ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2006
Seria: AUTOMATYKA z. 143 Nr kol. 1726

Adam NADOLSKI
Politechnika Gdańska

ZŁOŻONOŚĆ OBLICZENIOWA PROBLEMU SZEREGOWANIA ZADAŃ W CYLINDRYCZNYM SYSTEMIE PRZEPŁYWOWYM

Streszczenie. W pracy rozważamy złożoność obliczeniową szeregowania zadań w cylindrycznym systemie przepływowym. Konstruujemy algorytm wielomianowy dla problemu dwumaszynowego oraz wykazujemy, że problem staje się NP-trudny przy szeregowaniu na trzech maszynach oraz na dwóch maszynach przy wymuszeniu braku obustronnych przestojów.

COMPUTATIONAL COMPLEXITY OF THE PROBLEM OF SCHEDULING IN CYLINDRICAL FLOW SHOP

Summary. We consider the scheduling problem in a cylindrical flow shop to minimize the cycle time. We provide a polynomial time algorithm in the case of two processors and prove that the problem becomes NP-hard in the case of three processors. Moreover, we show that scheduling with no-wait and no-idle constraints is NP-hard in the case of two processors.

1. Wprowadzenie

System przepływowy jest jednym z podstawowych modeli szeregowania zadań. Składa się na niego zbiór m procesorów, które będziemy oznaczać przez P_1, ..., P_m oraz zbiór n zadań oznaczanych symbolami J_1, ..., J_n. Każde z zadań składa się z operacji, z których każda jest przypisana do ustalonego procesora. Operację zadania J_i, która ma być wykonywana na procesorze P_j, będziemy oznaczać symbolem T_{ij}, zaś czas jej przetwarzania symbolem t_{ij}. W obrębie każdego zadania, operacje muszą być przetwarzane w kolejności zgodnej z numeracją procesorów. Mówiąc precyzyjnie, operacja T_{is} musi być wykonana przed operacją T_{ir} dla wszystkich $s < r$.

W naszych rozważaniach będziemy przyjmować, że praca odbywa się w *ruchu ciągłym*. Oznacza to, że cykl pracy systemu jest wielokrotnie powtarzany, a pomiędzy poszczególnymi wykonaniami cykli nie ma przerw. W takim przypadku będziemy uznawać za dopuszczalne, aby procesor, który zakończył pracę nad i-tym zestawem, przeszedł do $(i+1)$-szego zestawu, nawet jeśli pozostałe procesory

przetwarzają wciąż zadania z i-tego cyklu. Przez T_{ij}^k będziemy oznaczali operację T_{ij} w k-tym cyklu pracy oraz przez x_{ij}^k czas rozpoczęcia wykonywania operacji T_{ij}^k. Przyjmujemy uproszczenie, polegające na założeniu, iż harmonogram jest *stabilny*, tzn. że istnieje stała μ, taka że $x_{ij}^{r+1} - x_{ij}^r = \mu$ dla wszystkich operacji T_{ij} oraz dowolnego r. Takie założenie wymusza, aby każdy cykl pracy był realizowany według tego samego schematu przesuniętego w czasie o odpowiednią wielokrotność μ. Sytuacja ta niesie za sobą wiele korzyści [6]. Harmonogram stabilny prowadzi do stałego obciążenia produkcyjnego przy, w przybliżeniu, stałych dostawach materiału i przepływie części. Co więcej, wystarczy wyznaczyć jedynie uszeregowanie dla jednego zestawu, dla pozostałych harmonogram otrzymujemy wykonując kopie harmonogramu dla pierwszego zestawu przesunięte o wielokrotność μ. Liczbę μ będziemy określać jako *długość cyklu CT* (ang. *cycle time*). Parametr ten, będący kryterium optymalizacyjnym w naszych rozważaniach, wyraża, z jaką częstością realizowane są kolejne zestawy zadań.

W pracy będziemy rozpatrywać tzw. cylindryczny wariant pracy cyklicznej. Występuje on w takich sytuacjach jak przydział zasobów ludzkich i maszyn przy produkcji wielozmianowej. W tym przypadku przystąpienie do przetwarzania operacji $(r+1)$ zestawu danego zadania wymaga uprzedniego ukończenia wszystkich operacji tego zadania z zestawu poprzedniego.

W notacji trójpolowej system przepływowy jest oznaczany symbolem F w polu α. W niniejszej pracy będziemy posługiwać się notacją wprowadzoną w [8], gdzie cylindryczny system przepływowy jest oznaczany symbolem F_{c1}. W kolejnych podrozdziałach będziemy zajmować się problemem minimalizacji długości cyklu, oznaczając to symbolem CT w polu γ.

