Wojciech BOŻEJKO, Andrzej GNATOWSKI, Radosław IDZIKOWSKI
Politechnika Wrocławska

Mieczysław WODECKI
Uniwersytet Wrocławski

ALGORYTM WIELOMIANOWY DLA CYKLICZNEGO PROBLEMU PRZYDZIAŁU OPERACJI W DWUMASZYNOWYM GNIEŹDZIE

Streszczenie. W artykule rozpatrujemy pewien wariant produkcji cyklicznej z przezbrojeniami maszyn wchodzących w skład dwumaszynowego gniazda. Jednym z etapów procesu optymalizacji czasu cyklu jest rozwiązywanie problemu przydziału każdej z operacji do jednej z maszyn gniazda, na której będzie ona wykonywana. Liczba wszystkich takich przydziałów jest wykładnicza. Przedstawiamy algorytm o wielomianowej złożoności obliczeniowej wyznaczający optymalny przydział operacji do maszyn.

POLYNOMIAL ALGORITHM FOR CYCLIC ASSIGNMENT PROBLEM IN TWO-MACHINE NEST

Summary. In the paper a variant of cyclic production with setups and two-machine nest is considered. One of the stages of the cycle time optimization process is solving the problem of assigning each operation to the machine on which it will be carried out. The total number of such assignments is exponential. We propose a polynomial time algorithm finding the optimal operations to machines assignment.

1. Wstęp

Elementami wielu systemów wytwarzania są gniazda wyposażone w maszyny tego samego typu, lecz o różnych wydajnościach. Obrabiane elementy, w ustalonej kolejności, przechodzą przez gniazdo i są wykonywane na jednej z tych maszyn. W cyklicznych problemach szeregowania, pewien ustalony zbiór zadań (w skrócie MPS, ang. *Minimal Part Set*) jest wykonywany wielokrotnie, tj. każda operacja zadania jest wykonywana na tej samej maszynie cyklicznie, co pewien okres czasu zwany czasem cyklu (obszerne wprowadzenie w problematykę szeregowania cyklicznego zawiera praca Kampmeyera [4]). Na długość czasu cyklu, która jest zazwyczaj minimalizowana, mają bezpośredni wpływ czasy wykonywania przez poszczególne maszyny przydzielonych do nich operacji. Jeżeli w rozpatrywanym problemie uwzględniamy dodatkowo przezbrojenia maszyn pomiędzy kolejno wykonywanymi operacjami, wówczas wyznaczenie minimalnego czasu pracy maszyny (permutacji operacji) jest problemem silnie NP-trudnym, bowiem sprowadza się do rozwiązania problemu komiwojażera (Bożej-

ko, Uchroński, Wodecki [1]). W przypadku, gdy operacja może być wykonywana na jednej z wielu maszyn tworzących gniazdo, problem wyznaczenia czasu wykonywania wszystkich operacji jest znacznie bardziej złożony. Należy bowiem dokonać: (a) przydziału operacji do maszyn, a następnie (b) ustalić kolejność wykonywania operacji przez każdą z maszyn tak, aby zminimalizować czas wykonywania wszystkich operacji (Sawik [6], Bożejko, Wodecki [2]). Przegląd cyklicznych problemów szeregowania pod kątem złożoności rozwiązujących je algorytmów oraz różnorodnych dodatkowych znaleźć można w pracy Levner i in. [5].

W pracy rozważamy problem minimalizacji czasu wykonywania operacji w dwumaszynowym gnieździe będącym elementem cyklicznego elastycznego systemu wytwarzania (dokładny opis przedstawiono w pracy Bożejko, Wodecki [2]). Czas ten ma bezpośredni wpływ na wielkość czasu cyklu. W tym przypadku wszystkich możliwych przydziałów jest 2^o, gdzie o jest liczbą operacji. Przedstawiamy algorytm o wielomianowej złożoności obliczeniowej wyznaczania minimalnego czasu wykonywania operacji w dwumaszynowym gnieździe. Może on być także stosowany do obliczania ograniczeń optymalnego czasu cyklu dla problemów z większą liczbą maszyn w gnieździe.

