ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1990
Seria: AUTOMATYKA z.100 Nr kol.1082

Joanna Józefowska
Politechnika Poznańska

DETERMINISTYCZNE PROBLEMY SZEREGOWANIA ZADAŃ W DWUSTANOWISKOWYM PRZEPŁYWOWYM SYSTEMIE OBSŁUGI

Streszczenie. Praca dotyczy złożoności obliczeniowej deterministycznych problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi. Na tle przeglądu dotychczasowych wyników przedstawiono nowe rezultaty w zakresie oceny złożoności obliczeniowej niektórych otwartych dotąd problemów szeregowania. W szczególności wykazano, że złożoność obliczeniowa problemów $F2|prec,p_j=1|C_{max}$ oraz $F2|prec,p_j=1|\bar{F}$ wynosi $O(n^2)$, natomiast złożoność problemu $F2|res111,p_j=1|\bar{F}$ wynosi $O(n)$.

1. Wstęp

Inspiracją do przeprowadzenia badań złożoności obliczeniowej, omawianej klasy problemów szeregowania zadań była analiza systemu bieżącego sterowania przebiegiem produkcji w zautomatyzowanej, dwustanowiskowej linii produkcyjnej. W konsekwencji skoncentrowano się na problemach deterministycznych dla dwustanowiskowego przepływowego systemu obsługi. Ograniczenie rozważań do struktury dwustanowiskowego systemu przepływowego wydaje się w tym kontekście naturalne, natomiast wybór modelu deterministycznego wymaga krótkiego uzasadnienia. Otóż w bieżącym sterowaniu przebiegiem produkcji w rozważanym systemie obróbczym wyróżniono dwie fazy:

- fazę budowy wzorca przebiegu produkcji, obejmującą ustalenie kolejności wykonywania zadań na obu stanowiskach oraz rozdział ograniczonych zasobów,
- fazę regulacji, czyli reagowania na odchylenia rzeczywistego przebiegu produkcji od aktualnego wzorca i/lub budowy nowego wzorca.

Faza budowy wzorca, czyli ustalenia uszeregowania zadań, realizowana jest z niewielkim wyprzedzeniem w stosunku do momentu rozpoczęcia wykonywania zadań objętych tym wzorcem. Regulację pracy systemu realizuje się w czasie rzeczywistym (poza przebudową wzorca). Wzorzec tworzy się zatem dla krótkich okresów czasu, co istotnie zmniejsza niepewność związaną ze stanem systemu (awarie maszyn, brak materiału lub narzędzia itp.). Równocześnie dane dotyczące momentów pojawiania się zadań w systemie oraz czasów wykonywania na poszczególnych stanowiskach są znane (wynikają z dokumentacji technologicznej i produkcyjnej). Takie rozwiązanie występuje w wielu systemach produkcyjnych w przemyśle maszynowym na różnych poziomach automatyzacji wytwarzania. Z punktu widzenia problemów szeregowania zadań można zatem system produkcyjny traktować jako deterministyczny.

2. Zadania niezależne

W dalszej części pracy dla oznaczenia problemów szeregowania zadań posłużono się notacją $\alpha|\beta|\gamma$, podaną m.in. w [1].

Rozważa się system obsługi złożony z dwóch różnych stanowisk M_1 i M_2 oraz zbiór zadań $\mathcal{J}=\{T_1, T_2, \ldots, T_n\}$. Każde zadanie $T_j \in \mathcal{J}$ składa się z dwóch operacji O_{j1} i O_{j2} o czasach wykonywania odpowiednio p_{j1}, p_{j2}, przy czym operacja O_{j1} jest wykonywana na stanowisku M_1, a operacja O_{j2} na stanowisku M_2. Ponadto operacja O_{j1} zadania T_j musi być zakończona przed rozpoczęciem operacji O_{j2} zadania T_j.

Spośród kryteriów optymalizacji uszeregowania wybrano trzy: długość uszeregowania (C_{max}), średni czas przebywania zadania w systemie ($\bar{F}$) oraz maksymalne opóźnienie (L_{max}), jako najczęściej stosowane dla oceny wzorca sterowania w systemach produkcyjnych.

Ponadto można wprowadzić ograniczenie polegające na tym, że zadanie T_j jest dostępne w systemie dopiero w chwili r_j, czyli operacja O_{j1} nie może być uszeregowana wcześniej niż w chwili r_j.

W poniższych zestawieniach pominięto problemy szeregowania zadań podzielnych, jako że w systemach obróbczych przerwanie wykonywania operacji w dowolnej chwili jest jak dotąd na ogół niemożliwe.

