ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2000
Seria: AUTOMATYKA z. 129 Nr kol. 1474

Michał MAŁAFIEJSKI
Politechnika Gdańska

MINIMALIZACJA ŚREDNIEGO CZASU OBSŁUGI ZADAŃ W SYSTEMIE RÓWNOLEGŁEGO PRZYDZIAŁU ZASOBÓW

Streszczenie. W pracy, obok podsumowania dotychczasowych wyników dotyczących problemu minimalizacji średniego czasu przepływu zadań w systemie równoległego przydziału zasobów, zaprezentowano własności sumy multichromatycznej, w szczególności nowe oszacowania. Ponadto zaproponowano rozszerzenie klasycznej notacji trójpolowej w celu uwzględnienia nowych wyników dotyczących złożoności wybranych problemów szeregowania zadań na podstawie wybranych klas grafów konfliktowych.

SCHEDULING MULTIPROCESSOR TASKS ON DEDICATED PROCESSORS TO MINIMIZE MEAN FLOW TIME

Summary. In the paper we consider the problem of scheduling multiprocessor tasks on dedicated processors. The author extends the notation $\alpha|\beta|\gamma$ to take into consideration new results concerning the complexity of the problem restricted to the special families of *conflicting graphs*. Moreover, some new properties of the multichromatic sum are presented.

1. Wstęp

W związku z rozwojem równoległych i rozproszonych systemów przetwarzania informacji szczególnego znaczenia nabiera model szeregowania zadań z systemem procesorów (maszyn, zasobów) działających niezależnie (równolegle). Rozważać będziemy systemy z zadaniami, dla których ściśle określono podzbiory zbioru procesorów wykonujących dane zadanie równocześnie. Z punktu widzenia zadania procesory będą wtedy zasobami niezbędnymi do jego wykonania. Przy interpretacji procesorów jako zasobów w systemach komputerowych, inne urządzenia peryferyjne, np. drukarki, stacje dysków czy nawet sekcje krytyczne programów, będą w modelu ujmowane jako procesory, czyli jednostki przetwarzające dane zadanie.

W pracy rozważany jest deterministyczny model przydziału zasobów zadaniom zgłaszającym żądania wyłącznego dostępu do dedykowanych zasobów. W zasadniczy sposób różni się on od prezentowanych w literaturze niedeterministycznych rozproszonych systemów synchronizacyjnych, np. szeroko opisywanego *Problemu Pięciu Filozofów* [7, 6, 20], gdzie główny nacisk położony jest na znalezienie rozwiązania eliminującego zjawisko głodowania oraz blokady, przy czym zastosowany algorytm rozwiązujący problem jest często algorytmem rozproszonym, wykonywanym lokalnie przez każdy proces (związany z danym zasobem). W literaturze problem (istnienia, nie optymalizacji) alokacji zasobów z ograniczonym jednoczesnym dostępem, np. [16], dotyczy sekwencyjnego (ang. *collecting resources one by one*) przydziału wszystkich zasobów niezbędnych do wykonania danego zadania bądź równoczesnego (ang. *collecting resources concurrently*), przy założeniu braku procesu zarządzającego przydziałem zasobów.

Prezentowane przez nas podejście opiera się na paradygmacie determinizmu oraz optymalizacji. Zakładamy bowiem pełną znajomość problemu przed jego rozwiązaniem, czyli pełne przyporządkowanie zasobów do zadań. Ponadto nie interesuje nas znalezienie jakiegoś rozwiązania, ale optymalnego, względem ustalonego kryterium optymalizacyjnego.

Systemy równoległe z zadaniami dwuprocesorowymi modelują zagadnienie współpracy procesorów wymaganej do wykonania zadania i znajdują zastosowanie w systemach komunikacyjnych, wymagających jednoczesnego udziału nadawcy i odbiorcy, np. w diagnostycznych systemach wieloprocesorowych [13]. Rozważa się również zadania z większą liczbą predefiniowanych procesorów [3, 2], głównie modele dwu- lub trzyprocesorowe z podzielnymi zadaniami..

Problem minimalizacji długości uszeregowania w systemach równoległych bez ograniczeń kolejnościowych jest obszernie opisany w literaturze, np. [13, 4, 3, 2]. Minimalizacja średniego czasu przepływu, czyli średniego czasu oczekiwania zadania w systemie na obsłużenie (przy założeniu jednoczesnego przybycia zadań do systemu), jest problemem obecnym w literaturze od niedawna [8, 11, 5, 1, 15, 2, 22].

