CYKLICZNY PROBLEM PRZEPŁYWOWY Z PRZEZBROJENIAMI MASZYN

Wojciech BOŻEJKO, Łukasz KACPRZAK, Mieczysław WODECKI

Streszczenie: W pracy zajmujemy się cyklicznym problemem przepływowym z przezbrojeniami maszyn. Polega on na wytwarzaniu w ustalonych odstępach czasu (czas cyklu) pewnego zestawu asortymentów. Optymalizacja procesu sprowadza się do minimalizacji czasu cyklu, tj. czasu po którym może nastąpić wykonywanie następnej partii elementów. Ponieważ problem jest *silnie NP-trudny*, więc do jego rozwiązywania będziemy stosowali algorytmy przybliżone. Przedstawiamy model grafowy problemu oraz „własności eliminacyjne bloków" umożliwiające znaczne zmniejszenie otoczeń stosowanych w algorytmach lokalnych poszukiwań.

Słowa kluczowe: problemy cykliczne, szeregowanie zadań, problem przepływowy, czas cyklu.

1. Wprowadzenie

Od wielu lat można zaobserwować rosnące zapotrzebowanie rynku na różnorodność (wieloasortymentowość) produkcji. Może być to zrealizowane, między innymi, poprzez wytwarzanie cykliczne. W ustalonych odstępach czasu (czas cyklu) jest produkowana pewna "partia" (mieszanka, zestaw) asortymentów. Optymalizacja procesu zazwyczaj sprowadza się do minimalizacji czasu cyklu. Właściwy dobór mieszanki oraz czasu cyklu pozwala zrealizować zapotrzebowanie, a ponadto zwiększyć wydajność oraz efektywności wykorzystania maszyn. W ostatnich latach widać wyraźnie, po liczbie ukazujących się publikacji, znaczne zwiększenie się zainteresowania cyklicznymi problemami teorii szeregowania zadań. Są to bowiem zazwyczaj problemy ciekawe i trudne (w większości *NP-trudne*), ważne zarówno z punktu widzenia teorii jak i praktyki.

W przeglądowej pracy Panwalkara i in. [12] dotyczącej szeregowania zadań stwierdzono, że 75% występujących w praktyce problemów wymaga przynajmniej jednego przezbrojenia zależnego od kolejności wykonywania zadań. Natomiast, w 15% problemów należy uwzględnić przezbrojenia pomiędzy wszystkimi zadaniami. Jednak w znaczącej większości prac z dziedziny szeregowania zadań nie uwzględnia się przezbrojeń. Dotyczy to zarówno problemów jedno, jak i wielomaszynowych oraz różnych funkcji celu. Tego typu zagadnienia są ważne ze względu teoretycznego i praktycznego.

Po optymalizacji czasu cyklu dalszą poprawę efektywności funkcjonowania systemu uzyskuje się poprzez eliminację lub ograniczenie magazynowania półproduktów w magazynach pośrednich lub pomiędzy stanowiskami; warunki te znane są w literaturze jako ,,zakaz składowania międzystanowiskowego" (ang. *no store*), „ograniczony obszar składowania" (ang. *limited store, buffer*) oraz „ograniczenie czasu oczekiwania" (ang. *waiting time constraints*). Problemy cykliczne stanowią unikalną, mało zbadaną podklasę problemów szeregowania, bowiem dotyczą zwykle nieregularnych kryteriów. Systemy cykliczne (w tym także z dodatkowymi ograniczeniami składowania) cieszą się ostatnio coraz większym zainteresowaniem zarówno praktyków jak i teoretyków. Są obiektem badań głównie ze względu na ich duże znaczenie praktyczne oraz kłopoty z uzyskaniem odpowiednio efektywnych algorytmów rozwiązywania.

