ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: AUTOMATYKA z. 117 Nr kol. 1337

Adam JANIAK
Politechnika Wrocławska

ANALIZA ZŁOŻONOŚCI OBLICZENIOWEJ JEDNOMASZYNOWYCH PROBLEMÓW SZEREGOWANIA Z TERMINAMI DOSTĘPNOŚCI ZADAŃ ZALEŻNYMI OD ZASOBÓW

Streszczenie. W pracy rozpatrzono jednomaszynowe problemy szeregowania zadań z terminami dostępności zadań zależnymi od zasobów przy dowolnych technologicznych ograniczeniach kolejnościowych pomiędzy zadaniami. Uwzględniono następujące kryteria: czas zakończenia wszystkich zadań oraz całkowite zużycie zasobów. Zbadano złożoność obliczeniową rozpatrywanych problemów oraz ich szczególnych przypadków. Zaproponowano także przybliżone algorytmy ich rozwiązania.

COMPUTATIONAL COMPLEXITY ANALYSIS OF SINGLE MACHINE SCHEDULING PROBLEMS WITH JOB RELEASE DATES DEPENDENT ON RESOURCES

Summary. In the paper single machine scheduling problems with job release dates dependent on resources and some technological precedence constraints among jobs were considered. The following criteria were taken into account: maximum job completion time, total resource consumption. Computational complexity of the considered problems and their special cases were examined. Some approximation algorithms were also proposed.

1. Wprowadzenie

W wielu dyskretnych lub dyskretno-ciągłych [3] procesach produkcyjnych mamy często do czynienia z tzw. "wąskim gardłem" na jednej maszynie, zwanej "maszyną krytyczną". Czasy wykonywania zadań na niekrytycznych maszynach poprzedzających maszynę krytyczną mogą być potraktowane jako czasy (terminy) gotowości poszczególnych zadań na maszynę krytyczną. W pracy zakłada się, że terminy te, jako czasy wykonywania zadań na maszynach poprzedzających maszynę krytyczną, nie są z góry ustalone, ale zależą od zasobów podzielnych w sposób ciągły, takich jak: gaz, paliwo, katalizator, środki finansowe. Ponadto w pracy przyjmuje się, że czasy wykonywania zadań na maszynie krytycznej są z góry ustalone. Przyjmuje się także, że pomiędzy zadaniami występują dowolne technologiczne

ograniczenia kolejnościowe. Jako kryterium przyjęto minimalizację czasu zakończenia wszystkich zadań, która jest równoważna minimalizacji przestojów maszyny krytycznej przy zadanym ograniczeniu na globalną ilość zasobów przeznaczoną do wykonania rozpatrywanych zadań na maszynach poprzedzających maszynę krytyczną. Rozpatrzono również kryterium będące minimalizacją całkowitego zużycia zasobów przy zadanym nieprzekraczalnym terminie wykonania wszystkich zadań. Z tego typu problemami mamy do czynienia np. podczas procesu produkcyjnego występującego w hutach stali na odcinku produkcyjnym stalownia konwertorowa - walcarka-zgniatacz, polegającym na podgrzewaniu wlewków za pomocą gazu w piecach wgłębnych, a następnie walcowaniu tych wlewków na walcarce-zgniataczu. Proces ten szczegółowo został opisany w [3].

Opis założeń rozpatrywanych w pracy problemów znajduje się w rozdziale 3. Precyzyjne sformułowanie dwu głównych problemów pracy jest w rozdziale 4. Złożoność obliczeniowa rozpatrywanych problemów oraz ich szczególnych przypadków jest dyskutowana w rozdziale 4. Opis algorytmów przybliżonych rozwiązujących badane problemy zamieszczono w rozdziale 5. Krótkie uwagi podsumowujące znajdują się natomiast w rozdziale 6.

W rozdziale następnym krótko, w zarysie, zostanie przypomniany przyjęty w literaturze [1] schemat notacji problemów szeregowania zadań. Schemat ten zostane ograniczony do jednomaszynowych problemów, a jednocześnie uogólniony w ten sposób, by mógł on obejmować wszystkie przypadki rozpatrywanych problemów z zasobami.

2. Schemat notacji problemów szeregowania zadań i rozdziału zasobów

Każdy problem szeregowania zadań i rozdziału zasobów można w sposób jednoznaczny zanotować za pomocą uporządkowanego ciągu parametrów podzielonych na trzy podciągi $E|\gamma|Q$, przy czym pewne parametry nie muszą mieć nadanych wartości, a ich znaczenie jest następujące.

Podciąg $E = E_1E_2E_3$ opisuje zbiór maszyn.

Parametr E_1 charakteryzuje rodzaje maszyn.

Poszczególne wartości tego parametru są następujące i oznaczają:

$E_1 := \emptyset$ – zbiór maszyn zawiera dokładnie jedną maszynę;

$E_1 := \neq \varnothing$ – zbiór maszyn zawiera więcej niż jedną maszynę, np. $E_1 := P$ oznacza wielomaszynowy system z identycznymi maszynami równoległymi.

Parametr E_2 oznacza liczbę maszyn, w naszym przypadku:

$E_2 := 1$.

Parametr E_3 charakteryzuje przepustowość maszyn.