2. Opis problemu i konstrukcja modelu

Rozważmy zbiór n zadań $J=\{J_1,\dots,J_n\}$ oraz m procesorów (zasobów) $M=\{M_1,\dots,M_m\}$. Każdemu z zadań przyporządkowany jest niepusty zbiór $M^j \subset M$

procesorów równolegle wykonujących zadanie J_j w czasie p_j. Zatem dane jest przyporządkowanie $F: J \to 2^M \times R_+$, gdzie $F(J_j) = (M^j, p_j)$ oraz R_+ oznacza zbiór liczb rzeczywistych dodatnich (oś czasu). Naszym celem jest uporządkowanie w czasie wszystkich zadań (ang. *no starvation*) tak, aby żadne dwa zadania nie korzystały jednocześnie ze wspólnego zasobu (ang. *mutual exclusion*). *Uszeregowaniem zadań* nazwiemy każdą funkcję $S: J \to 2^{R_+}$ taką, że $|S(J_j)| = p_j$, dla $j = 1,\dots,n$ oraz $|S(J_j) \cap S(J_k)| > 0 \Rightarrow M^j \cap M^k = \varnothing$, dla $1 \le j \ne k \le n$. Ponadto zakładamy, że każda ze skończonej ilości składowych spójności zbioru $S(J_j)$ jest przedziałem domkniętym. Funkcja S ustala zatem dla każdego zadania przedziały czasowe, w których korzysta z dedykowanych zasobów (procesorów) w sposób wyłączny. Pierwszy warunek określa łączny czas (sumę długości przedziałów) przetwarzania zadania. Drugi warunek wyklucza jednoczesny dostęp zadań do tych samych zasobów.

Definicja 2.1. *Grafem konfliktowym nazywamy graf*

$$G(F) := \left(J, \left\{\{J_i, J_j\} : 1 \le i \ne j \le n \wedge M^i \cap M^j \ne \varnothing\right\}\right).$$

Nietrudno zauważyć, że graf konfliktowy nie wyznacza jednoznacznie funkcji F (w odniesieniu do pierwszej współrzędnej). Wystarczy rozważyć jednozadaniowy zbiór J z różnymi zbiorami predefiniowanych procesorów dla tego samego zadania.

Definicja 2.2. ***Sumą chromatyczną*** *(wierzchołkową) grafu G nazywamy liczbę $\sum(G) := \min_c \sum(G,c)$, gdzie $\sum(G,c) = \sum_{v \in V(G)} c(v)$ oraz $c: V(G) \to N$ jest pokolorowaniem wierzchołkowym grafu G.*

Okazuje się, że dla zadań jednostkowych znalezienie uszeregowania S można równoważnie sprowadzić do pokolorowania wierzchołkowego grafu konfliktowego. Natomiast optymalne uszeregowanie minimalizujące średni czas przepływu odpowiada znalezieniu pokolorowania grafu konfliktowego, realizującego sumę chromatyczną. Równoważność wyznaczona jest przez równość $S(J_j) = [c(J_j) - 1, c(J_j)]$. Poprawność pokolorowania odpowiadającego uszeregowaniu w przypadku zadań jednostkowych $(p_j = 1)$ jest określona przez warunek różności kolorów sąsiednich wierzchołków.

Podsumowując, zebrane zgłoszenia żądań dostępu do zasobów (procesorów) pochodzące od zadań (procesów) bezpośrednio definiują funkcję F, zatem i graf konfliktowy. Zadaniem procesu zarządzającego (rys. 1) jest znalezienie optymalnego uszeregowania zadań, czyli rozdzielenia w czasie żądanych zasobów minimalizując średni czas przepływu albo inaczej - średni czas oczekiwania zadania na obsłużenie.

Rys.1. System optymalizujący równoległy przydział zasobów
Fig.1. The resource allocation system

3. Minimalizacja średniego czasu obsługi zadań jednostkowych

Sprowadzenie problemu optymalnego uszeregowania jednostkowych zadań do konstrukcji pokolorowania realizującego sumę chromatyczną grafu konfliktowego pozwala wykorzystać bogaty aparat teorii grafów. W niniejszym punkcie dokonamy przeglądu podstawowych wyników z punktu widzenia złożoności obliczeniowej, w szczególności rozgraniczymy przypadki wielomianowe oraz trudne obliczeniowo (NP-zupełne).

