ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 134 Nr kol. 1554

Robert JANCZEWSKI, Michał MALAFIEJSKI
Politechnika Gdańska

ALGORYTMY PRZYBLIŻONE DLA WYBRANYCH PROBLEMÓW RÓWNOLEGŁEGO PRZYDZIAŁU ZASOBÓW

Streszczenie. W pracy analizowane są własności uszeregowań dla wprowadzonego modelu systemu równoległego. W szczególnym przypadku, dla systemów równoległych z zadaniami dedykowanymi, ustaloną alokację zasobów możemy zamodelować wykorzystując tzw. grafy konfliktów. W chromatycznym modelu uporządkowań w czasie możemy sprowadzić znajdowanie uszeregowań optymalnych do znajdowania tych spośród multipokolorowań grafu konfliktów, których suma kolorów jest minimalna. Autorzy podają wstępną analizę własności modelu, wprowadzają algorytmy zachłanne oraz na podstawie uzyskanych lub cytowanych oszacowań pokazują algorytmy przybliżone.

APPROXIMATION ALGORITHMS FOR SELECTED PROBLEMS OF PARALLEL RESOURCE ALLOCATION

Summary. In the paper we consider the problem of scheduling multiprocessor tasks on dedicated processors. Assuming that there is only one alocation of resources the proposed general model can be reduced to the chromatic model of sequencing of tasks. The authors analyze the properties of the sum multicoloring problem of conflicting graphs and give some new bounds on the (multi)chromatic sum. Based on the introduced greedy algorithm we propose approximation algorithms to the problem $P|fix_j, G, r_j|\sum C_j$.

1. Definicja problemu

1.1. Wstęp

W pracy autorzy koncentrują się na analizie własności chromatycznego modelu uporządkowań w czasie zaproponowanego dla szczególnych przypadków deterministycznego szeregowania wieloprocesorowych zadań w środowisku maszyn działających równolegle. W ogólnym modelu rozpatruje się skończone zbiory zadań i procesorów. Zadania mogą

być wykonywane na procesorach dowolnych [4], dedykowanych [8, 10], czy na alternatywnych zbiorach procesorów dedykowanych [3, 6]. Przy interpretacji procesorów jako zasobów, w systemach komputerowych, inne urządzenia (np. zasoby pamięci, urządzenia peryferyjne, procesory specjalizowane czy sekcje krytyczne programów) będą w modelu traktowane jako procesory, czyli jednostki przetwarzające dane zadanie. Rozważać będziemy zatem systemy, w których zadania wymagają wyłącznego dostępu do procesorów określonego typu. W *systemach równoległych*, które będziemy dalej rozważać, zadanie będzie wykonywane na kilku procesorach równocześnie. Ponieważ przyjmiemy, że zadania będą autonomiczne (niezależne), nie wydaje się celowe ustalanie kolejności ich wykonywania. W pracy uwzględnione zostaną zarówno systemy z zadaniami jednostkowymi, jak również systemy z zadaniami o całkowitych czasach wykonania, zarówno podzielnych, jak i niepodzielnych.

Model będzie *zorientowany na zadania*. Po pierwsze, podstawowym warunkiem poprawności uszeregowania czasowego będzie wykonanie wszystkich zadań. Po drugie, każde zadanie będzie wymagało wyłącznego dostępu do zasobów określonego typu. Wreszcie najważniejsze, kryterium optymalizacyjne, czyli minimalizacja średniego czasu przepływu zadań, jest istotne właśnie z punktu widzenia zadań, które przybywając niezależnie do systemu chcą yć obsłużone jak najszybciej. Rówoważnie do średniego czasu przepływu można badać całkowitą sumę czasów zakończenia zadań.

Zadanie nazywamy *podzielnym* (ang. *preemptable*), jeżeli jego wykonywanie może zostać przerwane i później wznowione bez żadnych dodatkowych kosztów. W przeciwnym razie zadanie jest niepodzielne (ang. *nonpreemptable*). *Czasem wykonywania* zadania (ang. *processing time*) będziemy nazywać czas zawłaszczenia zasobów niezbędnych do jego wykonania. W przypadku alternatywnych zbiorów procesorów dedykowanych czasy wykonywania danego zadania mogą być różne. *Czas gotowości* zadania (ang. *ready time*) ogranicza z dołu moment przydzielenia zasobów dla tego zadania i jest często interpretowany jako moment przybycia zadania do systemu. W dalszej części pracy zakładać będziemy, że czasy wykonywania oraz gotowości są całkowite (wielokrotność czasu jednostkowego). *Czasem przepływu* zadania przez system nazywać będziemy różnice pomiędzy czasem zakończenia a czasem gotowości.