$E_3 \in \{\varnothing, \text{nieog. przep.}\}$, gdzie $E_3 := \varnothing$ oznacza, że każda z maszyn może wykonywać w każdej chwili tylko jedno zadanie, natomiast $E_3 :=$ nieog. przep. oznacza, że wszystkie rozpatrywane w problemie maszyny mogą wykonywać jednocześnie nieograniczoną liczbę zadań. (Gdy w danym problemie tylko część maszyn jest o nieograniczonej przepustowości, wówczas przed symbolem "nieog. przep." pojawią się numery tych maszyn).

Podciąg $Y = Y_1 Y_2 Y_3 Y_4 Y_5$ opisuje zbiór zadań oraz dodatkowe ograniczenie przyjęte dla zadań z tego zbioru.

Parametr $Y_1 \in \{\varnothing, \text{podz.}\}$ określa możliwość przerywania wykonywania zadań.

$Y_1 := \varnothing$ – zadania niepodzielne;

$Y_1 :=$ podz. – zadania podzielne.

Parametr $Y_2 \in \{\varnothing, <\}$ określa ograniczenia kolejnościowe w zbiorze zadań:

$Y_2 := \varnothing$ – brak ograniczeń kolejnościowych (zadania niezależne);

$Y_2 := <$ – dowolne ograniczenia kolejnościowe (zadania zależne).

Parametr $Y_3 \in \{\varnothing, r_j, r_j = b_j - a_j u_j, r_j = f_j(u_j)\}$ charakteryzuje terminy dostępności zadań:

$Y_3 := \varnothing$ – terminy dostępności są ustalone i równe, a zatem można przyjąć, że $r_j = 0$, $j = 1, 2, \ldots, n$;

$Y_3 := r_j$ – terminy dostępności są dowolne, ale ustalone;

$Y_3 := r_j = b_j - a_j u_j$ – terminy dostępności są (zmiennymi decyzyjnymi) nieujemnymi, malejącymi, liniowymi funkcjami zasobu;

$Y_3 := r_j = f_j(u_j)$ – terminy dostępności są (zmiennymi decyzyjnymi) nieujemnymi, ściśle malejącymi, ciągłymi funkcjami ilości zasobu (symbol $r_j = f(u_j)$ będzie oznaczać, że funkcje $f_j(\cdot)$ są jednakowe dla wszystkich zadań).

Parametr $Y_4 \in \{\varnothing, 1, p_j = b_j - a_j u_j, p_j = f_j(u_j)\}$ określa czasy wykonywania zadań:

$Y_4 := \varnothing$ – czasy wykonywania zadań (operacji) są dowolne, ale ustalone;

$Y_4 := p_j = b_j - a_j u_j$ – czasy wykonywania zadań są (zmiennymi decyzyjnymi) nieujemnymi, malejącymi, liniowymi funkcjami ilości zasobu.

Parametr $Y_5 \in \{\varnothing, \sum u_j \le \hat{U}, C_{max} \le \hat{C}\}$ określa dodatkowe ograniczenia:

$Y_5 := \varnothing$ – brak dodatkowych ograniczeń;

$Y_5 := \sum u_j \le \hat{U}$ – istnieje ograniczenie na globalną ilość zasobu przeznaczoną do wykonania wszystkich zadań;

$Y_5 := C_{\max} \le \hat{C}$ – istnieje ograniczenie na czas zakończenia wszystkich zadań.

Parametr $Q \in \{C_{\max}, \sum u_j, \sum u_j \wedge C_{\max}, \sum C_j\}$ oznacza symbol kryterium optymalizacji (w niniejszej pracy wszystkie kryteria będą minimalizowane):

$Q := C_{\max}$ – minimalizacja czasu zakończenia wykonania wszystkich zadań;

$Q := \sum u_j$ – minimalizacja całkowitej ilości zużytego zasobu;

$Q := \sum u_j \wedge C_{\max}$ – dwukryterialne podejście z jednoczesną minimalizacją $\sum u_j$ oraz $C_{\max}$;

$Q := \sum C_j$ – minimalizacja sumy czasów zakończenia wykonywania wszystkich zadań.

3. Opis założeń problemów

Dany jest zbiór n zadań $J = \{J_1, J_2, \ldots, J_j, \ldots, J_n\}$. Zadania te mają być wykonane za pomocą jednej maszyny stanowiącej gniazdo krytyczne. Zakłada się, że maszyna jest dostępna

w sposób ciągły i może wykonywać co najwyżej jedno zadanie w każdej chwili czasu. Zakłada się ponadto, że zadania są niepodzielne, tzn. każde zadanie musi być wykonywane bez przerwań od momentu rozpoczęcia aż do momentu zakończenia jego wykonywania. Zadane są kolejnościowe ograniczenia technologiczne "<" pomiędzy zadaniami. Ograniczenia te wyprowadzone są z acyklicznego skierowanego grafu G z wierzchołkami $\{1,2,\ldots,n\}$ odpowiadającymi poszczególnym zadaniom. Graf ten odzwierciedla technologiczne zależności kolejnościowe między zadaniami. Jeśli G zawiera skierowaną drogę z j do k, to pisze się, że $J_j < J_k$ i wymaga się, aby J_j było zakończone, zanim J_k może się rozpocząć. Zbiór wszystkich ograniczeń kolejnościowych pomiędzy zadaniami będzie oznaczony za pomocą zbioru P_0.

