ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2008
Seria: AUTOMATYKA z. 150 Nr kol. 1796

Paweł OBSZARSKI
Politechnika Gdańska

PARETO-OPTYMALNE SZEREGOWANIE ZADAŃ WIELOPROCESOROWYCH NA PROCESORACH DEDYKOWANYCH

Streszczenie. Problem szeregowania jednostkowych zadań wieloprocesorowych na maszynach dedykowanych można modelować za pomocą hipergrafów. Znamy kilka klas hipergrafów, dla których szeregowanie z kryterium kosztu całkowitego jest wielomianowe. Pokażemy, jak za pomocą modelu z kosztem całkowitym można rozwiązać problemy z innymi kryteriami znanymi z teorii szeregowania oraz jak rozwiązać problemy dwukryterialne.

PARETO-OPTIMAL SCHEDULING OF MULTIPROCESSOR TASKS ON DEDICATED MACHINES

Summary. Problem of scheduling multiprocessor tasks on dedicated machines can be modeled by hypergraphs. There are a few classes of hypergraphs for which polynominal time algorithms for scheduling with total cost criterion are known. Our aim is to show that other criteria and also bicriterial problems can be solved by the use of total cost criterion.

1. Wstęp

Rozważmy zbiór $J = \{J_1, J_2, ..., J_m\}$ zadań, które mogą być wykonywane na pewnym zbiorze procesorów (maszyn) $M = \{M_1, M_2, ..., M_n\}$. Procesory różnią się między sobą i każdy jest wyspecjalizowany w unikalnej czynności. Aby wykonać zadanie J_i, konieczny jest pewien zbiór procesorów $fix_i \in M$ dostępnych jednocześnie. Żaden procesor nie może pracować nad więcej niż jednym zadaniem w tym samym momencie. Czas jest podzielony na ponumerowane kolejnymi liczbami naturalnymi jednostki, które nazywać będziemy szczelinami czasowymi. Wszystkie zadania potrzebują jednakowej ilości czasu na wykonanie. Dla uproszczenia możemy przyjąć, że wystarcza im jedna jednostka czasu, czyli $p_i = 1$. Zadania mogą mieć ograniczenia dostępności wyrażone skończonymi listami $L(J_i) \subset \mathbb{N}$ szczelin czasowych, w których dane zadanie może być wykonane. Dla zadania J_i z jego wykonaniem w pewnej szczelinie czasowej $x \in L(J_i)$ związany jest koszt określony przez funkcję $f_{J_i}(x)$. Model ten będziemy także nazywać listowo-kosztowym lub modelem szeregowania zadań z kryterium kosztu całkowitego $\Sigma G_j(C_j)$ [1].

Istnieje wiele innych kryteriów optymalizacji szeregowania. W tej pracy rozważać będziemy kilka z nich. Najprostsze z obliczeniowego punktu widzenia jest kryterium minimalizujące długość harmonogramu C_{max}. Innym kryterium jest kryterium minimalizujące maksymalne spóźnienie T_{max} lub opóźnienie L_{max}. Najbliższe kryterium kosztu całkowitego są kryteria ważone sumacyjne $\Sigma w_j C_j$ i ważonej sumy spóźnień $\Sigma w_j T_j$. Ostatnim rozpatrywanym w tym opracowaniu kryterium jest ważona liczba spóźnień $\Sigma w_j U_j$. Szczegółowe definicje oznaczeń dotyczących szeregowania wykorzystywanych w tej pracy można znaleźć w [1].

Scharakteryzowany model szeregowania zadań jednostkowych na procesorach dedykowanych z ograniczeniami czasowymi (lub bez) można wygodnie opisać przy wykorzystaniu terminologii hipergrafów.

Hipergrafem H nazywamy uporządkowaną parę $H = (V, E)$, gdzie V (lub $V(H)$) oznacza zbiór *wierzchołków*, a E (lub $E(H)$) reprezentuje multizbiór *krawędzi* (*hiperkrawędzi*). Elementy zbioru E są pewnymi podzbiorami zbioru wierzchołków. *Poprawnym krawędziowym k-pokolorowaniem* (lub *pokolorowaniem*) hipergrafu H nazywamy funkcję $c : E(H) \rightarrow \{1, ..., k\}$ taką, że krawędzie mające wspólny wierzchołek otrzymują różne numery, zwane dalej *kolorami*. Jeśli każdej hiperkrawędzi $e \in E$ przypisana jest skończona lista dostępnych kolorów $L(e)$ i poprawne pokolorowanie krawędziowe spełnia dodatkowy warunek $c(e) \in L(e)$, to będziemy mówili o *listowym pokolorowaniu krawędzi* hipergrafu. Rozważmy funkcję $f_e : \mathbb{N} \rightarrow \mathbb{N}$ ($f_e : L(e) \rightarrow \mathbb{N}$). Jeśli dla każdej krawędzi $e \in E$ z wykorzystaniem koloru $x \in \mathbb{N}$ ($x \in L(e)$) związany jest pewien koszt wyrażony funkcją $f_e(x)$ i celem modelu jest minimalizacja sumy kosztów, to kolorowanie będziemy nazywać *kosztowym* (*listowo-kosztowym*).

