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

Konrad PIWAKOWSKI
Politechnika Gdańska

SZEREGOWANIE ZADAŃ Z OPTYMALIZACJĄ ŚREDNIEGO CZASU ZAKOŃCZENIA OPERACJI W SYSTEMIE OTWARTYM

Streszczenie. W niniejszej pracy rozważamy graniczne przypadki otwartego systemu obsługi zadań niepodzielnych NOSS (Non-Preemptive Open-Shop Scheduling), dla których problem szeregowania z optymalizacją średniego czasu zakończenia operacji przestaje być wielomianowy i staje się NP-trudny. W szczególności dowodzimy, że wielomianowy przypadek pojedynczego zadania składającego się z n operacji wykonywanych na n różnych procesorach staje się NP-trudny po dołączeniu drugiego zadania, składającego się z jednej operacji wykonywanej na jednym (z góry określonym) procesorze.

OPEN SHOP SCHEDULING TO MINIMIZE AVERAGE COMPLETION TIME OF THE OPERATIONS

Summary. In this paper we consider some special cases of the Non-Preemptive Open-Shop Scheduling model with the average completion time of all operations as the optimality criterion. In particular we prove that the polynomial solvable case where there is only one job consisting of n operations, becomes NP-hard when another job with only one operation sharing one of the processors is added.

1. Wstęp

Do najczęściej rozpatrywanych kryteriów optymalizacyjnych w otwartym systemie obsługi należy całkowity czas zakończenia oraz średni czas zakończenia zadania. W niniejszej pracy rozważamy inne kryterium – średni czas zakończenia operacji. A zatem w odróżnieniu od kryterium średniego czasu zakończenia zadania bierzemy pod uwagę momenty zakończenia wszystkich operacji, a nie tylko końcowych operacji poszczególnych zadań. Kryterium takie odpowiada np. sytuacji, gdy z każdą z operacji jest skojarzony identyczny nieodnawialny zasób, którego koszt magazynowania (rezerwowania) jest proporcjonalny do czasu magazynowania (np. trudny do utrzymania w stanie aktywności katalizator albo uciążliwe do przechowywania paliwo). Istotne jest tu jednak założenie o tym, że każda z operacji zużywa (zwalnia) taką samą część zasobu. Rozważmy przykład. Do szpitala przybywa m pacjentów (zadań), z których każdy musi przejść k operacji, po jednej na każdej z k sal operacyjnych (procesorów). Do czasu wykonania i-tej operacji pacjent musi być

podłączony do urządzenia (zasobu) podtrzymującego i-tą czynność życiową organizmu (natychmiast po przybyciu do szpitala pacjent jest podłączany do k urządzeń, które są kolejno odłączane w miarę wykonywania kolejnych operacji). Koszt wypożyczenia każdego urządzenia jest proporcjonalny do czasu eksploatacji, przy czym opłata za godzinę eksploatacji jest taka sama dla wszystkich urządzeń. W tym przypadku optymalne uszeregowanie przy minimalizacji średniego czasu zakończenia operacji minimalizuje koszty wypożyczenia urządzeń.

Problem szeregowania zadań w otwartym systemie obsługi bez możliwości przerywania wykonywania operacji dla każdego z wymienionych trzech kryteriów optymalizacji jest problemem NP-trudnym. Naturalnym celem jest więc próba charakteryzacji możliwie szerokich klas przypadków, dla których problem pozostaje wielomianowy. Ograniczenia nakładane na ogólny model zwykle polegają na ograniczeniu zbioru możliwych wartości czasu wykonania operacji (np. wszystkie operacje mają czas jednostkowy) lub na ograniczeniu zbioru operacji (np. każde zadanie składa się z co najwyżej dwóch operacji). Do tej pory znacznie więcej uwagi poświęcono wyznaczaniu takich klas dla czasu całkowitego zakończenia jako kryterium optymalizacji, np. [2], [3], [5], [7]. Tym razem zajmujemy się wyłącznie przypadkiem minimalizacji średniego czasu zakończenia operacji, przy czym nie stawiamy żadnych dodatkowych ograniczeń na zbiór wartości czasu wykonania operacji.