W tabeli 1 zestawiono wyniki dotyczące złożoności obliczeniowej problemów szeregowania zadań niezależnych w dwustanowiskowym przepływowym sys emie obsługi.

Tabela 1.

Złożoność obliczeniowa problemów szeregowania zadań niezależnych w dwustanowiskowym przepływowym systemie obsługi.

Lp.	Momenty gotowości	Czasy wykonywania	Kryterium	Złożoność obliczeniowa	Literatura
1			C_{max}	O(nlogn)	[8]
2	r_j		C_{max}	silnie NP-trudny	[7]
3			$\bar{F}$	silnie NP-trudny	[7]
4			L_{max}	silnie NP-trudny	[11]
5	r_j	$p_j=1$	$\bar{F}$	$O(n^3)$	[10]
6	r_j	$p_j=1$	L_{max}	$O(n^3)$	[10]

Jak widać z powyższego zestawienia, algorytm dokładny przy dowolnych czasach wykonywania zadań istnieje tylko dla kryterium C_{max}. Jest to algorytm Johnsona.

Przy założeniu jednostkowych czasów wykonywania (tj. $\forall j\ p_{j1}=p_{j2}=1$) zarówno dla kryterium średniego czasu przebywania zadań w systemie, jak i maksymalnego opóźnienia istnieją algorytmy dokładne o złożoności wielomianowej nawet przy dowolnych momentach gotowości zadań r_j.

3. Zadania zależne

Naturalnymi ograniczeniami przy tworzeniu wzorca sterowania przebiegiem produkcji są ograniczenia nałożone na kolejność wykonywania zadań i operacji, wynikające z planu procesu technologicznego. Stwierdzenie, że zadanie T_j poprzedza zadanie T_i ($T_j \{ T_i$), oznacza, że ostatnia operacja zadania T_j musi być zakończona zanim rozpocznie się pierwsza operacja zadania T_i. W naszym systemie kolejność operacji w ramach jednego zadania jest stała i wynika ze struktury przepływowego systemu obsługi, zgodnie z opisem przedstawionym w rozdziale drugim.

Ograniczenia kolejnościowe przedstawia się najczęściej w postaci grafu skierowanego i acyklicznego (dag). Zależnie od typu grafu wyróżnimy trzy typy ograniczeń kolejnościowych: łańcuch, drzewo i dowolne ograniczenia kolejnościowe.

Zestawienie wyników dotyczących złożoności obliczeniowej problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi przy ograniczeniach kolejnościowych zawarto w tabeli 2.

Tabela 2.

Złożoność obliczeniowa problemów szeregowania zadań zależnych w dwustanowiskowym przepływowym systemie obsługi.

Lp.	Typ ograniczeń	Czasy obsługi	Momenty gotowości	Kryterium	Złożoność obliczeniowa	Literatura
1	łańcuch			C_{max}	silnie NP-trudny	[11]
2	drzewo	$p_j=1$		C_{max}	$O(n)$	[10]
3	drzewo	$p_j=1$		$\bar{F}$	$O(n)$	[10]
4	dowolne	$p_j=1$		C_{max}	$O(n^2)$	[9]
5	dowolne	$p_j=1$		$\bar{F}$	$O(n^2)$	[9]
6.	drzewo	$p_j=1$	r_j	C_{max}	?	
7	drzewo	$p_j=1$	r_j	$\bar{F}$	?	
8	dowolne	$p_j=1$	r_j	L_{max}	?	

Jak widać, wprowadzenie ograniczeń kolejnościowych, nawet typu łańcuch, powoduje, że problem szeregowania zadań o dowolnych czasach wykonywania w celu minimalizacji C_{max} staje się silnie NP-trudny. Dla jednostkowych czasów wykonywania istnieją natomiast algorytmy dokładne o złożoności $O(n^2)$, nawet przy dowolnych ograniczeniach kolejnościowych, zarówno dla C_{max}, jak

2. Zadania niezależne

W dalszej części pracy dla oznaczenia problemów szeregowania zadań posłużono się notacją $\alpha|\beta|\gamma$, podaną m.in. w [1].

