ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: AUTOMATYKA z. 123 Nr kol. 1389

Jacek BŁAŻEWICZ, Maciej DROZDOWSKI, Piotr FORMANOWICZ
Politechnika Poznańska
Wiesław KUBIAK
Memorial University of Newfoundland

SZEREGOWANIE ZADAŃ W SYSTEMACH Z OGRANICZONĄ DOSTĘPNOŚCIĄ MASZYN

Streszczenie. W pracy przedstawiono wybrane problemy szeregowania zadań na maszynach z okresami niedostępności. Rozważane są zagadnienia dotyczące maszyn równoległych identycznych i dowolnych, a także przepływowego systemu obsługi. W przypadku maszyn równoległych omawiane problemy dotyczą zadań podzielnych bez ograniczeń kolejnościowych oraz z takimi ograniczeniami i kryterium C_{max} oraz L_{max}. Przedstawione są wyniki dotyczące złożoności obliczeniowej oraz algorytmy oparte na programowaniu liniowym. Dla problemów szeregowania w przepływowym systemie obsługi określono złożoność obliczeniową problemu szeregowania na dwóch maszynach z kryterium C_{max}. Ponadto przedstawiono własność tego problemu przydatną przy konstrukcji algorytmu podziału i ograniczeń oraz wyniki eksperymentu obliczeniowego.

SCHEDULING TASKS IN SYSTEMS WITH LIMITED MACHINE AVAILABILITY

Summary. The selected problems of scheduling tasks on machines with a limited availability are presented. Problems concerning parallel identical and unrelated machines and flow shop system are considered. In the case of parallel machines the problems presented are concerned with preemptive tasks with and without precedence constraints and C_{max} and L_{max} criteria. Computational complexity results and algorithms based on linear programming are presented. In the case of flow shop the computational complexity for two-machine problem with C_{max} criterion is analysed. Moreover, a property of this problem is presented which is useful for constructing a branch and bound algorithm and results of computational experiment are presented.

1. Wprowadzenie

W deterministycznej teorii szeregowania zadań przyjmowana jest na ogół ciągła dostępność maszyn. W rzeczywistych systemach produkcyjnych lub komputerowych założenie to nie zawsze jest spełnione. Niniejsza praca omawia problemy szeregowania zadań uwzględniające możliwość występowania okresów niedostępności maszyn. Niedostępność ta

wynikać może z wcześniejszego zaplanowania wykonania pewnych zadań, konieczności przeprowadzenia okresowych prac konserwacyjnych lub z nakładania się dwóch horyzontów czasowych w algorytmach z przesuwanym horyzontem. Ostatnia z wymienionych przyczyn niedostępności maszyn jest prawdopodobnie najistotniejsza ze względu na dynamiczną naturę problemów planowania produkcji. Dynamika ta oznacza, że dane wejściowe dla tego typu problemów są często uaktualniane. Zmiany w nich zachodzące powodują konieczność ustalenia nowego horyzontu czasowego dla planowanego procesu produkcyjnego. W sytuacji tej należy jednak wcześniej zaplanowaną część procesu uważać za ustaloną, ze względu na ograniczenia technologiczne utrudniające lub uniemożliwiające dokonanie zmian. Powoduje to wystąpienie ograniczonej dostępności maszyn dla zadań, których wykonanie nie zostało jeszcze zaplanowane.

W pracy przedstawione zostały wybrane problemy szeregowania zadań na maszynach z ograniczoną dostępnością. Dla oznaczenia tego typu problemów w pracy [4] wprowadzono do standardowej notacji $\alpha|\beta|\gamma$ symbol h_{ik} w jej pierwszym polu na oznaczenie możliwości występowania dowolnej liczby okresów niedostępności na każdej z maszyn. Indeks i określa maszynę, na której wspomniane okresy mogą występować, a indeks k jest liczbą tych okresów. W problemach szeregowania na maszynach z ograniczoną dostępnością rozpatruje się niekiedy zadania wznawialne (*resumable*) [5, 4], tj. takie, których wykonywanie może być kontynuowane po okresie niedostępności maszyny. Dla oznaczenia tego typu zadań zastosowano symbol *rs* w drugim polu standardowej notacji [4]. Należy odróżnić ten typ zadań od zazwyczaj występujących w teorii szeregowania zadań podzielnych, gdyż jedynym miejscem, w którym zadanie wznawialne może zostać przerwane, jest początek okresu niedostępności. Z drugiej strony zadania te różnią się również od zadań niepodzielnych, gdyż te ostatnie przerwane przez wystąpienie okresu niedostępności, po jego zakończeniu muszą być wykonywane od początku.

