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

Krzysztof GIARO
Politechnika Gdańska

SZEREGOWANIE ZADAŃ NA PROCESORACH DEDYKOWANYCH BEZ PRZESTOJÓW

Streszczenie. W praktycznym szeregowaniu zadań dość często spotykamy się z koniecznością zapewnienia nieprzerwanej pracy poszczególnym podmiotom naszego systemu. Szeregowanie takie nazywa się szeregowaniem bez przestojów. Zadaniem tej pracy jest zasygnalizowanie skali trudności obliczeniowej, jaką wymusza powyższe założenie. Wykażemy, że szereg podstawowych problemów decyzyjnych i optymalizacyjnych dotyczących istnienia lub najprostszych parametrów harmonogramu w szeregowaniu bez przestojów staje się NP-trudny, nawet dla systemów o grafach szeregowania tak prostych jak ścieżka i cykl. Rozważania nasze będą dotyczyć modeli open shop, flow shop oraz systemu zadań dwuprocesorowych.

SCHEDULING ON DEDICATED PROCESSORS WITHOUT WAITING PERIODS

Summary. In practical task scheduling it is sometimes required that the elements of a system perform consecutively. Such a scheduling is called scheduling without waiting periods or no-wait and/or no-idle. In this article we study the complexity of some simplified scheduling problems of this kind. In particular, we show that many trivial questions about scheduling become NP-hard, even if the scheduling graph of a system is a path, or a cycle. Our consideration concern the following models: open shop, flow shop and 2-procesor tasks system.

1. Wprowadzenie

Ogólnie, o szeregowaniu zadań bez przestojów mówimy wtedy, gdy układamy harmonogram dla pewnego systemu, w którym z pewnych względów konieczne jest zapewnienie warunków ciągłej pracy (brak przerw) poszczególnym jego podmiotom. Praca bez postojów jest warunkiem często pojawiającym się w praktycznym szeregowaniu zadań.

Brak przestojów może być rozważany w kontekście różnych teoretycznych modeli szeregowania. Weźmy dla przykładu klasyczny *system otwarty* (*open shop*). Mamy tu do czynienia ze skończonymi zbiorami procesorów i zadań, przy czym każde zadanie podzielone jest na operacje przypisane do procesorów, na których mają się one wykonywać. Każda operacja ma też ustaloną długość przedziału czasowego, jaki ma jej przypisać szukany harmonogram (zakładamy, że jest to liczba całkowita nieujemna). System otwarty narzuca warunek, by żadne dwie operacje w zadaniu nie były wykonywane równocześnie ani żaden procesor nie pracował jednocześnie nad dwiema operacjami. Szeregowanie bez przestojów oznacza, iż zarówno procesory mają pracować bez przerw, jak i pomiędzy poszczególnymi operacjami każdego zadania nie powinno być luk czasowych. Jest to dokładniej tzw. *szeregowanie bez obustronnych przestojów* - tylko takim będziemy się w niniejszej pracy zajmować (rozważa się również modele bez przestojów jednostronnych, np. kiedy przerwy w pracy nie są dopuszczalne tylko po stronie procesorów). System otwarty nie nakłada wymagań co do kolejności wykonywania się poszczególnych operacji, jeżeli jednak zażądamy by te wykonywane były w obrębie zadania wg porządku wzrastających numerów przypisanych procesorów - otrzymujemy tzw. *system przepływowy* (*flow shop*). Również i tu można zażądać obustronnego braku przestojów. Trzecim modelem, który będzie rozważany, jest *system zadań dwuprocesorowych* (*biprocessor tasks*). Składa się on z zestawu n zadań $J_1,...,J_n \in J$ wykonywanych na m procesorach $M_1,...,M_m \in M$. Zakładamy, że każde zadanie używa równocześnie dokładnie dwóch dedykowanych procesorów, przy czym procesor może w danej chwili obsługiwać co najwyżej jedno zadanie. Zadania mają też ustalone czasy wykonywania. Dodanie warunku szeregowania bez przestojów oznacza tym razem, iż każdy z procesorów ma pracować bez przerw, czyli odpowiadające mu zadania muszą wykonywać się jedno po drugim.