1.2. Model ogólnego systemu równoległego

Rozważmy zbiór n zadań $\mathcal{J} = \{J_1, ..., J_n\}$ oraz m procesorów $\mathcal{M} = \{M_1, ..., M_m\}$. *System równoległy* w ogólnej postaci zdefiniujemy jako $\mathcal{Q} = (\mathcal{J}, \mathcal{M}, (\mathcal{R}_J)_{J\in\mathcal{J}}, (P_J)_{J\in\mathcal{J}}, (r_J)_{J\in\mathcal{J}}, (w_J)_{J\in\mathcal{J}})$, gdzie $\mathcal{J}$ (ang. *jobs*) jest niepustym skończonym zbiorem autonomicznych zadań (bez ograniczeń kolejnościowych) oraz niepodzielnych na operacje, $\mathcal{M}$ (ang. *machines*) jest niepustym skończonym zbiorem maszyn (procesorów czy innych zasobów), natomiast rodzina $(\mathcal{R}_J)_{J\in\mathcal{J}}$ niepustych podzbiorów $2^{2^{\mathcal{M}}}$ przyporządkowuje każdemu zadaniu J zbiór d_J-elementowy $\mathcal{R}_J = \{D_{J,i}\}_{i=1}^{d_J}$ iepustych podzbiorów $2^{\mathcal{M}}$ (ang. *resources*), czyli alternatywnych zbiorów procesorów dedykowanych (ang. *alternative dedicated processors set*). Kolejne dwa parametry systemu $\mathcal{Q}$ odnoszą się do czasów przyporządkowanych danemu zadaniu. Odwzorowanie $P_J : \mathcal{R}_J \to \mathbb{N}$ przyporządkowuje każdemu zbiorowi zasobów dedykowanych $D_{J,i} \in \mathcal{R}_J$ dodatni całkowity czas wykonywania $p_{J,i}$. Ponadto każdemu zadaniu J przyporządkowany jest czas otowości $r_J \in \mathbb{N}_0$, gdzie $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Wreszcie ostatni element systemu $\mathcal{Q}$ zero-jedynkowy ciąg $(w_J)_{J\in\mathcal{J}}$ definiuje odzielność każdego z zadań, mianowicie wartość 1 lub 0 na pozycji w_J oznacza odpowiednio podzielność, bądź niepodzielność zadania J.

Przyjmiemy, że $|\cdot|$ jest standardową miarą Lebesgue'a określoną m.in. na domkniętych podzbiorach $\mathbb{R}$, co dla przedziału sprowadza się do policzenia jego długości (dla sumy rzedziałów, odpowiednio sumy długości). Zbiór wszystkich liczb rzeczywistych nieujemnych oznaczymy przez $\mathbb{R}_+$.

Definicja 1. *Uszeregowaniem zadań (ang. scheduling) w systemie równoległym* $\mathcal{Q} = (\mathcal{J}, \mathcal{M}, (\mathcal{R}_J)_{J\in\mathcal{J}}, (P_J)_{J\in\mathcal{J}}, (r_J)_{J\in\mathcal{J}}, (w_J)_{J\in\mathcal{J}})$ *nazywamy każde odwzorowanie* $S : \mathcal{J} \to \bigcup_{J\in\mathcal{J}} \mathcal{R}_J \times 2^{\mathbb{R}_+}$ *spełniające warunki*

1. $\forall_{1\le j\le n}\ S_1(J_j) \in \mathcal{R}_j$,
2. $\forall_{1\le j\le n}\ |S_2(J_j)| = P_j(S_1(J_j))$,
3. $\forall_{1\le j\ne k\le n}\ |S_2(J_j) \cap S_2(J_k)| > 0 \Rightarrow S_1(J_j) \cap S_1(J_k) = \emptyset$,
4. $\forall_{1\le j\le n}\ \min S_2(J_j) \ge r_j$,
5. $\forall_{1\le j\le n}\ w_j = 0 \Rightarrow \max S_2(J_j) - \min S_2(J_j) = |S_2(J_j)|$,

gdzie $S(J) = (S_1(J), S_2(J))$. *Pierwszą składową* $S_1(\cdot)$ *nazywać będziemy* ***alokacją zasobów***, *natomiast drugą składową* $S_2(\cdot)$ ***uporządkowaniem w czasie***. *Ponadto zakładamy, że każda ze skończonej liczby składowych spójności zbioru* $S_2(J_j)$ *jest przedziałem domkniętym o nieujemnych całkowitych końcach.*

Odwzorowanie S ustala dla każdego zadania J przedziały czasowe $(S_2(J))$, w których zadanie korzysta z przyporządkowanych zasobów $(S_1(J))$ w sposób wyłączny. Pierwszy warunek definicji specyfikuje zasoby zawłaszczane przez dane zadanie, drugi warunek określa łączny czas (sumę długości przedziałów) przetwarzania adania. Kolejny warunek wyklucza jednoczesny dostęp zadań do tych samych zasobów. Czwarty warunek ogranicza z dołu moment rozpoczęcia wykonywania zadania, wreszcie ostatni warunek zawiera restrykcję dotyczącą możliwości wykonywania zadania "na raty", czyli podzielności.

