia wierzchołka docelowego przez najwolniejszy obiekt był nie większy niż pewna ustalona chwila $T^{\max} \geq \tau^*$;

- możemy osłabić warunek wynikający z funkcji celu (12) przez zastąpienie tej funkcji dodatkową grupą ograniczeń (nieliniowych)

$$|\tau_p^{\max} - \tau_p(k)| \leq \tau^{\text{delay}}, \quad k = \overline{1, K}, \quad p = \overline{1, P_k}, \quad (36)$$

która gwarantuje, że wielkość opóźnienia (przyspieszenia) w każdym wierzchołku wyrównania dla każdego obiektu w stosunku do obiektu najwolniejszego będzie nie większa niż pewna ustalona wartość τ^{delay} ;

- możemy wprowadzić zmodyfikowany warunek (14) jako $v_{i^r(k), i^{r+1}(k)} \geq v^{\min}(k)$, gdzie $v^{\min}(k)$ jest minimalną dopuszczalną prędkością k -tego obiektu;

• definiujemy zadanie optymalizacji wielokryterialnej, używając podzbioru kryteriów z grupy C.1 oraz C.2 (tak jak to pokazano w rozdz. 3.2) i poszukując niezdominowanych wektorów dróg I (używając relacji dominowania w zbiorze wartości kryteriów, takiej jak np. relacja Pareto [3], [15], [16]).

Jednym z rozszerzeń omawianego problemu może być problem planowania przemieszczania wielu obiektów zgodnie z pewnym wzorcem ugrupowania.

Wzorcem ugrupowania (j -tego) K obiektów ponumerowanych od 0 do $K-1$ będziemy nazywać następujący wektor $2K$ -wymiarowy:

$$(x_0, y_0, \Delta x_1^j, \Delta y_1^j, \dots, \Delta x_{K-1}^j, \Delta y_{K-1}^j), \quad (37)$$

gdzie x_0, y_0 oznaczają współrzędne (w prostokątnym układzie współrzędnych) obiektu-prowadzącego (może to być np. pojazd dowódcy ugrupowania), względem którego ustalane jest położenie pozostałych obiektów we wzorcu ugrupowania, natomiast pary liczb $(\Delta x_i^j, \Delta y_i^j)$, $i = \overline{1, K-1}$ umożliwiają określenie współrzędnych i -tego obiektu w j -tym wzorcu ugrupowania w sposób następujący:

$$(x_i^j, y_i^j) = (x_0 + \Delta x_i^j, y_0 + \Delta y_i^j), \quad (38)$$

przy czym zakłada się, że istnieje pewien przedział tolerancji δ^j dla wartości $\Delta x_i^j, \Delta y_i^j$, $i = \overline{1, K-1}$, w związku z czym współrzędne obiektu i -tego we wzorcu j -tym mogą być zdefiniowane następująco:

$$(x_i^j, y_i^j) = (x_0 + \Delta x_i^j \pm \delta^j, y_0 + \Delta y_i^j \pm \delta^j). \quad (39)$$

Jeśli współrzędne każdego obiektu spełniają warunek (39), to uważa się, że j -ty wzorec ugrupowania jest zachowany. Należy dodać, że wzorec ugrupowania (37) jest definiowany przy założeniu, że kąt α , jaki tworzy wektor kierunku ruchu ugrupowania z osią Oy bazowego układu współrzędnych, jest równy 0° . Wobec tego współrzędne (38)–(39) też są wyliczane przy takim założeniu. Przykłady typowych wzorców ugrupowania przedstawiono na rysunku 2. Założono, że $\alpha = 0^\circ$, tzn. kierunek ruchu ugrupowania pokrywa się z osią Oy bazowego układu współrzędnych. Zwróćmy uwagę, że prawy dolny wzorec ugrupowania może odpowiadać sytuacji (opisywanej w rozdz. 3) równoczesnego dotarcia do punktów wyrównania wszystkich obiektów.