Wymienione wyżej modele dają się w zgodny z intuicją sposób wyrażać w języku teorii grafów. W systemach otwartym i przepływowym możemy utworzyć prosty graf dwudzielny, którego wierzchołki jednej partycji tworzą procesory, a drugiej zadania, krawędzie zaś łączące zadania z procesorami odpowiadają operacjom o niezerowych długościach. Każdej krawędzi przypisana jest waga - liczba naturalna zwana też grubością krawędzi, równa czasowi wykonania odpowiedniej operacji. Tworzenie harmonogramu w systemie otwartym odpowiada teraz interwałowemu kolorowaniu krawędziowemu grafu. Należy bowiem przypisać każdej krawędzi przedział o początku i końcu będący liczbami całkowitymi i o długości równej grubości tej krawędzi tak, by przedziały przyporządkowane krawędziom spotykającym się przy jednym wierzchołku nie zachodziły na siebie. System przepływowy dodatkowo żąda, by przy każdym wierzchołku w partycji zadań przedziały przypisane krawędziom wychodzącym do wierzchołków z partycji procesorów następowały po sobie w kolejności zgodnej z numeracją tychże procesorów. W obu przypadkach warunek obustronnego braku przestojów ogranicza nas do rozważania tzw. kolorowań zwartych, tj. takich, w których pomiędzy przedziałami odpowiadającymi krawędziom wychodzącym z każdego wierzchołka nie ma przerw. Pewną nieścisłość może wprowadzać fakt, że niekiedy przy kolorowaniu dopuścimy przedziały o ujemnych końcach, zwracając uwagę jedynie na ich całkowitość - możemy jednak przyjąć, iż każdemu kolorowaniu odpowiada szeregowanie przesunięte o taką stałą, by najniższe ograniczenie dolne użytego przedziału odpowiadało chwili zerowej harmonogramu. W tym kontekście istnienie harmonogramu bez przestojów dla systemu odpowiada istnieniu pokolorowania zwartego dla jego grafu, podobnie minimalizacja podstawowego parametru - długości harmonogramu (C_{max}) jest tym samym, co minimalizacja rozpiętości odpowiedniego kolorowania, czyli różnicy między maksymalnym z końców przypisanych krawędziom przedziałów a minimalnym z początków. Dla uproszczenia terminologii dalsze rozważania teoretyczne będą prowadzone w języku kolorowania zwartego grafów, jedynie ostateczne wyniki zostaną przetłumaczone na równoważne im zdania dotyczące szeregowania zadań.

Komentarza wymaga jeszcze sposób modelowania za pomocą grafu systemu zadań dwuprocesorowych. Wierzchołki tworzą tutaj procesory, każdemu zaś zadaniu odpowiada krawędź łącząca oba wykorzystywane przezeń procesory. Jest oczywiste, że jeśli przypiszemy krawędziom grubości równe długościom odpowiednich zadań, problemy szeregowania bez postojów stają się znów równoważne odpowiednim zagadnieniom zwartego kolorowania. Należy zauważyć, że tym razem w ogólnym przypadku system może być opisany za pomocą grafu innego niż dwudzielny, co więcej nie musi to być graf prosty - są bowiem dopuszczalne krawędzie wielokrotne.

Znane dotychczas wyniki zdają się nie pokazywać w pełni skali trudności obliczeniowej pojawiającej się wraz z wprowadzeniem warunku szeregowania bez przestojów. Dla przykładu - wiadomo, że optymalizacja wg kryterium C_{max} jest NP-trudna w systemie otwartym nawet w przypadku grafu o strukturze drzewa. W niniejszej pracy pójdziemy dalej - pokażemy NP-trudność szeregu podstawowych zagadnień w opisanych wyżej modelach, w sytuacji gdy grafem systemu są struktury tak proste, jak ścieżka i cykl. Okazuje się więc, że w systemie otwartym NP-trudne są problemy $O|no\text{-}wait\&idle, M{=}path\ or\ cycle|C_{max}$, podobnie w systemie zadań dwuprocesorowych $P|no\text{-}wait, fix_j{=}2, M{=}path\ or\ even\_cycle|C_{max}$ oraz dla systemu przepływowego $F|no\text{-}wait\&idle, M{=}path|C_{max}$. Dla systemu zadań dwuprocesorowych oraz systemu przepływowego o grafach cyklicznych NP-zupełne jest pytanie o samo tylko istnienie harmonogramu bez przestojów, mianowicie $P|no\text{-}wait,fix_j{=}2,M{=}odd\_cycle|\bullet$ oraz $F|no\text{-}wait\&idle,M{=}cycle|\bullet$.

W dalszym ciągu pozwolimy sobie na jeszcze jedno uproszczenie terminologiczne. Zauważmy, że zarówno ścieżkę n-krawędziową z krawędziami obciążonymi liczbami naturalnymi - ich grubościami, jak i podobny cykl można zadać, podając jedynie ciąg grubości kolejnych sąsiadujących ze sobą krawędzi $(c_1,\dots,c_n)$. Podobnie kolorowaniem zwartym opisanej w ten sposób ścieżki jest ciąg par liczb całkowitych $((a_1,b_1),\dots,(a_n,b_n))$, takich że $c_i = b_i - a_i$ dla $i=1,\dots,n$ oraz dla każdego $i=1,\dots,n-1$ ma zachodzić $a_i = b_{i+1}$ lub $b_i = a_{i+1}$. Kolorowanie zwarte cyklu jest analogicznym ciągiem par spełniającym powyższe warunki, dla którego dodatkowo musi zachodzić $a_1 = b_n$ lub $b_1 = a_n$. W obu przypadkach rozpiętość kolorowania jest równa $max(b_1,\dots,b_n) - min(a_1,\dots,a_n)$, nie może więc ona przekraczać $\sum_{i=1}^{n} c_i$.