Zdecydowana większość otrzymanych dotychczas interesujących wyników dla systemów z buforowaniem i ograniczeniami czasów oczekiwania dotyczy niecyklicznych wariantów tego typu [1]. Generalnie, deterministyczne niecykliczne problemy z ograniczeniami składowania należą do jednych z trudniejszych zagadnień optymalizacji dyskretnej i były dotychczas rozważane dla dość regularnych struktur wytwarzania, szeregowych lub szeregowo - równoległych. Z kolei problemy cykliczne z ograniczeniami składowania były dotychczas formułowane i analizowane wyłącznie dla systemów o stosunkowo prostej strukturze, np. system jednostanowiskowy lub przepływowy [13]. Zarówno stopień skomplikowania modelu jak i konstrukcja algorytmu dla systemu cyklicznego jest istotnie bardziej złożony. Punktem wyjścia do analizy prezentowanej w niniejszej pracy są wyniki z monografii [9], dla niecyklicznego systemu o strukturze szeregowo – równoległej z buforowaniem, pozwalające na ich uogólnienie na systemy o dowolnej strukturze. W badaniach wykorzystane są także oryginalne wyniki cząstkowe zamieszczone w pracach [14-16] dotyczące systemów cyklicznych. Silna NP-trudność już wielu najprostszych badanych wersji problemu, ogranicza zakres stosowania algorytmów dokładnych do instancji o małej liczbie zadań, choć w kontekście minimalizacji czasu cyklu konstrukcja i zastosowanie algorytmów dokładnych wydaje się całkowicie uzasadnione (i będzie brana pod uwagę). Niemniej, ze względu na NP-trudność, do wyznaczania satysfakcjonujących rozwiązań, stosuje się powszechnie szybkie algorytmy przybliżone oparte na technikach przeszukiwań lokalnych. Metody tego typu opierają się zazwyczaj na dwupoziomowej dekompozycji problemu: wyznaczenie optymalnej kolejności zadań (poziom górny) oraz wielokrotne wyznaczenie minimalnej wartości kryterium dla danej kolejności zadań (poziom dolny). O ile dla klasycznych problemów szeregowania rozwiązanie problemu dolnego poziomu można otrzymać w sposób efektywny czasowo przez analizę specyficznego grafu, to w przypadku postawionego zagadnienia rozwiązanie zagadnienia dolnego poziomu jest czasochłonne. Stąd wszelkie własności szczególne problemu, w tym pozwalające na bardziej efektywne wyliczanie czasu cyklu, poszukiwanie harmonogramu oraz ograniczenie wielkości lokalnie przeglądanego sąsiedztwa i/lub przyspieszenie szybkości jego przeglądania są bardzo pożądane.

Bogaty przegląd stanu wiedzy dotyczącego cyklicznych zagadnień szeregowania zadań znaleźć można w pracy Levner i in. [10], podejmującej zagadnienia złożoności obliczeniowej algorytmów rozwiązujących poszczególne typy cyklicznych zagadnień szeregowania. Autorzy rozpatrują w szczególności problematykę NP-trudności różnego rodzaju problemów cyklicznych z uwzględnieniem różnorodnych postaci funkcji kryterialnych oraz dodatkowych ograniczeń (*no wait, no buffer*, etc.).

Celem prowadzonych badań jest przede wszystkim identyfikacja i formalizacja modelu współczesnych przepływowych cyklicznych systemów wytwarzania, w szczególności z uwzględnieniem czasów przezbrojeń maszyn, a także opracowanie i weryfikacja metod rozwiązywania tak zdefiniowanych zagadnień optymalizacji dyskretnej z uwzględnieniem różnych funkcji celu (głównie czas cyklu). Przeprowadzona zostanie weryfikacja jakości uzyskiwanych rozwiązań w odniesieniu do wyników prezentowanych w specjalistycznych, zamieszczonych w Internecie przykładach testowych (*benchmarks*).

W pracy rozpatrujemy wielomaszynowy system produkcji cyklicznej, w którym dowolny element z ustalonej partii (mieszanki) przechodzi kolejno przez każdą z maszyn (permutacyjny system przepływowy (ang. *permutation flow shop*), zobacz Nowicki i Smutnicki [11] oraz Grabowski i Wodecki [8]). Ponadto, należy dokonać przezbrojenia każdej maszyny, przed wykonaniem kolejnego elementu. Problem polega na minimalizacji czasu cyklu, tj. czasu po którym może nastąpić wykonywanie następnej partii elementów.

Udowodnimy *silną NP-trudność* już znacznie uproszczonej wersji rozpatrywanego problemu. O trudności zagadnienia świadczy fakt, że jeden z etapów algorytmu sprowadza się do rozwiązania problemu komiwojażera. Z tego powodu, do wyznaczania satysfakcjonujących rozwiązań będziemy stosowali szybkie algorytmy przybliżone oparte na metodzie lokalnych przeszukiwań. Udowodnimy tzw. "własności eliminacyjne bloków", które zostaną zastosowane w konstrukcjach algorytmów. Umożliwiają one przegląd pośredni pewnych podzbiorów przestrzeni rozwiązań przyspieszając obliczenia i poprawiając jednocześnie jakość wyznaczanych rozwiązań.