Symbolem $N(e)$ oznaczać będziemy *sąsiedztwo* krawędzi e, czyli liczbę krawędzi, z którymi e ma przynajmniej jeden wspólny wierzchołek. $N(H)$ reprezentuje maksymalne sąsiedztwo w hipergrafie H.

Powyższe problemy szeregowania zadań i kolorowania krawędzi hipergrafów są równoważne. Maszyny utożsamiać będziemy z wierzchołkami, zadania z krawędziami oraz kolory z ich terminami wykonania. Automatycznie, poprawne k-pokolorowanie krawędzi reprezentuje pewne uszeregowanie. Pojęcia szeregowania zadań oraz kolorowania krawędzi hipergrafów będą używane zamiennie. W pracy przyjmujemy konwencję, że n oznacza liczbę wierzchołków hipergrafu oraz liczbę maszyn, a m liczbę krawędzi i liczbę zadań.

2. Szeregowanie zadań z kryterium kosztu całkowitego

Hipergrafem *przecinającym się* nazywamy hipergraf, w którym każde dwie krawędzie mają przynajmniej jeden wspólny wierzchołek.

Twierdzenie 1. *Problem listowo-kosztowego kolorowania krawędzi hipergrafów przecinających się można rozwiązać w czasie $O(m^3)$.*

Dowód. W pierwszej kolejności rozpatrzmy model kolorowania bez list. Model taki można rozumieć jako model z nieskończenie długimi listami. Zauważmy, że dla każdej krawędzi e_i jej listę możemy ograniczyć do $N(H) + 1$ najtańszych kolorów i nie wpłynie to na szeregowanie. Analogicznie postępujemy, gdy listy są dłuższe niż $N(H) + 1$. Inne rozwiązanie przewidziane jest, gdy listy są krótsze niż $N(H) + 1$. Możemy je wtedy rozszerzyć o kolory tak drogie, by ich wykorzystanie było jednoznacznie widoczne w całkowitej sumie kosztów. Dlatego przyjmujemy, że $|L(e_i)| = N(H) + 1$; wtedy mamy pewność, że hipergraf jest kolorowalny.

Rozważmy pewien przecinający się hipergraf H. H ma m krawędzi, z których żadne dwie nie mogą dostać tego samego koloru, ponieważ wszystkie krawędzie są ze sobą wzajemnie sąsiednie. Każdej krawędzi przypisana jest lista kolorów $L(e_i)$ długości $N(H) + 1 = m$ oraz funkcja kosztów $f_{e_i} : L(e_i) \to \mathbb{N}$. Zbudujemy dwudzielny obciążony graf $K^{E,L}$, który na jednej partycji ma m wierzchołków odpowiadających krawędziom, a na drugiej wierzchołki odpowiadające wszystkim kolorom występującym na listach. Krawędź $\{e_i, x\}$ w grafie $K^{E,L}$ występuje wtedy, gdy $x \in e_i$. Waga krawędzi w $K^{E,L}$ jest równa $f_{e_i}(x)$. Znalezienie pokolorowania krawędzi hipergrafu H jest równoważne znalezieniu minimalnego ważonego skojarzenia w grafie dwudzielnym. Dinits w pracy [2] pokazał, że maksymalne ważone skojarzenie można znaleźć w grafie dwudzielnym w czasie $O(p^3 + pq)$, gdzie p i q to rozmiary partycji i $p \leqslant q$. W przypadku naszego grafu dwudzielnego jedna partycja reprezentuje hiperkrawędzie, więc $p = m$, natomiast druga partycja reprezentuje kolory na listach, zatem w najgorszym przypadku $q = m(N(H) + 1)$. H jest hipergrafem przecinającym się, więc $= N(H) = m - 1$, zatem $q = m^2$. Dlatego złożoność algorytmu kolorowania krawędzi hipergrafów przecinających się szacujemy przez $O(m^3 + mm^2) = O(m^3)$. □

Hipergraf liniowy to hipergraf, w którym każde dwie krawędzie mają co najwyżej jeden wspólny wierzchołek. *Hiperdrzewem* nazywamy taki hipergraf H, że istnieje drzewo T (graf prosty) rozpięte na zbiorze wierzchołków $V(H)$, takie że każda hiperkrawędź $e \in E(H)$ indukuje spójne poddrzewo w T.