Przyjmuje się, że dla każdego zadania J_j zadany jest czas jego realizacji p_j, $j = 1,2,\ldots,n$, który jest stały i nie zależy od zasobów. Dla każdego zadania J_j, $j = 1,2,\ldots,n$, zadany jest również termin r_j jego gotowości (dostępności) do realizacji; innymi słowy, jest to czas, który musi upłynąć od chwili zerowej, aby zadanie to było gotowe do realizacji. Zakłada się przy tym, że czas ten nie jest zadany, ale zależy od ilości przydzielonego zasobu u_j i może zmieniać się w przedziale $\left[r_j^c, r_j^n\right]$, $j = 1,2,\ldots,n$. Czas ten jest opisany modelem:

$$r_j = f_j(u_j) = b_j - a_j u_j, \quad j = 1,2,\ldots,n, \tag{1}$$

gdzie a_j, b_j są zadanymi parametrami technologicznymi przyjętego liniowego modelu terminu dostępności zadania.

Zakłada się ponadto, że:

$$\alpha_j \le u_j \le \beta_j, \quad j = 1,2,\ldots,n, \tag{2}$$

gdzie α_j, β_j $(0 \le \alpha_j \le \beta_j \le b_j / a_j)$ są lokalnymi (dolnymi i górnymi) ograniczeniami technologicznymi na ilość zasobu przydzielonego do zadania J_j lub do maszyny (poprzedzającej rozpatrywaną maszynę krytyczną), na której zadanie to jest wcześniej, tzn. przed maszyną krytyczną, realizowane. Minimalna wartość terminu dostępności zadania J_j wynosi:

$$r_j^c = b_j - a_j\beta_j, \quad j = 1,2,\ldots,n, \tag{3}$$

natomiast maksymalna wartość wynosi:

$$r_j^n = b_j - a_j\alpha_j, \quad j = 1,2,\ldots,n. \tag{4}$$

Zadane może być również ograniczenie $\hat{U}$ na globalną ilość zasobu dysponowaną do rozdziału lub ograniczenie na nieprzekraczalny termin (tzw. "deadline") zakończenia wykonania wszystkich zadań $\hat{C}$.

Niech π oznacza dopuszczalną (tzn. spełniającą zadane technologiczne ograniczenie kolejnościowe P_0) permutację zadań z J, a $J_{\pi(j)}$ niech oznacza zadanie na pozycji j w permutacji π. Ponadto niech Π oznacza zbiór wszystkich dopuszczalnych permutacji π.

W przypadku gdy zadane jest ograniczenie $\hat{U}$ na globalną ilość zasobu, zbiór wszystkich dopuszczalnych rozdziałów u zasobu jest następujący:

$$U \equiv \left\{ u \in R_+^n : u \equiv \left[u_1, \ldots, u_n\right], \alpha_j \le u_j \le \beta_j, j = 1, \ldots, n, \sum_{j=1}^{n} u_j \le \hat{U} \right\}.$$

Natomiast w przypadku gdy nie ma ograniczenia na globalną ilość zasobu, tzn. np. gdy nie jest ono aktywne lub gdy globalna ilość zużytego zasobu jest minimalizowana przy zadanym ograniczeniu na nieprzekraczalny termin zakończenia wykonania wszystkich zadań, zbiór wszystkich dopuszczalnych rozdziałów u zasobu jest następujący:

$$U' \equiv \left\{ u \in R_+^n : u \equiv \left[u_1, \ldots, u_n\right], \alpha_j \le u_j \le \beta_j, j = 1, \ldots, n \right\}.$$

Dla ustalonej dopuszczalnej kolejności wykonywania zadań $\pi \in \Pi$ (dokładniej permutacji indeksów zadań) oraz zadanego dopuszczalnego rozdziału zasobu $u \in U$ lub $u \in U'$, czas zakończenia wykonywania zadania $J_{\pi(j)}$, $j = 1, 2, \ldots, n$, w uszeregowaniu π, oznaczany $C_{\pi(j)}(\pi, u)$, można zdefiniować w sposób następujący:

$$C_{\pi(1)}(\pi, u) \equiv b_{\pi(1)} - a_{\pi(1)} u_{\pi(1)} + p_{\pi(1)},$$
$$C_{\pi(j)}(\pi, u) \equiv max\left[b_{\pi(j)} - a_{\pi(j)} u_{\pi(j)}, C_{\pi(j-1)}(\pi, u)\right] + p_{\pi(j)}, \quad j = 2, \ldots, n.$$

Czas zakończenia wykonywania zadania $J_{\pi(j)}$, $j = 1, 2, \ldots, n$, $C_{\pi(j)}(\pi, u)$ można także wyznaczyć w następujący sposób:

$$C_{\pi(j)}(\pi, u) \equiv \max_{1 \le i \le j}\left[b_{\pi(i)} - a_{\pi(i)} u_{\pi(i)} + \sum_{k=i}^{j} p_{\pi(k)}\right], \quad j = 1, 2, \ldots, n.$$

Zatem czas zakończenia wykonywania wszystkich zadań $C_{max}(\pi, u)$ dla $\pi \in \Pi$ oraz $u \in U$ lub $u \in U'$ można wyznaczyć w następujący sposób:

$$C_{max}(\pi, u) \equiv \max_{1 \le j \le n}\left[b_{\pi(j)} - a_{\pi(j)} u_{\pi(j)} + \sum_{k=j}^{n} p_{\pi(k)}\right]. \tag{5}$$