2. Sformułowanie problemu

Rozpatrywany w pracy system wytwarzania jest pewnym rozszerzeniem *silnie NP-trudnego*, klasycznego w teorii szeregowania zadań, permutacyjnego problemu przepływowego (oznaczanego w literaturze prze $F^* \| C_{\max}$, zobacz: Nowicki i Smutnicki [11] oraz Grabowski i Wodecki [8]). Można go sformułować następująco:

Problem: Dany jest zbiór n zadań $J = \{1,2,\ldots,n\}$, które należy wykonać cyklicznie (w sposób powtarzalny) na maszynach ze zbioru $M = \{1,2,\ldots,m\}$. Dowolne zadanie należy wykonać kolejno, na każdej z m maszyn $1,2,\ldots,m$ (porządek technologiczny). Zadanie $j \in J$ jest ciągiem m operacji $O_{1,j}, O_{2,j},\ldots,O_{m,j}$. Operacja $O_{k,j}$ odpowiada czynności wykonywania zadania j na maszynie k, w czasie $p_{k,j}$ $(k = 1,2,\ldots,m)$ $j = 1,2,\ldots,n$. Po zakończeniu pewnej, a przed rozpoczęciem następnej operacji należy wykonać przezbrojenie maszyny. Niech $s_{i,j}^{k}$ $(k \in M, i \neq j\, i, j \in J)$ będzie czasem przezbrojenia pomiędzy operacją $O_{k,i}$ oraz $O_{k,j}$. Problem polega na wyznaczeniu kolejności wykonywania zadań (takiej samej na każdej maszynie), która minimalizuje czas cyklu, tj. termin rozpoczęcia wykonywania zadań ze zbioru J w następnym cyklu. Muszą być przy tym spełnione następujące ograniczenia:

(a) każda operacja może być wykonywana tylko przez jedną, określoną przez ciąg technologiczny, maszynę,
(b) żadna maszyna nie może wykonywać jednocześnie więcej niż jedną operację,
(c) zachowany musi być porządek technologiczny wykonywania operacji,
(d) wykonywanie żadnej operacji nie może być przerwane przed jej zakończeniem,
(e) pomiędzy kolejno wykonywanymi, na tej samej maszynie, operacjami należy dokonać przezbrojenia.

Zbiór zadań wykonywanych w pojedynczym cyklu nazywany jest MPS-em (*ang. minimal part set*). MPS-y są przetwarzane jeden po drugim w sposób cykliczny, dostarczając partię zadań (mieszankę produktów) w ilościach odpowiedniej dla każdego cyklu.

Problem sprowadza się więc do ustalenia momentów rozpoczęcia wykonywania zadań na maszynach spełniających ograniczenia (a)-(e), aby czas cyklu (czas po którym zadanie jest wykonywane w kolejnym MPS-ie) był minimalny. W skrócie problem ten będziemy oznaczali przez **CF**.

Przyjmujemy, że zadania ze zbioru J wykonywane są w *określonej kolejności*, takiej samej dla wszystkich MPS-ów, co zapewnia cykliczność sekwencji, ale niekoniecznie harmonogramów czasowych w ramach pojedynczego MPS-u.

W pracy (Grabowskiego i Pempery [7]) udowodniono, że w systemie przepływowym dopuszczalna kolejność wykonywania zadań musi być identyczna na każdej maszynie. Zatem, kolejność wykonywania zadań w każdym MPS-ie może być wyznaczona przez pojedynczą permutację.

Każde uszeregowanie zadań może być reprezentowane przez permutację $\pi = (\pi(1),\ldots, \pi(n))$ elementów ze zbioru J. Niech Φ oznacza zbiór wszystkich takich permutacji.

Wobec tego, rozpatrywany w pracy problem sprowadza się do wyznaczenia permutacji zadań zbioru J minimalizującej długość cyklu.