Twierdzenie 2. [4] *Istnieje algorytm znajdujący optymalne listowo-kosztowe pokolorowanie hiperdrzewa liniowego H w czasie $O(nN^2 \log NW)$, gdzie $N = N(H)$, $n = |V(H)|$, a W to maksymalny koszt po wszystkich krawędziach i elementach ich list.*

3. $\Sigma G_j(C_j)$ a inne kryteria

Zdecydowana większość klasycznych kryteriów szeregowania może być wielomianowo sprowadzona do zdefiniowanego wyżej problemu szeregowania listowo-kosztowego.

Lemat 1. *Modele szeregowania z kryteriami C_{max}, L_{max}, T_{max}, $\Sigma w_j C_j$, $\Sigma w_j T_j$, $\Sigma w_j U_j$, zarówno z ograniczeniami czasowymi, jak i bez, można wielomianowo zredukować do problemu listowo-kosztowego szeregowania zadań.*

Dowód. Przyjmujemy, że w modelu wszystkie krawędzie mają listy długości $N(H) + 1$. Możemy ustalić takie ograniczenie na takiej samej zasadzie jak w dowodzie twierdzenia 1.

Rozważymy różne kryteria szeregowania i pokażemy, jak w sposób wielomianowy można sprowadzić je do problemu listowo-kosztowego. Załóżmy, że znamy pewną wielomianową procedurę rozwiązywania problemu listowo-kosztowego szeregowania zadań dla pewnego typu instancji problemu. Ustalimy funkcje kosztu dla problemu listowo-kosztowego, tak by możliwe było rozwiązanie problemu z innym kryterium.

L_{max} – Przyjmijmy, że d_j jest ostatecznym terminem wykonania zadania J_j. Niech $C \in \mathbb{Z}$ będzie pewną liczbą. Ustalmy wartości funkcji kosztu:

$$f_{J_j}(x) = \begin{cases} 0 & x \leqslant d_j + C \\ 1 & x > d_j + C \end{cases}$$

Nietrudno zauważyć, że całkowity koszt uszeregowania będzie równy 0, o ile tylko $L_{max} \leqslant C$. Jeśli zastosujemy metodę dwupodziału, jesteśmy w stanie w wielomianowym czasie wyznaczyć optymalne uszeregowanie z kryterium L_{max}.

C_{max} – Kryterium to jest pewnym szczególnym przypadkiem L_{max}, w którym z góry przyjmujemy, że wszystkie zadania powinny się zakończyć przed pierwszą szczeliną czasową, czyli $d_j = 0$. Wtedy opóźnienie jest równe długości harmonogramu. Dlatego możemy powielić sposób postępowania z poprzedniego kryterium.

T_{max} – Ponownie stosujemy metodę wyznaczania optymalnego uszeregowania jak przy kryterium L_{max}, biorąc oczywiście pod uwagę, że $T_{max} = \max\{0, L_{max}\}$, zatem przyjmujemy, że $C \geqslant 0$.

$\Sigma w_j T_j$ – Zakładamy, że w_j jest wagą spóźnienia zadania J_j, a d_j ostatecznym terminem jego wykonania. Ustalamy wartości funkcji kosztu:

$$f_{J_j}(x) = \begin{cases} 0 & x \leqslant d_j \\ w_j(x - d_j) & x > d_j \end{cases}$$

Zauważmy, że wyznaczenie optymalnego listowo-kosztowego uszeregowania z takimi funkcjami kosztu jest równoważne wyznaczeniu szeregowania z kryterium $\Sigma w_j T_j$.

$\Sigma w_j C_j$ – Kryterium to jest szczególnym przypadkiem kryterium $\Sigma w_j T_j$. Aby wyznaczyć minimalny łączny czas zakończenia zadań, należy przyjąć, że $d_j = 0$ dla wszystkich zadań J_j i zastosować taką samą procedurę jak dla kryterium $\Sigma w_j T_j$.