W dalszej części badać będziemy systemy równoległe o jednoelementowych zbiorach $\mathcal{R}_j$, czyli tzw. *dedykowane systemy równoległe*, w których istnieje dokładnie jedna alokacja zasobów. Uszeregowanie zadań w dedykowanych systemach równoległych upraszcza się wtedy do uporządkowania zadań w czasie, co można równoważnie sprowadzić do multipokolorowania odpowiednio skonstruowanego grafu konfliktów. Okazuje się, że badanie własności uszeregowań w czasie z wykorzystaniem modeli teoriografowych pozwala z jednej strony zredukować liczbę instancji problemu (grafy konfliktów reprezentują w pewnym sensie zbiór systemów równoległych o identycznych konfliktach zadań w dostępie do współdzielonych zasobów), z drugiej umożliwia przeprowadzenie analizy modelu w efektywny sposób. Szerzej własności modelu systemu równoległego wraz z obszerną klasyfikacją problemów szeregowania zadań wieloprocesorowych podano w [14].

2. Chromatyczny model uporządkowań w czasie

Czwórkę (G, p, r, w) nazywać będziemy *grafem obciążonym*, jeżeli G jest grafem prostym, p jest odwzorowaniem przyporządkowującym każdemu wierzchołkowi v grafu G dodatnią całkowitą wagę $p(v)$. Odwzorowanie p nazywać będziemy *funkcją krotności* (*krotnością*) lub *obciążeniem* (wierzchołkowym) grafu, natomiast liczbę $p(v)$ nazywać będziemy *krotnością wierzchołka* $v \in V(G)$ lub jego *wagą*. Ponadto dany jest wektor nieujemnych liczb całkowitych $(r(v))$ oraz wektor zero-jedynkowy $(w(v))$.

Definicja 2. *Multipokolorowaniem wierzchołkowym (krawędziowym) grafu prostego G nazywamy dowolną funkcję $c : V(G) \to 2^N$ ($c : E(G) \to 2^N$) spełniającą warunek, że dla dowolnej pary sąsiednich wierzchołków (krawędzi) a, b skończone zbiory $c(a)$ oraz $c(b)$ są rozłączne. Multipokolorowanie c nazywamy **zwartym**, jeżeli każdy zbiór $c(a)$ jest interwałem, czyli $|c(v)| = \max c() - \min c(v) + 1$. Jeżeli wszystkie zbiory $c(a)$ są jednoelementowe, to c nazywamy **pokolorowaniem**. Elementy zbioru $c(a)$ nazywać będziemy **kolorami**.*

Definicja 3. *Dany jest graf obciążony (G, p, r, w). Każde multipokolorowanie c grafu G spełniające poniższe warunki:*

1. $|c(v)| = p(v)$,
2. $\min c(v) > r(v)$,
3. $c(v)$ *jest interwałem, o ile* $w(v) = 0$,

*nazywamy **multipokolorowaniem grafu obciążonego** (G, p, r, w). **Sumą multichromatyczną** nazywać będziemy*

$$\sum(G, p, r, w) := \min_c \sum(G, p, r, w; c),$$

*gdzie $\sum(G, p, r, w; c) := \sum_{v \in V(G)} \max c(v)$ oraz c jest multipokolorowaniem grafu obciążonego (G, p, r, w). Jeżeli $\sum(G, p, r, w; c) = \sum(G, p, r, w)$, to c nazywać będziemy **minimalnym multipokolorowaniem**.*

Własności sumy chromatycznej oraz multichromatycznej oraz jej związki z szeregowaniem zadań rozważane były w [1, 2] oraz w [11, 12, 13]. Szczegółowy model przedstawiono w [14].

Definicja 4. *Grafem konfliktów skojarzonym z alokacją zasobów S_1 w systemie równoległym Q nazywamy graf prosty*

$$G(Q, S_1) := (\mathcal{J}, \{\{J_i, J_j\} : 1 \le i \ne j \le n \wedge S_1(J_i) \cap S_1(J_j) \ne \emptyset\}).$$

Związek pomiędzy uporządkowaniami w czasie oraz multipokolorowaniami grafu konfliktów dany jest przez następujące równania (zob. [14]):

1. $c(J_j) = \{a \in I\!N : [a-1, a] \subset S_2(J_2)\}$,

2. $S_2(J_j) = \bigcup_{a \in c(J_j)} [a-1, a]$.

Twierdzenie 1. *Dany jest graf obciążony (G, p), ciąg nieujemnych liczb całkowitych $r = (r(v))_{v \in V(G)}$ oraz zero-jedynkowy wektor $w = (w(v))_{v \in V(G)}$. Dla dowolnego minimalnego multipokolorowania c oraz dowolnego wierzchołka $v \in V(G)$ zachodzi*

$$\max c(v) \leq w(v) \cdot a_1(v) + (1 - w(v)) \cdot a_0(v), \tag{1}$$

gdzie $a_1(v) := r(v) + p_{sum}(v)$, natomiast $a_0(v) := r(v) + p(v) \cdot (p_{sum}(v) - p(v) + 1)$, przy czym $p_{sum}(v) := p(v) + \sum_{u \in N(v)} p(u)$.