2. Definicja problemu

Niech $M = \{M_1, M_2, ..., M_k\}$ oznacza zbiór procesorów, a $J = \{J_1, J_2, ..., J_m\}$ - zbiór zadań, gdzie $k, m \in N$. Każde zadanie J_i składa się co najwyżej z k operacji $O_i = \{o_{i,1}, ..., o_{i,k}\}$, które muszą być wykonywane odpowiednio na procesorach $M_1, M_2, ..., M_k$. Czas wykonania operacji $o_{i,j}$ wynosi $t_{i,j} \in Q$. Ponadto zakłada się, że żadne dwie operacje wchodzące w skład tego samego zadania nie mogą być wykonywane równocześnie oraz że każdy procesor wykonuje co najwyżej jedną operację w tym samym czasie. Operacje nie mogą być przerywane (Non-Preemptive Open-Shop Scheduling – NOSS), a ich kolejność wykonywania jest dowolna. Niech O będzie zbiorem wszystkich operacji. Rozwiązanie problemu polega na określeniu momentów rozpoczęcia wykonywania każdej operacji $f : O \rightarrow Q$ w taki sposób, aby suma czasów zakończenia poszczególnych zadań $\sum_{1 \le i \le m} \sum_{1 \le j \le k} (f(o_{i,j}) + t_{i,j})$ była najmniejsza z możliwych.

Zdefiniujemy dwa problemy, z których pierwszy jest przypadkiem problemu szeregowania NOSS z optymalizacją średniego czasu zakończenia parametryzowanym przez $2n$ liczb naturalnych, a drugi jest dobrze znanym problemem "odd-even partition".

Problem Φ

Dane są liczby naturalne $a_1 < a_2 < ... < a_{2n}$. Znaleźć optymalne uszeregowanie przy optymalizacji średniego czasu zakończenia operacji dla następującej instancji NOSS: $M = \{M_1, M_2, ..., M_{2n+2}\}$, $J = \{J_1, J_2\}$, $O_1 = \{o_{1,1}, ..., o_{1,2n+2}\}$, $O_2 = \{o_{2,2n+2}\}$, $t_{2,2n+2} = nL$, $t_{1,2n+2} = \frac{2n+1}{2(n+1)}L$, $t_{1,2n+1} = M$, $t_{1,i} = L + \varepsilon_i$, gdzie $i = 1,..., 2n$, $\varepsilon_i = a_i - \left(\frac{1}{2n}\sum_{i=1}^{2n} a_i\right)$, $\varepsilon = \max\{|\varepsilon_i| : 1 \le i \le 2n\}$, $L = 36(n+1)n^2\varepsilon$, $M = 20n^2L$.

Niech Σ_X oznacza w skrócie sumę $\sum_{x \in X} x$.

Problem Θ

Dany jest zbiór liczb naturalnych $A = \{a_1, a_2, ..., a_{2n}\}$, gdzie $a_i < a_{i+1}$, dla $i = 1,...,2n-1$. Czy istnieje zbiór $B \subseteq A$ spełniających warunki:

(1) $\forall_{1 \le i \le n}((a_{2i} \in B \wedge a_{2i-1} \notin B) \vee (a_{2i} \notin B \wedge a_{2i-1} \in B))$,

(2) $\Sigma_B = \Sigma_{A \setminus B}$.

3. Redukowalność problemu Θ do problemu Φ

Przedstawiamy teraz trzy kolejne lematy opisujące własności uszeregowań stanowiących rozwiązanie problemu Φ. Własności te są niezależne od wartości parametrów $a_1, a_2, ..., a_{2n}$. Szczegółowe dowody tych lematów można znaleźć w [4].

Dla danego uszeregowania f oznaczmy sumę momentów rozpoczęcia operacji przez $S(f) = \sum_{1 \le i \le m} \sum_{1 \le j \le k} f(o_{i,j})$.

Lemat 1. *W uszeregowaniu stanowiącym rozwiązanie problemu Φ operacja $o_{1,2n+1}$ jest ostatnią operacją zadania J_1.*

Szkic dowodu

Dość łatwo można wykazać istnienie uszeregowania f spełniającego $S(f) < M$. Jednocześnie oczywiste jest, że uszeregowanie g nie spełniające tezy implikuje $S(g) > M$. A zatem g nie może być optymalne. Sprzeczność.