2. System otwarty i system zadań dwuprocesorowych

Twierdzenie 1. *Istnieją liniowe algorytmy zwracające 2-przybliżone rozwiązania problemów* $O|no\text{-}wait\&idle,M{=}path\ or\ cycle|C_{max}$ *oraz* $P|no\text{-}wait\&idle,fix_j{=}2,M{=}path\ or\ even\_cycle|C_{max}$.

Dowód. Niech $(c_1,\dots,c_n)$ będzie grafem naszego systemu, tj. cyklem parzystym lub ścieżką. Wówczas kolorowanie za pomocą ciągu $((0,c_1),(-c_2,0),(c_3,0),(-c_4,0)\dots)$ jest zwarte i ma rozpiętość nie przekraczającą $2max(c_1,\dots,c_n)$. Wystarczy zauważyć, że każde kolorowanie musi mieć rozpiętość nie mniejszą niż maksymalna grubość krawędzi grafu.□

Definicja. Niech dany będzie ciąg liczb naturalnych $(a_1,\dots,a_n)$ uporządkowany niemalejąco i złożony z wyrazów większych niż jeden. Jego *rozwinięciem* nazwiemy ciąg liczb naturalnych $(b_1,\dots,b_{n+1})$ spełniający układ równań:

$$b_1 = 2,\ a_1 = (b_1+b_2)/2,\ a_2 = (b_2+b_3)/2,\ \dots\ ,\ a_n = (b_n+b_{n+1})/2. \qquad (*)$$

Można łatwo sprawdzić, że rozwiązaniem układu są liczby:

$$\begin{aligned} b_1 &= 2\\ b_2 &= 2a_1 - b_1\\ b_3 &= 2(a_2 - a_1) + b_1\\ &\dots \qquad\qquad\qquad (**)\\ b_i &= 2(a_{i-1} - a_{i-2} + \dots + (-1)^i a_1) - (-1)^i b_1 \end{aligned}$$

Zatem liczby b_i są parzyste oraz $b_i \geq 2$, gdyż dla i nieparzystych $b_i \geq b_1$ oraz $b_i \geq 2a_1 - b_1 \geq 2$ dla parzystych i. W dalszym ciągu będziemy też korzystać z NP-zupełności znanego problemu połowienia zbioru (na użytek tej pracy nazwiemy go problemem *PZ*):

Twierdzenie 2. *Problem polegający na rozstrzygnięciu, czy z danego ciągu $(c_1,\ldots,c_n)$ liczb naturalnych o sumie S można wybrać podciąg o sumie S/2, jest problemem NP-zupełnym.* □

Problem powyższy można oczywiście przeformułować do pytania o istnienie ciągu liczb całkowitych $(a_1,\ldots,a_n)$, takiego że $|a_i| = c_i$ oraz $\sum_{i=1}^{n} a_i = 0$ - wystarczy przyjąć, że liczbom z wybranego podciągu mają odpowiadać dodatnie a_i i odwrotnie. Poniższy lemat pozwala natomiast sprowadzić **PZ** do zagadnień związanych ze zwartym kolorowaniem ścieżek i cykli.

Lemat 3. *Niech dany będzie ciąg liczb naturalnych większych od jeden $(c_1,\ldots,c_n)$ uporządkowany niemalejąco oraz niech $(b_1,\ldots,b_{n+1})$ stanowi jego rozwinięcie. Dla danej liczby całkowitej x istnienie ciągu $(a_1,\ldots,a_n)$ liczb takich, że $|a_i| = c_i$ oraz $\sum_{i=1}^{n} a_i = x$ jest równoważne istnieniu takiego kolorowania zwartego $((p_1,k_1),\ldots,(p_{n+1},k_{n+1}))$ ścieżki $(b_1,\ldots,b_{n+1})$, że spełniony jest warunek $(p_{n+1}+k_{n+1})/2-(p_1+k_1)/2 = x$.*

Dowód.

$\Leftarrow$ Załóżmy, że dane jest odpowiednie pokolorowanie. Definiujemy $s_i = (p_i+k_i)/2$. Skoro b_i są parzyste, to s_i są liczbami całkowitymi. Dalej mamy $x = s_{n+1}-s_1 = (s_2-s_1)+\ldots+(s_{n+1}-s_n) = a_1+\ldots+a_n$, gdzie oznaczyliśmy $a_i = s_{i+1}-s_i$. Zauważmy dalej, że z warunków zwartego kolorowania mamy $|a_i| = |s_{i+1}-s_i| = |b_i+b_{i+1}|/2 = c_i$, czyli $a_i = \pm c_i$.