Dowód. Zauważmy, że powyższy warunek (1) jest równoważny układowi nierówności

$$\begin{aligned} \max c(v) &\leq a_1(v) = r(v) + p_{sum}(v), \text{ dla } w(v) = 1, \\ \max c(v) &\leq a_0(v) = r(v) + p(v) \cdot (p_{sum}(v) - p(v) + 1), \text{ dla } w(v) = 0. \end{aligned}$$

Niech $\max c(v) > a_1(v)$ dla pewnego wierzchołka $v \in V(G)$, dla którego $w(v) = 1$. Oznaczmy $x_a := 1 - |c_N(v) \cap \{a\}|$, gdzie $c_N(v) := \bigcup_{u \in N(v)} c(u)$. Ponieważ

$$|c_N(v)| \leq \sum_{u \in N(v)} |c(u)| = \sum_{u \in N(v)} p(u) = p_{sum}(v) - p(v),$$

więc $\sum_{a=r(v)+1}^{r(v)+p_{sum}(v)} x_a \geq p_{sum}(v) - |c_N(v)| \geq p(v)$. Zdefiniujmy multipokolorowanie $c'(u) = c(u)$, dla $u \neq v$ oraz $c'(v) = \{a \geq r(v) + 1 : x_a = 1 \wedge a \leq b\}$, gdzie b spełnia warunek $\sum_{a=r(v)+1}^{b} x_a = p(v)$. Nietrudno zauważyć, że c' jest dobrze określone, czyli $|c'(v)| = p(v)$ oraz $\max c'(v) = b \leq a_1(v) < \max c(v)$, co daje sprzeczność z minimalnością multipokolorowania c.

Niech teraz $\max c(v) > a_0(v)$ dla pewnego wierzchołka $v \in V(G)$, dla którego $w(v) = 0$. Niech $x_a := |c_N(v) \cap \{a\}|$ dla $a \geq r(v) + 1$ oraz $x_{r(v)} := 1$, ponadto oznaczmy $y_a := \min\{b \geq 0 : x_{a-b} = 1\}$. Jeżeli istnieje $r(v) + p(v) \leq a \leq a_0(v)$ takie, że $y_a \geq p(v)$, to istnieje dobrze określone multipokolorowanie c' równe c dla $u \neq v$ oraz $c'(v) := \{a - p(v) + 1, ..., a\}$, dla którego $\max c'(v) \leq a_0(v) < \max c(v)$, co jest sprzeczne z minimalnością multipokolorowania c. Załóżmy zatem, że $y_a < p(v)$ dla dowolnego $r(v) + p(v) \leq a \leq a_0(v)$. Wtedy dla dowolnych $1 \leq i \neq j \leq p_{sum}(v) - p(v) + 1$ mamy, że $a_i - y_{a_i} \neq$

$a_j - y_{a_j}$ oraz $x_{a_i - y_{a_i}} = x_{a_j - y_{a_j}} = 1$, gdzie $a_i = i \cdot p(v) + r(v)$ oraz $a_j = j \cdot p(v) + r(v)$, skąd $\sum_{b \in B} x_b = p_{sum}(v) - p(v) + 1$, gdzie $B := \{a_i - y_{a_i} : 1 \leq i \leq p_{sum}(v) - p(v) + 1\}$. Z drugiej strony $\sum_{b \in B} x_b \leq |c_N(v)| \leq p_{sum}(v) - p(v)$, sprzeczność. □

Analizując szczegółowe własności multipokolorowań grafów konfliktów można wyznaczyć oszacowania wartości ograniczeń funkcji celu $(\sum C_j, \sum c(J_j))$ oraz skonstruować algorytmy optymalne i przybliżone wykorzystujące szczególne postaci grafów konfliktów.

3. Algorytmy zachłanne

W twierdzeniu 1 pokazano górne ograniczenia wartości kolorów dowolnego minimalnego multipokolorowania grafu obciążonego (G, p, r, w), mianowicie

$$\max c(v) \leq w(v) \cdot (r(v) + p_{sum}(v)) + (1 - w(v)) \cdot p(v) \cdot (p_{sum}(v) - p(v) + 1) = a_{w(v)}(v),$$

gdzie $p_{sum}(v) := p(v) + \sum_{u \in N(v)} p(u)$. W dalszej części wprowadzimy oznaczenie

$$A_{w(v)}(v) = \{r(v) + 1, ..., a_{w(v)}(v)\}. \tag{2}$$

Zauważmy następujący prosty

Fakt 1. *Graf skierowany D jest acykliczny wtedy i tylko wtedy, gdy każdy jego podgraf posiada co najmniej jedno źródło.* □