W kolejnych rozdziałach przedstawione zostaną wyniki dotyczące szeregowania zadań na maszynach równoległych oraz dedykowanych.

2. Szeregowanie zadań na maszynach równoległych z okresami niedostępności

Problemy szeregowania zadań na równoległych maszynach identycznych z ograniczoną dostępnością były badane w kilku pracach. Jednymi z pierwszych, w których poruszony został

ten problem, były prace Schmidta z 1984 i 1988 roku [8, 9]. W pierwszej z nich podany został algorytm rozwiązujący problem szeregowania zadań podzielnych na maszynach identycznych przy kryterium $C_{\max}$, z okresami niedostępności tworzącymi pewien określony wzorzec (*staircase pattern*). Algorytm ten można jednak zastosować do rozwiązania problemu z dowolnym układem okresów niedostępności, gdyż każdy z nich można sprowadzić do wspomnianego wzorca. Łatwo można również pokazać, że algorytm zaproponowany przez Schmidta może być wykorzystany do rozwiązania problemu, w którym występują ograniczenia kolejnościowe typu łańcuch. W tym celu wystarczy każdy z łańcuchów zadań traktować jako jedno podzielne zadanie [1]. W drugiej ze wspomnianych prac algorytm ten został dostosowany do szeregowania zadań z liniami krytycznymi. Inne modele były również badane w pracach [6, 7]. W pracy [1] zaproponowany został algorytm rozwiązujący problem $P, h_{ik} \mid pmtn, chains, r_j \mid L_{\max}$. Algorytm ten oparty jest na sekwencji problemów programowania liniowego. W tej samej pracy pokazane zostało, że problem $P, h_{ik} \mid pmtn, intree \mid C_{\max}$ jest silnie NP-trudny.

Prawie wszystkie prace dotyczące szeregowania zadań na maszynach z ograniczoną dostępnością dotyczą problemów z równoległymi maszynami identycznymi. Problemy z maszynami dowolnymi zostały zbadane w niewielkim stopniu. Poniżej przedstawione zostaną dwa algorytmy rozwiązujące takie problemy z zadaniami podzielnymi przy kryteriach $C_{\max}$ i $L_{\max}$. Oba oparte są na programowaniu liniowym. Przed ich zaprezentowaniem konieczne jest wprowadzenie niezbędnych oznaczeń:

n - liczba zadań

m - liczba maszyn

r_j - moment gotowości zadania j

d_j - linia krytyczna zadania j

p_{ij} - czas przetwarzania zadania j na maszynie i

$0 = t_0 < t_1 < \ldots < t_q$ - lista różnych momentów, w których liczba dostępnych maszyn zmienia się

$0 = e_0 \le e_1 \le \ldots \le e_K$ - lista różnych momentów w zbiorze $\{r_j, d_j : j = 1, \ldots, n\} \cup \{t_j : j = 1, \ldots, q\}$

$A_j = \{k : e_k \le d_j\}$

$B_k = \{j : d_j \geq e_k\}$

x_{ij}^k - część zadania j przetwarzana na maszynie i w przedziale $[e_{k-1}, e_k]$

$x_{ij}^k \in [0,1]$

$x_{ij}^k = 0$ dla wszystkich k, dla których maszyna i jest niedostępna w przedziale $[e_{k-1}, e_k]$

$$a_i = \begin{cases} 1 & \text{jeśli } e_i \text{ jest linią krytyczną} \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