Natomiast całkowite zużycie globalnej ilości zasobów $U(\pi, u)$ dla ustalonej dopuszczalnej kolejności wykonywania zadań $\pi \in \Pi$ oraz dopuszczalnego rozdziału zasobów $u \in U'$ będzie oznaczane w następujący sposób:

$$U(\pi, u) \equiv \sum_{j=1}^{n} u_{\pi(j)}. \tag{6}$$

4. Sformułowanie problemów

PROBLEM I $(1|<, r_j = b_j - a_j u_j, C_{max} \le \hat{C} \left| \sum u_j \right.)$

Minimalizacja całkowitego zużycia globalnej ilości zasobów przy zadanym nieprzekraczalnym terminie zakończenia wykonywania wszystkich zadań

Należy znaleźć taką dopuszczalną kolejność wykonywania zadań $\pi^* \in \Pi$ oraz taki dopuszczalny rozdział zasobu $u^* \in U'$, tzn. takie sterowanie (π^*, u^*), aby całkowite zużycie globalnej ilości zasobów było minimalizowane, tzn.

$$U(\pi^*, u^*) \equiv \min_{\pi \in \Pi} \min_{u \in U} U(\pi, u),$$

przy zadanym ograniczeniu na nieprzekraczalny termin zakończenia wykonywania wszystkich zadań $\hat{C}$, tzn.:

$$C_{max}(\pi^*, u^*) \le \hat{C}.$$

Niech $\alpha := [\alpha_1, \alpha_2, \ldots, \alpha_n]$, $\beta = [\beta_1, \beta_2, \ldots, \beta_n]$, $\underline{C} \equiv \min_{\pi \in \Pi}\{C_{max}(\pi, \beta)\}$ oraz $\overline{C} \equiv \min_{\pi \in \Pi}\{C_{max}(\pi, \alpha)\}$.

Oczywiste jest, że wartość $\underline{C}$ oraz $\overline{C}$ można znaleźć w $O(n \log n)$ krokach poprzez uszeregowanie zadań odpowiednio zgodnie z niemalejącymi wartościami $f_j(\beta)$ oraz $f_j(\alpha)$.

Łatwo można wykazać, że zachodzą następujące własności:

Własność 1

a) optymalne uszeregowanie dla problemu $1|<, r_j = b_j - a_j u_j, C_{\max} \le \hat{C} \left| \sum u_j \right.$ istnieje wtedy i tylko wtedy, gdy $\hat{C} \ge \underline{C}$. Ponadto, jeśli $\hat{C} \ge \underline{C}$, wówczas:

b) jeśli $U(\pi^*, u^*) > \sum_{j=1}^{n} \alpha_j$, wtedy $C_{max}(\pi^*, u^*) < \overline{C}$ oraz

c) jeśli $\hat{C} \le \overline{C}$, wówczas $C_{max}(\pi^*, u^*) = \hat{C}$.

Dzięki powyższym własnościom analizę problemu można ograniczyć do przypadku, w którym:

$$\underline{C} \le \hat{C} \le \overline{C}.$$

Dla ustalonej permutacji $\pi \in \Pi$, przez $u^*_\pi \in U$ będzie oznaczany rozdział zasobów, dla którego całkowite zużycie zasobu jest minimalizowane przy zadanym ograniczeniu $\hat{C}$ na czas zakończenia wykonywania wszystkich zadań, tzn. zachodzi:

$$U(\pi, u^*_\pi) = \min_{u \in U'} \{U(\pi, u)\} \quad przy$$
$$C_{max}(\pi, u^*_\pi) \le \hat{C}.$$

Fakt istnienia takiego rozdziału zasobów u^*_π dla dowolnego ustalonego uszeregowania $\pi \in \Pi$ oraz zadanego ograniczenia $\hat{C}$ może być sprawdzony w $O(n)$ krokach poprzez zweryfikowanie, czy zachodzi:

$$C_{max}(\pi, \beta) \le \hat{C}.$$

Łatwo można wykazać, że zachodzi następująca własność:

Własność 2

Jeśli dla ustalonego $\pi \in \Pi$ oraz $\hat{C}$ optymalny rozdział zasobu $u^*_\pi \in U'$ istnieje, wówczas może być on otrzymany w $O(n)$ krokach za pomocą następującego algorytmu:

Algorytm 1

$$u^*_{\pi(j)} = max\left\{\alpha_{\pi(j)}, min\left\{\beta_{\pi(j)}, \left(b_{\pi(j)} + \sum_{i=j}^{n} p_{\pi(i)} - \hat{C}\right) / a_{\pi(j)}\right\}\right\}.$$

PROBLEM II $(1|<, r_j = b_j - a_j u_j, \sum u_j \le \hat{U} | C_{max})$

Minimalizacja czasu zakończenia wykonywania wszystkich zadań przy zadanym ograniczeniu na całkowitą globalną ilość zasobów

Należy znaleźć taką dopuszczalną kolejność wykonywania zadań $\pi^* \in \Pi$ oraz taki dopuszczalny rozdział zasobu $u^* \in U$, tzn. takie sterowanie (π^*, u^*), aby czas wykonywania wszystkich zadań był minimalny, tzn.:

$$C_{max}(\pi^*, u^*) = \min_{\pi \in \Pi} \min_{u \in U} C_{max}(\pi, u)$$

(przy zadanym ograniczeniu na dopuszczalną globalną ilość zasobów $\hat{U}$, tzn.:

$$\sum_{j=1}^{n} u^*_{\pi(j)} \le \hat{U}.)$$

W powyższym problemie przez $u^*_\pi \in U$ będzie oznaczany optymalny rozdział zasobu dla ustalonej permutacji π, $\pi \in \Pi$, tzn. rozdział, dla którego zachodzi

$$C_{max}(\pi, u^*_\pi) = \min_{u \in U} C_{max}(\pi, u).$$

Można wykazać, że zachodzi następująca własność:

Własność 3

Dla ustalonej permutacji zadań π, $\pi \in \Pi$, optymalny rozdział zasobu $u^*_\pi \in U$ może być uzyskany w $O(n^2)$ krokach za pomocą następującego algorytmu:

Algorytm 2

Krok 1. Podstaw $u^*_{\pi(i)} := \alpha_{\pi(i)}$, $\hat{U} := \hat{U} - \sum_{j=1}^{n} \alpha_j$, $C_{\pi(i)} := b_{\pi(i)} - a_{\pi(i)} u_{\pi(i)} + \sum_{j=i}^{n} p_{\pi(j)}$, dla $i = 1, 2, \ldots, n$. Następnie podstaw $I := \{\pi(i) : i = 1, 2, \ldots, n\}$, $l := 0$, $C_0 := 0$, $P_0 := 0$ i przejdź do Kroku 2.

Krok 2. Podstaw $l := l + 1$ i znajdź zbiór $P_l := \left\{ \pi(i) : \pi(i) \in I, C_{\pi(i)} = \min_{\pi(j) \in I} C_{\pi(j)} \right\}$. Następnie podstaw $I := I - P_l$. Jeśli I jest pusty, wówczas podstaw $Q := -P_l$ i przejdź do Kroku 3, w przeciwnym przypadku przejdź do Kroku 2.

Krok 3. Jeśli $\hat{U} = 0$ lub $l = 0$ lub $\min_{j \in Q} \{\beta_j - u^*_j\} = 0$, wówczas Stop – u^*_π jest optymalnym rozdziałem zasobów dla π; w przeciwnym przypadku przejdź do Kroku 4.

Krok 4. Podstaw $x := \min \left\{ C_q - C_p, \hat{U} / \sum_{j \in Q} (1/a_j), \min_{j \in Q} \{a_j (\beta_j - u^*_j)\} \right\}$, gdzie q i p są indeksami zadań odpowiednio ze zbiorów Q oraz P_{l-1}. Następnie podstaw $u^*_j := u^*_j + x / a_j$ dla $j \in Q$, $\hat{U} := \hat{U} - \sum_{j \in Q}^{n} x / a_j$, $l := l - 1$, $Q := Q \cup P_l$ i przejdź do Kroku 3.

5. Złożoność obliczeniowa problemów

Obecnie wykażemy, że problem $1|<, r_j = b_j - a_j u_j, C_{max} \le \hat{C} | \sum u_j$ jest NP-trudny, tzn. decyzyjna wersja (oznaczana DFPI) rozpatrywanego problemu (optymalizacyjnego) jest NP-zupełna nawet dla przypadku z identycznymi modelami terminów dostępności zadań, tzn. dla $b_j = b$, $a_j = a$, $\alpha_j = \alpha$, $\beta_j = \beta$, $j = 1, 2, \ldots, n$.

Decyzyjna wersja rozpatrywanego problemu jest sformułowana następująco:

DFPI: Dany jest jednomaszynowy problem szeregowania n niepodzielnych zadań z dowolnymi ograniczeniami kolejnościowymi " < ", z czasami wykonywania zadań p_j, $j = 1, 2, \ldots, n$ oraz modelami (1) terminów gotowości zadań z parametrami $a_j, b_j, \alpha_j, \beta_j$ oraz dane są wielkości

$\hat{C}$ i y, gdzie n, $a_j, b_j, \alpha_j, \beta_j$, $j = 1,2,\ldots,n$, są nieujemnymi liczbami całkowitymi, a $\hat{C}$ oraz y są nieujemnymi liczbami wymiernymi.

Pytanie: Czy istnieje sterowanie (π, u), $\pi \in \Pi$, $u \in U'$, takie że $U(\pi,u) \le y$?

Własność 4

Decyzyjna wersja DFPI problemu $1|<, r_j = b_j - a_j u_j, C_{\max} \le \hat{C} \,|\, \sum u_j$ jest NP-zupełna, nawet dla przypadku z identycznymi modelami terminów gotowości zadań, tzn. dla $a_j = a$, $b_j = b$, $\alpha_j = \alpha$, $\beta_j = \beta$, $j = 1,2,\ldots,n$.

Dowód:

Oznaczmy przez $DF(1|\rightarrow|\sum C_i)$ decyzyjną wersję klasycznego (tzn. bez zasobów) jednomaszynowego problemu szeregowania zadań z ograniczeniami kolejnościowymi "$\rightarrow$" i z kryterium będącym sumą czasów zakończenia wykonywania wszystkich zadań.

NP-zupełność DFPI będzie dowodzona poprzez wykazanie, że $DF(1|\rightarrow|\sum C_i)$ (będąca NP-zupełną [4]) jest wielomianowo transformowalna do DFPI. Rozważmy:

$DF(1|\rightarrow|\sum C_i)$: Dany jest jednomaszynowy problem szeregowania m niepodzielnych zadań z dowolnymi ograniczeniami kolejnościowymi "$\rightarrow$", z czasami wykonywania zadań p'_i, $i = 1,2,\ldots,m$, oraz liczbą z (gdzie m, z, p'_i, $i = 1,2,\ldots,m$, są nieujemnymi liczbami całkowitymi).