W ogólnym przypadku, jeśli kąt $\alpha \neq 0$, czyli aktualny układ współrzędnych OXY jest obrócony względem układu bazowego Oxy o kąt α zgodnie z rysunkiem 3, to współrzędne (38) w nowym układzie współrzędnych są wyliczane następująco (patrz Rys. 3):

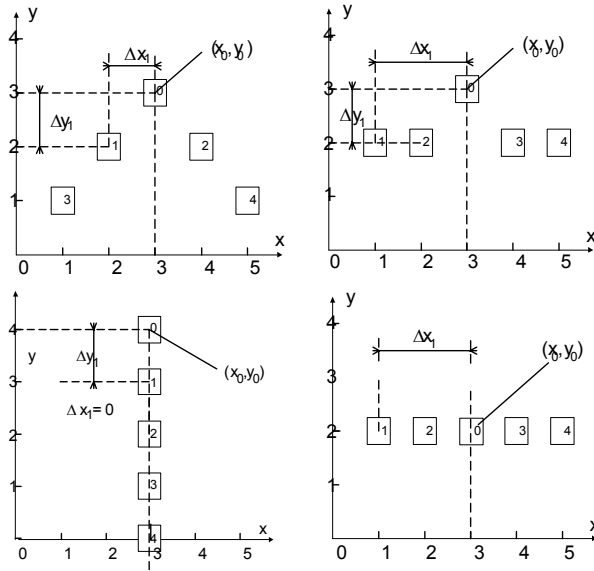
$$(X_i^j, Y_i^j) = (x_i^j \cdot \cos \alpha + y_i^j \cdot \sin \alpha, -x_i^j \cdot \sin \alpha + y_i^j \cdot \cos \alpha). \quad (40)$$

Mając dane współrzędne (X_i^j, Y_i^j) w układzie obróconym o kąt α , współrzędne w układzie bazowym możemy wyznaczyć następująco:

$$(x_i^j, y_i^j) = (X_i^j \cdot \cos \alpha - Y_i^j \cdot \sin \alpha, X_i^j \cdot \sin \alpha + Y_i^j \cdot \cos \alpha). \quad (41)$$

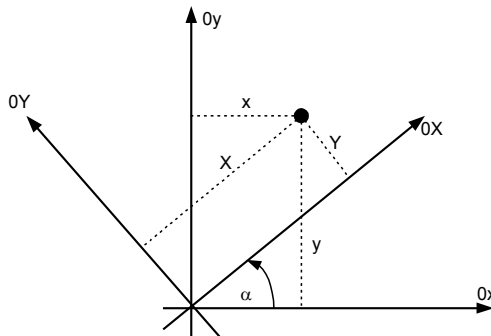
Kąt α będziemy nazywać *kierunkiem ruchu ugrupowania*. Niech w chwili t aktualne położenie ugrupowania będzie zdefiniowane następująco:

$$(X_0(t), Y_0(t), X_1(t), Y_1(t), \dots, X_{K-1}(t), Y_{K-1}(t), \alpha). \quad (42)$$



Rys. 2. Przykłady definicji wzorców ugrupowania dla grupy przemieszczanych obiektów

Zauważmy, że współrzędne (42) są wyznaczone w układzie współrzędnych obróconym o kąt α . Aby uzyskać współrzędne $(x_i(t), y_i(t))$, $i = \overline{0, K-1}$ w układzie bazowym, należy skorzystać z zależności (41).



Rys. 3. Układ współrzędnych bazowy Oxy i obrócony o kąt $\alpha - OXY$

Jeżeli położenie obiektu prowadzącego w chwili t oznaczymy przez $(x_0(t), y_0(t))$, to aktualne, wzorcowe położenie rozpatrywanych K obiektów ugrupowanych według wzorca o numerze j w układzie bazowym współrzędnych będzie zdefiniowane następująco:

$$(x_0(t), y_0(t), x_1^j(t), y_1^j(t), \dots, x_{K-1}^j(t), y_{K-1}^j(t)), \quad (43)$$

gdzie:

$$x_i^j(t) = x_0(t) + \Delta x_i^j, \quad (44)$$

$$y_i^j(t) = y_0(t) + \Delta y_i^j \quad (45)$$

określają współrzędne i -tego obiektu w ugrupowaniu według j -tego wzorca w chwili t w bazowym układzie współrzędnych.