$\Rightarrow$ Niech $(a_1,\ldots,a_n)$ będzie ciągiem jak w założeniach. Wtedy $a_1+\ldots+a_n = x$ i definiując ciąg $s_1=0$, $s_i = a_1+\ldots+a_{i-1}$ dla $i > 1$ mamay $s_{n+1} = x$. Oczywiście s_i są całkowite. Pokolorowanie $((p_1,k_1),\ldots,(p_{n+1},k_{n+1}))$ tworzymy przyjmując $(p_i,k_i) = (s_i-b_i/2, s_i+b_i/2)$. Oczywiście $p_1+k_1 = 0$ oraz $p_{n+1}+k_{n+1} = 2x$, wreszcie $k_i-p_i = b_i$. Wystarczy pokazać, że $p_i = k_{i+1}$, lub $k_i = p_{i+1}$ dla $i = 1,\ldots,n$. Ale $k_{i+1}-p_i = s_{i+1}+b_{i+1}/2-s_i+b_i/2 = a_i+c_i$, podobnie $p_{i+1}-k_i = s_{i+1}-b_{i+1}/2-s_i-b_i/2 = a_i-c_i$. Jedna z tych liczb musi być zerem, gdyż $a_i = \pm c_i$. □

Lemat 4. *Następujący problem W1 jest NP-zupełny: "Rozstrzygnąć, czy dla danej ścieżki opisanej przez ciąg $(c_1,\ldots,c_n)$ liczb naturalnych o c_1 i c_n parzystych istnieje kolorowanie zwarte $((a_1,b_1),\ldots,(a_n,b_n))$ spełniające warunek $a_1+b_1 = a_n+b_n$?"*

Dowód. Do powyższego problemu można łatwo sprowadzić problem połowienia **PZ**. Niech $(d_1,...,d_n)$ będzie ciągiem wejściowym dla **PZ**. Możemy założyć, że wszystkie jego wyrazy są większe od 1, w przeciwnym razie wystarczy podwoić każdy z nich. W czasie wielomianowym sortujemy ten ciąg niemalejąco, uzyskując jego permutację $(e_1,...,e_n)$, dla niej zaś korzystając ze wzorów (**) znów w czasie wielomianowym liczymy rozwinięcie $(c_1,...,c_{n+1})$ - jak wiemy c_i są parzyste. Na mocy lematu 3 pozytywna odpowiedź na problem **PZ** jest równoważna pozytywnej odpowiedzi na **W1** z danymi $(c_1,...,c_{n+1})$. □

Lemat 5. *Niech dany będzie ciąg liczb naturalnych* $L = (c_1,...,c_n)$ *o* c_1 *i* c_n *parzystych i sumie wyrazów* S. *Istnienie dla niego rozwiązania problemu W1 jest równoważne istnieniu kolorowania zwartego o rozpiętości* $\leq 4S$ *dla ścieżki opisywanej przez ciąg* $P = (4S-1,1,2S-c_1/2-1,c_1,c_2,...,c_n,2S-c_n/2-1,1,4S-1)$.

Dowód.

$\Rightarrow$ Niech $((a_1,b_1),...,(a_n,b_n))$ będzie rozwiązaniem dla **W1**. Dodając odpowiednią stałą do wszystkich liczb w tych parach, możemy zażądać, by $a_1+b_1 = a_n+b_n = 0$. Kolorowanie zwarte ciągu P można zadać w postaci ciągu par: $((-2S+1,2S), (-2S,-2S+1), (-2S+1,a_1), (a_1,b_1), ..., (a_n,b_n), (-2S+1,a_n), (-2S,-2S+1), (-2S+1,2S))$. Skoro $((a_1,b_1),...,(a_n,b_n))$ jest kolorowaniem zwartym dla L i $b_1 > 0$, to $a_i \geq -S$, tak samo uzyskujemy $b_i \leq S$. Zatem minimalną i maksymalną liczbą w uzyskanym kolorowaniu jest odpowiednio $-2S$ i $2S$, a rozpiętość wynosi $4S$.

$\Leftarrow$ Niech dane będzie zwarte kolorowanie $((x_1,y_1),(x_2,y_2),(x_3,y_3),(a_1,b_1),...,(a_n,b_n),(x_4,y_4),(x_5,y_5),(x_6,y_6))$ ciągu P o rozpiętości $\leq 4S$. Ponieważ suma długości dwóch pierwszych krawędzi P wynosi $4S$ - tyle też musi być równa sama rozpiętość. Możemy więc założyć, że minimalną liczbą w tym kolorowaniu jest $-2S$, a maksymalną $2S$. Biorąc to pod uwagę oraz rozważając warunki poprawności kolorowania zwartego uzyskujemy łatwo, iż istnieją tylko dwa możliwe układy par liczb $(x_1,y_1),(x_2,y_2),(x_3,y_3)$ równe $(-2S+1,2S), (-2S,-2S+1), (-2S+1,-c_1/2)$ lub $(-2S,2S-1),(2S-1,2S),(c_1/2,2S-1)$. W obu przypadkach dostajemy $(a_1,b_1) = (-c_1/2, c_1/2)$. Podobne rozumowanie dotyczące ostatnich trzech krawędzi ścieżki pozwala stwierdzić, że $(a_n,b_n) = (-c_n/2,c_n/2)$. Zatem $((a_1,b_1),...,(a_n,b_n))$ jest rozwiązaniem problemu **W1** dla ciągu L. □

Lemat 6. *Niech dany będzie ciąg liczb naturalnych* $L = (c_1,...,c_n)$ *o* c_1 *i* c_n *parzystych i sumie wyrazów* S. *Istnienie dla niego rozwiązania problemu W1 jest równoważne istnieniu* *kolorowania zwartego o rozpiętości* $\leq 4S$ *dla cyklu parzystego opisywanego przez ciąg* $Q = (4S-1, 1, 2S-c_1/2-1, c_1,c_2,...,c_n,2S-c_n/2-1, 1, 4S-1, 1, 2S-c_n/2-1, c_n,c_{n-1},...,c_1, 2S-c_1/2-1, 1)$.