Trudność zagadnienia w przypadku ogólnym zmusza nas do przyjęcia założeń upraszczających model. W celu przejrzystości zapisu założeń wykorzystamy *notację trójpolową* $\alpha|\beta|\gamma$, w znacznej mierze zgodną z [11]. Wyrażenie fix_j w drugim polu notacji będzie wskazywać na systemy z zadaniami wykonywanymi na procesorach dedykowanych. Dysponując funkcją F można skonstruować graf konfliktowy. Odwrotnie, zakładając szczególną postać grafu konfliktowego dostajemy odpowiadającą mu klasę funkcji. Zatem jednym ze sposobów uproszczenia problemu jest ograniczenie klasy rozpatrywanych grafów konfliktowych. Wstawienie wyrażenia $G = graph$ w drugim polu notacji wskazywać będzie na postać grafu konfliktowego. Brak tego wyrażenia oznaczać będzie, że graf konfliktowy jest dowolny. Kryterium optymalizacyjnym będzie minimalizacja średniego czasu przepływu, co będzie oznaczane wyrażeniem $\sum C_j$ w trzecim polu notacji.

Pojęcie sumy chromatycznej wprowadzone zostało w pracy [14] jako nowa charakteryzacja liczbowa grafu pochodząca od sumy chromatycznej. O ile w przypadku liczby chromatycznej istotna jest liczba kolorów (etykiet przyporządkowanych wierzchołkom), w przypadku sumy chromatycznej istotne są własności algebraiczne etykiet (liczb naturalnych). Podstawowe własności sumy chromatycznej, w szczególności oszacowania dolne i górne, zostały opisane w [14, 21, 9, 17].

Odwołując się do zapisu problemu w notacji trójpolowej, przez $P \mid fix_j, G = class, UET \mid \sum C_j$ oznaczać będziemy problem optymalizacyjny znalezienia uszeregowania minimalizującego średni czas przepływu zadań jednostkowych dla grafów konfliktowych należących do zbioru grafów *class*, natomiast problem decyzyjny istnienia uszeregowania o własności $\sum(G,c) \leq k$ zapisywać będziemy:

$$P \mid fix_j, G = class, UET \mid \sum C_j \leq k.$$

3.1. NP-zupełne problemy istnienia uszeregowania

Korzystając z NP-zupełności problemu sumy chromatycznej dla grafów ogólnych [14, 17] możemy sformułować:

Twierdzenie 3.1. Problem decyzyjny $P \mid fix_j, UET \mid \sum C_j \leq k$ jest NP-zupełny. ■

Okazuje się, że przedstawiony problem pozostaje wciąż NP-zupełny nawet wtedy, gdy grafy konfliktowe ograniczymy do grafów podkubicznych, tzn. grafów z $\Delta \leq 3$.

Twierdzenie 3.2. [17, 18]. Problem decyzyjny $P \mid fix_j, G = subcubic, UET \mid \sum C_j \leq k$ jest NP-zupełny. ■

Powyższy wynik można wzmocnić, ograniczając klasę grafów do grafów kubicznych.

Twierdzenie 3.3. [17, 18]. Problem decyzyjny $P \mid fix_j, G = cubic, UET \mid \sum C_j \leq k$ jest NP-zupełny. ■

Okazuje się, że twierdzenie 3.3 można uogólnić dla innych grafów *k*-regularnych .

Twierdzenie 3.4. [17, 18]. Problem decyzyjny

$P \mid fix_j, G = r - regular, UET \mid \sum C_j \leq k$ jest NP-zupełny dla dowolnej liczby $r \geq 3$. ■