Lemat 2. *W uszeregowaniu stanowiącym rozwiązanie problemu Φ operacja $o_{1,2n+2}$ jest wykonywana po operacji $o_{2,2n+2}$.*

Szkic dowodu

Przypuśćmy, że f jest uszeregowaniem stanowiącym rozwiązanie problemu Φ, które nie spełnia tezy lematu. Możemy udowodnić, że wówczas $S(f) > L(n+1)(2n+1) - n(2n+1)\varepsilon$.

Niech B będzie dowolnym n-elementowym podzbiorem zbioru $\{1, 2, ..., 2n\}$ takim, że $\sum_{i\in B} t_{1,i} > \frac{1}{2}\sum_{1\le i\le 2n} t_{1,i}$. Niech f_2 będzie pokolorowaniem określonym w ten sposób, że operacje o indeksach ze zbioru B są wykonywane w pierwszej kolejności, po nich operacja $o_{1,2n+2}$, następnie wszystkie pozostałe operacje zadania J_1, z tym że operacja $o_{1,2n+1}$ jest wykonywana na końcu. Operacja $o_{2,2n+2}$ rozpoczyna się w chwili 0. Operacje zadania J_1 rozpoczynają się bezzwłocznie jedna po drugiej.
Można wykazać, że $S(f_2) < L(n(2n+3)+\frac{1}{2}) + n(2n+2)\varepsilon$.
Stąd $S(f) - S(f_2) > \frac{1}{2}\left(L - (8n^2+6n)\varepsilon\right) > 0$. Sprzeczność.

Lemat 3. *Jeżeli f jest rozwiązaniem problemu Φ, to $f(o_{1,2n+2}) = nL$ oraz istnieje dokładnie n operacji $o_{1,i}$ poprzedzających operację $o_{1,2n+2}$.*

Szkic dowodu

Załóżmy, że f jest rozwiązaniem problemu Φ. Niech k będzie liczbą operacji $o_{1,i}$ takich, że $f(o_{1,i}) > f(o_{1,2n+2})$. Rozpatrując kolejno przypadek $k < n$ oraz $k > n$ dla obu znajdujemy lepsze uszeregowanie niż f podobnie jak w dowodach powyższych lematów. Stąd mamy $k = n$.
Z lematu 2 mamy $f(o_{1,2n+2}) \ge nL$. Przypadek $f(o_{1,2n+2}) > nL$ wykluczamy prowadząc do lepszego uszeregowania niż f poprzez "zamianę" operacji zadania J_1 poprzedzających operację $o_{1,2n+2}$ z operacjami rozpoczynającymi się po operacji $o_{1,2n+2}$, z pominięciem operacji $o_{1,2n+1}$, co tworzy luz dla operacji $o_{1,2n+2}$. Po przyspieszeniu rozpoczęcia $o_{1,2n+2}$ i wszystkich późniejszych operacji zadania J_1 o interwał $f(o_{1,2n+2}) - nL$ otrzymujemy lepsze uszeregowanie niż f. Sprzeczność.

Przedstawione własności rozwiązań problemu Φ prowadzą do następującego lematu, którego dowód można znaleźć w [4].

Lemat 4. *Jeżeli $A = \{a_1, a_2, ..., a_{2n}\}$, $a_1 < a_2 < ... < a_{2n}$ i f jest rozwiązaniem problemu Φ dla A, to odpowiedź dla problemu Θ, dla zbioru A jest pozytywna wtedy i tylko wtedy, gdy*
$$S(f) = \sum_{i=1}^{n-1}(n-i)(a_{2i-1}+a_{2i}) + (n^2+4n+\tfrac{1}{2})L.$$

Twierdzenie 1. *Problem szeregowania zadań z optymalizacją średniego czasu zakończenia operacji w otwartym systemie obsługi NOSS z dwoma zadaniami, z których jedno składa się tylko z jednej operacji, jest problemem NP-trudnym.*

Dowód

Z lematu 4 wynika, że problem Θ, który jest problemem NP-zupełnym znanym jako "odd-even partition problem", może być wielomianowo sprowadzony do problemu Φ, który jest szczególnym przypadkiem problemu wymienionego w tezie twierdzenia.

Procesory i zadania pełnią analogiczną, równorzędną rolę w ograniczeniu dozwolonych uszeregowań. Średni czas zakończenia operacji jako kryterium optymalizacji (w odróżnieniu od średniego czasu zakończenia zadania) także nie wyróżnia żadnego z elementów tej pary.