Dowód. Korzystając z lematu poprzedniego wystarczy zauważyć, że cykl Q to dwie kopie ścieżki P sklejone odpowiadającymi sobie krawędziami końcowymi. Zatem każde pokolorowanie zwarte ścieżki można przenieść na jej kopie uzyskując pokolorowanie zwarte cyklu Q o tej samej rozpiętości, podobnie każde pokolorowanie zwarte cyklu można obciąć do takiegoż pokolorowania ścieżki P bez zwiększenia rozpiętości. Minimalne rozpiętości pokolorowania zwartego dla ścieżki P i cyklu Q są więc równe. □

Twierdzenie 7. *Oba poniższe problemy są NP-zupełne: "Czy dla danej ścieżki (Czy dla cyklu parzystego)* $(c_1,...,c_n)$ *i liczby naturalnej d istnieje zwarte kolorowanie o rozpiętości* $\leq d$*?"*

Dowód. Lematy 5 i 6 dają proste wielomianowe sprowadzenia problemu **W1** z lematu 4 do powyższych problemów decyzyjnych. □

Tłumacząc powyższy wynik na język szeregowania zadań uzyskujemy:

Twierdzenie 8. *Problemy optymalizacyjne z twierdzenia* 1 *są NP-trudne.* □

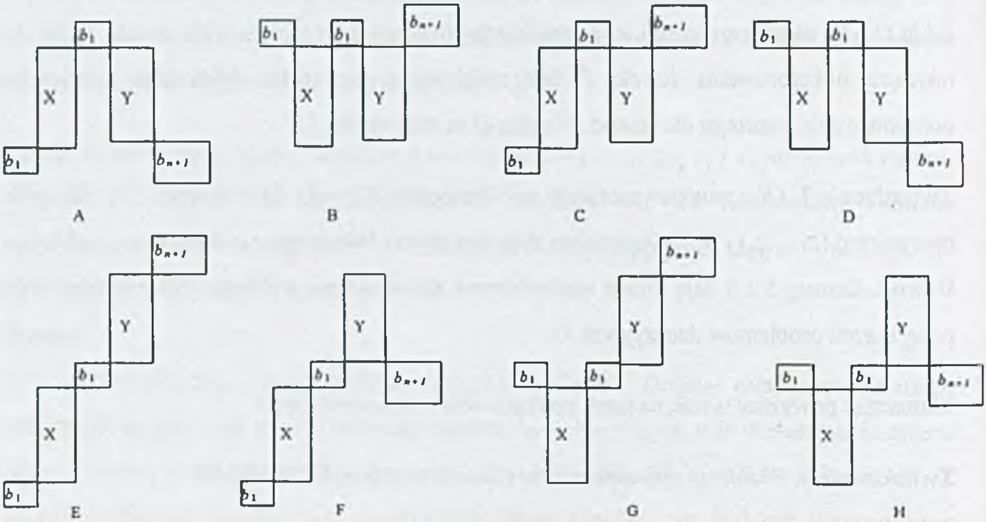
Twierdzenie 9. *Problem polegający na stwierdzeniu, czy dany cykl nieparzysty opisywany przez ciąg* $(c_1,...,c_n)$ *można pokolorować w sposób zwarty, jest NP-zupełny.*

Dowód. Sprowadzimy do niego wielomianowo problem połowienia zbioru **PZ**. Niech więc $L = (a_1,...,a_n)$ będzie ciągiem danych wejściowych dla **PZ**, posortowanym niemalejąco. Prostym ćwiczeniem jest stwierdzenie, że **PZ** pozostaje NP-zupełny, jeżeli ograniczymy się do ciągów o długości nieparzystej, możemy ponadto przyjąć, że wszystkie a_i są podzielne przez 3 (w przeciwnym razie potrajamy każdą z tych liczb). Niech $(b_1,...,b_{n+1})$ będzie rozwinięciem ciągu L. Dla dowodu twierdzenia pokażemy, że istnieje rozwiązanie problemu PZ dla L wtedy i tylko wtedy, gdy cykl nieparzysty opisany ciągiem $C = (b_1,...,b_{n+1},2S,b_1,2S+(b_{n+1}-b_1)/2)$ ma pokolorowanie zwarte, gdzie przez S oznaczono sumę $b_1+...+b_{n+1}$.