Rozważa się system obsługi złożony z dwóch różnych stanowisk M_1 i M_2 oraz zbiór zadań $\mathcal{J}=\{T_1, T_2, \ldots, T_n\}$. Każde zadanie $T_j \in \mathcal{J}$ składa się z dwóch operacji O_{j1} i O_{j2} o czasach wykonywania odpowiednio p_{j1}, p_{j2}, przy czym operacja O_{j1} jest wykonywana na stanowisku M_1, a operacja O_{j2} na stanowisku M_2. Ponadto operacja O_{j1} zadania T_j musi być zakończona przed rozpoczęciem operacji O_{j2} zadania T_j.

Spośród kryteriów optymalizacji uszeregowania wybrano trzy: długość uszeregowania (C_{max}), średni czas przebywania zadania w systemie ($\bar{F}$) oraz maksymalne opóźnienie (L_{max}), jako najczęściej stosowane dla oceny wzorca sterowania w systemach produkcyjnych.

Ponadto można wprowadzić ograniczenie polegające na tym, że zadanie T_j jest dostępne w systemie dopiero w chwili r_j, czyli operacja O_{j1} nie może być uszeregowana wcześniej niż w chwili r_j.

W poniższych zestawieniach pominięto problemy szeregowania zadań podzielnych, jako że w systemach obróbczych przerwanie wykonywania operacji w dowolnej chwili jest jak dotąd na ogół niemożliwe.

W tabeli 1 zestawiono wyniki dotyczące złożoności obliczeniowej problemów szeregowania zadań niezależnych w dwustanowiskowym przepływowym systemie obsługi.

Tabela 1.

Złożoność obliczeniowa problemów szeregowania zadań niezależnych w dwustanowiskowym przepływowym systemie obsługi.

Lp.	Momenty gotowości	Czasy wykonywania	Kryterium	Złożoność obliczeniowa	Literatura
1			C_{max}	O(nlogn)	[8]
2	r_j		C_{max}	silnie NP-trudny	[7]
3			$\bar{F}$	silnie NP-trudny	[7]
4			L_{max}	silnie NP-trudny	[11]
5	r_j	$p_j=1$	$\bar{F}$	$O(n^3)$	[10]
6	r_j	$p_j=1$	L_{max}	$O(n^3)$	[10]

Jak widać z powyższego zestawienia, algorytm dokładny przy dowolnych czasach wykonywania zadań istnieje tylko dla kryterium C_{max}. Jest to algorytm Johnsona.

Przy założeniu jednostkowych czasów wykonywania (tj. $\forall j\ p_{j1}=p_{j2}=1$) zarówno dla kryterium średniego czasu przebywania zadań w systemie, jak i maksymalnego opóźnienia istnieją algorytmy dokładne o złożoności wielomianowej nawet przy dowolnych momentach gotowości zadań r_j.

3. Zadania zależne

Naturalnymi ograniczeniami przy tworzeniu wzorca sterowania przebiegiem produkcji są ograniczenia nałożone na kolejność wykonywania zadań i operacji, wynikające z planu procesu technologicznego. Stwierdzenie, że zadanie T_j poprzedza zadanie T_i ($T_j \{ T_i$), oznacza, że ostatnia operacja zadania T_j musi być zakończona, zanim rozpocznie się pierwsza operacja zadania T_i. W naszym systemie kolejność operacji w ramach jednego zadania jest stała i wynika ze struktury przepływowego systemu obsługi, zgodnie z opisem przedstawionym w rozdziale drugim.

Ograniczenia kolejnościowe przedstawia się najczęściej w postaci grafu skierowanego i acyklicznego (dag). Zależnie od typu grafu wyróżnimy trzy typy ograniczeń kolejnościowych: łańcuch, drzewo i dowolne ograniczenia kolejnościowe.

Zestawienie wyników dotyczących złożoności obliczeniowej problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi przy ograniczeniach kolejnościowych zawarto w tabeli 2.

Tabela 2.

Złożoność obliczeniowa problemów szeregowania zadań zależnych w dwustanowiskowym przepływowym systemie obsługi

Lp.	Typ ograniczeń	Czasy obsługi	Momenty gotowości	Kryterium	Złożoność obliczeniowa	Literatura
1	łańcuch			C_{max}	silnie NP-trudny	[11]
2	drzewo	$p_j=1$		C_{max}	$O(n)$	[10]
3	drzewo	$p_j=1$		$\bar{F}$	$O(n)$	[10]
4	dowolne	$p_j=1$		C_{max}	$O(n^2)$	[9]
5	dowolne	$p_j=1$		$\bar{F}$	$O(n^2)$	[9]
6	drzewo	$p_j=1$	r_j	C_{max}	?	
7	drzewo	$p_j=1$	r_j	$\bar{F}$	?	
8	dowolne	$p_j=1$	r_j	L_{max}	?	

