ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1988
Seria: AUTOMATYKA z.94 Nr kol.970

Czesław Smutnicki

Politechnika Wrocławska
Instytut Cybernetyki Technicznej

ZAGADNIENIE SZEREGOWANIA Z OGRANICZENIAMI CZASÓW OCZEKIWANIA

Streszczenie. W pracy rozważany jest jednomaszynowy problem szeregowania z kryterium w postaci maksymalnego terminu zakończenia wykonywania wszystkich zadań. W problemie tym przyjmuje się dodatkowo, że istnieją ograniczenia kolejnościowe wykonywania zadań, przy czym z każdym ograniczeniem tego typu są związane dolne i górne ograniczenie czasu oczekiwania pomiędzy zadaniami. Dla problemu przeprowadzana jest analiza złożoności obliczeniowej oraz proponowany algorytm rozwiązania.

1. Wstęp

W pracy rozważa się jednomaszynowy problem szeregowania z kryterium w postaci maksymalnego terminu zakończenia wykonywania wszystkich zadań. Zakłada się, że dodatkowe ograniczenia kolejnościowe wprowadzane są poprzez określenie dolnego i górnego ograniczenia czasu oczekiwania pomiędzy parą zadań związanych tym ograniczeniem. Problem ten w ogólnym przypadku jest silnie NP-trudny. W celu jego rozwiązania możliwe jest zastosowanie kilku "typowych" podejść. Pierwsze polega na zastosowaniu algorytmu aproksymacyjnego (przybliżonego). Drugie, na zastosowaniu metody podziału i ograniczeń lub metody programowania dynamicznego. Oba podane podejścia posiadają ogólnie znane wady i zalety. Wydaje się, że dla zagadnień o większych rozmiarach jedynym sensownym podejściem jest zastosowanie algorytmów aproksymacyjnych. Aktualny stan badań dotyczących zagadnień jednomaszynowych, opracowany na podstawie 204 pozycji literaturowych, przedstawiono w pracy [10]. Z pracy tej wynika, że nie są znane żadne algorytmy rozwiązywania omawianego problemu w przypadku ogólnym.

W prezentowanej pracy przeprowadzono analizę złożoności obliczeniowej kilku szczególnych przypadków problemu oraz przedstawiono metody wyznaczania górnego i dolnego ograniczenia dla proponowanego algorytmu rozwiązania opartego na metodzie podziału i ograniczeń.

Wyniki otrzymane w pracy mogą być bezpośrednio wykorzystane w algorytmach rozwiązywania problemów znacznie ogólniejszych, np. [1], [2].

Praca była częściowo finansowana przez RP.I.02 "Teoria sterowania i optymalizacji układów dynamicznych i procesów dyskretnych".

2. Sformułowanie problemu

Dany jest zbiór n niezależnych zadań $J = \{1,2,\dots,n\}$, które należy wykonać na jednej maszynie. Dla każdego zadania określony jest czas wykonywania $p_j > 0$, $j \in J$. Dodatkowo dana jest także relacja $R \subset J \times J$, której graf jest acykliczny, oraz liczby $d_{ij} \geq 0$ i $D_{ij} \geq 0$, $(i,j) \in R$, $i,j \in J$. Relacja R przedstawia ograniczenia kolejności wykonywania zadań wynikające z porządku technologicznego. Wielkości d_{ij} oraz D_{ij} reprezentują, odpowiednio, dolne i górne ograniczenie czasu oczekiwania pomiędzy terminem zakończenia zadania i-tego oraz terminem rozpoczęcia zadania j-tego, tzn.