Pytanie: Czy istnieje uporządkowanie π' zadań $J_1,\ldots,J_m$, spełniające zadane ograniczenia kolejnościowe "$\rightarrow$", takie że:

$$\sum_{i=1}^{m} C_{\pi'(i)} \le z \ ?$$

Odpowiednia instancja DFP jest zdefiniowana następująco:

$$n = m,\ p_j = p'_j,\ b_j = b = \sum_{i=1}^{m} p'_i,\ \beta_j = \beta = \sum_{i=1}^{m} p'_i,\ a_j = a = 1,\ j = 1,2,\ldots,n,\ y = z,\ \hat{C} = \sum_{i=1}^{m} p'_i,$$

a ograniczenia kolejnościowe "<" pomiędzy zadaniami w DFPI należy przyjąć dokładnie odwrotne do ograniczeń "$\rightarrow$" z problemu $1|\rightarrow|\sum C_i$.

Obecnie zostanie pokazane, że dla DFPI zachodzi $U(\pi,u) \le y$ ($\pi \in \Pi$, $u \in U'$) wtedy i tylko wtedy, gdy dla $DF(1|\rightarrow|\sum C_i)$ można znaleźć uporządkowanie zadań π' (spełniające zadane ograniczenia kolejnościowe "$\rightarrow$"), takie że zachodzi:

$$\sum_{i=1}^{m} C_i \le z\ .$$

(i) Jeżeli dla pewnego dopuszczalnego uporządkowania zadań π' zachodzi $\sum_{i=1}^{m} C_i \le z$, tzn. jeśli $DF(1|\to|\sum C_i)$ ma rozwiązanie, to kolejność wykonywania zadań $\hat{\pi} \in \Pi$ oraz rozdział zasobu $\hat{u} \in U'$, dla którego dla DFPI zachodzi $U(\hat{\pi},\hat{u}) \le y$, są otrzymywane przez wykonywanie zadań w następujący sposób:

$$\hat{\pi}(i) = \pi'(n-(i-1)), \quad i = 1,2,\ldots,n, \tag{7}$$

(każda uporządkowana para zadań $J_i \to J_j$ w $DF(1|\to|\sum C_i)$ jest zamieniana na odwrotną w DFPI, tzn. na parę $J_j < J_i$) $\hat{u}_{\hat{\pi}(1)} = \beta = \sum_{i=1}^{m} p'_{\pi'(i)}$, $\hat{u}_{\hat{\pi}(i)} = \sum_{j=1}^{m-i+1} p'_{\pi'(j)}$, $i = 2,3,\ldots,n$.

Łatwo sprawdzić, że dla sterowania $(\hat{\pi},\hat{u})$ zachodzi:

$\sum_{i=1}^{n} \hat{u}_{\hat{\pi}(i)} = \sum_{i=1}^{m} C_i \le z = y$ oraz $C_{max}(\hat{\pi},\hat{u}) = \sum_{i=1}^{n} p_i = \sum_{i=1}^{m} p'_i = \hat{C}$, a rozdział zasobu $\hat{u}$ jest optymalny dla permutacji $\hat{\pi}$, tzn. $\hat{u} = u^*_{\hat{\pi}}$.

Zatem pokazaliśmy, że jeśli $DF(1|\to|\sum C_i)$ ma rozwiązanie, to DFPI ma je również.

(ii) Należy teraz jeszcze pokazać, że jeśli DFPI ma rozwiązanie, to $DF(1|\to|\sum C_i)$ ma je także. Jest to równoważne wykazaniu, że jeśli $DF(1|\to|\sum C_i)$ nie ma rozwiązania, tzn. zachodzi $\sum_{i=1}^{m} C_{\pi'(i)} > z$, dla wszystkich dopuszczalnych π', to – jak zostanie pokazane – dla każdego uporządkowania π o postaci (7) (które wyczerpuje wszystkie możliwe dopuszczalne uporządkowania zadań) i dla każdego dopuszczalnego rozdziału zasobów $u \in U'$ zachodzi $U(\pi,u) > y$ przy spełnieniu ograniczenia:

$$C_{\max}(\pi,u) \le \hat{C}. \tag{8}$$

Łatwo zauważyć, że dla dowolnego uporządkowania zadań π (o postaci (7)), minimalna całkowita ilość zasobów, zapewniająca spełnienie ograniczenia (8) jest zużyta przy rozdziale zasobów $\hat{u}_\pi \in U'$, dla którego zachodzi:

$$r_{\pi(1)} = f_{\pi(1)}\left(u_{\pi(1)}\right) = 0,$$
$$r_{\pi(2)} = f_{\pi(2)}\left(u_{\pi(2)}\right) = p_{\pi(1)},$$
$$\vdots$$
$$r_{\pi(i)}\left(\hat{u}_{\pi(i)}\right) = \sum_{j=1}^{i-1} p_{\pi(j)}, \quad i = 3,4,\ldots,n,$$

tj. dla następującego rozdziału zasobów:

$$\dot{u}_{\pi(1)} = \sum_{j=1}^{m} p'_{\pi'(j)},$$
$$\dot{u}_{\pi(2)} = \sum_{j=1}^{m-1} p'_{\pi'(j)},$$
$$\vdots$$
$$\dot{u}_{\pi(i)} = \sum_{j=1}^{m-(i-1)} p'_{\pi'(j)}, \quad i = 3,4,\ldots,n.$$

Zatem, dla dowolnego uporządkowania π (o postaci (7)), minimalna całkowita ilość zasobu $\dot{U}$, przy której ograniczenie (8) jest spełnione, wynosi:

$$\dot{U} = \sum_{i=1}^{n} \dot{u}_{\pi(i)} = \sum_{i=1}^{m} C_{\pi'(i)} .$$

Z założenia mamy, że dla każdego uporządkowania π' zachodzi $\sum_{i=1}^{m} C_{\pi'(i)} > z$, a przy konstrukcji transformacji przyjęto, że $z = y$, więc $\dot{U} > y$.