W niektórych zastosowaniach przemysłowych systemu przepływowego, takich jak metalurgia, przemysł chemiczny i spożywczy, a także w przypadku przydziału zasobów ludzkich, często pojawia się dodatkowy wymóg braku przestojów oznaczanych odpowiednio jako *no-idle* w przypadku braku przestojów dla procesorów i *no-wait* w przypadku braku przestojów dla zadań. W przypadku systemu cylindrycznego będziemy mówić, że harmonogram spełnia ograniczenie *no-idle*, o ile każdy z procesorów w pojedynczym cyklu pracy ma co najwyżej jeden cykliczny przedział bezczynności. Podobnie powiemy, że harmonogram spełnia wymóg *no-wait*, jeśli każde z zadań w pojedynczym cyklu ma co najwyżej jeden cykliczny przedział bezczynności. W swoich rozważaniach będziemy analizować przypadek jednoczesnego wymuszenia braku przestojów dla zadań i procesorów, co będziemy sygnalizować oznaczeniem *no-wait&idle* w polu β notacji trójpolowej.

2. Złożoność obliczeniowa przy ograniczonej liczbie procesorów

Na podstawie wcześniejszych wyników wiemy, że ogólny problem szeregowania w cylindrycznym systemie otwartym, który jest uproszczeniem systemu przepływowego, jest NP-trudny [5]. Tak więc spodziewamy się, że także NP-trudne będzie szeregowanie w cylindrycznym systemie przepływowym. Tak jest

istotnie. Poniżej podajemy klasyfikację złożoności w zależności od liczby procesorów. Najpierw pokażemy, że, podobnie jak w przypadku klasycznego systemu przepływowego, przypadek dwóch procesorów jest wielomianowy [4]. Jednak algorytm konstrukcji harmonogramu jest zgoła inny i gwarantuje, że długość cyklu będzie zawsze wynosić $T = \max\{\max_i \sum_j t_{ij}, \max_j \sum_i t_{ij}\}$, czyli większej z liczb: maksymalne obciążenie procesora i maksymalne obciążenie zadania. Następnie zaś podamy dowód wskazujący, że problem szeregowania staje się trudny obliczeniowo w przypadku trzech procesorów.

Twierdzenie 1. *Optymalny harmonogram dla problemu $F_{c1}2||CT$ może zostać wyznaczony w czasie $O(n)$. Optymalna długość cyklu dla takiego przypadku wynosi $T = \max\{\max_i \sum_j t_{ij}, \max_j \sum_i t_{ij}\}$.*

Dowód. Algorytm konstrukcji harmonogramu jest oparty na algorytmie konstrukcji harmonogramu dla dwuprocesorowego klasycznego systemu otwartego [2].

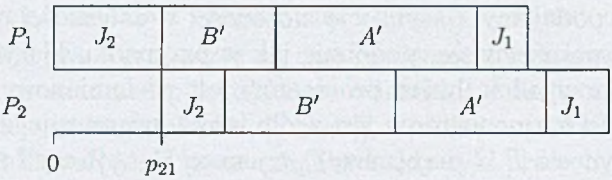
Oznaczmy przez W_1 obciążenie procesora P_1, zaś przez W_2 obciążenie P_2. Przyjmijmy, że $T = \max\{W_1, W_2\}$ (w przypadku, gdy parametr T jest realizowany dla pewnego zadania, bez trudu można skonstruować odpowiednie uszeregowanie). Oznaczmy przez A zbiór tych zadań J_i, dla których $t_{i1} \geqslant t_{i2}$, zaś przez B zbiór pozostałych zadań. Załóżmy dodatkowo, nie zmniejszając ogólności, że zadania J_1 oraz J_2 spełniają odpowiednio:

$$t_{11} \geqslant t_{i2}, \quad \text{dla } J_i \in A,$$
$$t_{22} \geqslant t_{i1}, \quad \text{dla } J_i \in B.$$

Oznaczmy $A' = A \setminus \{J_1, J_2\}$ oraz $B' = B \setminus \{J_1, J_2\}$.