Problem $R, h_{ik} \mid pmtn \mid C_{\max}$ może zostać rozwiązany za pomocą zmodyfikowanej metody dwufazowej [1, 2]. W pierwszej fazie należy rozwiązać następujący problem programowania liniowego:

zminimalizować

$$C_{\max} \tag{1}$$

przy ograniczeniach

$$\sum_{i=1}^{m} p_{ij} x_{ij}^k \leq t_k - t_{k-1} \qquad j = 1,\ldots,n; k = 1,\ldots,K \tag{2}$$

$$\sum_{j=1}^{n} p_{ij} x_{ij}^k \leq t_k - t_{k-1} \qquad i = 1,\ldots,m; k = 1,\ldots,K \tag{3}$$

$$\sum_{i=1}^{m} \sum_{k=1}^{K} x_{ij}^k = 1 \qquad j = 1,\ldots,n \tag{4}$$

W wyniku rozwiązania powyższego problemu programowania liniowego otrzymuje się optymalny przydział zadań do poszczególnych przedziałów $[t_{k-1}, t_k]$, $k = 1,\ldots,K$, w których liczba dostępnych procesorów nie zmienia się. Optymalne uszeregowanie otrzymywane jest w wyniku zastosowania drugiej fazy metody do każdego z przedziałów.

Problem $R, h_{ik} \mid r_j, pmtn \mid L_{\max}$ również może zostać rozwiązany przy zastosowaniu odmiany metody dwufazowej (por. [2]). W pierwszej fazie stosuje się procedurę przeszukiwania binarnego w celu minimalizacji maksymalnego opóźnienia. W każdym kroku tej procedury należy rozwiązać problem programowania liniowego dla danego przedziału zmienności wartości $L_{\max}$ takiego, dla którego liczba dostępnych maszyn i sekwencja zdarzeń określających przedziały nie zmienia się.

Zmiany wartości $L_{\max}$ mogą powodować zmiany sekwencji momentów e_i, w sytuacji gdy dla pewnego zadania $d_i + L_{\max}$ staje się równe jednemu z momentów gotowości bądź momentowi początkowemu, lub końcowemu jednego z okresów niedostępności. Liczba przedziałów $[L_{l-1}, L_l]$ wartości $L_{\max}$, dla których sekwencja momentów e_i nie zmienia się, wynosi najwyżej $(n+q)n$. Można w związku z tym sformułować następujący problem programowania liniowego:

zminimalizować

$$L_{\max} \tag{5}$$

przy ograniczeniach

$$\sum_{i=1}^{m} p_{ij} x_{ij}^{k} \le e_k - e_{k-1} + L_{\max}(a_k - a_{k-1}) \qquad j = 1,\ldots,n;\ k \in A_j \setminus \{0\} \tag{6}$$

$$\sum_{j \in B_k} p_{ij} x_{ij}^{k} \le e_k - e_{k-1} + L_{\max}(a_k - a_{k-1}) \qquad i = 1,\ldots,m;\ k = 1,\ldots,K \tag{7}$$

$$L_{l-1} \le L_{\max} < L_l \tag{8}$$

$$\sum_{i=1}^{m} \sum_{k \in A_j \setminus \{0\}} x_{ij}^{k} = 1 \qquad j = 1,\ldots,n \tag{9}$$

W powyżej sformułowanym problemie programowania liniowego nierówności (6) i (7) zapewniają, że suma części zadań przetwarzanych w każdym z przedziałów nie przekracza jego długości. Nierówności (8) gwarantują, że sekwencja momentów e_i nie zmienia się, a równania (9) zapewniają, że wszystkie zadania zostaną w całości wykonane. Jeśli rozwiązanie dopuszczalne istnieje, należy sprawdzić mniejszą wartość $L_{\max}$, jeśli nie istnieje, większą.

Druga faza metody jest stosowana do ostatniego z problemów programowania liniowego w sekwencji i daje optymalne uszeregowanie minimalizujące maksymalne opóźnienie.

3. Szeregowanie zadań wznawialnych w przepływowym systemie obsługi

Przepływowym systemem obsługi z ograniczoną dostępnością maszyn jako pierwszy zajął się Lee w swojej pracy z 1995 roku [5]. Rozpatrywane były w niej problemy dwumaszynowe z jednym okresem niedostępności na jednej z maszyn. We wspomnianej pracy wykazano, że problem szeregowania zadań w takim systemie przy kryterium $C_{\max}$ jest NP-trudny zarówno dla zadań niewznawialnych (*nonresumable*), jak i wznawialnych (*resumable*).