$$d_{ij} \leq S_j - (S_i + p_i) \leq D_{ij}, \qquad (i,j) \in R \tag{1}$$

gdzie S_j jest terminem rozpoczęcia zadania j, $j \in J$. Przyjmuje się, że w każdej chwili czasowej maszyna może realizować co najwyżej jedno zadanie oraz realizacji zadania nie można przerywać. Problem polega na znalezieniu uszeregowania (reprezentowanego przez zestaw terminów rozpoczęcia S_j, $j \in J$), które minimalizuje maksymalny termin zakończenia wykonywania wszystkich zadań oraz spełnia ograniczenie (1). Łatwo pokazać, że jeżeli istnieje co najmniej jedno uszeregowanie spełniające ograniczenie (1), to w zbiorze uszeregowań optymalnych istnieje uszeregowanie, dla którego zadania rozpoczynają się najwcześniej jak mogą. Zatem nasze rozważania możemy ograniczyć do uszeregowań opisanych permutacją π zadań ze zbioru J. Dla danej permutacji π terminy rozpoczęcia zadań wyznaczane są z zależności

$$S_{\pi(i)} = \max\{S_{\pi(i-1)}+p_{\pi(i-1)},\ \max_{j\in B_{\pi(i)}} (S_j+p_j+d_{j\pi(i)})\},\quad i=1,2,\dots,n, \tag{2}$$

gdzie $S_{\pi(0)} = 0 = p_{\pi(0)}$ oraz $B_j = \{i: (i,j) \in R\}$. Kolejność π, dla której spełnione jest także ograniczenie (1) (dla terminów S_j wyznaczonych z (2)), będziemy nazywać kolejnością dopuszczalną. Zbiór kolejności dopuszczalnych oznaczony będzie dalej przez Ω. Zatem powyższy problem formułuje się następująco. Znaleźć kolejność wykonywania zadań $\pi^* \in \Omega$, dla której

$$C_{max}(\pi^*) = \min_{\pi\in\Omega} C_{max}(\pi), \tag{3}$$

gdzie

$$C_{max}(\pi) = \max_{1 \le i \le n} (S_{\pi(i)} + p_{\pi(i)}) \tag{4}$$

oraz $S_{\pi(i)}$, $i = 1,2,...,n$ są terminami wyznaczonymi z zależności (2) dla permutacji π. Problem ten nie występuje w ogólnie przyjętym systemie klasyfikacyjnym problemów szeregowania [9], zatem w dalszym ciągu pracy będzie oznaczany przez $1|prec,d_{ij},D_{ij}|Cmax$. Odpowiednie problemy otrzymane przez relaksacje postaci $d_{ij} = 0$ oraz $D_{ij} = \infty$, $(i,j) \in R$ będą oznaczane jako $1|prec|C_{max}$, $1|prec,d_{ij}|C_{max}$, $1|prec,D_{ij}|C_{max}$.

Problem wyznaczenia terminów S_j z zależności (2) może być sprowadzony do problemu wyznaczania dróg w grafie $G_\pi = (J,R \cup E_\pi)$, gdzie J jest zbiorem wierzchołków, zaś $R \cup E_\pi$ zbiorem łuków, $E_\pi = \bigcup_{i=1}^{n-1} \{(\pi(i),\pi(i+1))\}$. Wszystkie wierzchołki posiadają obciążenie równe p_j, łuki $(i,j) \in R$ obciążenie d_{ij}, zaś łuki $(i,j) \in E_\pi$ obciążenie zerowe. Terminy S_j przedstawiają długość najdłuższej drogi dochodzącej do wierzchołka j, zaś $C_{max}(\pi)$ jest długością najdłuższej drogi w tym grafie. Zatem złożoność obliczeniowa algorytmu realizującego wzór (2) jest $O(n+|R|)$.

W dalszym ciągu pracy relacja R będzie przedstawiana również za pomocą zbiorów bezpośrednich poprzedników $B_j = \{i: (i,j) \in R\}$ oraz zbiorów bezpośrednich następników $A_j = \{i: (j,i) \in R\}$, $j \in J$. Odpowiednio niech $\tilde{B}_j$, $\tilde{A}_j$ będą zbiorami wszystkich (niekoniecznie bezpośrednich) poprzedników i następników zadania j, $j \in J$.