Przypuśćmy, że $W_1 - t_{21} \leqslant W_2 - t_{12}$. Wskażemy uszeregowanie dla operacji pierwszego zestawu. Najpierw układamy harmonogram dla procesora P_1, rozpoczynając od momentu 0. Szeregujemy bez przestojów: najpierw operację zadania J_2, następnie operacje zadań ze zbioru B', kolejno operacje zadań z A' i na koniec operację zadania J_1. Na procesorze P_2 szeregujemy zadania w tej samej kolejności bez przestojów, przy czym zaczynamy od momentu t_{21}. Diagram Gantta dla tego harmonogramu jest zilustrowany na rys. 1. Z warunku $t_{22} \geqslant t_{i1}$ dla $P_i \in B'$ wynika, że pierwsze zadanie ze zbioru B' jest poprawnie uszeregowane. Natomiast poprawność uszeregowania dla pozostałych zadań ze zbioru B' wynika stąd, że dla $J_i \in B'$ zachodzi $t_{i1} < t_{i2}$. Legalność uszeregowania dla zadania J_1 wynika z nierówności $W_1 - t_{21} \leqslant W_2 - t_{12}$. Natomiast poprawność harmonogramu dla zadań A' wynika z analogicznych argumentów jak dla zadań B'. Uszeregowanie dla kolejnych cykli produkcyjnych otrzymujemy przesuwając kopie uszeregowania dla pierwszego zestawu o wielokrotność $W = \max\{W_1, W_2\}$.

W przypadku gdy $W_1 - t_{21} > W_2 - t_{12}$, konstruujemy harmonogram w sposób analogiczny jak dla poprzedniej sytuacji, przy czym szeregowanie na procesorze P_2 rozpoczynamy od momentu $W_1 - (W_2 - t_{12})$. □



Rys. 1. Harmonogram dla systemu dwuprocesorowego w przypadku $W_1 - t_{21} \leqslant W_2 - t_{12}$

Twierdzenie 2. *Problem $F_{c1}3||CT$ jest NP-trudny.*

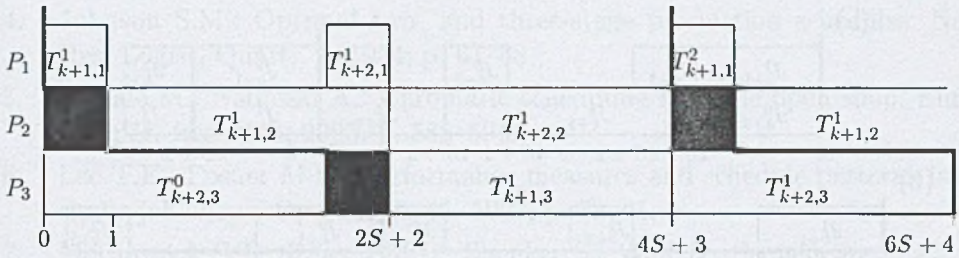
Dowód. Konstruujemy redukcję z problemu podziału zbioru. W tym celu, mając zbiór $A = \{a_1, \ldots, a_k\} \subset \mathbb{N}$, $a_1 + \ldots + a_k = 2S$, rozważamy system o $m = 3$ procesorach i $n = k + 2$ zadaniach. Każde z zadań J_i, $i = 1 \ldots, n$ składa się z jednej operacji T_{i1} o czasie wykonania $2a_i$, zaś zadania J_{k+1}, J_{k+2} mają operacje na wszystkich procesorach o długościach odpowiednio $t_{i,1} = 1$, $t_{i,2} = t_{i,3} = 2S + 1$, dla $i = k + 1, k + 2$.

Pokażemy, że istnienie harmonogramu o długości cyklu $4S + 3$ jest równoważne istnieniu podziału zbioru A. Istotnie, jeśli istnieje harmonogram o $CT = 4S + 3$, to możemy bez utraty ogólności przyjąć $0 = x^1_{k+1,1} < x^1_{k+2,1}$. Wtedy, ponieważ obciążenie J_{k+1} wynosi $4S + 3$, otrzymujemy $x^1_{k+1,2} = 1$, $x^1_{k+1,3} = 2S + 2$. Zatem operacje J_{k+1} są uszeregowane tak, jak ilustruje to rys. 2. Zauważmy teraz, że w analogiczny sposób operacje J_{k+2} muszą być przetwarzane bez przestojów. Implikuje to, że $x^1_{k+2,1} \geqslant 2S + 2$. Z drugiej strony musi zachodzić $x^1_{k+2,1} \leqslant 2S + 3$, gdyż inaczej przetwarzanie pierwszego zestawu na procesorze P_2 trwałoby ponad $4S + 3$ jednostek czasu. Rozważmy teraz podzbiór $B \subset A$ złożony z takich a_i, że T_{i1} jest uszeregowane wewnątrz przedziału $[x^1_{k+1,1} + 1, x^1_{k+2,1}]$, tzn.