Niech $D(G)$ będzie dowolną orientacją acykliczną grafu G. Niech P_v będzie zbiorem wszystkich ścieżek w digrafie $D(G)$ o końcu w wierzchołku v. Każdemu wierzchołkowi $v \in V(D)$ przypiszemy wagę

$$d(v) = \begin{cases} 0, & \text{jeżeli } v \text{ jest źródłem w } D(G) \\ \max_{P \in P_v} |P|, & \text{jeżeli } v \text{ nie jest źródłem w } D(G) \end{cases} \tag{3}$$

gdzie $|P|$ jest długością ścieżki P (liczbą łuków). Oczywiście funkcja d jest dobrze określona. Dla podkreślenia związku z digrafem D będziemy pisać czasem $d_D(v)$. Wprowadźmy dodatkowe oznaczenie $d(D) := \max_{v \in V(D)} d(v)$. Niech $V_i := \{v \in V(D) : d(v) = i\}$, stosując indukcję mamy wtedy dla dowolnego wierzchołka $v \in V(D) \setminus V_0$

Fakt 2. $d_D(v) = d_{D \setminus V_0}(v) + 1$. □

Fakt 3. *Jeżeli $(v,w) \in A(D)$, to $d(v) < d(w)$.*

Dowód. Załóżmy, że $d(v) \geq d(w)$ oraz $(v,w) \in A(D)$. Wtedy ze wzorów (3) oraz z acykliczności digrafu D dostajemy dla dowolnej ścieżki $P \in P_v$, $d(w) \geq |P| + 1$, skąd $d(w) \geq d(v) + 1$, sprzeczność. □

Wniosek 1. *Każdy ze zbiorów V_i jest zbiorem niezależnym w G.* □

Każdy ostry porządek liniowy $(V(D), <)$ spełniający warunek

$$\forall_{v \neq w}\ v < w \Rightarrow d(v) \leq d(w)$$

nazywamy *zgodnym* z orientacją $D(G)$ (digrafem D). Można zauważyć, że porządek częściowy w zbiorze $V(D)$ zadany przez łuki digrafu D można uzupełnić do dowolnego porządku liniowego zgodnego z D.

Fakt 4. *Jeżeli D jest digrafem acyklicznym, to istnieje liniowy ostry porządek $(V(D), <)$ zgodny z D.*

Dowód. Na mocy faktu 1 wystarczy częściowe uporządkowanie, zadane jak następuje, $\forall_{v,w \in V(D)}\ d(v) < d(w) \Rightarrow v < w$ uzupełnić w dowolny sposób do porządku liniowego. □

Na mocy faktu 3 mamy natychmiast

Fakt 5. *Jeżeli $v_1 < \ldots < v_n$ jest ostrym porządkiem liniowym zgodnym z acyklicznym digrafem D, to*

$$\forall_{v \neq w}\ (v,w) \in A(D) \Rightarrow v < w. \ \square$$

Wniosek 2. *Jeżeli $v_1 < \ldots < v_n$ jest ostrym porządkiem liniowym zgodnym z acykliczną orientacją $D(G)$, to*

$$\forall_{v \neq w}\ (v,w) \in A(D) \iff (v < w) \wedge \{v,w\} \in E(G). \ \square$$

Fakt 6. *Niech $v_1 < \ldots < v_n$ będzie ostrym liniowym porządkiem zgodnym z $D(G)$. Jeżeli $v \in V_i$ oraz $w \in V_j$, to $i < j \Rightarrow v < w$.*

Dowód. Jeżeli $i < j$, to $d(v) < d(w)$, skąd jeżeli $v > w$, to z definicji zgodności porządku mamy $d(v) \geq d(w)$, sprzeczność. □

Na mocy powyższych faktów mamy

Wniosek 3. *Każdy porządek liniowy zgodny z digrafem acyklicznym D ma postać*

$$v_1^0 < ... < v_{k_0}^0 < v_1^1 < ... < v_{k_1}^1 < ... < v_1^{d(D)} < ... < v_{k_{d(D)}}^{d(D)},$$

gdzie $k_i := |V_i|$, $\{v_j^i : j = 1, ..., k_i\} = V_i$, natomiast $v_1^i < ... < v_{k_i}^i$ jest zadany przez dowolną permutację zbioru niezależnego V_i. □

Załóżmy, że każdemu wierzchołkowi $v \in V(D)$ przyporządkowano *zbiór kolorów dopuszczalnych* $A(v) \subset I\!N$, czyli dane jest odwzorowanie $A : V(G) \to 2^{I\!N}$, którego wartości dla wierzchołka v_i będziemy oznaczać przez A_i.