Inną możliwością ograniczenia problemu istnienia uszeregowania sformułowanego w ogólnej postaci dla zadań jednostkowych $P \mid fix_j, UET \mid \sum C_j$, obok rozpatrywania szczególnych postaci grafów konfliktowych, jest przyporządkowanie zadaniom dedykowanych zbiorów maszyn o określonej strukturze. Rozpatrzmy system, w którym każdemu zadaniu przyporządkowany jest dwuelementowy zbiór procesorów, przy czym żadnym dwóm zadaniom nie zostanie przyporządkowana identyczna para procesorów. Systemy takie znajdują zastosowanie w modelowaniu komunikacji między procesorami, np. [13]. Problem znalezienia optymalnego uszeregowania zadań jednostkowych dwuprocesorowych w notacji trójpolowej oznaczać będziemy $P \mid fix_j = 2, M = graph, UET \mid \sum C_j$, gdzie w odróżnieniu od sytuacji ogólnej, zakładając znajomość funkcji F, skonstruujemy graf M, którego graf krawędziowy będzie grafem konfliktowym utworzonym dla funkcji F. Formalnie zatem, $V(M) := \{M_i : i = 1,\dots,m\}$ oraz $E(M) := \{J_i : i = 1,\dots,n\}$.

W pracy [10] pokazano NP-zupełność problemu istnienia uszeregowania zadań dwuprocesorowych dla grafów dwudzielnych podkubicznych.

Twierdzenie 3.5. [10]. Problem decyzyjny istnienia uszeregowania $P \mid fix_j = 2, M = bipartite \,\&\, \Delta \leq 3, UET \mid \sum C_j \leq k$ jest NP-zupełny. ■

3.2. Klasyfikacja problemów wielomianowych

W zastosowaniach niejednokrotnie pojawiają się grafy konfliktowe o regularnej strukturze. Dla niektórych z nich jesteśmy w stanie podać wzór na sumę chromatyczną, dla innych istnieje wielomianowy algorytm wyznaczający ją najczęściej poprzez konstrukcję pokolorowania optymalnego.

Dla podstawowych klas grafów, np. ścieżek, cykli, kół, pełnych dwudzielnych i grafów pełnych można podać szczegółowe wzory na wartości sumy chromatycznej (zob. np. [17]). Podobnie dla grafów dwudzielnych regularnych zachodzi następujące

Twierdzenie 3.6. Niech B_k będzie spójnym dwudzielnym grafem regularnym stopnia k. Wtedy $\sum(B_k) = \frac{3 \cdot n}{2}$.

Dowód. Kolorując graf B_k dwoma kolorami 1, 2 dostajemy $\sum(B_k) \leq \frac{3 \cdot n}{2}$. Niech c będzie pokolorowaniem wykorzystującym co najmniej trzy kolory. Nietrudno zauważyć, że liczba wierzchołków w każdej partycji jest jednakowa i wynosi $\frac{n}{2}$ (liczba krawędzi wychodząca z każdej składowej jest identyczna). Przez $k_{i,j}$ oznaczymy liczbę krawędzi łączących wierzchołki ze zbiorów $C_i := c^{-1}(\{i\})$ oraz C_j, ponadto każdej krawędzi

$\{v_i, v_j\} \in E(B_k)$ $(v_i \in C_i, v_j \in C_j)$ przyporządkujemy wagę równą $i + j$. Ponieważ $\sum_{1 \le i < j \le \max c(V)} k_{i,j} = m = k \cdot \frac{n}{2}$, więc $k \cdot \sum(B_k, c) = \sum_{1 \le i < j \le \max c(V)} (i + j) \cdot k_{i,j} > 3 \cdot k \cdot \frac{n}{2}$, skąd $\sum(B_k, c) > 3 \cdot \frac{n}{2}$. ■

Wniosek 3.7. Dla dowolnego grafu dwudzielnego k-regularnego każde optymalne pokolorowanie wykorzystuje co najwyżej dwa kolory. ■

Okazuje się, że można dokładniej scharakteryzować grafy dwudzielne, których każde optymalne pokolorowanie korzysta z co najwyżej dwóch kolorów.

Twierdzenie 3.8. Jeżeli w spójnym grafie dwudzielnym G istnieje skojarzenie pokrywające jedną z dwóch partycji, to każde optymalne pokolorowanie grafu G korzysta z dwóch kolorów oraz $\sum(G) = n_1 + n_2 + \min\{n_1, n_2\}$, gdzie n_i jest liczbą wierzchołków partycji V_i.