2. Sformułowanie problemu

Rozpatrywany problem polega na znalezieniu takiego przyporządkowania operacji do jednej z dwóch maszyn, by czas wykonywania w cyklicznej produkcji pojedynczego MPSa był najkrótszy. Bardziej formalnie, problem można sformułować w następujący sposób: dany jest zbiór operacji $\mathcal{O} = \{1, 2, \ldots, o\}$, które należy wykonywać na maszynach ze zbioru $\mathcal{M} = \{1, 2\}$ tworzących gniazdo. Kolejność wykonywania operacji przedstawić można za pomocą permutacji $\pi = (\pi(1), \pi(2), \ldots, \pi(o))$. Dla uproszczenia, bez utraty ogólności przyjęto $\pi = (1, 2, \ldots, o)$. Każdą operację $i \in \mathcal{O}$ należy wykonywać nieprzerwanie na maszynie $l \in \mathcal{M}$ przez p_i^l czasu, przy czym maszyna może wykonywać jednocześnie co najwyżej jedną operację. Przydział operacji do maszyn $P = (\mathcal{Z}_1, \mathcal{Z}_2)$, określony jest przez dwa rozłączne zbiory $\mathcal{Z}_1, \mathcal{Z}_2$ operacji, wykonywanych odpowiednio na maszynach 1 i 2, $\mathcal{Z}_1 \cup \mathcal{Z}_2 = \mathcal{O}$. Zbiór wszystkich możliwych przydziałów

$$\mathcal{P} = \bigcup_{I \in \mathcal{P}(\mathcal{O})} \{(\mathcal{Z}_1 = I,\ \mathcal{Z}_2 = \mathcal{O} \setminus \{I\})\}, \tag{1}$$

gdzie $\mathcal{P}(\mathcal{O})$ jest zbiorem potęgowym o mocy $|\mathcal{P}| = 2^o$. Dla ustalonego przydziału P, przez krotkę

$$\pi_P^l = (\pi_P^l(1), \pi_P^l(2), \ldots, \pi_P^l(|\pi_P^l|)),\ l \in \mathcal{M} \tag{2}$$

oznaczono kolejność wykonywania operacji na maszynie l, a przez $v_P(i)$ maszynę na której wykonywana jest operacja i. W pojedynczym MPSie, pomiędzy wykonywaniem każdych dwóch kolejnych operacji na maszynie $l \in \mathcal{M}$, należy wykonać przezbrojenie. Czas przezbrojenia

$$s^l_{\pi_P^l(i),\pi_P^l(i+1)},\ l \in \mathcal{M},\ i \in \{1, 2, \ldots, |\pi_P^l| - 1\}. \tag{3}$$

Dodatkowo, ze względu na cykliczny charakter problemu, przed rozpoczęciem wykonywania pierwszej operacji na każdej z maszyn, musi zostać wykonane przezbrojenie $s^l_{\pi_P^l(|\pi_P^l|),1},\ l \in \mathcal{M}$ z ostatniej operacji MPSu na pierwszą z następnego. Niech $s_P^\alpha(i)$

będzie przezbrojeniem maszyny przed wykonaniem operacji $i \in \mathcal{O}$. Wartość tego przezbrojenia na maszynie $v_P(i)$ dla przydziału P

$$s_P^{\alpha}(\pi_P^l(i)) = \begin{cases} s^l_{\pi_P^l(i),\pi_P^l(i+1)} & \text{dla } i = \{1,2,\ldots,|\pi_P^l|-1\} \\ s^l_{\pi_P^l(i),\pi_P^l(1)} & \text{dla } i = |\pi_P^l| \end{cases}, \quad l \in \mathcal{M}. \tag{4}$$

Rozwiązanie rozpatrywanego problemu sprowadza się do wyznaczenia przydziału operacji do maszyn oraz czasów rozpoczęcia i zakończenia wykonywania operacji w kolejnych MPSach. Przez $S^x = (S_1^x, S_2^x, \ldots, S_o^x)$ oznaczono momenty rozpoczęcia, a przez $C^x = (C_1^x, C_2^x, \ldots, C_o^x)$ zakończenia wykonywania operacji w x-tym MPSie. Ciągi te muszą spełniać następujące ograniczenia:

$$\forall i \in \mathcal{O} \quad C_i^x = S_i^x + p_i^{v_P(i)}, \tag{5}$$

$$\forall i \in \mathcal{O} \quad S_i^{x+1} = S_i^x + T, \tag{6}$$

$$\forall i \in \mathcal{O} \setminus \{1\} \quad S_i^x \geqslant C_{i-1}^x + s_P^{\alpha}(i-1), \tag{7}$$

$$S_1^{x+1} \geqslant C_o^x + s_P^{\alpha}(o), \tag{8}$$

gdzie x oznacza numery kolejnych MPSów, a T jest czasem cyklu (niekoniecznie optymalnym). Ograniczenie (5) wymusza ciągłość wykonywania operacji, równanie (6) cykliczność, nierówność (7) dotyczy przezbrojeń w ramach MPS, a nierówność (8) przezbrojeń pomiędzy MPSami.

Dla przydziału P, niech $T(P)$ oznacza minimalny czas pracy gniazda (czas cyklu), dla którego istnieją ciągi S^x i C^x spełniając ograniczenia (5)–(8). Łatwo zauważyć, że

$$T(P) = \sum_{i=1}^{o} \left(p_i^{v_P(i)} + s_P^{\alpha}(i) \right). \tag{9}$$

Problem sprowadza się do wyznaczenia $P^* \in \mathcal{P}$, minimalizującego (9).

3. Model grafowy

Niech P^Z oznacza zbiór przydziałów w których wszystkie operacje są wykonywane na dokładniej jednej z dwóch maszyn. W tym przypadku obliczenie $T(P)$ może być wykonane w czasie $O(o)$. W dalszej części pracy, o ile nie napisano inaczej, rozważane będą przydziały ze zbioru $\mathcal{P} \setminus P^Z$.

Poniżej przedstawiamy konstrukcję grafu, który wykorzystamy do wyznaczenia przydziału ze zbioru $\mathcal{P} \setminus P^Z$ minimalizującego (9). Graf skierowany $\mathcal{A}$ definiujemy następująco:

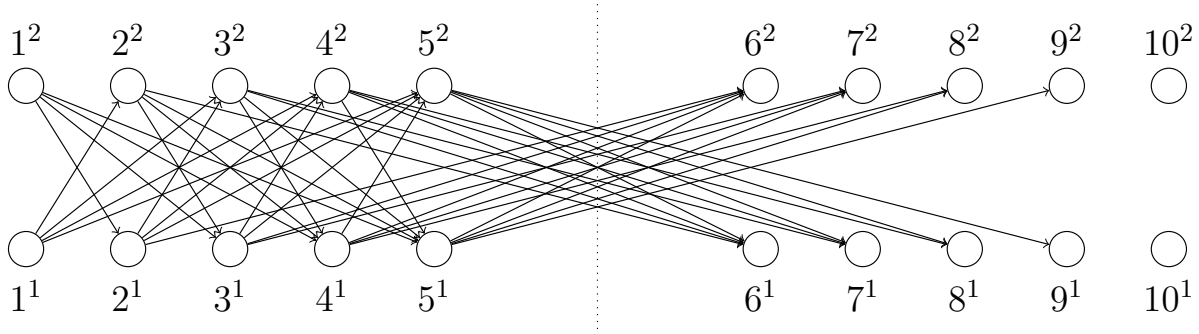
$$\mathcal{A} = (\mathcal{W} \cup \mathcal{W}',\ \mathcal{E} \cup \mathcal{E}'). \tag{10}$$

gdzie $\mathcal{W}$ i $\mathcal{W}'$ są zbiorami wierzchołków, a $\mathcal{E}$ i $\mathcal{E}'$ zbiorami łuków, przy czym

$$\mathcal{W} = \bigcup_{i \in \mathcal{M}} \bigcup_{j=1}^{o} \left\{ j^i \right\}, \quad \mathcal{W}' = \bigcup_{i \in \mathcal{M}} \bigcup_{j=o+1}^{2o} \left\{ j^i \right\}, \tag{11}$$

$$\mathcal{E} = \bigcup_{a \in \mathcal{M}} \bigcup_{b \in \mathcal{M} \setminus \{a\}} \bigcup_{i=1}^{o-1} \bigcup_{j=i+1}^{o} \left\{ \left(i^a, j^b \right) \right\}, \quad \mathcal{E}' = \bigcup_{a \in \mathcal{M}} \bigcup_{b \in \mathcal{M} \setminus \{a\}} \bigcup_{i=2}^{o} \bigcup_{j=o}^{o-1+i} \left\{ \left(i^a, j^b \right) \right\}. \tag{12}$$