3. Pewne własności problemu

Znane w literaturze problemy $1|prec|C_{max}$, $1|prec|L_{max}$, $1|prec,r_j|C_{max}$, $1|prec,r_j|L_{max}$, $1|prec,C_i \le d_i|*$ mogą być, poprzez odpowiednie zdefiniowanie danych, sprowadzone do problemu $1|prec,d_{ij},D_{ij}|C_{max}$. Zatem rozważany problem jest znacznie ogólniejszy od każdego z ww. wymienionych. Co więcej, można pokazać, że jego wersja decyzyjna jest problemem NP-zupełnym, nawet jeśli poszukujemy tylko dowolnej kolejności dopuszczalnej. Przeprowadzona analiza złożoności obliczeniowej dla szczególnych przypadków ogólnego problemu $1|prec,d_{ij},D_{ij}|C_{max}$ doprowadziła do uzyskania następujących rezultatów.

Problem A.

Jest to problem $1|prec|C_{max}$. Optymalną kolejność można otrzymać przez uporządkowanie grafu (J,R) relacji R w $O(n+|R|)$ krokach.

Problem B.

Jest to problem $1|prec,d_{ij}|C_{max}$ z następującymi dodatkowymi założeniami:

istnieje dokładnie jedno zadanie $k \in J$ i podzbiór $K \subseteq J$ taki, że $R' = \bigcup_{j\in K}\{(k,j)\} \subseteq R$, oraz $d_{ij} > 0$ dla $(i,j) \in R'$, $d_{ij} = 0$ dla $(i,j) \in R \setminus R'$. Opisana poniżej procedura wyznacza optymalną kolejność dla tego problemu:

(a) wyznacz optymalną kolejność σ wykonywania zadań dla problemu postaci $1|prec|C_{max}$ ze zbiorem zadań $\tilde{B}_k$ oraz relacją R ograniczoną do tego zbioru (patrz Problem A),

(b) wyznacz optymalną kolejność ω wykonywania zadań dla problemu postaci $1|prec|C_{max}$ ze zbiorem zadań $C_k = J \setminus (\tilde{B}_k \cup \tilde{A}_k)$ oraz relacją R ograniczoną do tego zbioru (patrz Problem A),

(c) wyznacz optymalną kolejność wykonywania δ zadań dla problemu postaci $1|prec,r_j|C_{max}$ ze zbiorem zadań $\tilde{A}_k$, relacją R ograniczoną do tego zbioru, wartościami $r_j = d_{kj}$ dla $j \in K$ oraz $r_j = 0$ dla $j \in \tilde{A}_k \setminus K$. Kolejność δ można wyznaczyć algorytmem podanym w pracy [11] w $O(n^2)$ krokach,

(d) optymalna kolejność π^* dla Problemu B ma postać $\pi^* = \sigma k \omega \delta$.

Łatwo zauważyć, że wyznaczenie π^* wymaga $O(n^2)$ kroków. W celu uzasadnienia poprawności przedstawionego powyżej postępowania zauważmy wpierw, że dla problemu $1|prec|C_{max}$ zachodzi $C_{max}(\pi) = \sum_{j\in J} p_j$. Stąd $C_{max}(\sigma) = \sum_{j\in\tilde{B}_k} p_j$ jest dolnym ograniczeniem terminu S_k w Problemie B. Podobnie $C_{max}(\delta)$ wyznaczone w p. (b) jest najkrótszym terminem zakończenia zadań ze zbioru $\tilde{A}_k$. Zatem jeśli $C_{max}(\delta)-\sum_{j\in\tilde{A}_k} p_j \geq \sum_{j\in C_k} p_j$, to zadania ze zbioru C_k należy umieścić "pomiędzy" zadaniem k oraz zbiorem $\tilde{A}_k$ i wtedy $C_{max}(\pi^*) = C_{max}(\sigma)+p_k+C_{max}(\delta)$. W przeciwnym przypadku umieszczając zadania ze zbioru C_k jak poprzednio otrzymamy $C_{max}(\pi^*) = C_{max}(\sigma)+p_k+C_{max}(\omega)+\sum_{j\in\tilde{A}_k} p_j = \sum_{j\in J} p_j$, co kończy dowód optymalności π^*.