Jak widać, wprowadzenie ograniczeń kolejnościowych, nawet typu łańcuch, powoduje, że problem szeregowania zadań o dowolnych czasach wykonywania w celu minimalizacji C_{max} staje się silnie NP-trudny. Dla jednostkowych czasów wykonywania istnieją natomiast algorytmy dokładne o złożoności $O(n^2)$, nawet przy dowolnych ograniczeniach kolejnościowych, zarówno dla C_{max}, jak

i $\bar{F}$ [9]. Oba kryteria minimalizuje algorytm szeregowania listowego (Algorytm 1), dla którego lista tworzona jest zgodnie z regułą przydzielania etykiet zastosowaną przez Coffmana i Grahama przy szeregowaniu zadań o jednostkowych czasach wykonywania i dowolnych ograniczeniach kolejnościowych na dwóch maszynach identycznych ($P2|prec,p_j=1|C_{max}$) [4]. Dla problemu $P2|prec,p_j=1|C_{max}$ został znaleziony przez Gabowa i Tarjana [5,6] dokładny algorytm liniowy. Nadal otwarte pozostają problemy określenia złożoności obliczeniowej przy dowolnych czasach przybywania zadań do systemu.

Można wykazać następujące twierdzenie [9]:

TWIERDZENIE 1

Niech C_k^A oznacza czas zakończenia zadania T_k^A w uszeregowaniu S^A, otrzymanym zgodnie z algorytmem listowym opartym na etykietach Coffmana-Grahama. Dla każdego uszeregowania dopuszczalnego S oraz dla każdego $k = 1,2,\ldots,n$ zachodzi nierówność: $C_k^A \le C_k$.

W dowodzie twierdzenia posłużymy się następującym lematem, którego prosty dowód zostanie pominięty:

LEMAT 1

Załóżmy, że etykiety zostały przydzielone wszystkim zadaniom w zbiorze $\mathcal{T}$ zgodnie z regułą Coffmana-Grahama. Oznaczmy przez $\mathcal{T}'$ zbiór zadań, które nie mają następników w $\mathcal{T}$. Istnieje przyporządkowanie etykiet w zbiorze $\mathcal{T}\setminus\mathcal{T}'$ zgodne z regułą etykietowania i takie, że kolejność zadań na liście pozostaje bez zmian.

Dowód Twierdzenia 1

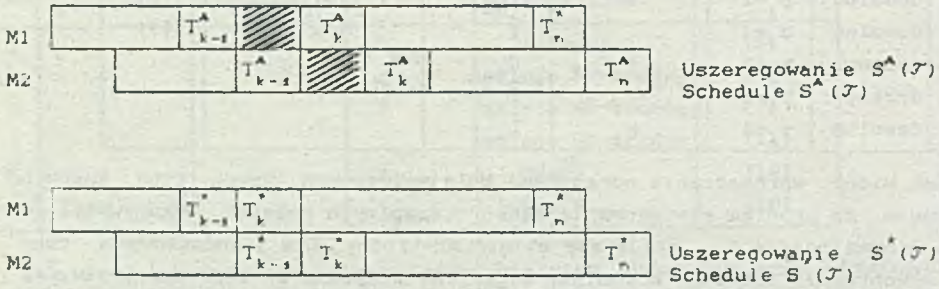
Twierdzenie zostanie udowodnione nie wprost. Niech $S^*(\mathcal{T})$ oznacza optymalne uszeregowanie zbioru $\mathcal{T}$. Niech $\mathcal{T}$ będzie najmniejszym zbiorem, dla którego uszeregowanie $S^A(\mathcal{T})$ zawiera zadanie T_k^A, takie że

$$C_k^A > C_k^*. \tag{1}$$

Załóżmy ponadto, że $C_k^A \le C_k^*$ (można przyjąć takie założenie, gdyż $C_1^A = C_1^*$).

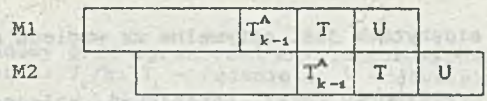
Należy rozważyć dwa przypadki: $k<n$ oraz $k=n$.

(i) Przypuśćmy, że $k<n$ (rys.1.)



Rys.1. Uszeregowania $S^A(\mathcal{T})$ i $S^*(\mathcal{T})$ $(k<n)$.
Fig.1. Schedules $S^A(\mathcal{T})$ and $S^*(\mathcal{T})$ $(k<n)$.