Wniosek 1. *Problem szeregowania zadań z optymalizacją średniego czasu zakończenia operacji w otwartym systemie obsługi NOSS z dwoma procesorami, z których jeden jest wykorzystywany tylko przez jedno zadanie, jest problemem NP-trudnym.*

4. Podsumowanie

Przedstawione negatywne rezultaty wyznaczają istotną granicę możliwości określenia "szerokiej" klasy przypadków NOSS z optymalizacją średniego czasu zakończenia operacji, o wielomianowej złożoności czasowej (o ile $P \neq NP$). Naturalne staje się pytanie o status klasy przypadków o ograniczonej liczbie operacji dla każdego zadania i procesora przez funkcję $o(n)$, a w szczególności przez pewną stałą c.

Z [1] wiadomo, że dla $c = 3$ problem pozostaje NP-trudny. Dla $c = 2$ problem staje się wielomianowy, co wynika z prostego uogólnienia rozumowania przedstawionego w [6], gdzie przyjęto dodatkowe założenie wymagające acykliczności grafu, którego wierzchołki odpowiadają zadaniom i procesorom, a krawędzie są wyznaczone przez operacje o niezerowej długości.

LITERATURA

1. Giaro K., Kubale M., Małafiejski M., Piwakowski K.: Chromatic scheduling of dedicated 2-processor UET tasks to minimize mean flow time, ETFA'99, Barcelona, 1999, pp. 343-347.
2. Kubale M., Piwakowski K.: A linear time algorithm for edge-coloring of binomial trees, Disc. Math., 150, 1996, 247-256.
3. Kubale M., Piwakowski K.: Open shop problems with simplified time operations, System Modelling Control VIII, vol. 1, Zakopane 1995, pp. 457-461.
4. Piwakowski K.: Interwałowe kolorowanie krawędzi nadgrafów gwiazdy przy optymalizacji sumy chromatycznej, Rap. tech. 66/99, Politechnika Gdańska, Wydział ETI, 1999.
5. Piwakowski K.: Szeregowanie zadań w otwartym systemie duoprocesorowym, Zeszyty Naukowe Politechniki Śląskiej, s. Automatyka, z. 114, Gliwice 1994, 231-239.

6. Piwakowski K.: Szeregowanie zadań z optymalizacją średniego czasu przepływu przy ograniczonej możliwości zrównoleglania przez graf ścieżki wykluczeń, Rap. tech. 28/97, Politechnika Gdańska, Wydział ETI, 1999.
7. de Werra D.: Some Graph-Theoretical Models for Scheduling in Automated Production Systems. Networks, Vol. 23, 1993, pp. 651-660.

Recenzent: Prof.dr hab.inż. A.Janiak

Abstract

The traditional optimization goal is to minimize the total completion time or the average completion time of the jobs. In this paper we consider another objective – the average completion time of the operation. In contrast to the average completion time of the jobs in this case we take into consideration completion times of all operations and not only the latest one of each job. Such objective corresponds for example with the situation, where there is a constant amount of not renewable resources assigned to each operation and for which the cost of storing (renting) is proportional to the time of storing (for example dangerous toxic fuel or catalyst). The important assumption here is that each operation consumes the same amount of resources. Consider the following example: m patients come to the hospital and each of them has to undergo k surgeries – one in each of k operating rooms. Until the i-th surgery is done the patient must be connected to a special piece of equipment (monitor) responsible for the i-th body action (at the beginning each patient is immediately connected to k monitors and then successively, after each surgery one monitor is disconnected). The cost of renting monitor is relative to the renting time. In this case the optimal scheduling minimizes the total cost of renting the equipment.

The Non-Preemptive Open Shop Scheduling Problem for each of the three mentioned optimization objectives is *NP*-hard. Thus the natural goal is to look for a possibly wide classes of cases for which the problem remains polynomial solvable. Usually the restrictions for the general model reduce the possible execution times of operations (for example all operations have unit execution time) or reduce the set of operations (for example each task consists of at most two operations). Until now more attention were focussed on the establishing such classes for the total completion time objective – for example [2], [3], [5], [7]. In this paper we consider only the average operation completion time and we do not assume any restriction on the execution time of the operations. The main result of this paper is presented in Theorem 1, which states that the NOSS problem to minimize operation average completion time remains *NP*-hard even if we restrict the model to two tasks and one of them requires only one operation.