$\Rightarrow$ Jeśli istnieje rozwiązanie **PZ** dla L, to na mocy lematu 3 istnieje pokolorowanie zwarte ścieżki $(b_1,...,b_{n+1})$ postaci $((c_1,d_1),...,(c_{n+1},d_{n+1}))$ takie, że $c_1+d_1 = c_{n+1}+d_{n+1}$. Można je rozszerzyć do kolorowania zwartego całego cyklu C nadając dodatkowym trzem krawędziom kolejne przedziały: $(d_{n+1},d_{n+1}+2S),(d_{n+1}+2S,d_{n+1}+2S+b_1),(d_1,d_{n+1}+2S)$.

$\Leftarrow$ Załóżmy istnienie zwartego kolorowania cyklu C i rozważmy przedziały przypisane kolejnym pięciu krawędziom tego cyklu o grubościach kolejno: b_1, $2S+(b_{n+1}-b_1)/2$, b_1, $2S$ oraz

b_{n+1}. Możliwych jest 16 sposobów wzajemnego względnego ułożenia tych przedziałów, po odrzuceniu rozwiązań symetrycznych względem odwrócenia osi liczbowej pozostaje 8 zaprezentowanych na rys. 1. Grubość krawędzi X wynosi $2S+(b_{n+1}-b_1)/2$, a krawędzi Y jest równa $2S$. Zastanówmy się, które z tych możliwości są dopuszczalne.



Rys. 1. Możliwe uszeregowania
Fig. 1. Possible schedulings

Pierwsza z narysowanych krawędzi i ostatnia są końcami ścieżki $(b_1,\dots,b_{n+1})$ o rozpiętości kolorowania nie większej, niż S. Zatem grubości krawędzi X i Y są za duże i od razu można odrzucić jako niemożliwe przypadki C,D,E,F,G. Niech $((c_1,d_1),\dots,(c_{n+1},d_{n+1}))$ będzie kolorowaniem zwartym ścieżki $(b_1,\dots,b_{n+1})$ powstałym przez obcięcie naszego kolorowania cyklu. Z lematu 3 wynika istnienie takiego ciągu liczb całkowitych $(e_1,\dots,e_n)$, że $|e_i| = a_i$ oraz $(c_{n+1}+d_{n+1})/2-(c_1+d_1)/2 = \sum_{i=1}^{n} e_i$. W przypadkach B i H mamy jednak: $(c_{n+1}+d_{n+1})/2-(c_1+d_1)/2 = \pm(b_{n+1}-b_1)/2 = \pm(a_n-a_{n-1}+\dots+a_1-b_1) \equiv \pm 2 \ mod\ 3$, gdyż $b_1 = 2$ i wszystkie a_i dzieliły się przez 3. Z tej samej przyczyny $\sum_{i=1}^{n} e_i \equiv 0 \ mod\ 3$ - sprzeczność. Pozostaje nam jedynie przypadek A, kiedy $c_{n+1}+d_{n+1} = c_1+d_1$, co na mocy lematu 3 jest równoważne istnieniu rozwiązania problemu **PZ** dla ciągu L. □

Dla szeregowania zadań oznacza to, że zachodzi:

Twierdzenie 9. *Problem $P|no\text{-}wait, fix_j=2, M=odd\_cycle|\bullet$ jest NP-zupełny.* □

3. Zagadnienia w systemie przepływowym

W całym bieżącym rozdziale dopuszczamy występowanie w harmonogramach liczb ujemnych, co oczywiście nie ma wpływu na ich istnienie lub długość. Zauważmy, że do opisu systemu przepływowego o grafie ścieżki lub cyklu nie wystarcza podanie samego tylko ciągu grubości krawędzi - dla poprawności szeregowania poza warunkami poprawności kolorowania zwartego dochodzi konieczność następowania po sobie operacji w obrębie pracy w kolejności odpowiadającej numeracji procesorów. Na początek odnotujmy oczywisty fakt:

Twierdzenie 10. *Zawsze istnieje rozwiązanie problemu F|no-wait&idle,M=path|•*. □

Następny lemat będzie przydatny przy dowodach NP-zupełności problemów związanych z systemem przepływowym.

Lemat 11. *Następujący problem W2 jest NP-zupełny: "Dany jest system przepływowy o grafie ścieżki parzystej opisanej przez ciąg liczb parzystych* $(d_1,...,d_n)$ *i sumie S', w której dwa skrajne wierzchołki są procesorami, zaś idąc wzdłuż tej ścieżki od początku do końca mijamy procesory w kolejności rosnącej numeracji. Czy istnieje szeregowanie bez postojów* $((a_1,b_1),...,(a_n,b_n))$ *takie, że* $(b_n+a_n)-(b_1+a_1) = S'$*?"*