Usuwajmy ze zbioru $\mathcal{T}$ wszystkie zadania bez następników tak długo, aż zostaną usunięte wszystkie następniki zadania T^A_{k-1}. Oznaczmy usunięty zbiór przez $\mathcal{T}'$. Oczywiście $\mathcal{T}'$ zawiera co najmniej 2 zadania ($k<n$). Z Lematu 1 wynika, że uszeregowanie $S^A(\mathcal{T}\setminus\mathcal{T}')$ można otrzymać szeregując przerwy w miejsce zadań ze zbioru $\mathcal{T}'$. Wszystkie zadania uszeregowane później niż T^A_{k-1} w $S^A(\mathcal{T})$ należą do zbioru $\mathcal{T}'$ i zadanie T^A_{k-1} poprzedza wszystkie zadania uszeregowane za nim. Z rysunku 1 wynika, że istnieje zadanie $T\in\mathcal{T}\setminus\mathcal{T}'$ i zadanie P uszeregowane później niż T^A_{k-1}, takie że P nie poprzedza T i usuwając T ze zbioru $\mathcal{T}\setminus\mathcal{T}'$ można otrzymać uszeregowanie o długości C^A_{k-2}. Z reguły przydzielania etykiet wynika natychmiast, że $e(T) < e(T^A_{k-1})$. Dodajmy teraz do zbioru $\mathcal{T}\setminus\mathcal{T}'$ zadanie U takie, że U jest następnikiem tylko tych zadań, które mają etykiety wyższe niż $e(T^A_{k-1})$. Przydzielmy zadaniu U etykietę $e(U) = 0$. Przydzielmy teraz zadaniom bez następników oraz zadaniom, których jedynym następnikiem jest U, te same etykiety co poprzednio. Jest to zgodne z przyjętą regułą, gdyż zadania nie poprzedzające U i tylko takie miały etykiety niższe niż $e(T_{k-1})$. Zauważmy, że przydzielając pozostałym zadaniom te same etykiety co poprzednio, również nie naruszamy przyjętej reguły etykietowania, gdyż ciągi następników tych zadań nie uległy zmianie. Uszeregowanie $S^{A1}(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ zgodne z algorytmem i uszeregowanie $S^C(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ skonstruowane przy założeniu istnienia zadania T przedstawiono na rys. 2. Znaleziono zbiór $\mathcal{T}\setminus\mathcal{T}'\cup\{U\}$ mniejszy niż $\mathcal{T}$ i uszeregowanie zgodne z algorytmem takie, że $C^{A1}_k > C^C_k$, co jest sprzeczne z założeniem, że $\mathcal{T}$ jest najmniejszym zbiorem zawierającym zadanie spełniające nierówność (1)



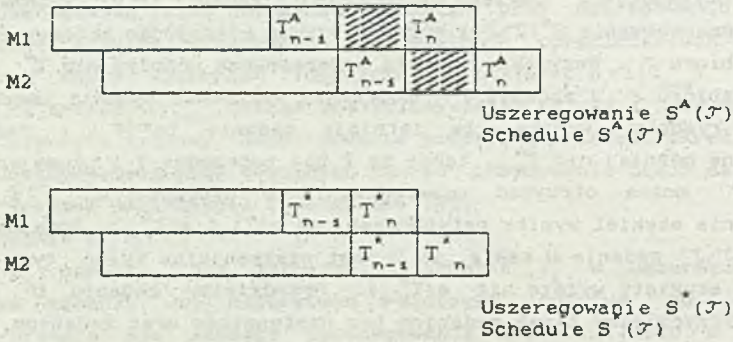
Uszeregowanie $S^C(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ skonstruowane w oparciu o założenie istnienia zadania T.

Schedule $S^C(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ constructed basing on the assumption of the existence of task T.

Uszeregowanie $S^{A1}(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ zgodne z algorytmem.
Schedule $S^{A1}(\mathcal{T}\setminus\mathcal{T}'\cup\{U\})$ obtained following the algorithm.

Rys.2. Uszeregowania zbioru $\mathcal{T}\setminus\mathcal{T}'\cup\{U\}$
Fig.2. Schedule of set $\mathcal{T}\setminus\mathcal{T}'\cup\{U\}$.

(ii) Załóżmy, że k=n (rys. 3).



Rys.3. Uszeregowania $S^A(\mathcal{T})$ i $S^*(\mathcal{T})$ (k=n).
Fig.3. Schedules $S^A(\mathcal{T})$ i $S^*(\mathcal{T})$ (k=n).