Dowód. Niech A będzie skojarzeniem pokrywającym partycję V_1 oraz zbiór $V_2^A \subset V_2$, ponadto niech c będzie pokolorowaniem optymalnym wykorzystującym więcej niż dwa kolory. Wtedy z własności skojarzenia dostajemy:

$$\sum(G,c) = \sum_{\{v_1,v_2\} \in A} (c(v_1) + c(v_2)) + \sum_{v \in V_2 \setminus V_2^A} c(v) > 3 \cdot n_1 + n_2 - n_1 = n_2 + 2 \cdot n_1,$$

skąd wobec istnienia pokolorowania kolorami 1, 2 o sumie równej $n_2 + 2 \cdot n_1$ dostajemy tezę. ■

Wiedząc, że najliczniejsze skojarzenie w dowolnym grafie można znaleźć w czasie wielomianowym (zob. [19]), powyższą klasę grafów możemy zaliczyć do przypadków wielomianowych.

Nie dla wszystkich grafów istnieje prosta zależność między wartością sumy chromatycznej a liczbą wierzchołków lub krawędzi. Przykładem są drzewa, dla których można pokazać

Twierdzenie 3.9. [14]. Dla dowolnego drzewa T o $n \ge 1$ wierzchołkach zachodzi $n + 1 \le \sum(T) \le \left\lfloor \frac{3 \cdot n}{2} \right\rfloor$. Ponadto dla dowolnej całkowitej wartości k z tego przedziału istnieje drzewo T o n wierzchołkach, dla którego $\sum(T) = k$. ■

Twierdzenie 3.10. [14]. Istnieje algorytm liniowy znajdujący optymalne pokolorowanie dowolnego drzewa. ■

W grafach gęstych, w których maksymalny zbiór niezależny ma co najwyżej dwa wierzchołki (dopełnienia grafów nie zawierających trójkątów (*triangle-free graphs*)) znajdując najliczniejsze skojarzenie A w grafie dopełniającym dostaniemy pokolorowanie optymalne o sumie $\sum_{i=1}^{|A|} 2\cdot i + \sum_{i=|A|+1}^{|V|-|A|} i$, skąd dostajemy

Twierdzenie 3.11. [12]. Optymalne pokolorowanie grafów będących dopełnieniami grafów bez trójkątów można znaleźć w czasie $O(m\cdot\sqrt{n})$. ■

W przypadku systemów z zadaniami dwuprocesorowymi skonstruowano algorytm wielomianowy dla drzew.

Twierdzenie 3.12. [10]. Istnieje algorytm znajdujący optymalne uszeregowanie zadań dla problemu $P\,|\,fix_j = 2, M = tree, UET\,|\sum C_j$ w czasie $O(n^{4.5}\cdot\log n)$. ■

4. Suma multichromatyczna

Modelowanie systemów z zadaniami niejednostkowymi o całkowitych długościach wymaga rozszerzenia klasycznej definicji pokolorowania wierzchołkowego.

Definicja 4.1. *Multipokolorowaniem wierzchołkowym nazywamy dowolną funkcję $c: V(G)\to 2^N$ spełniającą warunek, że dla każdej pary sąsiednich wierzchołków $v,w\in V(G)$ skończone zbiory $c(v)$ oraz $c(w)$ są rozłączne. Multipokolorowanie c nazywamy **zwartym multipokolorowaniem wierzchołkowym**, jeżeli każdy zbiór $c(v)$ jest interwałem, czyli $|c(v)| = \max c(v) - \min c(v) + 1$.*

Definicja 4.2. *Funkcją krotności (**krotnością**) grafu G nazywamy dowolną funkcję $p: V(G)\to N$. Dla ustalonej krotności p (zwartą) **sumą multichromatyczną** nazywamy liczbę $\sum_p(G) := \min_c \sum_p(G,c)$, gdzie $\sum_p(G,c) = \sum_{v\in V(G)} \max c(v)$ oraz c jest (zwartym) multipokolorowaniem grafu G spełniającym warunek $|c(v)| = p(v)$. Zwartą sumę multichromatyczną będziemy oznaczać symbolem $\sum_p^I(G)$. W przypadku, gdy funkcja p jest stale równa k, będziemy pisać $\sum_{p=k}(G)$ lub $\sum_k(G)$.*

Znalezienie uszeregowania podzielnych zadań całkowitych można równoważnie sprowadzić do znalezienia multipokolorowania grafu konfliktowego. Optymalne uszeregowanie odpowiada multipokolorowaniu realizującemu sumę multichromatyczną.