$$B = \{a_i : x^1_{i,1} \in [x^1_{k+1,1} + 1, x^1_{k+2,1}]\}.$$

Zwróćmy uwagę, że obciążenie P_1 wynosi $4S + 2$, skąd wynika, że w każdym cyklu przez jedną jednostkę czasu procesor ten pozostaje wolny. Z drugiej strony długość przedziału $(x^1_{k+1,1} + 1, x^1_{k+2,1})$ jest liczbą z zakresu $[2S, 2S + 1]$, zaś czasy wykonania zadań T_{i1} są parzyste dla $i = 1, \ldots, k$. To implikuje, że sumaryczny czas operacji wykonywanych w rozważanym przedziale wynosi $2S$, a co za tym idzie, B stanowi podział A.

Jeśli zaś istnieje podział A, to konstruujemy odpowiedni harmonogram. Wpierw szeregujemy operacje zadań J_{k+1} oraz J_{k+2} zgodnie z diagramem Gantta przedstawionym na rys. 2. Wtedy na procesorze P_1 w każdym cyklu pracy mamy dwie luki o długościach $2S$ oraz $2S + 1$, wewnątrz których możemy uszeregować operacje zadań $J_1, \ldots, J_k$. □



Rys. 2. Harmonogram systemu skonstruowanego w dowodzie twierdzenia 2

3. Szeregowanie bez obustronnych przestojów

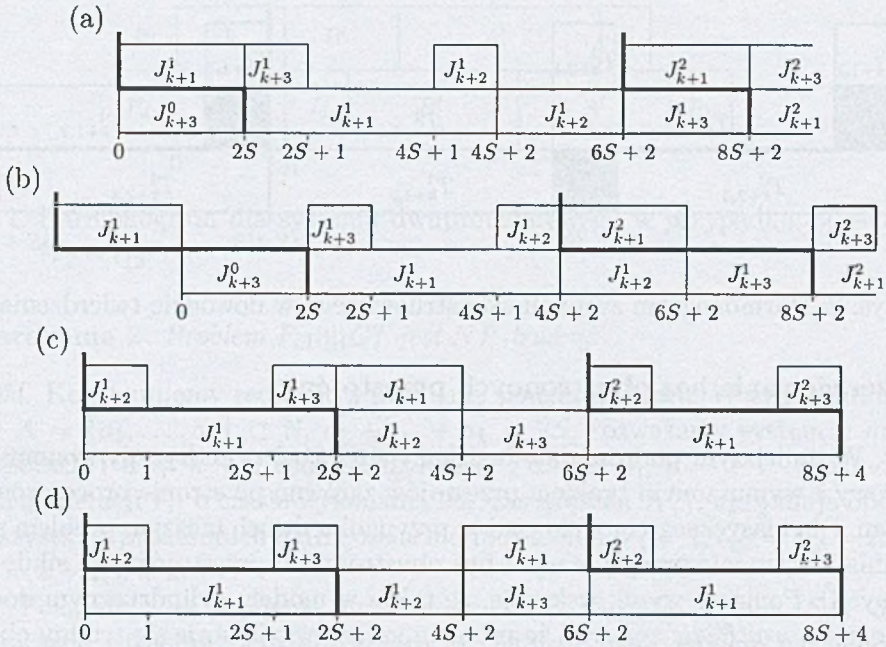
W niniejszym podrozdziale będziemy rozważać cylindryczny system przepływowy z wymuszonym brakiem przestojów zarówno po stronie procesorów, jak i zadań. Dla klasycznego modelu , już w przypadku dwóch maszyn, problem szeregowania w systemie przepływowym bez obustronnych przestojów jest silnie NP-trudny [1]. Poniższy wynik wskazuje, że także w modelu cylindrycznym dodanie wymogu *no-wait&idle* sprawia, że problem szeregowania staje się trudny obliczeniowo już dla dwóch procesorów.

Twierdzenie 3. *Problem $F_{c1}2|$no-wait&idle$|CT$ jest NP-trudny.*

Dowód. Podobnie jak w poprzednim dowodzie skonstruujemy redukcję z problemu podziału zbioru. W tym celu, mając zbiór $A = \{a_1, \ldots, a_k\} \subset \mathbb{N}$, $a_1 + \ldots + a_k = 2S$, rozważamy system o $m = 2$ procesorach i $n = k + 3$ zadaniach. Każde z zadań J_i, $i = 1 \ldots, n$, składa się z jednej operacji T_{i1} o czasie wykonania $2a_i$, zaś zadania J_{k+1}, J_{k+2} oraz J_{k+3} mają operacje na obu procesorach o długościach odpowiednio $t_{i,1} = 1$, $t_{i,2} = 2S$, dla $i = k+2, k+3$ i $t_{k+1,1} = 2S$, $t_{k+1,2} = 2S + 2$. Wykażemy, że A ma podział wtedy i tylko wtedy, gdy w skonstruowanym systemie istnieje uszeregowanie bez obustronnych przestojów o długości cyklu $6S + 2$.