Dowód. Sprowadzimy wielomianowo do powyższego problem **W1** z lematu 4. Weźmy daną wejściową dla **W1** - ścieżkę $L = (c_1,...,c_n)$ o sumie elementów S (bez zmniejszenia ogólności można przyjąć, że wszystkie c_i są parzyste) i zbudujmy system przepływowy F' jak w treści lematu, o ścieżce długości $2n$ zadanej przez ciąg $L' = (c_1,c_1,c_2,c_2,...,c_n,c_n)$. Definiujemy też liczby $s_0 = 0$, $s_i = c_1+...+c_i$ i dla dowolnego pokolorowania zwartego ścieżki L postaci $((e_1,f_1),...,(e_n,f_n))$ budujemy szeregowanie F' zadane przez następujące kolorowanie: krawędziom o numerach $2i-1$ dla $i = 1,...,n$ przypisujemy przedział $(e_i+s_{i-1},f_i+s_{i-1})$ a tym o numerach $2i$ przedział (e_i+s_i,f_i+s_i). Można łatwo sprawdzić, że jest to poprawne szeregowanie bez przestojów dla F' i co więcej każde szeregowanie bez przestojów F' jest związane powyższą relacją z pewnym pokolorowaniem zwartym ścieżki L. Zatem **W1** ma rozwiązanie wtedy i tylko wtedy, gdy istnieje takie szeregowanie dla F', w którym różnica sum ograniczeń przedziałów krawędzi skrajnych w ścieżce L' wynosi $2s_n$ - to zaś jest równe sumie jej elementów. □

Twierdzenie 12. *Problem $F|no\text{-}wait\&idle,M=path|C_{max}$ jest NP-trudny.*

Dowód. Sprowadzamy wielomianowo do powyższego problem **W2**. Niech dany będzie system F jak w treści **W2** opisany ścieżką $(c_1,...,c_n)$ o sumie S. Tworzymy F' dodając jeden procesor o numerze większym od wszystkich użytych w F oraz jeden o numerze mniejszym od wszystkich, wreszcie dwa zadania, takie by grafem F' była ścieżka $L' = (2S,2S,c_1,...,c_n, (3S+c_1-c_n)/2, (S+c_n-c_1)/2)$, przy czym pierwszym wierzchołkiem tej ścieżki jest procesor o numerze maksymalnym, a ostatnim - minimalnym.

Załóżmy istnienie rozwiązania problemu **W2** postaci $((a_1,b_1),...,(a_n,b_n))$, możemy przyjąć, że $a_1 = 0$. Wtedy ciąg: $((0,2S), (-2S,0), (a_1,b_1), ..., (a_n,b_n), (-2S,a_n), (a_n,2S))$ stanowi poprawne szeregowanie bez przestojów dla F'. Jego rozpiętość jest równa $4S$, gdyż żadna z liczb a_i i b_i co do modułu nie przekracza S.

Odwrotnie, mając dane uszeregowanie bez przestojów dla F' o rozpiętości $\leq 4S$ widzimy, że suma grubości pierwszych dwóch krawędzi L' wynosi $4S$, rozpiętość musi więc być też tyle równa. Możemy przyjąć, że maksymalną liczbą w tym szeregowaniu jest $2S$, a minimalną $-2S$ i wtedy pierwsze trzy krawędzie ścieżki muszą mieć przypisane przedziały $(0,2S),(-2S,0),(0,c_1)$, a ostatnie trzy $((S+c_1-c_n)/2,(S+c_1+c_n)/2)$, $(-2S,(S+c_1-c_n)/2)$, $((S+c_1-c_n)/2,2S)$. A zatem odrzucając po dwie skrajne pary z tego uszeregowania, uzyskamy rozwiązanie problemu **W2**. W ten sposób **W2** został wielomianowo sprowadzony do pytania o istnienie dla danego systemu przepływowego o grafie ścieżki uszeregowania bez przestojów o długości nie przekraczającej zadanej liczby. □

Twierdzenie 13. *Problem $F|no\text{-}wait\&idle,M=cycle|\bullet$ jest NP-zupełny.*

Dowód. Sprowadzamy wielomianowo do powyższego problem **W2**. Niech dany będzie system F jak w treści **W2** o grafie ścieżki opisanej ciągiem $(c_1,...,c_n)$ z sumą S. Bez zmniejszenia ogólności można przyjąć, że wszystkie c_i są podzielne przez 4. Tworzymy system F' dodając jedno zadanie zawierające operację długości 1 na procesorze o najmniejszym numerze i operację długości $(S+c_1-c_n)/2$ na procesorze maksymalnego numeru. Wtedy F' ma graf o strukturze cyklu parzystego opisanego ciągiem $(c_1,...,c_n, (S+c_1-c_n)/2,1)$.

Załóżmy istnienie uszeregowania $((a_1,b_1),...,(a_n,b_n))$ będącego rozwiązaniem problemu **W2**. Możemy przyjąć, że $a_1 = 0$, wtedy $a_n = (S+c_1-c_n)/2$. Nadając nowej operacji o długości 1 przedział $(-1,0)$, a drugiej nowej operacji przedział $(0,(S+c_1-c_n)/2)$, dostajemy szeregowanie bez przestojów dla F'.