$\Sigma w_j U_j$ – Ponownie w_j oznacza wagę, a d_j ostateczny termin zakończenia zadania J_j. Dla tego kryterium funkcję kosztu definiujemy w następujący sposób:

$$f_{J_j}(x) = \begin{cases} 0 & x \leqslant d_j \\ w_j & x > d_j \end{cases}$$

Zastosowanie procedury listowo-kosztowego szeregowania zadań odpowie na pytanie, ile wynosi ważona liczba spóźnień. Zauważmy, że jeśli przyjmiemy $w_j = 1$, to dowiemy się, ile zadań przekroczyło terminy.

□

4. Problemy dwukryterialne

Procedura listowo-kosztowego szeregowania zadań może być pomocna nie tylko przy rozwiązywaniu problemów jednokryterialnych, ale także dwukryterialnych.

Mówimy, że $(a, b) \prec (c, d)$, jeśli $(a, b) \neq (c, d)$ i $a \leqslant c$ oraz $b \leqslant d$. Niech $g : X \to \mathbb{N}^2$ będzie funkcją wyznaczającą optymalizowane wartości dla pewnego rozwiązania x z przestrzeni stanów X. O rozwiązaniu $x \in X$ mówimy, że jest *Pareto-optymalne*, jeśli nie istnieje $x' \in X$, taki że $g(x') \prec g(x)$. Szczegółowy opis teorii można znaleźć w [3].

Lemat 2. *Problem znajdowania wszystkich optimów Pareto-optymalnego szeregowania zadań z pierwszym kryterium C_{max}, L_{max} albo T_{max} i drugim kryterium C_{max}, L_{max}, T_{max}, $\Sigma w_j C_j$, $\Sigma w_j T_j$ albo $\Sigma w_j U_j$ można wielomianowo sprowadzić do problemu listowo-kosztowego szeregowania zadań.*

Dowód. Zakładamy, że znamy wielomianowy algorytm listowo-kosztowego szeregowania zadań dla rozpatrywanych hipergrafów. Z lematu 1 wynika, że potrafimy uszeregować owe hipergrafy szeregowania z kryteriami: C_{max}, L_{max}, T_{max}, $\Sigma w_j C_j$, $\Sigma w_j T_j$ i $\Sigma w_j U_j$. W pierszym kroku znajdujemy optymalne uszeregowanie z pierwszym z kryteriów. Wiemy, że jest to C_{max}, L_{max} lub T_{max}. Możemy teraz skrócić listy dostępnych kolorów dla poszczególnych krawędzi, by nie było możliwe uszeregowanie gorsze ze względu na rozpatrywane kryterium. Czyli dla kryterium C_{max} z list skreślamy wszystkie szczeliny czasowe o indeksach większych niż C^*_{max}, dla kryterium L_{max} usuwamy z list wszystkie szczeliny czasowe o indeksach większych niż $d_j + L^*_{max}$ i analogicznie dla T_{max} większe niż $d_j + T^*_{max}$(* oznacza optymalną długość uszeregowania dla danego kryterium). Na w ten sposób przygotowanych hipergrafach możemy uruchomić procedurę szeregowania z dowolnym kryterium C_{max}, L_{max}, T_{max}, $\Sigma w_j C_j$, $\Sigma w_j T_j$ albo $\Sigma w_j U_j$. Dzięki temu uzyskamy pierwsze graniczne Pareto-optymalne uszeregowanie. Aby wyliczyć pozostałe optima, musimy kolejno powiększać listy i sprawdzać, czy możliwe jest lepsze uszeregowanie ze względu na drugie kryterium. Jeśli jest możliwe, to dodajemy kolejne rozwiązanie do rozwiązań Pareto-optymalnych, jeśli nie jest, dalej powiększamy listy. Listy powiększamy tak długo, aż drugie kryterium osiągnie wartość optymalną z hipergrafu szeregowania z nienaruszonymi listami. □

Zauważmy, że metody takiej nie można zastosować do sytuacji, gdy oba kryteria są typu sumacyjnego. Niemniej jednak możemy znaleźć graniczne rozwiązania takiego problemu innym sposobem.