„Odległością” $d^j(t)$ aktualnego ugrupowania od j -tego wzorca ugrupowania w chwili t będziemy nazywali następującą liczbę:

$$d^j(t) = \sum_{i=1}^{K-1} \sqrt{s_{x_i}^2(t) + s_{y_i}^2(t)}, \quad (46)$$

gdzie:

$$s_{x_i}(t) = \begin{cases} x_i(t) - x_i^j(t), & \text{gdy } x_i(t) \notin [x_i^j(t) - \delta^j, x_i^j(t) + \delta^j], \\ 0, & \text{w przeciwnym przypadku,} \end{cases} \quad (47)$$

$$s_{y_i}(t) = \begin{cases} y_i(t) - y_i^j(t), & \text{gdy } y_i(t) \notin [y_i^j(t) - \delta^j, y_i^j(t) + \delta^j], \\ 0, & \text{w przeciwnym przypadku.} \end{cases} \quad (48)$$

Naszym celem może być takie zaplanowanie ruchu poszczególnych obiektów ugrupowania, aby minimalizować odległość $d^j(t)$ ugrupowania od j -tego, założonego wzorca w każdej chwili t (lub dla ciągu ustalonych chwil, w których kontrolujemy stan ugrupowania) przy zapewnieniu realizacji pozostałych celów.

Zwróćmy uwagę, że „równoległość” przemieszczania zgodnie ze wzorcem ugrupowania ma wymiar geometryczny (zdeterminowany przez położenie obiektów), podczas gdy w rozdziale 3 definiowaliśmy ten problem w wymiarze czasowym (dotarcie do pewnych punktów w tych samych chwilach). Związek między tymi dwoma sposobami „zrównoleglania” przemieszczania autor szczegółowo opisał w pracach (12) i (13).

5. Metody rozwiązywania sformułowanych problemów harmonogramowania

W rozdziale 3.1 zasugerowano, że jednym ze sposobów rozwiązania sformułowanych w rozdziale 3.2 zadań dwukryterialnych (*PM1* i *PM2*) może być metoda poszukiwania rozwiązań leksykograficznych. Dlatego też przedstawione w rozdziale 3.2 wnioski z twierdzenia 1 są istotne z punktu widzenia poszukiwania tego rozwiązania, gdyż rozwiązanie obu zadań może przebiegać dwuetapowo: najpierw będziemy poszukiwać dróg najkrótszych dla K obiektów w zbiorze dróg dopuszczalnych, przy założeniu, że obiekty poruszają się z maksymalnymi dopuszczalnymi prędkościami, tzn. rozwiążemy zadanie programowania liniowego z funkcją celu (16) dla zadania *PM1* lub (25) dla zadania *PM2*, z ograniczeniami (18)–(22) oraz $x_{ijnk} \geq 0$ dla *PM1* lub (27)–(32) oraz $x_{jn} \geq 0$ dla *PM2*, a następnie – mając wyznaczone w poprzednim etapie drogi najkrótsze dla K obiektów – rozwiążemy zadanie z funkcją celu (17) (dla *PM2*: (26)) z ograniczeniami (23) oraz $v_{ijk} > 0$ (dla *PM2*: z ograniczeniami (33) oraz $v_{jk} > 0$). Otrzymanie rozwiązania leksykograficznego tym sposobem nie jest zagadnieniem trywialnym i bardziej szczegółowo autor omówił ten problem w pracy [14]. Wykazanie natomiast, że macierz współczynników ograniczeń zadań *PM1* i *PM2* jest całkowicie unimodularna chroni nas przed koniecznością rozwiązywania zadań programowania binarnego (w ogólnym przypadku dużo bardziej złożonych obliczeniowo niż zadania PL).