Wierzchołek $i^a \in \mathcal{W}$ odpowiada operacji i wykonywanej na maszynie a, natomiast wierzchołek $j^a \in \mathcal{W}'$ odpowiednio kopii operacji $j-o$ z kolejnego MPSa. Zbiór $\mathcal{E}$ to zbiór łuków pomiędzy wierzchołkami ze zbioru $\mathcal{W}$, natomiast $\mathcal{E}'$ to zbiór łuków pomiędzy wierzchołkami ze zbiorów $\mathcal{W}$ i $\mathcal{W}'$. Przykład takiego grafu $\mathcal{A}$ dla liczby operacji $o=5$ przestawiono na rysunku 1.



Rys. 1. Graf $\mathcal{A}$ dla $n=5$.

Wierzchołki w grafie $\mathcal{A}$ nie są obciążone wagami. Z kolei waga łuku $\left(i^a, j^b\right) \in \mathcal{E} \cup \mathcal{E}'$ jest równa sumie czasów wykonywania oraz przezbrojeń dla operacji (lub ich kopii) od i do $j-1$, i wyraża się wzorem

$$d\left(i^a, j^b\right) = p'^a_i + s'^a_{i,j+1} + \sum_{k=i+1}^{j-1} \left(p'^b_k + s'^b_{k,k+1}\right), \tag{13}$$

gdzie $s'^a_{i,j}$ jest przezbrojeniem między operacjami odpowiadającymi wierzchołkom i^a i j^b

$$s'^a_{i,j} = s^a_{((i-1) \bmod o)+1,\ ((j-1) \bmod o)+1},\ a \in \mathcal{M},\ i \in \mathcal{O},\ j \in \mathcal{O} \setminus \{i\}, \tag{14}$$

natomiast p'^a_i jest czasem wykonywania operacji reprezentowanej przez wierzchołek i^a

$$p'^a_i = p^a_{((i-1) \bmod o)+1}, \quad a \in \mathcal{M},\ i \in \mathcal{O}. \tag{15}$$

Dla dowolnego przydziału $P \in \mathcal{P} \setminus P^Z$ przez $\pi'_P = (\pi'_P(1), \pi'_P(2), \ldots, \pi'_P(|\pi'_P|))$ oznaczono krotkę zawierającą wszystkie operacje po których następuje zmiana maszyny

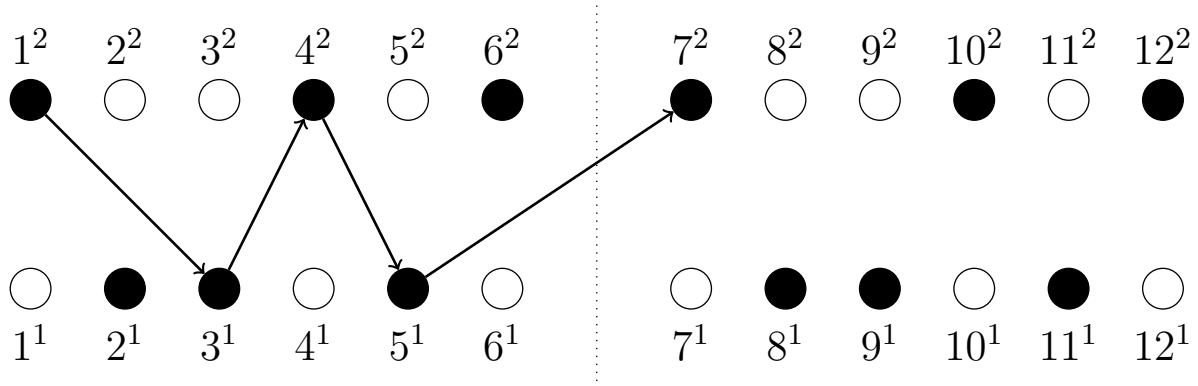
$$\forall i \in \mathcal{O} \setminus \{o\} \quad v_P(i) \neq v_P(i+1) \Rightarrow i \in \pi'_P, \tag{16}$$