3. Model matematyczny

Niech $[S^k]_{m\times n}$ będzie macierzą terminów rozpoczęcia wykonywania zadań k-tego MPS-a (dla ustalonej kolejności π), gdzie $S^k_{i,j}$ oznacza termin rozpoczęcia wykonywania zadania j na maszynie i. Zakładamy, że nie tylko kolejność zadań jest cyklicznie powtarzana ale także, że harmonogram czasowy pracy systemu (tj. wykonywanie kolejnych MPS-ów) jest w pełni cykliczny. Oznacza to, że istnieje stała $T(\pi)$ (okres) taka, że

$$S^{k+1}_{i,\pi(j)} = S^k_{i,\pi(j)} + T(\pi),\ i=1,\ldots,m,\ j=1,\ldots,n,\ k=1,2,\ldots \tag{1}$$

Okres $T(\pi)$ zależy oczywiście od permutacji π i jest nazywany *czasem cyklu* systemu. Minimalną wartość $T(\pi)$, dla ustalonej π, będziemy nazywać *minimalnym czasem cyklu* i oznaczać przez $T^*(\pi)$. Ponieważ klejność wykonywania zadań w ramach pojedynczego MPS-u jest taka sama, więc wystarczy wyznaczyć kolejność zadań π dla *jednego* MPS-a i dokonać jego przesunięcia o wielkość $k\cdot T(\pi)$, k=1,2,... na osi czasu. Optymalną wartość czasu cyklu $T^*(\pi^*)$ dla pierwszego MPS-a problemu **CF** można wyznaczyć rozwiązując następujące zadanie optymalizacyjne:

wyznaczyć

$$T^*(\pi^*) = \mathbf{min}\{T^*(\pi) : \pi \in \Phi\}, \tag{2}$$

przy ograniczeniach:

$$S_{i,\pi(j)} + p_{i,\pi(j)} \le S_{i+1,\pi(j)},\ i=1,\ldots,m-1,\ j=1,\ldots,n, \tag{3}$$

$$S_{i,\pi(j)} + p_{i,\pi(j)} \le S_{i,\pi(j+1)},\ i=1,\ldots,m,\ j=1,\ldots,n-1, \tag{4}$$

$$S_{i,\pi(n)} + p_{i,\pi(n)} \le S_{i,\pi(1)} + T,\ i=1,\ldots,m, \tag{5}$$

$$S_{i+1,\pi(n)} \le S_{i,\pi(1)} + T,\ i=1,\ldots,m-1. \tag{6}$$

Bez straty ogólności możemy przyjąć, że termin rozpoczęcia wykonywania pierwszego zadania na pierwszej maszynie $S_{1,\pi(1)} = 0$.

Niech

$$T_k(\pi) = \sum_{i=1}^{n-1}(p_{k,\pi(i)} + s^k_{\pi(i),\pi(i+1)}) + p_{k,\pi(n)}. \tag{7}$$

będzie czasem wykonywania zadań w kolejności π, wraz z przezbrojeniami, na k-tej maszynie.

4. Problem z zerowymi czasami przezbrojeń

Zakładamy, że czasy przezbrojenia maszyn, pomiędzy kolejno wykonywanymi operacjami, są równe zero, więc dla dowolnej permutacji $\pi \in \Phi$, $s^k_{\pi(i),\pi(i+1)} = 0$, $j,k = 1,2\ldots,$ m, $i = 1,2,\ldots,n$**.** Wówczas, czas (7) wykonywania operacji przez k-tą maszynę

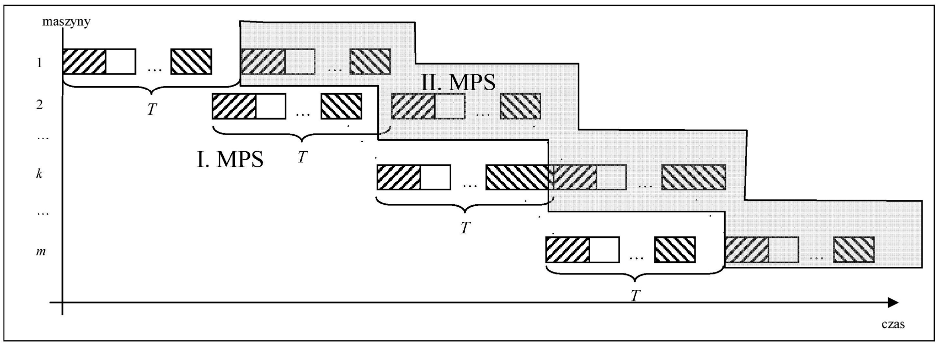
$$T^k(\pi) = \sum_{i=1}^{n} p_{k,\pi(i)} \tag{8}$$

i nie zależy od kolejności wykonywania zadań π. Łatwo zauważyć, że w tym przypadku minimalny czas cyklu

$$T^*(\pi^*) = min\{T^k(\pi) : k = 1,2,\ldots,m\}, \tag{9}$$

gdzie π jest dowolną permutacją zadań ze zbioru J.