Równoważność wyznaczają równości $p(J_j)=p_j$ oraz $S(J_j)=\bigcup_{a\in c(J_j)}[a-1,a]$ dla $j=1,\dots,n$. Uszeregowaniom zadań niepodzielnych odpowiadają zwarte multipokolorowania grafu konfliktowego.

Niech dana będzie funkcja krotności $p:V(G)\to N$. Multipokolorowanie c grafu G, dla którego $|c(v)|=p(v)$ oznaczać będziemy c_p. *Multipokolorowaniem optymalnym* c_p grafu G nazywać będziemy każde multipokolorowanie realizujące sumę $\sum_p(G)$. Ponadto przez $p(G)$ oznaczać będziemy sumę $\sum_{v\in V(G)}p(v)$, natomiast dla ustalonego słabego liniowego porządku $\prec$ w zbiorze $V(G)$ przez $p(G,\prec)$ oznaczać będziemy sumę $\sum_v\sum_{w\prec v}p(w)$. Każdy porządek $\prec$, dla którego zachodzi $v\prec w\Rightarrow p(v)\le p(w)$, nazwiemy porządkiem *SF* (*shortest first*) i oznaczymy $\prec_{SF}$.

Fakt 4.1. $p(G,\prec)$ osiąga wartość najmniejszą dla $\prec=\prec_{SF}$. ■

Twierdzenie 4.2. Dla dowolnej krotności p oraz dowolnego grafu G zachodzi:

$$p(G)\le\sum_p(G)\le\sum_p^{I}(G)\le p(G,\prec_{SF})\le\frac{n+1}{2}\cdot p(G).$$

Dowód. Niech $n=n(G)$.

Lewa nierówność jest konsekwencją faktu $\max c(v)\ge|c(v)|=p(v)$, dla $v\in V(G)$, natomiast $\sum_p(G)\le\sum_p^{I}(G)$ jest oczywista. Niech $(v_1,\dots,v_n)$ będzie porządkiem $\prec_{SF}$. Zauważmy, że przydzielając v_j zbiór $\{\sum_{i=1}^{j-1}p(v_i)+1,\dots,\sum_{i=1}^{j}p(v_i)\}$ dostaniemy legalne multipokolorowanie o własności $\sum_p(G,c_p)\le\sum_{j=1}^{n}\sum_{i=1}^{j}p(v_i)=p(G,\prec_{SF})$, przy czym $p(G,\prec_{SF})$ nie zależy od porządku $\prec_{SF}$. Ponadto średnia arytmetyczna $p(G,\prec_{SF})$ oraz $p(G,\prec_{LF})$, gdzie *LF* (*largest first*) jest porządkiem odwrotnym do *SF*, wynosi $\frac{n+1}{2}\cdot p(G)$ i jest nie mniejsza niż $p(G,\prec_{SF})$. ■

W [M1] można znaleźć konstrukcje grafów rozstrzygające, że $\sum_p(G)$ oraz $\sum_p^{I}(G)$ są istotnie różne.

Przez $p:=p_1+p_2$ będziemy oznaczać krotność, dla której $p(v)=p_1(v)+p_2(v)$.

Twierdzenie 4.3. Niech p_1 oraz p_2 będą dowolnymi krotnościami grafu G. Wtedy

$$\frac{1}{2}\cdot\left(\sum_{p_1}(G)+\sum_{p_2}(G)\right)+\frac{1}{2}\cdot(p_1+p_2)(G)\leq\sum_{p_1+p_2}(G).$$

Dowód. Wystarczy pokazać, że $\sum_{p_1}(G)+p_2(G)\leq\sum_{p_1+p_2}(G)$. Niech $c_{p_1+p_2}$ będzie multipokolorowaniem grafu G. Nietrudno zauważyć, że biorąc z każdego zbioru $c_{p_1+p_2}(v)$ pierwsze $p_1(v)$ elementów dostaniemy multipokolorowanie c_{p_1} takie, że $\max c_{p_1}(v)\leq\max c_{p_1+p_2}(v)-p_2(v)$, skąd mamy tezę. ∎

Okazuje się, że istnieją drzewa T,T' oraz odpowiednio krotności p_1,p_2,p_1',p_2' takie, że $\sum_{p_1}(T)+\sum_{p_2}(T)<\sum_{p_1+p_2}(T)$ oraz $\sum_{p_1'}(T')+\sum_{p_2'}(T')>\sum_{p_1'+p_2'}(T')$. Konstrukcję można znaleźć w [17].