W pracy [15] opisano szereg metod rozwiązywania wielokryterialnych problemów dróg najkrótszych, które możemy zastosować do rozwiązania naszego problemu dwukryterialnego (*PM1* lub *PM2*). Jednym ze sposobów praktycznego podejścia do rozwiązania problemu może być sposób polegający na zrezygnowaniu z funkcji celu (12) na rzecz wprowadzenia dodatkowej grupy ograniczeń (nieliniowych), reprezentowanych przez (1). Problem w ten sposób formułowany znany jest w literaturze jako problem RSPP (ang. *Restricted Shortest Path Problem*) i polega na tym, że niektóre z funkcji kryteriów w sformułowaniu wielokryterialnym przechodzą do zbioru ograniczeń oraz ogranicza się wartości tych funkcji od dołu lub od góry wartościami parametrów.

Szczegółowo o algorytmach rozwiązywania sformułowanych problemów harmonogramowania traktuje praca [14].

Wnioski

W artykule przedstawiono modele harmonogramowania zsynchronizowanego przemieszczania wielu (K) obiektów, wykorzystywane w wielu zagadnieniach: routingu w sieciach komputerowych, planowania przemieszczania mobilnych robo-

tów, przetwarzania zadań w systemach równoległych (rozproszonych), sterowania ramionami wielu niezależnych robotów, planowania i synchronizacji przemieszczania wielu obiektów w symulacyjnych grach komputerowych (systemy typu CGF i SAF). Zdefiniowano dwie podstawowe kategorie miar oceny jakości harmonogramu: (1) ze względu na szybkość dotarcia do celu K obiektów, (b) ze względu na „równoległość” przemieszczania się K obiektów. Dodatkowo opisane w rozdziale 4 rozszerzenia problemu podstawowego umożliwiają elastyczne dopasowanie się do różnych praktycznych wymagań w tego typu zagadnieniach.

Przedstawione modele harmonogramowania wpisują się w etap planowania działań (ruchu obiektów) [6], [9]. Na etapie realizacji przemieszczania (zwłaszcza w środowiskach podlegających dynamicznym zmianom) istnieje potrzeba wykorzystywania metod sterowania ruchem i reagowania na odstępstwa od założonego harmonogramu [12], [13]. Problemy te są istotne m.in. w systemach typu CGF [8] oraz SAF [7], w których sterowanie ruchem (położeniem) wielu obiektów w czasie symulacji przemieszczania jest bardzo ważne z punktu widzenia realizacji celu działań.

W pracy wspomniano jedynie o sposobach rozwiązania sformułowanych problemów. Wykazanie, że macierz współczynników ograniczeń zadań $PM1$ i $PM2$ jest całkowicie unimodularna uchroniło nas przed koniecznością rozwiązywania zadań programowania binarnego (w ogólnym przypadku dużo bardziej złożonych obliczeniowo niż zadania PL). Szczegółowa prezentacja algorytmów służących rozwiązaniu opisywanych problemów została przedstawiona w innej pracy autora [14].