Na rysunku 1 przedstawiono fragment diagramu Gantta dwóch pierwszych MPS-ów, dla problemu CF z zerowymi czasami przezbrojeń. Przyjęliśmy, że maksimum w równości (9) osiągnięto dla k-tej maszyny.



Rysunek 1. Diagram Gantta dla dwóch pierwszych MPS-ów, dla problemu cyklicznego bez przezbrojeń maszyn

5. Własności problemu cyklicznego z przezbrojeniami

Niech $\pi \in \Phi$ będzie pewną ustaloną permutacją zadań. Przez $S^h_{k,\pi(i)}$ oznaczamy moment rozpoczęcia zadania $\pi(i)$ z h-tego MPS-u na k-tej maszynie. Dla ustalenia uwagi rozpatrujemy dwa pierwsze MPS-y. Z ograniczeń (3)-(6) wynika, że

$$S^2_{k,\pi(1)} \geq S^1_{k,\pi(i)} + \sum_{i=1}^{n} p_{k,\pi(i)} + \sum_{i=1}^{n-1} s^k_{\pi(i),\pi(i+1)} + s^k_{\pi(n),\pi(1)}, \tag{10}$$

dla $k = 1,2,\ldots,m$. Podobne zależności zachodzą dla kolejnych MPS-ów, tj. pomiędzy $S^h_{k,\pi(1)}$ oraz $S^{h+1}_{k,\pi(1)}$ $(h = 2,3,\ldots)$.

Podobnie jak (9), czas wykonywania zadań ze zbioru J w kolejności $\pi \in \Phi$ na k-tej maszynie w ramach jednego MPS-u (wraz z przezbrojeniem maszyny po wykonaniu ostatniego zadania)

$$T_k(\pi) = \sum_{i=1}^{n-1}(p_{k,\pi(i)} + s^k_{\pi(i),\pi(i+1)}) + p_{k,\pi(n)} + s^k_{\pi(n),\pi(1)}, \tag{11}$$

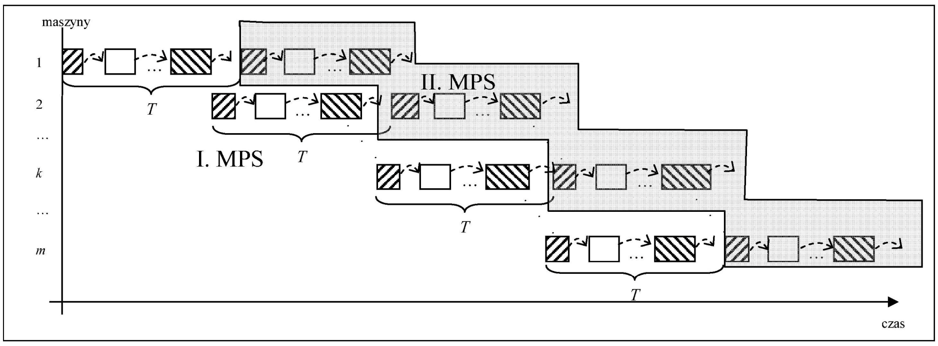
dla $k = 1,2,\ldots,m$.

W tym przypadku

$$T^*(\pi) = \mathbf{max}\{T_k(\pi) : k = 1,2,\ldots\}, \tag{12}$$

spełnia wszystkie warunki długości cyklu.

Na rysunku 2 przedstawiono fragment diagramu Gantta dwóch pierwszych MPS-ów. Przyjęliśmy, że maksimum w równości (11) osiągnięto dla k-tej maszyny.



Rysunek 2. Diagram Gantta dwóch pierwszych MPS-ów dla problemu cyklicznego z przezbrojeniami maszyn.