Twierdzenie 4.4. Dla dowolnej krotności $p\equiv k\in N$ oraz grafu G mamy:

$$\sum_p^I(G)=k\cdot\sum(G).$$

Dowód. Niech c będzie pokolorowaniem optymalnym grafu G. Wtedy $c_p(v):=\{(c(v)-1)\cdot k+1,\ldots,(c(v)-1)\cdot k+k\}$ jest legalnym zwartym multipokolorowaniem grafu G z krotnością tożsamościowo równą k. Weźmy bowiem dowolne sąsiednie wierzchołki v i w oraz załóżmy, że $c(v)>c(w)$, skąd $c(v)\geq c(w)+1$.

Załóżmy przeciwnie, że $(c(v)-1)\cdot k+i=(c(w)-1)\cdot k+j$, dla $1\leq i,j\leq k$. Wtedy $j-i=k\cdot(c(v)-c(w))\geq k$, co jest niemożliwe. Zatem $\sum_p^I(G)\leq k\cdot\sum(G)$. Odwrotnie, niech c_p będzie optymalnym zwartym multipokolorowaniem grafu G. Zdefiniujemy pokolorowanie grafu G jak następuje $c(v):=\left\lfloor\frac{\max c_p(v)}{k}\right\rfloor$. Widać, że $k\cdot c(v)=k\cdot\left\lfloor\frac{\max c_p(v)}{k}\right\rfloor\leq\max c_p(v)$, skąd $k\cdot\sum(G)\leq\sum_p^I(G)$. Wystarczy jeszcze zauważyć, że c jest legalnym pokolorowaniem grafu G. ∎

5. Podsumowanie

Tabela 1 zawiera charakteryzację złożoności obliczeniowej problemu minimalizacji średniego czasu obsługi dla wybranych klas grafów (konfliktowych).

Tabela 1

Klasyfikacja złożoności problemu minimalizacji średniego czasu przepływu

Graf konfliktowy G	Problem	Złożoność	Referencja
dowolny	$P \mid fix_j, UET \mid \sum C_j \leq k$	NPC	Tw. 3.1. [14]
podkubiczny	$P \mid fix_j, G = subcubic, UET \mid \sum C_j \leq k$	NPC	Tw. 3.2. [18]
kubiczny	$P \mid fix_j, G = cubic, UET \mid \sum C_j \leq k$	NPC	Tw. 3.3. [18]
r-regularny	$P \mid fix_j, G = r - regular, UET \mid \sum C_j \leq k$	NPC	Tw. 3.4. [18]
* dwudzielny, podkubiczny	$P \mid fix_j = 2, M = bipartite \,\&\, \Delta \leq 3, UET \mid \sum C_j \leq k$	NPC	Tw. 3.5. [10]
$P_n, C_n, W_n, B_k, K_{r,s}, K_n$	$P \mid fix_j, G, UET \mid \sum C_j$	$O(n)$ lub $O(n^2)$	[17]
dwudzielny ze skojarzeniem	$P \mid fix_j, G, UET \mid \sum C_j$	$O(m \cdot \sqrt{n})$	Tw. 3.8.
drzewa	$P \mid fix_j, G = tree, UET \mid \sum C_j$	$O(n)$	Tw. 3.10. [14]
dopełnienie jest bez trójkątów	$P \mid fix_j, G, UET \mid \sum C_j$	$O(m \cdot \sqrt{n})$	Tw. 3.11. [12]
* drzewa * dotyczy grafu M	$P \mid fix_j = 2, M = tree, UET \mid \sum C_j$	$O(n^{4.5} \cdot \log n)$	Tw. 3.12. [10]

LITERATURA

1. Bar-Noy A., Shachnai H., Tamir T.: On chromatic sums and distributed resource allocation, The 4th Israeli Symp. On Theory of Computing and Systems (ISIC'96), 1996, pp.119-128.
2. Bianco L., Błażewicz J., Dell'Olmo P., Drozdowski M.: Preemptive multiprocessor task scheduling with release times and times windows, Annals of Operations Research 70, 1997, pp.43-55.
3. Błażewicz J., Dell'Olmo P., Drozdowski M., Speranza M.G.: Scheduling multiprocessor tasks on three dedicated processors, Information Processing Letters 41, 1992, pp.275-280.