Problem C.

Jest to problem $1|prec,d_{ij}|C_{max}$ z założeniami jak w Problemie B, z tym że $R' = \bigcup_{j\in K}\{(j,k)\} \subseteq R$. Opisana poniżej procedura wyznacza optymalną kolejność dla tego problemu:

(a) wyznacz optymalną kolejność σ wykonywania zadań dla problemu postaci $1|prec|C_{max}$ ze zbiorem zadań $\tilde{A}_k$ oraz relacją R ograniczoną do tego zbioru (patrz Problem A),

(b) wyznacz optymalną kolejność ω wykonywania zadań dla problemu postaci $1|prec|C_{max}$ ze zbiorem zadań $C_k = J \setminus (\tilde{B}_k \cup \tilde{A}_k)$ oraz relacją R ograniczoną do tego zbioru (patrz Problem A),

(c) wyznacz optymalną kolejność wykonywania δ zadań dla problemu postaci

$1|prec,q_j|C_{max}$ ze zbiorem zadań $\tilde{B}_k$, relacją R ograniczoną do tego zbioru, wartościami $q_j = d_{jk}$ dla $j \in K$ oraz $q_j = 0$ dla $j \in \tilde{A}_k \setminus K$. Optymalną kolejność można wyznaczyć algorytmem podanym w pracy [11] w $O(n^2)$ krokach,

(g) optymalna kolejność π^* dla Problemu B ma postać $\pi^* = \delta\omega k\sigma$.

Podobnie jak poprzednio wyznaczenie π^* wymaga $O(n^2)$ kroków, zaś uzasadnienie poprawności postępowania jest w pełni analogiczne.

Problem D.

Jest to problem $1|prec,d_{ij}|C_{max}$ z dodatkowym założeniem, że istnieje podzbiór $K = \{u,v,w\} \subseteq J$, $(u,v),(v,w) \in R$, dla którego $d_{uv} > 0$, $d_{vw} > 0$. Problem ten jest silnie NP-trudny. Do wykazania tego faktu można skorzystać z problemu podziału (2-podziału) [12].

Problem E.

Jest to problem $1|prec,D_{ij}|C_{max}$ z następującymi dodatkowymi założeniami: (i) graf (J,R) relacji R można zdekomponować na pewną liczbę rozłącznych łańcuchów $c_1,\dots,c_s$ postaci $c_i = (c_{i1},\dots,c_{iv_i})$, gdzie $c_{ij} \in J$ oraz v_i jest długością łancucha i, $i=1,\dots,s$, (ii) $D_{c_{ij}c_{ij+1}} < \infty$, $j=1,\dots v_i-1$, $i=1,\dots,s$. Problem ten posiada rozwiązanie optymalne postaci $\pi^* = c_1c_2\dots c_s\sigma$, gdzie σ jest dowolną kolejnością zadań ze zbioru $J \setminus (c_1 \cup \dots \cup c_s)$. Dowód tego faktu jest oczywisty.

Problem F.

Jest to problem $1|prec,D_{ij}|C_{max}$ z następującymi dodatkowymi założeniami: istnieje zadanie $k \in J$ i podzbiór $K \subseteq J$, taki, że $R' = \bigcup_{j \in K} \{(k,j)\} \subseteq R$, oraz $D_{ij} < \infty$ dla $(i,j) \in R'$, $D_{ij} = \infty$ dla $(i,j) \in R \setminus R'$. Rozważmy problem $1|prec|L_{max}$ ze zbiorem zadań $\tilde{A}_k$, relacją R ograniczoną do tego zbioru, oraz $d_j = D_{kj}$, $j \in K$, $d_j = \infty$, $j \in \tilde{A}_k \setminus K$. Jeśli dla optymalnej kolejności δ dla tego problemu zachodzi $L_{max}(\delta) \le 0$, to optymalna kolejność dla Problemu F ma postać $\pi^* = \sigma\omega k\delta$, gdzie kolejności σ i ω są wyznaczone jak w Problemie B. W przeciwnym przypadku $\Omega = \emptyset$. Kolejność δ można wyznaczyć algorytmem podanym w pracy [3] w $O(n^2)$ krokach. Uzasadnienie optymalności π^* jest analogiczne do przedstawionego dla Problemu B. Podobny rezultat uzyskuje się, gdy relacja R' jest postaci $R' = \bigcup_{j \in K} \{(j,k)\} \subseteq R$.