$$v_P(o) \neq v_P(1) \Rightarrow o \in \pi'_P, \tag{17}$$

w której zachowana została także kolejność wykonywania operacji z π. Z kolei przez $\nu(P) = (\nu_1(P), \nu_2(P), \ldots, \nu_{|\pi'_P|+1}(P))$ oznaczono ścieżkę w grafie $\mathcal{A}$, gdzie wierzchołki ścieżki

$$\nu_k(P) = \begin{cases} \pi'_P(k)^{v_P(\pi'_P(k))} & \text{dla } k \in \{1, 2, \ldots, |\pi'_P|\}, \\ (\pi'_P(1)+o)^{v_P(\pi'_P(1))} & \text{dla } k = |\pi'_P|+1. \end{cases} \tag{18}$$

Definicja 1. *Ścieżkę* $(i^a, (i+1)^b, (i+2)^a, \ldots, (i+o)^a),\ a, b \in \mathcal{M},\ a \neq b,\ i \in \mathcal{O} \setminus \{o\}$ *w grafie* $\mathcal{A}$ *nazywamy ścieżką wyróżnioną. Zbiór wszystkich takich ścieżek oznaczamy przez* $\mathcal{N}$.



Rys. 2. Przykład *wyróżnionej* ścieżki z grafu $\mathcal{A}$.

Przykład *wyróżnionej* ścieżki dla przydziału $P = (\{1, 4, 6\},\ \{2, 3, 5\})$ oraz liczby operacji $o = 5$ pokazano na rysunku 2. Czarnymi punktami zaznaczono przydział.

Lemat 1. *Każda wyróżniona ścieżka $\mu \in \mathcal{N}$ odpowiada pewnemu przydziałowi $P \in \mathcal{P} \setminus P^Z$. tj.*

$$\forall \mu \in \mathcal{N},\ \exists P \in \mathcal{P} \setminus P^Z \quad \nu(P) = \mu. \tag{19}$$

Dowód. Weźmy dowolną wyróżnioną ścieżkę $\mu = (\mu_1 = i^a, \mu_2, \ldots, \mu_{|\mu|} = (i+o)^a)$ łączącą wierzchołki i^a i $(i+o)^a,\ 1 \leqslant i \leqslant o,\ a \in \mathcal{M}$. Definiujemy ciągi φ oraz γ w taki sposób, że dowolny wierzchołek ścieżki $\mu_k \in \mu$ można wyrazić za ich pomocą jako $\mu_k = \varphi_k^{\gamma_k},\ k \in \{1, 2, \ldots, |\mu|\}$. Łatwo zauważyć, że dla przydziału $P = (\mathcal{Z}_a, \mathcal{Z}_b)$, gdzie

$$\mathcal{Z}_a = \{1, \ldots, \varphi_1\} \cup \{\varphi_{|\varphi|-1} + 1, \ldots, n\} \cup \bigcup_{k=1}^{|\varphi|/2} \bigcup_{l=\varphi_{2k-1}}^{\varphi_{2k}} \{l\},\ \mathcal{Z}_b = \mathcal{O} \setminus \mathcal{Z}_a, \tag{20}$$

oraz $a = \gamma_1$, zachodzi $\nu(P) = \mu$.

■

Lemat 2. *Dla dowolnego przydziału $P \in \mathcal{P} \setminus P^Z$, ścieżka $\nu(P)$ w grafie $\mathcal{A}$ jest ścieżką wyróżnioną, tj.*

$$\forall P \in \mathcal{P} \setminus P^Z,\ \exists \mu \in \mathcal{N} \quad \nu(P) = \mu. \tag{21}$$

Dowód. Lemat 2. w sposób oczywisty wynika z definicji ścieżki $\nu(P)$. ■

Twierdzenie 1. *Funkcja $f : \mathcal{P} \setminus P^Z \to \mathcal{N},\ f(P) = \nu(P)$ jest wzajemnie jednoznaczna.*