4. Błażewicz J., Drabowski M., Węglarz J.: Scheduling multiprocessor tasks to minimize schedule length, IEEE Transactions on Computers C-35, 1986, pp.389-393.
5. Brücker P., Kramer A.: Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems, European Journal of Operational Research 90, 1996, pp.214-226.
6. Chandy K.M., Misra J.: The drinking philosophers problem, ACM Trans. Programing Languages Syst. 6, 1984, pp.632-646.
7. Dijkstra E.W.: Hierarchical ordering of sequential processes, Acta Informatica 1(2), 1971, pp.115-138.
8. Dobson G., Karmarkar U.S.: Simultaneous resource scheduling to minimize weighted flow times, Operational Research 37, 1989, pp.592-600.
9. Erdös P., Kubicka E., Schwenk A.J.: Graphs that require many colors to achieve their chromatic sum, Congressus Numerantium 71, 1990, pp.17-28.
10. Giaro K., Kubale M, Małafiejski M., Piwakowski K.: Chromatic scheduling of 2-dedicated UET tasks to minimize mean flow time, Emerging Technologies and Factory Automation (EFTA'99), 1999.
11. Hoogeven J.A., van de Velde S.L., Veltman B.: Complexity of scheduling multiprocessor tasks with prespecified processor allocations, Discrete Applied Mathematics 55, 1994, pp.259-272.
12. Jansen K.: Complexity results for the optimum cost chromatic partition problem, Forschungsbericht Nr. 96-41, Universität Trier, 1996.
13. Krawczyk H., Kubale M.: An approximation algorithm for diagnostic test scheduling in multiprocessor systems, IEEE Transactions on Computers C-34, 1985, pp.869-872.
14. Kubicka E., Schwenk A.J.: An introduction to chromatic sums, Proceedings of ACM Computer Science Conference, 1989, pp.39-45.
15. Lee C.Y., Lei L., Pinedo M.: The current trend of deterministic scheduling, Annals of Operations Research 70, 1997, pp.1-41.
16. Lynch N.: Upper bounds for static resource allocation in a distributed system, Journal of Computer and System Sciences 23, 1981, pp.254-278.
17. Małafiejski M.: Uszeregowania zadań wieloprocesorowych minimalizujące średni czas przepływu, Praca dyplomowa, Politechnika Gdańska 1999.
18. Małafiejski M.: The complexity of the chromatic sum problem on subcubic, cubic and regular graphs, Discrete Applied Mathematics, 2000 (praca wysłana), pp.1-13.
19. Micali S., Vaziriani V.V.: An $O(m \cdot \sqrt{n})$ algorithm for finding maximum matching in general graphs, 21st Ann. IEEE Symp. On Found. Of Comp. Sci., 1980, pp.17-27.
20. Papatriantafilou M., Tsigas P.: On distributed resource handling: dining, drinking and mobile philosophers, OPODIS'97, 1997, pp.293-307.
21. Thomassen C., Erdös P., Alavi Y., Malde P.J., Schwenk A.J.: Tight bounds on the chromatic sum of a connected graph, Journal of Graph Theory 13, 1989, pp.353-357.
22. Xiaoqiang C., Lee C.-Y., Li C.-L.: Minimizing total completion time in two-processor tasks systems with prespecified processor allocations, Naval Research Logistics 45, 1998, pp.231-242.

Recenzent: Prof.dr hab.inż. J.Grabowski

Abstract

In the paper the author considers the following problem: Given n jobs and m processors (resources), each job requires exclusively some dedicated processors (resources) to be performed. The competition of tasks collecting resources concurrently is represented by the *conflicting graph* with tasks represented as vertices which are adjacent if and only if there exists conflicting resource requirement. Our goal is to schedule all tasks minimizing their average completion time assuming that no two conflicting tasks are executed simultaneously. The minimized criterion is important from the point of view of tasks, in contrast to the makespan, which favors the system.

The author investigates the computational complexity of the problem subject to special families of conflicting graphs and presents some new polynonomial and NP-complete results. Scheduling arbitrary jobs is modeled by the graph multicoloring problem. The lower and upper bounds on the multichromatic sum are presented.