Odwrotnie, niech dane będzie uszeregowanie bez przestojów dla F' postaci $((a_1,b_1),...,(a_n,b_n), (e_1,f_1),(e_2,f_2))$ i znów przyjmijmy $a_1 = 0$, ponadto mamy $f_2 = e_1$. Z konstrukcji systemu wynika, że $a_n \geq b_1 = c_1$, czyli $b_n > c_1+1$, gdy tymczasem $f_2 \leq c_1+1$, czyli $e_1 \neq b_n$, a zatem $f_1 = a_n$ i $e_1 = b_n-(S+c_1-c_n)/2$. Gdyby było $e_2 = b_1$, wtedy $c_1+1 = f_2 = e_1 = b_n-(S+c_1-c_n)/2$, czyli b_n byłoby nieparzyste. Jest to sprzeczność, gdyż wszystkie liczby a_i i b_i muszą być tej samej parzystości. Musi więc być $f_2 = a_1 = 0$, czyli $b_n = (S+c_1-c_n)/2$ - wtedy jednak ciąg $(a_1,b_1),...,(a_n,b_n)$ jest rozwiązaniem problemu **W2**. □

4. Podsumowanie

Powyższe wyniki negatywne dotyczące bardzo ogólnych systemów skłaniają do zapytania o takie szczegółowe założenia upraszczające, które dopuszczałyby istnienie pewnych wielomianowych algorytmów szeregujących. Jednym z najpowszechniej stosowanych uproszczeń jest przyjmowanie równych długości dla wszystkich niezerowych operacji lub zadań w systemie. Tak zwany model zero-jedynkowy (*Zero-Unit Execution Times*) pozwala na usunięcie z grafu systemu wag krawędzi (wszystkie bowiem są równe jeden) i sprowadzenie szeregowania zadań w systemie otwartym oraz zadań dwuprocesorowych do klasycznego kolorowania krawędziowego grafu liczbami naturalnymi jako kolorami. Odpowiednik szeregowania bez przestojów nosi nazwę zwartego kolorowania krawędziowego i jest od pewnego czasu badany w teorii grafów. Istnieją więc np. wielomianowe algorytmy kolorujące pewne klasyczne rodziny grafów w sposób optymalny (np. drzewa [4], pełne dwudzielne [4,6,7], dwudzielne podkubiczne [1,4,6], dwudzielne regularne i biregularne-$(2,k)$ [4,6,7], kraty [2]) bądź suboptymalny (np. dwudzielne kaktusy [4], grafy mające w jednej partycji co najwyżej 3 wierzchołki [4]). Niestety, znów problem istnienia kolorowania zwartego dla dowolnego grafu dwudzielnego jest NP-zupełny (Sevastianow [8]), co daje NP-zupełność zagadnień $O|no\text{-}wait\&idle,ZUET|\bullet$ oraz $P|no\text{-}wait\&idle,fix_j=2,ZUET|\bullet$. Znane też są rodziny systemów dających się uszeregować bez przestojów wielomianowo, których optymalizacja względem kryteriów C_{max} ([1]) lub ΣC_i jest NP-trudna. Podobne wyniki negatywne prawdziwe są dla systemu przepływowego.

LITERATURA

1. Giaro K.: The complexity of consecutive Δ-coloring of bipartite graphs: 4 is easy, 5 is hard. Ars Combinatoria 47 (1997), pp. 287-300.

2. Giaro K., Kubale M.: Consecutive edge-colorings of complete and incomplete Cartesian products of graphs, Congr. Numer. 128 (1997), pp. 143-149.

3. Giaro K., Kubale M., Małafiejski M.: Szeregowanie zadań jednostkowych w systemie otwartym bez przestojów. Zesz. Nauk. Pol. Śl., Ser. Automatyka 117 (1996), pp. 29-36.

4. Giaro K., Kubale M., Małafiejski M.: Compact scheduling in open shop with zero-one time operations (wysłane do druku).

5. Hall N. G., Sriskandarajah C.: A survey of machine scheduling problems with blocking and no-wait in process. Operation Research 44 (1996), pp. 510-525.

6. Hansen H. M.: Scheduling with minimum waiting periods (po duńsku), Master Thesis, Odense University, Odense, Denmark, 1992.

7. Hanson P., Loten C, Toft B.: On interval colorings of bi-regular bipartite graphs. Ars Combinatoria. (w druku).

8. Sevast'janov S.V.: On interval colorability of a bipartite graph (po rosyjsku). Met. Diskret Analiz. 50 (1990), pp. 61-72.

Recenzent:Prof.dr hab.inż. Józef Grabowski

Abstract

In practical task scheduling it is sometimes required that the elements of a system perform consecutively. Such a scheduling is called no-wait or no-idle. In this article it is shown that the complexity of some simplified problems of this kind remains NP-hard. In particular, the following problems: $O|no\text{-}wait\&idle,M{=}path\ or\ cycle|C_{max}$, $P|no\text{-}wait,fix_j{=}2,M{=}path\ or\ even\_cycle|C_{max}$ and $F|no\text{-}wait\&idle,M{=}path|C_{max}$ are NP-hard. For 2-procesor tasks problem $P|no\text{-}wait,fix_j{=}2,M{=}odd\_cycle|\bullet$ and for flow shop system $F|no\text{-}wait\&idle,M{=}cycle|\bullet$ the corresponding decision problems remain NP-complete.