Zatem wykazano, że dla dowolnego dopuszczalnego rozdziału zasobów $u \in U'$, spełniających ograniczenie (8), zachodzi: $U(\pi, u) > y$.

Reasumując, udowodniono, że $DF(1|\rightarrow|\sum C_i)$ jest wielomianowo transformowalna do DFPI. (Łatwo sprawdzić, że wszystkie warunki wielomianowej transformacji są spełnione.) Wykazanie, że DFPI $\in NP$ jest trywialne. ■

Obecnie wykażemy, że problem $1|<, r_j = b_j - a_j u_j, \sum u_j \le \hat{U}|C_{\max}$ jest NP-trudny.

Decyzyjna wersja (oznaczana DFPII) rozpatrywanego problemu jest następująca:

DFPII: Dany jest jednomaszynowy problem szeregowania n niepodzielnych zadań z dowolnymi ograniczeniami kolejnościowymi "<", z czasami wykonywania zadań p_j, $j = 1,2,\ldots,n$, oraz modelami (1) terminów gotowości zadań z parametrami a_j, b_j, α_j, β_j oraz dane są wielkości $\hat{U}$, y, gdzie n, a_j, b_j, α_j, β_j, $j = 1,2,\ldots,n$, są nieujemnymi liczbami całkowitymi, a $\hat{U}$ oraz y są nieujemnymi liczbami wymiernymi.

Pytanie: Czy istnieje sterowanie (π, u), $\pi \in \Pi$, $u \in U$, takie że $C_{\max}(\pi, u) \le y$?

Własność 5

Decyzyjna wersja DFPII problemu $1|<, r_j = b_j - a_j u_j, \sum u_j \le \hat{U}|C_{\max}$ jest NP-zupełna, nawet dla przypadku z identycznymi modelami terminów gotowości zadań, tzn. dla $a_j = a$, $b_j = b$, $\alpha_j = \alpha$, $\beta_j = \beta$, $j = 1,2,\ldots,n$.

Dowód:

Dowód przeprowadza się podobnie jak dowód Własności 4, tzn. konstruuje się transformację wielomianową z problemu $DF(1|\rightarrow|\sum C_i)$ do DFPII. Transformacja ta w tym przypadku jest następująca:

$$n=m,\ p_j=p'_j,\ b_j=b=\sum_{j=1}^{m} p'_j,\ \beta_j=\beta=\sum_{j=1}^{m} p'_j,\ a_j=a=1,\ (j=1,2,\ldots,n),\ \hat{U}=z,\ y=\sum_{j=1}^{m} p'_j,$$

a ograniczenia kolejnościowe "<" pomiędzy zadaniami w DFPII należy przyjąć dokładnie odwrotnie do ograniczeń "$\rightarrow$" z problemu $1|\rightarrow|\sum C_i$. Pierwsza część dowodu tej własności jest bardzo podobna do pierwszej części dowodu Własności 4, a druga część jest natomiast zbliżona do drugiej części dowodu Własności 4. ■

Łatwo sprawdzić, że prawdziwe są następujące własności:

Własność 6

Problem $1|<,\ r_j=b-au_j,\ \beta_j=\beta,\ \alpha_j=\alpha,\ p_j=p,\ C_{\max}\le\hat{C}|\sum u_j$ można rozwiązać w $O(n^2)$ krokach, szeregując zadania w sposób dowolny, aczkolwiek zgodny z relacją "<" oraz rozdzielając zasoby zgodnie z Algorytmem 1.

Własność 7

Problem $1|<,\ r_j=b_j-a_ju_j,\ \beta_j=\beta,\ \alpha_j=\alpha,\ p_j=p,\ \sum u_j\le\hat{U}|C_{\max}$ można rozwiązać w $O(n^2)$ krokach, szeregując zadania w sposób dowolny, aczkolwiek zgodny z relacją "<" oraz rozdzielając zasoby zgodnie z Algorytmem 2.

Jeśli założy się, że zbiór technologicznych ograniczeń kolejnościowych pomiędzy zadaniami P_0 jest pusty, tzn. $P_0=\emptyset$, to wówczas można wykazać następujące własności:

Własność 8 [2]

Decyzyjna wersja problemu $1|r_j=b_j-a_ju_j,\ C_{\max}\le\hat{C}|\sum u_j$ jest silnie NP-zupełna, natomiast przypadek z identycznymi nachyleniami modeli, tzn. przypadek z $a_i=a$, $i=1,2,\ldots,n$, jest NP-zupełny (w słabym sensie).

Własność 9 [3]

Decyzyjna wersja problemu $1|r_j=b_j-a_ju_j,\ \sum u_j\le\hat{U}|C_{\max}$ jest silnie NP-zupełna, natomiast przypadek z identycznymi nachyleniami modeli, tzn. przypadek z $a_i=a$, $i=1,2,\ldots,n$, jest NP-zupełny (w słabym sensie).