Dowód. Funkcja f musi spełniać następujące własności wzajemnej jednoznaczności:

$$\forall \mu \in \mathcal{N},\ \exists P \in \mathcal{P} \setminus P^Z \quad f(P) = \nu(P) = \mu, \tag{22}$$

$$\forall P \in \mathcal{P} \setminus P^Z,\ \exists \mu \in \mathcal{N} \quad f(P) = \nu(P) = \mu, \tag{23}$$

$$\forall P_1, P_2 \in \mathcal{P} \setminus P^Z \quad f(P_1) = f(P_2) \Leftrightarrow P_1 = P_2. \tag{24}$$

Z lematów 2. i 1., równania (22) oraz (23) są spełnione. Rozważmy pierwsze $|\nu(P)| - 1$ wierzchołków ścieżki $\nu(P)$. Z definicji $\nu_k(P) = \pi'_P(k)^{v_P(\pi'_P(k))},\ k \in \{1, 2, \ldots, |\nu_P| -$

$1\}$. Ponieważ kolejność π'_P zawiera wszystkie operacje po których następuje zmiana maszyny, więc łatwo zauważyć, że

$$\forall P_1, P_2 \in \mathcal{P} \quad (P_1 = (\mathcal{Z}_1, \mathcal{Z}_2) \wedge P_2 = (\mathcal{O} \setminus \mathcal{Z}_1, \mathcal{O} \setminus \mathcal{Z}_2) \vee P_1 = P_2) \Leftrightarrow \pi'_{P_1} = \pi'_{P_2}. \quad (25)$$

Dla przypadków $P_1 = (\mathcal{Z}_1, \mathcal{Z}_2),\ P_2 = (\mathcal{O} \setminus \mathcal{Z}_1, \mathcal{O} \setminus \mathcal{Z}_2)$ różne wartości przyjmuje z kolei $v_{P_1}(k) \neq v_{P_2}(k),\ k \in \mathcal{O}$. Stąd warunek z równania (24) jest spełniony. ∎

Konsekwencją twierdzenia 1. jest istnienie funkcji odwrotnej do f, $g(f(P)) = P,\ P \in \mathcal{P} \setminus P^Z$. Funkcja g wzajemnie jednoznacznie przyporządkowuje dowolną wyróżnioną ścieżkę przydziałowi ze zbioru $\mathcal{P} \setminus P^Z$.

Lemat 3. *Dla dowolnego przydziału $P \in \mathcal{P} \setminus P^Z$, suma wag łuków wyróżnionej ścieżki $\nu(P)$ jest równa minimalnemu czasowi pracy gniazda $T(P)$*

$$d(\nu(P)) = T(P). \quad (26)$$

Dowód. Dowód polega na obliczeniu sumy wag łuków ścieżki $\nu(P)$

$$d(\nu(P)) = \underbrace{\sum_{k=1}^{|\pi'|-1} d((\nu_k(P), \nu_{k+1}(P)))}_{X(P)} + \underbrace{d((\nu_{|\pi'|}(P), \nu_{|\pi'|+1}(P)))}_{Y(P)}, \quad (27)$$

gdzie $X(P)$ jest sumą wag łuków należących do zbioru $\mathcal{E}$, a $Y(P)$ do zbioru $\mathcal{E}'$. Wartości $X(P)$ i $Y(P)$ można wyznaczyć za pomocą równania (13). Po przekształceniach (szczegółowo opisanych w raporcie [3]), otrzymujemy:

$$d(\nu(P)) = X(P) + Y(P) = \sum_{k=\pi'_P(1)}^{\pi'_P(|\pi'_P|)-1} \left(p_k^{v_P(k)} + s_P^{\alpha}(k)\right) + \sum_{k=\pi'_P(|\pi'_P|)}^{o} \left(p_k^{v_P(k)} + s_P^{\alpha}(k)\right) + \\ + \sum_{k=1}^{\pi'_P(1)-1} \left(p_k^{v_P(k)} + s_P^{\alpha}(k)\right) = \sum_{k=1}^{o} \left(p'^{v(k)}_k + s_P^{\alpha}(k)\right). \quad (28)$$

Porównując równanie (28) z (9), otrzymujemy $d(\nu(P)) = T(P)$. ∎

Twierdzenie 2. *W grafie $\mathcal{A}$, wyróżniona ścieżka o najmniejszej wadze odpowiada przydziałowi minimalizującemu czas pracy gniazda $T(P)$ dla $P \in \mathcal{P} \setminus P^Z$, tj.*

$$\arg\max_{\mu \in N}\{d(\mu)\} = \nu(\arg \max_{P \in \mathcal{P} \setminus P^Z}\{T(P)\}). \quad (29)$$