Łatwo zauważyć, że zadanie T^*_{n-1} nie ma następników w $\mathcal{T}$. Usuńmy zbiór $\mathcal{T}'$ wszystkich zadań bez następników z $\mathcal{T}$. Uszeregowanie zgodne z algorytmem można otrzymać przez zastąpienie zadań ze zbioru $\mathcal{T}'$ przerwami w $S^A(\mathcal{T})$ (zgodnie z Lematem 1). Stąd

$$C_{max}(S^A(\mathcal{T}\setminus\mathcal{T}')) = C^A_{n-1} > C^*_{n-2} \geq C_{max}(S^*(\mathcal{T}\setminus\mathcal{T}'))$$

Zatem znaleźliśmy zbiór $\mathcal{T}\setminus\mathcal{T}'$, mniejszy niż $\mathcal{T}$ i uszeregowanie $S^A(\mathcal{T}\setminus\mathcal{T}')$ zgodne z algorytmem, takie że $C^A_k > C^*_k$, co jest sprzeczne z założeniem.

WNIOSEK 1

Uszeregowanie zgodne z algorytmem jest optymalne ze względu na C_{max} i $\bar{F}$.

Dowód jest oczywisty.

Rysunek 4 przedstawia przykładowy graf ograniczeń kolejnościowych i uszeregowanie optymalne otrzymane dla tego grafu zgodnie z omawianym algorytmem.

4. Szeregowanie zadań przy ograniczeniach zasobowych

Ograniczenia zasobowe są również naturalne w przypadku modelowania przemysłowych systemów produkcyjnych. Takie ograniczone zasoby stanowią specjalne uchwyty, narzędzia, palety itp.

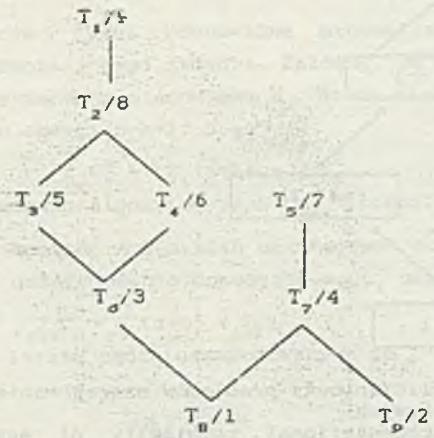
W teorii szeregowania zadań typ ograniczeń zasobowych definiowany jest jako trójka (λ,σ,ρ), gdzie:

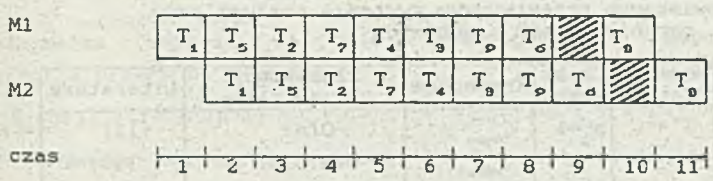
λ - oznacza liczbę rodzajów dodatkowych zasobów,

σ - oznacza ograniczenie liczby jednostek każdego zasobu,

ρ - oznacza maksymalne żądanie zasobowe.

Każdy z parametrów może mieć wartość dowolną (oznaczaną ·) lub ustaloną-k.





Rys.4.a. Przykładowy graf ograniczeń kolejnościowych G.
Oznaczenia: T_i/k: T_i - zadanie i, k - etykieta zadania i.

b. Uszeregowanie optymalne dla grafu G.

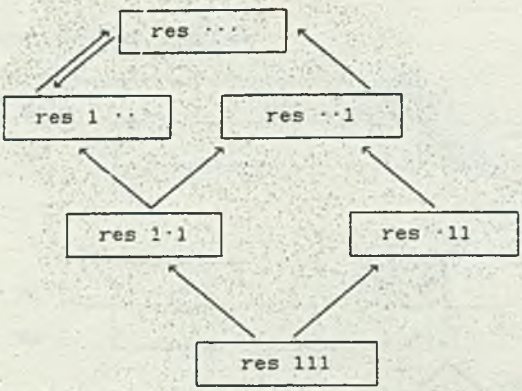
Fig.4.a. Graph of precedence constraints G - an example.
Notation: T_i/k: T_i - task i, k - label of task i.

b. Optimal schedule of graph G.

Rysunek 5 przedstawia zależność złożoności obliczeniowej problemów szeregowania zadań od typu ograniczeń zasobowych [1].