Lemat 3. *Problem znajdowania granicznych rozwiązań Pareto-optymalnych z dowolnymi dwoma z trzech kryteriów ΣC_j, ΣT_j i ΣU_j można wielomianowo sprowadzić do problemu listowo-kosztowego szeregowania zadań.*

Dowód. Przyjmijmy, że pierwsze kryterium jest kryterium silniejszym. Ustalimy funkcje kosztu w taki sposób, by uszeregowanie było optymalne z pierwszym kryterium i dodatkowo w miarę możliwości z drugim oraz by część całkowita dzielenia kosztu całkowitego przez Y wyznaczała wartość optimum wg pierwszego kryterium, a reszta z dzielenia dawała warunkowe optimum drugiego.

$$\Sigma C_j,\ \Sigma U_j - f_{J_i}(x) = \begin{cases} xY & x \leqslant d_i \\ xY + 1 & x > d_i \end{cases}$$

gdzie $Y = m + 1$, ponieważ wiadomo, że spóźnionych zadań nie może być więcej niż liczba zadań.

$$\Sigma U_j,\ \Sigma C_j - f_{J_i}(x) = \begin{cases} x & x \leqslant d_i \\ x + Y & x > d_i \end{cases}$$

gdzie Y to duża liczba większa niż dowolna suma kolorów. Przyjmijmy na przykład, że $Y = m^2$.

$$\Sigma U_j,\ \Sigma T_j - f_{J_i}(x) = \begin{cases} 0 & x \leqslant d_i \\ Y + \max(0, x - d_j) & x > d_i \end{cases}$$

gdzie Y to ponownie na przykład m^2.

$$\Sigma T_j, \Sigma U_j - f_{J_i}(x) = \begin{cases} 0 & x \leqslant d_i \\ Y \max(0, x - d_j) + 1 & x > d_i \end{cases}$$

Ponownie $Y = m + 1$.

$$\Sigma T_j, \Sigma C_j - f_{J_i}(x) = \begin{cases} x & x \leqslant d_i \\ Y \max(0, x - d_j) + x & x > d_i \end{cases}$$

gdzie $Y = m^2$.

$$\Sigma C_j, \Sigma T_j - f_{J_i}(x) = \begin{cases} Yx & x \leqslant d_i \\ Yx + \max(0, x - d_j) & x > d_i \end{cases}$$

dla $Y = m^2$. □

Zauważmy, że ostateczne terminy nie muszą być takie same dla kryteriów ΣU_j i ΣT_j. Rozumowanie to można rozszerzyć na modele z różnymi terminami zakończenia. Można także rozszerzyć na podwójne kryteria ΣU_j i ΣT_j z różnymi terminami zakończenia.

BIBLIOGRAFIA

1. Błażewicz J., Ecker K.K., Pesch E., Schmidt G., Węglarz J.: Scheduling Computer and Manufacturing Processes. Springer (1996).
2. Dinits E. A.: O reshenii dvukh zadach o naznachenii. Issledovaniya po Diskretnoi Optimizatsii, Izdatel'stvo "Nauka", Moskwa, 1976, 333–348.
3. Ehrgott M.: Multicriteria optimization. Springer, (2000).
4. Giaro K., Kubale M., Obszarski P.: Graph coloring approach to scheduling of multiprocessor tasks on dedicated machines with availability constraints.

Recenzent: Dr hab. inż. Zdzisław Duda, prof. Politechniki Śląskiej

Abstract

Let $J = \{J_1, J_2, ..., J_m\}$ be a set of tasks which can be executed on a set of processors $M = \{M_1, M_2, ..., M_n\}$. All processors are distinct, and they can execute only one specified task at a time. Each task J_i requires the simultaneous use of a nonempty set $fix_i \subseteq M$ of processors for its execution. All tasks are independent, nonpreemptable, and of the same length. For the sake of simplicity we assume that the execution time of J_i, i.e. $p_i = 1$ for each $i = 1, ..., m$. Every processor can work on not more than one task at the same time. Time is divided into unit length slots numbered with successive integers. Tasks have availability constraints prespecified by lists $L(J_i) \subseteq \mathbb{N}$ (where $\mathbb{N}$ is the set of nonnegative integers) of available time slots in which J_i can be executed. With each task we associate a function $f_{J_i}(x)$ which assigns to each $x \in L(J_i)$ the cost of executing J_i if it is executed in time slot x. Our aim is to find a schedule with minimum total cost. We call this model cost list scheduling of multiprocessor tasks on dedicated machines.

Classical scheduling theory provides various criteria of scheduling like C_{max}, L_{max}, T_{max}, $\Sigma w_j C_j$, $\Sigma w_j U_j$ and many others. All of them can be reduced to the cost list scheduling model. We also present some reductions for different bicriterial problems.