Wobec tego, wyznaczenie minimalnego czasu cyklu, dla problemu przepływowego z przezbrojeniami maszyn, sprowadza się do wyznaczenia permutacji optymalnej π^* dla której

$$T^*(\pi^*) = \mathbf{min}\{T^*(\pi) : \pi \in \Phi\}. \tag{13}$$

Dla ustalonej permutacji zadań π minimalny czas cyklu $T^*(\pi)$ został określony w (12). Załóżmy, że minimum to zostało osiągnięte dla k-tej maszynie, tj. $T^*(\pi) = T_k(\pi)$. Rozpatrujemy ciąg czasów $T_1(\pi), T_2(\pi), \ldots, T_m(\pi)$. Symbolicznie jest on przedstawiony na rysunku 3.

Dla ustalonej maszyny k, $k = 1,2,\ldots,m$ konstruujemy symetryczny graf pełny (sieć)

$$H_k = \langle V, E; p; s\rangle, \tag{14}$$

z obciążonymi wierzchołkami i krawędziami, gdzie

zbiór wierzchołków: $V = J$,

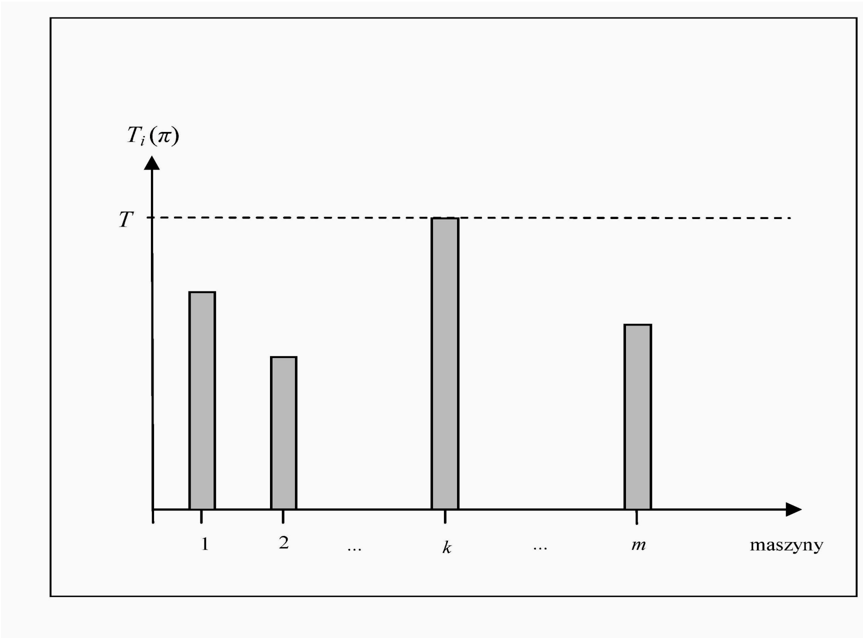
zbiór krawędzi: $E = \{\{v,u\} : v \neq u,\ v,u \in J\}$,

wagi wierzchołków: $p : V \to R$,

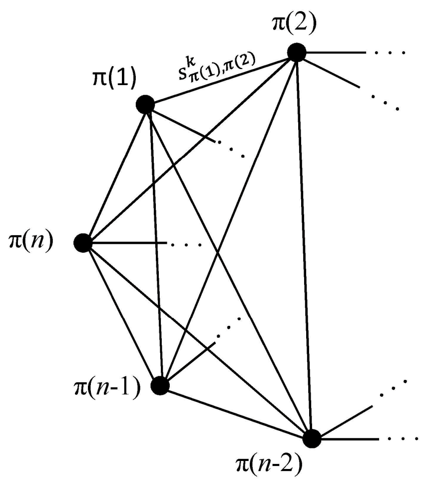
$p(v) = p_{k,v}\ v \in V$,

wagi krawędzi: $s : E \to R$, $s(e) = s^k_e\ e \in E$.

Wagi wierzchołków są równe czasom wykonywania zadań, a krawędzi - czasom przezbrojeń. Fragment grafu H_k, bez czasów wykonywania zadań, przedstawiono na rysunku 4.



Rysunek 3: Wartości $T_k(\pi)$, $k = 1, 2, \ldots, m$.



Rysunek 4. Fragment grafu H_k.

Lemat 1. *Dla ustalonej permutacji zadań* $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$*, czas wykonywania zadań* $T_k(\pi)$ *(definicja (11)) na maszynie* $k, k = 1,2,\ldots,m$ *jest równy długości* $L_k(\pi)$ *cyklu komiwojażera* $P_k(\pi) = (\pi(1), \pi(2), \ldots, \pi(n), \pi(1))$ *w grafie* H_k.