Zestawienie wyników dotyczących złożoności obliczeniowej problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi przy ograniczeniach zasobowych przedstawia tabela 3.

W przypadku ograniczeń zasobowych wielomianowe algorytmy dokładne istnieją dla najprostszych ograniczeń (typu res 111) i jednostkowych czasów wykonywania. Wyjątek stanowi $F2|res1\cdot\cdot|C_{max}$, gdzie algorytm o złożoności O(nlogn) znajduje uszeregowanie optymalne.



Rys.5. Zależność złożoności obliczeniowej problemów szeregowania zadań od typu ograniczeń zasobowych.
Fig.5. Dependence of the computational complexity of scheduling problems on the type of resource constraints.

Tabela 3

Złożoność obliczeniowa problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi przy ograniczeniach zasobowych

Lp.	Typ ogr. zasobowych	Momenty gotow.	Czasy wykon	Kryterium	Złożoność obliczeniowa	Literatura
1	res 111		$p_j=1$	C_{max}	O(n)	[3]
2	res 1··			C_{max}	O(nlogn)	[13]
3	res 2··		$p_j=1$	C_{max}	silnie NP-trudny	[14]
4	res 1··	r_j	$p_j=1$	C_{max}	silnie NP-trudny	[13]
5	res ·11		$p_j=1$	C_{max}	silnie NP-trudny	[11]
6	res 111			C_{max}	silnie NP-trudny	[3]
7	res 111		$p_j=1$	$\bar{F}$	O(n)	Stw.1
8	res 1··		$p_j=1$	$\bar{F}$	silnie NP-trudny	[14]
9	res 111		$p_j=1$	L_{max}	$O(n\log^2 n)$	[2]
10	res 1··		$p_j=1$	L_{max}	silnie NP-trudny	[12]

W [1] podano algorytm liniowy dla problemu $F2|res111,p_j=1|C_{max}$ i równocześnie wskazano $F2|res111,p_j=1|\bar{F}$ jako problem otwarty. Uszeregowanie, jakie znajduje wspomniany algorytm, jest optymalne również ze względu na średni czas przebywania zadania w systemie, co wykazano poniżej.

TWIERDZENIE 2

Złożoność obliczeniowa problemu $F2|res111,p_j=1|\bar{F}$ wynosi O(n).

Dowód

Minimalizacja $\bar{F}$ jest równoważna minimalizacji $\sum_{j=1}^{n} C_j$, gdzie C_j jest czasem zakończenia j-tego zadania. Załóżmy, że uszeregowanie wg algorytmu J^A zawiera k przerw na stanowisku M_1. Wtedy pierwsza przerwa na M_1 jest uszeregowana w chwili n-k+1. Stąd

$$\sum_{j=1}^{n} C_j^A = \sum_{j=1}^{n} (j+1) + \sum_{i=1}^{k} i$$

Uszeregowanie wg algorytmu jest C_{max}-minimalne, więc zawiera minimalną liczbę przerw spośród wszystkich uszeregowań dopuszczalnych. Wartość $\sum_{j=1}^{n} C_j$ dla dowolnego uszeregowania dopuszczalnego, zawierającego k przerw, wynosi

$$\sum_{j=1}^{n} C_j = \sum_{j=1}^{n} (j+1) + \sum_{i=1}^{k} n_i$$

gdzie n_i jest liczbą zadań uszeregowanych po i-tej przerwie na M_1. Łatwo zauważyć, że:

$$(n_1)_{min} = k$$
$$(n_2)_{min} = k-1$$
$$\dots\dots\dots\dots$$
$$(n_k)_{min} = 1$$

Zatem minimalna wartość $\sum_{j=1}^{n} C_j$ dla uszeregowania dopuszczalnego, zawierającego k przerw na M_1, wynosi

$$\sum_{j=1}^{n} C_j = \sum_{j=1}^{n} (j+1) + (k + (k-1) + \dots + 1) = \sum_{j=1}^{n} (j+1) + \sum_{i=1}^{k} i = \sum_{j=1}^{n} C_j^A$$

Z powyższej równości wynika, że wartość $\sum_{j=1}^{n} C_j^A$ jest minimalna, a zatem algorytm o złożoności O(n) znajduje uszeregowanie optymalne ze względu na $\bar{F}$ dla dwustanowiskowego przepływowego systemu obsługi o jednostkowych czasach wykonywania zadań i ograniczeniach zasobowych typu res 111.