Twierdzenie 2. *Niech $v_1 < v_2 < ... < v_n$ będzie dowolnym uporządkowaniem zgodnym z orientacją acykliczną $D(G)$, ponadto dany jest ciąg zbiorów kolorów dopuszczalnych $A := (A_i)_{i=1}^n$. Odwzorowanie c zadano kolejno dla $v_1, ..., v_n$, jak następuje:*

$$c(v_i) = \begin{cases} \bigcup_{j=1}^{p(v_i)} \{c_j(v_i)\} & dla \quad w(v) = 1 \\ \{c_{\min}(v_i), ..., c_{\min}(v_i) + p(v_i) - 1\} & dla \quad w(v) = 0 \end{cases} \tag{4}$$

gdzie

$$c_1(v_i) := \min\{x : x > r(v_i) \wedge x \in A_i \wedge x \notin \bigcup_{k:(v_k,v_i)\in A(D)} c(v_k)\}$$

$$c_{\min}(v_i) := \min\{x : x > r(v_i) \wedge \{x, ..., x + p(v_i) - 1\} \subset A_i \setminus \bigcup_{k:(v_k,v_i)\in A(D)} c(v_k)\}$$

oraz dla $j = 2, ..., p(v_i)$

$$c_j(v_i) := \min\{x : x > c_{j-1}(v_i) \wedge x \in A_i \wedge x \notin \bigcup_{k:(v_k,v_i)\in A(D)} c(v_k)\}\}$$

Jeżeli c jest dobrze określone, to c jest multipokolorowaniem grafu (G, p, r, w). Co więcej, c nie zależy od wyboru ostrego porządku liniowego zgodnego z $D(G)$.

Dowód. Odwzorowanie c jest dobrze określone, o ile istnieją elementy minimalne, te zaś istnieją, o ile odpowiednie zbiory są niepuste. Ponadto, ponieważ porządek liniowy $v_1 < ... < v_k$ jest zgodny z orientacją $D(G)$, więc dla danego wierzchołka v_i wszystkie wierzchołki u, takie że $(u, v_i) \in A(D)$, mają już określone zbiory kolorów $c(u)$. Załóżmy zatem, że dla dowolnego $i = 1, ..., n$ oraz $j = 1, ..., p(v_i)$ wartości $c_j(v_i)$ oraz $c_{\min}(v_i)$ istnieją. Wtedy oczywiście warunki (2) i (3) definicji 3 są spełnione, co wynika wprost ze wzorów (4). Ponadto, zarówno dla $w(v_i) = 0$, jak i $w(v_i) = 1$ mamy, że $|c(v_i)| = p(v_i)$, co

również nietrudno zauważyć. Na mocy wniosku 3 dwa różne porządki zgodne z orientacją $D(G)$ różnią się kolejnością wyłącznie w obrębie zbiorów niezależych V_i, skąd dostajemy, że multipokolorowania im odpowiadające określone wzorami (4) muszą być równe. □

Jeżeli dla ustalonej orientacji acyklicznej $D(G)$ oraz zbiorów $A_1, ..., A_n$ istnieje multipokolorowanie c grafu (G, p, r, w) określone wzorami jak w fakcie 2, to c nazywamy *multipokolorowaniem zachłannym*. W dalszej części pracy *algorytmem zachłannym* nazywać będziemy każde odwzorowanie przyporządkowujące danej orientacji acyklicznej $D(G)$ ciąg zbiorów $(A_i)_{i=1}^n$ taki, że istnieje multipokolorowanie zachłanne grafu (G, p, r, w) określone wzorami (4). W szczególnym przypadku ciąg zbiorów może być zawsze taki sam. Z twierdzenia 1 wynika, że poszukując minimalnych multipokolorowań możemy zbiory kolorów dopuszczalnych ograniczyć wyłącznie do zbiorów $A_1(v)$ dla $w(v) = 1$, bądź $A_0(v)$ dla $w(v) = 0$. W najprostszym przypadku dla jednostkowych wag i braku dolnych ograniczeń wartości kolorów mamy

Fakt 7. *Dany jest graf G o n wierzchołkach oraz niech $p \equiv 1$ i $r \equiv 0$. Wtedy przyporządkowanie danej orientacji acyklicznej $D(G)$ ciągu zbiorów $A_v := \{1, ..., \rho(v) + 1\}$ definiuje algorytm zachłanny. Wzory (4) podane w twierdzeniu 2 redukują się wtedy do postaci*

$$c(v_i) = \min(A_i \backslash \{c(v_j) : j < i \wedge (v_j, v_i) \in A(D)). \tag{5}$$

□

Własności algorytmów zachłannych opisano w [1, 2, 14].

4. Algorytmy przybliżone

Okazuje się, że dla wielu klas grafów konfliktów potrafimy dla algorytmów zachłannych podać dobre oszacowania ekstremalizowanej funkcji celu (sumy multichromatycznej).

Przyjmijmy, że r oraz w są ustalone. Nietrudno zauważyć, że suma multichromatyczna ma własność monotoniczności zarówno dla grafów, jak i krotności.