Lemat 2. *Permutacja* $\beta \in \Phi$ *jest optymalną kolejnością wykonywania zadań na* k *-tej maszynie (tj. minimalizuje funkcję (11)) wtedy i tylko wtedy, gdy* $P_k^* = (\beta(1), \beta(2), \ldots, \beta(n), \beta(1))$ *jest optymalnym cyklem komiwojażera w grafie* H_k.

Niech L_k^* będzie długością optymalnego cyklu komiwojażera P_k^* w grafie symetrycznym H_k, $k = 1,2,\ldots,m$.

Uwaga 1. *Dla dwóch różnych maszyn* k *i* l *nie musi zachodzić równość* $L_k^* = L_l^*$.

Uwaga 2. *Optymalna wartość czasu cyklu*

$$T^*(\pi^*) \geq \mathbf{min}\{L_k^* : k = 1,2,\ldots,m\}. \tag{15}$$

Twierdzenie 1 *Jeżeli czasy przezbrojeń pomiędzy operacjami są na każdej maszynie takie same, to problem wyznaczenia optymalnego czasu cyklu jest silnie NP-trudny.*

Dowód. W tym przypadku każde dwa dowolne grafy H_k i H_l $(k \neq l, l = 1,2,\ldots,m)$ są izomorficzne, więc długości optymalnych cykli komiwojażera są takie same. Łatwo zauważyć (korzystając z nierówności (15)), że wyznaczenie optymalnego czasu cyklu sprowadza się do wyznaczenia optymalnego cyklu komiwojażera w grafie H_1. Jest więc problemem silnie *NP-trudnym*.

Wniosek. Rozpatrywany w pracy problem **CF** jest silnie NP-trudny.

6. Bloki zadań

Niech π_k^* będzie optymalnym cyklem Hamiltona w grafie $H_k(\pi)$ $(k = 1,2,\ldots,m)$. Jest to optymalna (tj. minimalna ze względu na czas wykonywania) kolejność zadań. Permutację tą będziemy nazywali *wzorcem* dla *k*-tej maszyny.

Niech

$$B = \{\pi(a), \pi(a+1), \ldots, \pi(b)\}, \tag{16}$$

będzie ciągiem bezpośrednio występujących po sobie zadań w permutacji $\pi \in \Phi$, π_k^* *wzorcem* dla *k*-tej maszyny oraz u, v $(u \neq v, u, v = 1,\ldots,n)$ parą liczb takich, że:

W1: $\pi(a) = \pi^*(u), \pi(a+1) = \pi^*(u+1), \ldots, \pi(b-1) = \pi^*(v-1), \pi(b) = \pi^*(v)$, lub

W2: $\pi(b) = \pi^*(u), \pi(b-1) = \pi^*(u+1), \ldots, \pi(a+1) = \pi^*(v-1), \pi(a) = \pi^*(v)$

W3: *B* jest maksymalnym podciągiem ze względu na zawieranie, tj. nie można go powiększyć ani o element $\pi(a-1)$, ani o $\pi(b+1)$, spełniającym ograniczenia **W1** lub **W2**),

Jeżeli ciąg zadań (16) spełnia warunki **W1** i **W3** lub **W2** i **W3**, to nazywamy go *blokiem* na k-tej maszynie ($k \in M$).

Z definicji wynika, że blok może zawierać tylko jeden element. Jeżeli liczba elementów $b - a \geq 2$, to taki blok bez pierwszego i ostatniego elementu nazywamy *blokiem wewnętrznym*.

Własność 1. *Permutację zadań $\pi \in \Phi$ można, na dowolnej maszynie $k \in M$, rozbić na rozłączne bloki.*

Twierdzenie 2 [17]. *Algorytm rozbicia n elementowej permutacji zadań na bloki ma złożoność obliczeniową $O(n)$.*

Twierdzenie 3 [17]. *Jeżeli permutacja β została wygenerowana z permutacji π przez zmianę kolejności elementów w pewnym bloku wewnętrznym na maszynie $k \in M$, to*

$$T_k(\beta) \geq T_k(\pi).$$

Uwaga 3. *Aby zmniejszyć bieżącą długość cyklu $T^*(\pi)$ należy pewne zadanie z pewnego bloku wewnętrznego przestawić przed pierwszy lub za ostatni element tego bloku.*

Z twierdzenia 3 wynika tzw. "własność eliminacyjna bloków". Dzięki niej, przy generowaniu otoczeń w algorytmach typu popraw, można pominąć rozwiązania nie dające bezpośredniej poprawy. Przyspiesza to znacznie działanie algorytmu.