Przypuśćmy, że A posiada podział. Wtedy najpierw szeregujemy operacje zadań J_{k+1}, J_{k+2} i J_{k+3}, tak jak ilustruje to diagram Gantta na rysunku 3(a). Wówczas na procesorze P_1 pozostają dwie luki długości $2S$, w których możemy uszeregować operacje zadań $J_1, \ldots, J_k$.

Załóżmy teraz, że istnieje uszeregowanie bez obustronnych przestojów o długości cyklu $CT = 6S + 2$. Rozważmy wszystkie możliwe sposobu uszeregowania zadań J_{k+1}, J_{k+2}, J_{k+3}. Jeśli odrzucimy sytuacje symetryczne oraz takie, w których na procesorze P_1 pozostają luki długości $2S - 1$ (wówczas nie ma możliwości uszeregowania w tych lukach zadań $J_1, \ldots, J_k$ ze względu na parzystość długości T_{i1} dla $i = 1, \ldots, k$), to mamy cztery możliwe konfiguracje przedstawione na rys. 3. We wszystkich przypadkach na procesorze P_1 pozostaje luka długości $2S$, która będzie generować podział A w sposób analogiczny jak w dowodzie twierdzenia 2. □



Rys. 3. Harmonogramy bez przestojów z dowodu twierdzenia 3

4. Podsumowanie

Uzyskane wyniki wskazują, że granica pomiędzy przypadkami łatwymi i trudnymi obliczeniowo przebiega podobnie dla modeli klasycznego i cyklicznego systemu przepływowego. Tym nie mniej należy zaznaczyć, że nie można w prosty sposób przenieść algorytmów wielomianowych ani dowodów NP-zupełności pomiędzy klasycznym a cylindrycznym wariantem szeregowania. W rozprawie [8] zostały skonstruowane zarówno przykłady, dla których problem w wersji standardowej jest NP-trudny, zaś w wersji cyklicznej posiada wielomianowy algorytm, jak i takie, dla których wariant klasyczny jest łatwy, zaś cykliczny trudny obliczeniowo.

LITERATURA

1. Giaro K.: NP-hardness of compact scheduling in simplified open and flow shops. Europ. J. Oper. Res., 130, 2001, p. 90–98.
2. Gonzalez T., Sahni S.: Open shop scheduling to minimize finish time. J. ACM, 3, 1976, p. 665–679.
3. Hall N.G., Lee T.E., Posner M.E.: The complexity of cyclic shop scheduling problems. J. Scheduling, 5, 2002, p. 307–327.

4. Johnson S.M.: Optimal two- and three-stage production schedules. Naval Res. Logist. Quart., 1, 1954, p. 61–68.
5. Kubale M., Nadolski A.: Chromatic scheduling in cyclic open shop. Europ. J. Oper. Res., 164, 2005, p. 585–591.
6. Lee T.E., Posner M.E.: Performance measures and schedule patterns in periodic job shops. Oper. Res., 45, 1997, p. 72–91.
7. McCormick S.T., Pinedo M.L., Shenker S., Wolf B.: Sequencing in an Assembly Line with Blocking to Minimize Cycle Time. Oper. Res., 37, 1989, p. 925–935.
8. Nadolski A.: Chromatyczne szeregowanie w cyklicznych systemach produkcyjnych. 2005. Rozprawa doktorska, Politechnika Gdańska.

Recenzent: Prof. dr hab. inż. Czesław Smutnicki

Abstract

In the paper we study the problem of scheduling in a flow shop in which production occurs in a cyclic manner, i.e. there is a number of identical job sets and whenever one set is completed the following one is started immediately. We consider a model of cylindrical flow shop, where no task of a given job can be performed prior to the completition of all the tasks of that job in the previous job set. The main optimization criterion is the cycle time. First we analyze the problem of scheduling with a few processors. We prove that the problem is hard in the case of three processors and we provide a polynomial time algorithm for scheduling on two processors. Then we study the complexity of scheduling without two-sided waiting periods. We show that no-wait and no-idle constraints make the problem hard even in the case of two processors.