Podano tam również pseudowielomianowy algorytm oparty na programowaniu dynamicznym, charakteryzujący się dużą złożonością obliczeniową, silnie uzależnioną od długości uszeregowania.

W dalszej części tego punktu podane zostaną wyniki dotyczące przepływowego systemu obsługi z dowolną liczbą okresów niedostępności na każdej z maszyn i zadań wznawialnych.

Twierdzenie 1

Problemy $F2, h_{1k} \mid rs \mid C_{\max}$ i $F2, h_{2k} \mid rs \mid C_{\max}$ są silnie NP-trudne [4].

Ponadto prawdziwe są również następujące dwa twierdzenia [4]:

Twierdzenie 2

Rozwiązanie problemu $F2, h_{1k} \mid rs \mid C_{\max}$ z błędem względnym równym w najgorszym przypadku 2 może zostać znalezione w czasie $O(n \log n)$.

Twierdzenie 3

Jeżeli $P \neq NP$, to nie istnieje wielomianowy algorytm dla problemu $F2, h_{2k} \mid rs \mid C_{\max}$ dający rozwiązanie ze skończonym błędem względnym w najgorszym przypadku.

Aby przekonać się o prawdziwości twierdzenia 2, wystarczy przeanalizować działanie algorytmu Johnsona [3, 4] dla problemu z niedostępnością maszyn. Twierdzenie 3 jest prawdziwe również dla problemu z dwoma okresami niedostępności, jeśli drugi z nich znajduje się na drugiej maszynie.

Przytoczony w twierdzeniu 1 wynik dotyczący złożoności obliczeniowej każe szukać rozwiązań omawianego problemu za pomocą algorymów przybliżonych lub wyliczeniowych. Okazuje się, że omawiany problem posiada pewną właściwość bardzo przydatną przy konstrukcji algorytmu podziału i ograniczeń. Polega ona na tym, iż wśród rozwiązań optymalnych znajduje się przynajmniej jedno takie, w którym zadania uszeregowane w przedziałach wyznaczonych przez okresy niedostępności maszyn są uszeregowane zgodnie z regułą Johnsona. Zmniejsza to znacząco liczbę rozwiązań, które należy przejrzeć, gdyż nie jest

poszukiwana optymalna permutacja zadań, lecz optymalny ich przydział do wspomnianych przedziałów „między" okresami niedostępności.

Poniżej przedstawione są wyniki eksperymentu obliczeniowego ukazujące efektywność zaprojektowanego algorytmu podziału i ograniczeń wykorzystującego wspomnianą właściwość problemu. Obliczenia przeprowadzone zostały dla problemów z okresami niedostępności odpowiednio tylko na pierwszej maszynie, tylko na drugiej oraz na obu maszynach. Dla każdej pary [*liczba zadań, liczba okresów niedostępności*] stworzonych zostało 10 instancji. Długości operacji zadań oraz długości okresów niedostępności maszyn wygenerowane zostały losowo z rozkładem jednorodnym z przedziału $[1,1000]$. Obliczenia zostały przeprowadzone na komputerze HP Apollo 9000 Series 700. Czas obliczeń dla każdej instancji został ograniczony do 1000 sekund. W poniższych tabelach przedstawione są dla każdej pary [*liczba zadań, liczba okresów niedostępności*] liczby instancji, które zostały rozwiązane w zadanym limicie czasu.

Tablica 1

Liczba rozwiązanych instancji dla problemu z okresami niedostępności na pierwszej maszynie

liczba zadań	liczba okresów niedostępności									
	1	2	3	4	5	6	7	8	9	10
10	10	10	10	10	10	10	10	10	10	9
25	10	10	9	7	5	4	1	2	1	0
40	10	9	6	3	1	1	0	0	1	0
55	10	5	4	3	3	2	1	1	1	1
70	10	4	4	3	3	1	1	1	0	1
85	8	5	2	2	1	3	2	1	1	1
100	7	5	3	4	4	5	2	2	1	1