Dla obu problemów rozpatrywanych we Własnościach 8 i 9 można znaleźć szereg przypadków rozwiązywalnych wielomianowo. Przypadki te zebrano w następujących dwu własnościach, które można łatwo udowodnić.

Własność 10

Następujące przypadki problemu $1|r_j = b_j - a_j u_j,\ C_{\max} \le \hat{C} \left| \sum u_j \right.$ można rozwiązać wielomianowo:

(1) $a_i = a,\ \alpha_i = \alpha,\ \beta_i = \beta,\ i = 1,2,\ldots,n,$

(2) $\alpha_i = \alpha,\ \beta_i = \beta,\ p_i = p,\ i = 1,2,\ldots,n,$

(3) $a_i = a,\ \beta_i = \beta,\ p_i = p,\ i = 1,2,\ldots,n,$

(4) $a_i = a,\ \alpha_i = \alpha,\ p_i = p,\ i = 1,2,\ldots,n$

w $O(n \log n)$ krokach poprzez następujące uporządkowanie zadań: (1) według nierosnących p_i, (2) według nierosnących a_i, (3) według nierosnących α_i, (4) według nierosnących β_i oraz (dla uzyskanych uporządkowań) poprzez rozdział zasobów zgodny z Algorytmem 1.

Własność 11

Następujące przypadki problemu $1|r_j = b_j - a_j u_j,\ \sum u_j \le \hat{U} | C_{\max}$ można rozwiązać wielomianowo:

(1) $a_i = a,\ \alpha_i = \alpha,\ \beta_i = \beta,\ i = 1,2,\ldots,n,$

(2) $\alpha_i = \alpha,\ \beta_i = \beta,\ p_i = p,\ i = 1,2,\ldots,n,$

(3) $a_i = a,\ \beta_i = \beta,\ p_i = p,\ i = 1,2,\ldots,n,$

(4) $a_i = a,\ \alpha_i = \alpha,\ p_i = p,\ i = 1,2,\ldots,n$

w $O(n^2)$ krokach poprzez następujące uporządkowanie zadań: (1) według nierosnących p_i, (2) według nierosnących a_i, (3) według nierosnących α_i, (4) według nierosnących β_i oraz (dla uzyskanych uporządkowań) poprzez rozdział zasobów zgodny z Algorytmem 2.

6. Algorytmy przybliżone

Ponieważ zarówno Problem I, jak i Problem II są NP-trudne, do ich rozwiązania zaproponowano algorytmy przybliżone. A mianowicie, do rozwiązania Problemu I zaproponowano następujący algorytm o złożoności $O(n^2)$:

Algorytm 3

Krok 1. Ze zbioru wszystkich zadań J wybierz zadanie J_i z najdłuższym czasem wykonywania zadań nie posiadające jeszcze nie ustawionych poprzedników i ustaw je na kolejnej od początku pozycji poszukiwanego uporządkowania. Następnie zmodyfikuj zbiór J eliminując J_i z J i powtarzaj Krok 1 aż do wyczerpania wszystkich zadań z J.

Krok 2. Dla uzyskanego w Kroku 1 uporządkowania wyznacz optymalny rozdział zasobów za pomocą Algorytmu 1.

Analiza najgorszego przypadku tego algorytmu jest trudnym problemem i pozostaje otwartym zagadnieniem.

Dla Problemu II zaproponowano natomiast następujący algorytm o złożoności również $O(n^2)$:

Algorytm 4

Krok 1. Taki sam jak w Algorytmie 3.

Krok 2. Dla uzyskanego w Kroku 1 uporządkowania wyznacz optymalny rozdział zasobów za pomocą Algorytmu 2.

Dla powyższego algorytmu można znaleźć osiągalny współczynnik najgorszego przypadku równy 2.

7. Uwagi końcowe

Analiza złożoności obliczeniowej rozpatrywanych problemów wykazuje, że drobna zmiana założeń odnośnie do parametrów problemu może spowodować natychmiastowe przejście danego problemu z klasy problemów rozwiązywalnych wielomianowo do klasy problemów NP-trudnych lub nawet silnie NP-trudnych. W dalszych pracach warto się zająć podejściem wielokryterialnym do rozpatrywanych problemów.

LITERATURA

1. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math., vol. 5, 1979, pp. 287-326.
2. Janiak A.: Single machine scheduling problem with a common deadline and resource dependent release dates. European Journal of Operational Research, vol. 55, No 3, 1991, pp. 317-325.

3. Janiak A.: Dokładne i przybliżone algorytmy szeregowania zadań i rozdziału zasobów w dyskretnych procesach przemysłowych. Seria: Monografie No 20, Wyd. Polit. Wrocławskiej, Wrocław 1991.

4. Lawler E.L.: Sequencing jobs to minimize total weighted completion time subject to precedence constraints. Ann. Discrete Math., No 2, 1978, pp. 75-90.

Recenzent: Prof. dr hab. inż. Jan Węglarz

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

In the paper single machine scheduling problems with job release dates and some technological precedence constraints among jobs were considered. It was assumed that job release dates depend on continuously-divisible resources such as: energy, fuel, catalyzer, raw materials, financial outlay. The following optimization criteria were considered: makespan, total resource consumption. Detailed computational complexity analysis of the considered problems and their special cases were conducted. It was appeared that both problems are NP-complete even for the same models of release dates for all jobs. Several special cases with polynomial time algorithms were found. Some approximate algorithms were also presented.