Dowód. Z lematu 3. i twierdzenia 1., mamy:

$$\arg\max_{\mu \in N}\{d(\mu)\} \overset{\text{tw. 1.}}{=} \arg \max_{\mu \in \bigcup_{P \in \mathcal{P} \setminus P^Z} \{\nu(P)\}}\{d(\mu)\} = \\ = \nu\Big(\arg \max_{P \in \mathcal{P} \setminus P^Z}\{d(\nu(P))\}\Big) \overset{\text{lm. 3.}}{=} \nu\Big(\arg \max_{P \in \mathcal{P} \setminus P^Z}\{T(P)\}\Big).$$

∎

Oczywistym następstwem twierdzenia 2. jest równanie

$$\max_{\mu \in N}\{d(\mu)\} = \max_{P \in \mathcal{P} \setminus P^Z}\{T(P)\}. \quad (30)$$

4. Metoda wyznaczania optymalnego przydziału

Algorytm wyznaczający optymalny przydział $P^* \in \mathcal{P}$ składa się z trzech kroków:

Krok 1 Zbudować graf $\mathcal{A}$ oraz wyznaczyć wagi wszystkich łuków.

Krok 2 Dla każdej pary wierzchołków $(i^a, (i+o)^a),\ i \in \mathcal{O} \setminus \{o\},\ a \in \mathcal{M}$ wyznaczyć ścieżkę o najmniejszej wadze z i^a do $(i+o)^a$. Z wyznaczonych ścieżek, wybrać tą o najmniejszej wadze. Ustalić odpowiadający jej przydział $P_1 \in \mathcal{P} \setminus P^Z$.

Krok 3 Wyznaczyć czas pracy gniazda $T(P)$ dla osobno rozpatrywanych przypadków $P_2 = (\emptyset,\ \mathcal{O})$ oraz $P_3 = (\mathcal{O},\ \emptyset)$. Obliczyć $P^* = \arg \min_{P \in \{P_1, P_2, P_3\}} \{T(P)\}$.

W kroku 2. algorytmu, na mocy twierdzenia 2., wyznaczana jest wyróżniona ścieżka (a tym samym przydział) o wadze

$$\min_{P \in \mathcal{P} \setminus P^Z} \{d(\nu(P))\} = \min_{P \in \mathcal{P} \setminus P^Z} \{T(P)\}. \tag{31}$$

W kroku 3. obliczane są wartości $T(P)$ dla $P \in P^Z$, stąd algorytm wyznacza rozwiązanie optymalne.

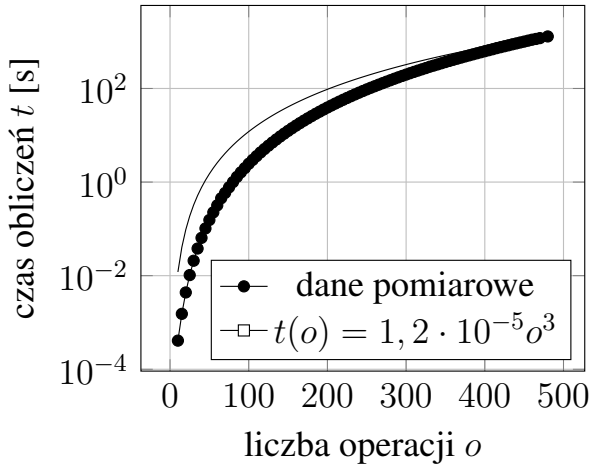
Twierdzenie 3. *Dla problemu przydziału operacji do maszyn, przydział P^* minimalizujący czas pracy dwumaszynowego gniazda $T(P)$ można wyznaczyć w czasie $O(o^3)$.*