Tablica 2

Liczba rozwiązanych instancji dla problemu z okresami niedostępności na drugiej maszynie

liczba zadań	liczba okresów niedostępności									
	1	2	3	4	5	6	7	8	9	10
10	10	10	10	10	10	10	10	10	10	9
25	10	10	9	9	8	9	9	9	9	10
40	10	9	7	7	7	9	8	9	8	8
55	10	8	7	7	8	8	9	8	9	8
70	10	6	7	5	8	9	8	7	8	9
85	8	6	5	6	7	8	9	9	10	9
100	7	5	6	5	8	9	7	9	10	8

Tablica 3

Liczba rozwiązanych instancji dla problemu z okresami niedostępności na obu maszynach

liczba zadań	liczba okresów niedostępności									
	1	2	3	4	5	6	7	8	9	10
10	10	10	10	10	10	10	10	10	10	10
25	10	10	9	7	8	5	4	4	4	4
40	10	8	6	4	6	6	4	5	2	6
55	10	7	6	5	5	4	6	3	4	4
70	9	5	6	5	6	6	5	5	5	5
85	6	5	3	6	5	5	5	6	3	6
100	9	5	5	4	6	7	4	4	6	3

Eksperyment ten pokazał, że liczba instancji, które zostały rozwiązane w zadanym limicie czasu, była mniejsza dla problemów z okresami niedostępności tylko na pierwszej maszynie niż dla problemów z niedostępnością drugiej maszyny. Zjawisko to można wytłumaczyć przez asymetrię maszyn. Okres niedostępności na pierwszej z nich może zmniejszyć liczbę zadań znajdujących się w buforze między maszynami, co może doprowadzić do przestoju na drugiej maszynie. Z drugiej strony, żaden okres niedostępności na drugiej maszynie nie może spowodować przestoju na maszynie pierwszej.

LITERATURA

1. Błażewicz J., Drozdowski M., Formanowicz P., Kubiak W., Schmidt G., Scheduling preemptable tasks on parallel processors with limited availability. Wysłane do publikacji.
2. Błażewicz J., Ecker K., Pesch E., Schmidt G., Węglarz J., Scheduling Computer and Manufacturing Processes, Springer, Berlin, 1996.
3. Johnson S. M., Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1, 1954, 61-68.
4. Kubiak W., Błażewicz J., Formanowicz P., Schmidt G. Flow shops with limited machine availability. Wysłane do publikacji.
5. Lee C.-Y., Machine Scheduling with an Availability Constraint. Research report 95-3, Department of Industrial & Systems Engineering, University of Florida.
6. Lee C.-Y., Parallel machine scheduling with non-simultaneous machine available time. Discrete Applied Mathematics, 30, 1991, 53-61.
7. Liu Z., Sanlaville E., Preemptive scheduling with variable profile, precedence constraints and due dates. Discrete Applied Mathematics, 58, 1995, 253-280.
8. Schmidt G., Scheduling on Semi-Identical Processors. ZOR - Zeitschrift für Operations Research, Vol. 28, 1984, 153-162.

9. Schmidt G., Scheduling Independent Tasks with Deadlines on Semi-identical Processors. Journal of Operational Research Society, Vol. 39, No. 3, 271-277.

Recenzent: Prof.dr hab.inż. Marek Kubale

Abstract

In the paper selected problems of scheduling on machines with a limited availability are presented. The limited availability of machines may result from preschedules, preventive maintenance activities, or the overlap of two consecutive time horizons in the rolling horizon planning algorithm. The rolling horizons are used mainly because most of the real world production planning problems are dynamic. This means that the input data are being frequently updated. A natural approach to handling this dynamic is to trigger a new planning horizon when the changes in the data justify it. However, due to many constraints, as for example process preparation, it is necessary to take earlier plans as fixed which obviously limits availability of machines for any subsequent schedules.

The presented problems concern parallel identical and unrelated machines and flow shops. Two algorithms based on linear programming are presented for problems with unrelated machines and C_{max} and L_{max} criteria. Moreover there is indicated that two-machine flow shop with one unavailability period is NP-hard for the C_{max} criterion, while an arbitrary number of such periods results in NP-hardness in the strong sense. For the problem with unavailability periods on the first machine there exists a simple polynomial-time heuristic with error bound equal to 2. On the other hand for the problem with limited availability of the second machine there is no polynomial-time heuristic with finite error bound unless $P = NP$. Finally, a useful property of the problem which may be used for constructing a branch and bound algorithm is described and results of computational experiments are shown.