Fakt 8. *Jeżeli $H \subset G$ oraz $V(H) = V(G)$, to $\sum(H, p, r, w) \leq \sum(G, p, r, w)$.* □

Dla krotności p_1 oraz p_2 (grafu G) przyjmiemy oznaczenie $p_1 \leq p_2$, jeżeli dla każdego wierzchołka v jest $p_1(v) \leq p_2(v)$. Analogicznie przyjmiemy $r_1 \leq r_2$, o ile dla dowolnego

$v \in V(G)$ mamy $r_1(v) \leq r_2(v)$. Z uwagi na fakt, że dysponując multipokolorowaniem c_2 grafu obciążonego (G, p_2, r_2, w) nietrudno skonstruować multipokolorowanie c_1 grafu obciążonego (G, p_1, r_1, w), mamy

Fakt 9. *Dla dowolnych dwóch grafów obciążonych (G, p_1, r_1, w) oraz (G, p_2, r_2, w) takich, że $p_1 \leq p_2$ oraz $r_1 \leq r_2$ mamy $\sum(G, p_1, r_1, w) \leq \sum(G, p_2, r_2, w)$.* □

Notacja 1. *Dla dowolnego grafu G, krotności p oraz wektora r będziemy pisać $(G, p, r, 1) := (G, p, r, w \equiv 1)$. Analogicznie dla $w \equiv 0$.*

Z uwagi na ograniczenie przez postać wektora w zbioru legalnych multipokolorowań dostajemy

Fakt 10. $\sum(G, p, r, 1) \leq \sum(G, p, r, w) \leq \sum(G, p, r, 0)$. □

Wniosek 4. *Jeżeli dla dowolnego grafu G, krotności p oraz wektora r mamy $\sum(G, p, r, 1) = \sum(G, p, r, 0)$, to dla dowolnego wektora w mamy $\sum(G, p, r, 1) = \sum(G, p, r, w)$.* □

Wykorzystując własności algorytmów zachłannych w [14] pokazano

Twierdzenie 3. ([14]) *Dla dowolnej orientacji acyklicznej D mamy*

$$\sum(G, p, r, 1) \leq p(G) + r(G) + \sum_{(u,v) \in A(D)} p(u). \ \square$$

W przypadku wszystkich niepodzielnych zadań dostajemy z kolei

Twierdzenie 4. ([14]) *Dla dowolnego grafu obciążonego $(G, p, r, 0)$ mamy*

$$\sum(G, p, r, 0) \leq r(G) + p^2(G). \ \square$$

Za [7] zdefiniujemy liczbę

$$col\,(G) = 1 + \min_{\pi \in S_n} \max_i \left\{ d\left(x_{\pi(i)}, G_{\pi(i)}\right) \right\},$$

przy czym $G_{\pi(i)}$ jest grafem indukowanym przez $x_{\pi(1)}, \ldots, x_{\pi(i)}$, natomiast $d\,(x, H)$ oznacza stopień wierzchołka x w grafie H. Można pokazać (zob. [7]), że

$$col(G) = 1 + \max \delta(G[x_{p(1)}, x_{p(2)}, \ldots, x_{p(i)}]),$$

gdzie $x_{p(1)}, ..., x_{p(n)}$ jest porządkiem typu "ostatni najmniejszy" (ang. *smallest last*) zdefiniowanym następująco: $x_{p(i)}$ jest wierzchołkiem z minimalnym stopniem $\delta(G_i)$ w podgrafie

$$G_i := G \backslash \{x_{p(i+1)}, ..., x_{p(n)}\}.$$

Ponieważ $D(G) := (V(G), \{(x_{p(i)}, x_{p(j)}) : i > j \wedge \{x_{p(i)}, x_{p(j)}\} \in E(G)\}$ jest poprawnie zdefiniowaną acykliczną orientacją grafu G, na mocy twierdzenia 3 mamy

Twierdzenie 5. *Dla dowolnego grafu obciążonego* $(G, p, r, 1)$ *zachodzi*

$$\sum(G, p, r, 1) \leq r(G) + col(G) \cdot p(G). \ \square \tag{6}$$

Ponieważ $col(T) \leq 2$ (zob. [7]), dla dowolnego drzewa dostajemy

Wniosek 5. *Dla dowolnego drzewa obciążonego* $(T, p, r, 1)$ *mamy*

$$\sum(G, p, r, 1) \leq r(G) + 2 \cdot p(G). \ \square$$

Przy założeniu $r \equiv 0$ dostajemy lepsze oszacowanie od (6), bowiem $\chi(G) \leq col(G)$, dla dowolnego grafu G (zob. [7]).

Twierdzenie 6. *Dla dowolnego grafu obciążonego* $(G, p, 0, 1)$ *mamy*

$$\sum(G, p, 0, 1) \leq \chi(G) \cdot p(G). \ \square$$

Dowód. Niech V_i będzie zbiorem wierzchołków pokolorowanych i-tym kolorem w pewnym χ-pokolorowaniu grafu G. Wystarczy zauważyć, że kolorując V_1 pierwszym kolorem, V_2 drugim kolorem, ..., V_χ χ-tym kolorem, V_1 $(\chi+1)$-kolorem, i tak dalej, dostaniemy dla dowolnego wierzchołka v $\max c(v) \leq \chi(G) \cdot p(v)$, skąd mamy natychmiast tezę. $\square$

Dysponując wyprowadzonymi oszacowaniami możemy skonstruować algorytmy przybliżone dla wybranych problemów szeregowania zadań. Wykorzystamy znaną notację trójpolową ([4, 10, 14]).