Bibliografia

- [1] BLAZEWICZ J., ECKER K.H., PESCH E., SCHMIDT G., WEGLARZ J., *Scheduling Computer and Manufacturing Processes*, Springer-Verlag, Heidelberg-Berlin-New York 2001.
- [2] COFFMAN E. Jr. (red.), *Teoria szeregowania zadań*, WNT, Warszawa 1980.
- [3] EHRGOTT M., GANDIBLEUX X. (eds.), *Multiple Criteria Optimization – state of the art annotated bibliographic surveys*, Kluwer Academic Publishers, Boston 2002.
- [4] EPPSTEIN D., *Finding the K shortest Paths*, SIAM J. Computing, 28(2), 1999, s. 652–673.
- [5] IBARAKI T., *Algorithms for obtaining shortest paths visiting specified nodes*, SIAM Review, Vol. 15, No. 2, Part 1 (Apr. 1973), s. 309–317.
- [6] LOGAN B., *Route planning with ordered constraints*, Proceedings of the 16th Workshop of the UK Planning and Scheduling Special Interest Group, December, Durham (UK, 1997).
- [7] LONGTIN M., MEGHERBI D., *Concealed routes in ModSAF*, in *Proceedings of the 5th Conference on Computer Generated Forces and Behavioural Representation*, Technical Report, Institute for Simulation and Training, 1995, s. 305–314.
- [8] PETTY M.D., *Computer generated forces in Distributed Interactive Simulation*, Proceedings of the Conference on Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment, 19-20 April, Orlando (USA), 1995, s. 251–280.
- [9] RAJPUT S., KARR C., *Unit Route Planning*, Technical Report IST-TR-94-42, Institute for Simulation and Training, Orlando (USA, 1994).

- [10] SCHRIJVER A., *Combinatorial Optimization. Polyhedra and Efficiency*, Springer-Verlag, Berlin–New York 2004.
- [11] SCHRIJVER A., SEYMOUR P., *Disjoint paths in a planar graph – a general theorem*, SIAM Journal of Discrete Mathematics, 5, 1992, s. 112–116.
- [12] TARAPATA Z., *Modelling, optimisation and simulation of groups movement according to group pattern in multiresolution terrain-based grid network*, Proceedings of the 3rd NATO Regional Conference on Military Communication and Information Systems, 10–12 October, Zegrze (Poland) 2001, Vol. I, s. 241–251.
- [13] TARAPATA Z., *Synchronization method of many objects movement in Computer Generated Forces systems*, Proceedings of the 7th NATO Regional Conference on Military Communication and Information Systems, 04–05 October, Zegrze (Poland) 2005, s. 93–99.
- [14] TARAPATA Z., *Algorytmy harmonogramowania zsynchronizowanego przemieszczania wielu obiektów*, Badania Operacyjne i Decyzje, 2007 (złożony do druku).
- [15] TARAPATA Z., *Selected multicriteria shortest path problems: an analysis of complexity, models and adaptation of standard algorithms*, International Journal of Applied Mathematics and Computer Science, 2007, Vol. 17, No. 2, s. 269–287.
- [16] WARBURTON A., *Approximation of Pareto optima in multiple-objective, shortest-path problems*, Operations Research, 35, 1987, s. 70–79.

Models of scheduling synchronized movement of many objects

The paper deals with the problem of determining movement schedule of many objects, used in many domains such as: routing in computer networks, movement planning of mobile robots, tasks processing in parallel or distributed computing systems, arms control of independent robots, planning and synchronization of the movement of many objects in computer simulation games (e.g., in Computer Generated Forces (CGF) systems or Semi-Automated Forces (SAF) systems). A lot of movement scheduling models are discussed. Two groups of criteria which are essential from the point of view of schedule estimation are described: a group connected with movement time of all objects and a group connected with “parallelization” of their movement (in the sense of location and times of reaching specified checkpoints). A nonlinear movement scheduling problem in order to minimize the sum of delays of all objects at checkpoints with some additional constraints is defined. Two equivalent formulations of two-criteria mathematical programming problems are also presented. It is proved that constraint coefficient matrices for both problems are totally unimodular and we can use effective algorithms for solving linear programming problems to find lexicographic solution of two-criteria problems. Similarities and differences between the defined problem and classical tasks scheduling problem before critical lines on parallel processors are discussed. Some extensions of the problem are presented, one of which is the scheduling movement problem of many objects according to a group pattern. Methods of solving formulated problems are indicated.

Keywords: *movement scheduling and synchronization, shortest paths, disjoint paths, multicriteria shortest paths problems*