Dowód. Dowód polega na analizie złożoności obliczeniowej opisanego wyżej algorytmu. Krok 1. polega na wyznaczeniu $O(o^2)$ wag łuków, przy czym każda z tych wag jest sumą $O(o)$ składników, stąd jego złożoność obliczeniowa $O(o^3)$. Krok 2. może być zrealizowany przez wyznaczanie najdłuższych ścieżek od wierzchołków początkowych do kolejnych wierzchołków (według kolejności topologicznej). Dla każdego z $O(o)$ wierzchołków początkowych, obliczana jest najkrótsza ścieżka do każdego z $O(o)$ wierzchołków. Ponieważ do każdego wierzchołka dochodzi $O(o)$ łuków, złożoność kroku 2. wynosi więc $O(o^3)$. Krok 3. sprowadza się do wyznaczenia czasu pracy gniazda dla osobno rozpatrywanych przydziałów z P^Z. Można je obliczyć ze wzoru (9) w czasie $O(o)$. Ostatecznie, złożoność obliczeniowa całego algorytmu wynosi $O(o^3) + O(o^3) + O(o) = O(o^3)$. ∎

5. Eksperymenty obliczeniowe

Przeprowadzono eksperymenty obliczeniowe zaproponowanej metody. Algorytm zaimplementowano w języku C++, jako kompilatora użyto Microsoft Visual Studio 2015. Program uruchomiono na komputerze PC wyposażonym w procesor Intel Core i7-4930K CPU@3,4GHz oraz 32GB pamięci RAM. Wygenerowano losowe przykłady o liczbie operacji od 10 do 480. Wyniki pomiarów czasów obliczeń przedstawiono na wykresie 3 oraz w tabeli 1.

Otrzymane wyniki są zgodne z wyznaczoną teoretycznie złożonością obliczeniową. Czasy obliczeń dla liczby operacji $o < 80$ są krótsze od 1s, co pozwala na stosowanie metody jako podprocedur wielopoziomowych algorytmów rozwiązywania problemów cyklicznych.



Rys. 3. Zależność czasu wykonywania algorytmu t od liczby operacji.

Tabela 1
Zależność czasu wykonywania algorytmu t od liczby operacji.

o	t [s]	o	t [s]	o	t [s]
10	0.0004	120	5.0841	320	256.40
20	0.0044	140	9.4156	340	326.64
30	0.0209	160	16.053	360	410.53
40	0.0642	180	25.709	380	509.12
50	0.1548	200	39.187	400	625.23
60	0.3183	220	57.334	420	762.52
70	0.5870	240	81.167	440	917.27
80	1.0000	260	111.50	460	1101.4
90	1.6141	280	149.93	480	1293.2
100	2.4623	300	198.09		

6. Podsumowanie

W artykule rozważamy cykliczną pracę dwumaszynowego gniazda produkcyjnego z uwzględnieniem czasów przezbrojeń. Przy założeniu, że ustalona jest kolejność wykonywania operacji udowodniliśmy, iż optymalny przydział operacji do maszyn można wyznaczyć w czasie $O(o^3)$, gdzie o jest liczbą operacji, choć możliwych przydziałów jest wykładnicza liczba. W dalszych badaniach planujemy rozszerzyć nasze rozważania na gniazda z większą niż dwa liczbą maszyn.

LITERATURA

1. Bożejko W., Uchroński M., Wodecki M.: Parallel metaheuristics for the cyclic flow shop scheduling problem. Computers & Industrial Engineering 95, 2016, p. 156–163.
2. Bożejko W., Wodecki M.: Problemy cykliczne z równoległymi maszynami. W: Automatyzacja procesów dyskretnych, Teoria i zastosowania (red. A. Świerniak, J. Krystek), Gliwice, 2014, p. 27–36.
3. Gnatowski A.: Opracowanie własności problemu przydziału do zastosowania w cyklicznych problemach szeregowania zadań. Raporty Katedry Automatyki, Mechatroniki i Systemów Sterowania Politechniki Wrocławskiej, Ser. PRE nr 5, 2016.
4. Kampmeyer T.: Cyclic Scheduling Problems. Praca doktorska, University Osnabriick, 2006.
5. Levner E., Kats V, Lopez A.P., Cheng T.C.E.: Complexity of cyclic scheduling problems: A state-of-the-art survey. Computers and Industrial Engineering 59, 2010, 352–361.
6. Sawik T.: A mixed integer programming for cyclic scheduling of flexible flow lines. Bulletin of the Polish Academy of Sciences, Technical Sciences, 62(1), 2014, p. 121–128.