Na podstawie faktu 8 (dolne ograniczenie) oraz twierdzeń 5 i 6 (górne ograniczenia) dostajemy

Twierdzenie 7. *Dla problemu optymalizacyjnego*

$$P|fix_j, G, r_j, pmtn|\sum C_j,$$

istnieje $col(G)$*-przybliżony algorytm zachłanny o złożoności* $O(np\log p)$*, ponadto dla problemu optymalizacyjnego*

$$P|fix_j, G, pmtn|\sum C_j,$$

istnieje $\chi(G)$*-przybliżony algorytm zachłanny o złożoności* $O(np\log p)$.

Dowód. Wystarczy zauważyć, że

$$r(G) + p(G) \leq \sum(G, p, r, 1) \leq r(G) + col(G) \cdot p(G)$$

oraz zwrócić uwagę na fakt, że algorytm zachłanny przydzielając wierzchołkowi v wartości kolorów (dla danej orientacji) sortuje tablicę kolorów o rozmiarze $O(p)$ w czasie $O(p\log p)$ otrzymując w czasie $O(p)$ zbiór $c(v)$. Analogicznie jest dla χ-przybliżonego algorytmu przy założeniu identycznych czasów gotowości. □

Zwróćmy uwagę, że powyższe algorytmy mają złożoności proporcjonalne do wartości $p(G)$, czyli są pseudowielomianowe.

LITERATURA

1. Bar-Noy A., Bellare M., Halldórsson M.M., Shachnai H., Tamir T.: On chromatic sums and distributed resource allocation. Information and Computation 140, 183-202 (1998).
2. Bar-Noy A., Halldórsson M.M., Kortsarz G., Salman R., Shachnai H.: Sum multicoloring of graphs. Journal of Algorithms 37, 422-450 (2000).
3. Cai X., Lee C.-Y., Wong T.-L.: Multiprocessor task scheduling to minimize the maximum tardiness and the total completion time. IEEE Transactions on Robotics and Automation, Vol. 16, No. 6, 824-830 (2000).
4. Drozdowski M.: Scheduling multiprocessor tasks - An overview. European Journal of Operational Research 94, 215-230 (1996).
5. Drozdowski M., Dell'Olmo P.: Scheduling multiprocessor tasks for mean flow time criterion. Comput. Oper. Research 27, 571-585 (2000).
6. Jansen K., Prokolab L.: Preemptive scheduling on dedicated processors: applications of fractional graph coloring, manuscript (2001).
7. Jensen T.R., Toft B.: Graph Coloring Problems. John Wiley & Sons, New York (1995).

8. Kubale M.: The complexity of scheduling independent two-processor tasks on dedicated processors. Information Processing Letters 24, 141-147 (1987).
9. Kubale M.: Introduction to Computational Complexity and Algorithmic Graph Coloring. Wydawnictwo Gdańskiego Towarzystwa Naukowego, Gdańsk (1998).
10. Kubale M.: Preemptive versus nonpreemptive scheduling of biprocessor tasks on dedicated processors. European Journal on Operational Research 94, 242-251 (1996).
11. Giaro K., Kubale, M., Małafiejski M., Piwakowski K.: Dedicated scheduling of biprocessor tasks to minimize mean flow time. (2001) (praca zaakceptowana do druku w LNCS).
12. Małafiejski M.: Minimalizacja średniego czasu obsługi zadań w systemie równoległego przydziału zasobów. Zeszyty Naukowe Politechniki Śląskiej, Ser. Automatyka z. 129, 273-285 (2000).
13. Małafiejski M.: Wybrane problemy harmonogramowania procesów w oparciu o dwudzielne grafy konfliktowe. Zesz. Nauk. AGH, Ser. Automatyka, Tom 5, z. 1/2, 403-409 (2001).
14. Małafiejski M.: Uszeregowania zadań wieloprocesorowych minimalizujące średni czas przepływu. Rozprawa doktorska, Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska (2002).

Recenzent: Prof. dr hab. inż. Andrzej Świerniak

Abstract

In this paper the authors consider some properties of the proposed model of the general and dedicated parallel systems. Scheduling multiprocessor tasks on dedicated machines is equivalent to sum multicoloring of so-called *conflicting graph*, whose vertices represent tasks and two vertices are adjacent if and only if there exists a conflicting resource requirement. Our goal is to schedule all tasks minimizing their average completion time under stipulation that no two conflicting tasks are executed simultaneously. The main results concern the bouding of the sum of colors used in any optimal multicoloring (sum of completion times in a schedule). The authors introduce greedy algorithms and use them to construct some approximation algorithms in scheduling multiprocessor tasks problems. The main result is an pseduopolynomia $col(G)$-approximation algorithm to the multichromatic sum or equivalently to the problem $P|fix_j, G, r_j, pmtn|\sum C_j$.