7. Podsumowanie

W pracy rozpatrywano cykliczny problem przepływowy z przezbrojeniami maszyn. Należy od do klasy problemów *silnie NP-trudnych.* Przedstawiono model matematyczny oraz udowodniono pewne specyficznych własności problemu, w tym „eliminacyjne własności blokowe". Będą one bezpośrednio zastosowane w konstrukcjach algorytmów rozwiązywania przedstawionego problemu.

Praca została częściowo sfinansowana ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC-2012/05/B/ST7/00102.

Literatura

1. Bożejko W., Makuchowski M., Solving the no-wait job shop problem by using genetic algorithm with automatic adjustement, International Journal of Advanced Manufacturing Technology, Volume 57, Issue 5, 2011, 735-752.
2. Chaar J.K., Davidson E.S. Cyclic Job Shop Scheduling Using Reservation Tables. Department of Electrical Engineering and Computer Science. The University of Michigan. 1990, 2128-2135.
3. Cavory G., Dupas R., Goncalves G. A genetic approach to solving the problem of cyclic job shop scheduling with linear constraints. European Journal of Operational Research 161, 2005, 73-85.
4. Caggiano K., Jackson P.L. Finding minimum flow time cyclic schedules for non-identical, multistage jobs. IIE Transactions, 40, 2008, 45-65.

5. Dąbrowski P., Pempera J., Smutnicki C.: Minimizing cycle time of the flow line-genetic approach with gene expression, Lecture Notes in Computer Science, vol. 4431, 2007, 194-201.
6. Fournier O., Lopez, P., Lan Sun Luk J.D. Cyclic scheduling following the social behavior of ant colonies. Conference Publications Systems, Man and Cybernetics, 2002 IEEE International. 2002.
7. Grabowski J., Pempea J., Sequencing of jobs in some production system, EJOR, 2000,125, 535–550.
8. Grabowski J., Wodecki M., A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion, *Computers and Operations Research*, 31, 2004, 1891–1909.
9. Kampmeyer T., Cyclic Scheduling Problems, Ph.D. Dissertation, Universitat Osnabruck, 2006.
10. 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.
11. Nowicki E., C. Smutnicki, A fast tabu search algorithm for the permutation flow-shop problem, European Journal of Operational Research, 1996;91:160–175.
12. Panwalkar S.S., Dudek R.A., Smith M.L.: Sequencing research and the industrial scheduling problem. Symposium on the theory of scheduling and its applications (ed. Elmaghraby S.E.), Springer-Verlag, Berlin, 1973.
13. Pempera J., Smutnicki C., Minimalizacja czasu cyklu wytwarzania na linii: Podejście genetyczne z ekspresją genów. Automatyka 2005, t. 9, z. 1/2, 189-199.
14. Smutnicki C.: Scheduling with high variety of customized compound products. Decision Making in Manufacturing and Services, 1, 1/2, 2007, 91-110.
15. Smutnicki C., Smutnicki A.: Nowe własności harmonogramów cyklicznych w systemie przepływowym, Automatyka, t. 11, z. 1/2, 2007, 275-285.
16. Smutnicki C., Smutnicki A.: Harmonogramowanie cykliczne w systemie gniazdowym, Zastosowania teorii systemów, AGH, 2007, 105-115.
17. Wodecki M., Identyfikacja własności cyklicznego problemu przepływowego pod kątem ich wykorzystania w konstrukcji efektywnych algorytmów rozwiązania, Raport Instytutu Informatyki, Automatyki i Robotyki Politechniki Wrocławskiej, 2013 (w przygotowaniu).

Prof. nadzw. dr hab. Wojciech BOŻEJKO
Mgr inż. Łukasz KACPRZAK
Instytut Informatyki Automatyki i Robotyki Politechniki Wrocławskiej
ul. Janiszewskiego 11/17, 50-372 Wrocław
e-mail: wojciech.bozejko@iiar.pwr.wroc.pl

Prof. nadzw. dr hab. Mieczysław WODECKI
Instytut Informatyki Uniwersytetu Wrocławskiego
ul.Joliot-Curie 15, 50-383 Wrocław
e-mail: mwd@ii.uni.wroc.pl