5. Podsumowanie

W rzeczywistych systemach produkcyjnych ograniczenia kolejnościowe występują równocześnie z ograniczeniami zasobowymi. Problem $F2|res111, łańcuch, p_j=1|C_{max}$ jest jednak NP-trudny [1], a zatem NP-trudne są również problemy szeregowania zadań dla innych typów ograniczeń zasobowych i kolejnościowych. Wobec faktu, że większość algorytmów dokładnych wymaga założenia jednostkowych czasów wykonywania zadań, co w praktyce zdarza się niezwykle rzadko, pozostaje poszukiwanie heurystycznych metod tworzenia wzorca sterowania dla rozważanego systemu produkcyjnego.

Niezależnie, z teoretycznego punktu widzenia, interesująca jest ocena złożoności obliczeniowej, wymienionych w tabeli 2, otwartych problemów szeregowania zadań w dwustanowiskowym przepływowym systemie obsługi.

Ponadto przedmiotem dalszych badań jest poszukiwanie liniowego algorytmu

dla problemu $F2|prec,p_j=1|C_{max}$ analogicznego do proponowanego przez Gabowa i Tarjana [5,6] dla $P2|prec,p_j=1|C_{max}$.

LITERATURA

[1] Błażewicz J., Cellary W., Słowiński R., Węglarz J.: Scheduling under resource constraints - deterministic models.", Annals of Operations Research, Vol.7 (1986)

[2] Błażewicz J., Kubiak W., Szwarcfiter J.: "Scheduling unit-time jobs under resource constraints.", Annals of Op. Res. 16 (1986), 255-266

[3] Błażewicz J., Lenstra J.K., Rinnooy Kan A.H.G.: "Scheduling subject to resource constraints, classification and complexity.", Discrete Applied Mathematics 5, No.1,(1983), 11-24

[4] Coffman E.G.,Jr., Graham R. L.: "Optimal scheduling for two processor systems.", Acta Informatica 1(1972), 200-213

[5] Gabow H.N.: "An almost linear algorithm for two-processor scheduling.", J. Assoc. Comp. Math. 29(1982),766-780

[6] Gabow H.N., Tarjan R.E.: "A linear-time algorithm for a special case of disjoint set unions.", Proc. 15th Annual ACM Symp. Theory of Computing, 1983, 246-254

[7] Garey M.R., Johnson D.S., Sethi R.: "The complexity of flow shop and job shop scheduling", Mathematics of Operations Research 1, No.2, 1976

[8] Johnson S.M.: "Optimal two- and three-stage production schedules", Naval Research Logistic Quarterly 1,No.1,1954

[9] Józefowska J.: "Scheduling unit-length two-machine flow shop", w przyg.

[10] Lageweg B. - wynik niepublikowany

[11] Lenstra J.K., Rinnooy Kan A.H.G., Brucker P.: "Complexity of machine scheduling problems.", in. Discrete Math. 1(1977),343-362

[12] Rock H., - wynik niepublikowany

[13] Röck H.: "Some new results in flowshop scheduling", Zeitschrift für Op. Res. 28(1984), 1-1616

[14] Rock H.: "Scheduling unit task shops with resource constraints and excess usage costs, Report 83-21, Fachbereich Informatik, TU Berlin

Recenzent: Doc.dr h.inż.J.Klamka

Wpłynęło do Redakcji do 1990-04-30.

DETERMINISTIC PROBLEMS OF SCHEDULING TASKS IN A TWO-MACHINE FLOW-SHOP

S u m m a r y

A survey of computational complexity of deterministic problems of scheduling tasks in a two-machine flow-shop is presented. The survey is supplemented with new results. Namely, the polynomial complexity of problems $F2|prec,p_j=1|C_{max}$, $F2|prec,p_j=1|\bar{F}$ and $F2|res111,p_j=1|\bar{F}$ has been proved.

ДЕТЕРМИНИСТИЧЕСКИЕ ПРОБЛЕМЫ УПОРЯДОЧНЕНИЯ ЗАДАЧ В ДВУХПОСТОВОЙ ПРОТОЧНОЙ СИСТЕМЕ ОБСЛУЖИВАНИЯ

Р е з ю м е

В работе рассмотрены проблемы упорядочнения задач в двухстовой проточной системе обслуживания. На основании обзора результатов полученных прежде, представлены новые результаты в области оценки вычислительной сложности некоторых нерешенных до настоящего времени проблем упорядочения. Подробно изложено, что вычислительная сложность проблемы $F2|prec,p_j=1|C_{max}$, $F2|prec,p_j=1|\bar{F}$