Problem G.

Jest to problem $1|prec,d_{ij},D_{ij}|C_{max}$ z dodatkowym założeniem, że istnieją dwa zadania $u,v \in J$, $(u,v) \in R$, dla których $d_{uv} > 0$ oraz $D_{uv} < \infty$. Problem ten jest silnie NP-trudny. Do wykazania tego faktu można skorzystać z problemu podziału (2-podziału).

3. Dolne ograniczenia

Wyznaczenie dolnego ograniczenia w metodzie podziału i ograniczeń można sprowadzić do problemu $1|prec,d_{ij},D_{ij}|C_{max}$ z odpowiednio zdefiniowaną relacją R, wynikającą z przyjętej reguły podziału drzewa rozwiązań. Do wyznaczania dolnego ograniczenia dla ww. problemu proponuje się następujące podejścia:

(a) zastosowanie odpowiednich relaksacji umożliwiających sprowadzenie rozważanego problemu do jednego z problemów A-C, E-F opisanych w rozdz.3, dla których istnieje algorytm wielomianowy,

(b) zastosowanie odpowiednio sformułowanego problemu $1|prec,r_j,q_j|C_{max}$ (lub jego dolnego ograniczenia).

Podejście (a) nie wymaga komentarza. Idea podejścia (b) polega na przyjęciu $r_k = 0$, jeśli $B_k = \emptyset$, $k \in J$. Następnie zakładając, że dla każdego zadania $i \in \tilde{B}_k$ została ustalona wartość r_i, to wartość r_k jest równa optymalnej wartości funkcji celu dla problemu $1|prec,r_j,q_j|C_{max}$ ze zbiorem zadań $\tilde{B}_k$, relacją R ograniczoną do tego zbioru, oraz $q_j = d_{jk}$, $j \in B_k$, $q_j = 0$, $j \in \tilde{B}_k \backslash B_k$. Ponieważ tak sformułowany problem jest w dalszym ciągu NP-trudny, zatem należy go zastąpić odpowiednim dolnym ograniczeniem. Najlepszy efekt można osiągnąć stosując jako dolne ograniczenie problem $1|prec,pmtn,r_j,q_j|C_{max}$ z algorytmem o złożoności $O(n^2)$ [3]. Niewiele słabszy rezultat można otrzymać stosując jako dolne ograniczenie problem $1|pmtn,r_j,q_j|C_{max}$ z algorytmem o złożoności $O(n\log n)$ [4]. Dolne ograniczenie tego typu jest rekomendowane do praktycznych zastosowań [13]. Przedstawione dwa przypadki prowadzą do otrzymania procedur wyznaczania wartości r_k, $k \in J$ o złożonościach $O(n^3)$ oraz $O(n^2\log n)$ /odpowiednio/. Jako alternatywne dolne ograniczenie proponuje się problem postaci $1|r_j,q_j=q_s|C_{max}$. Wprowadzając niewielkie zmiany w Algorytmie modyfikacyjnym opisanym w pracy [15] otrzymamy procedurę wyznaczania wartości r_k, $k \in J$ o złożoności $O(n^2)$. Wyznaczenie wartości q_k, $k \in J$ jest w pełni analogiczne.

Jako dolne ograniczenie dla problemu $1|prec,d_{ij},D_{ij}|C_{max}$ proponuje się przyjąć problem $1|prec,pmtn,r_j,q_j|C_{max}$ lub $1|pmtn,r_j,q_j|C_{max}$ z danymi r_k, q_k, $k \in J$ określonymi powyżej. Złożoność takiego dolnego ograniczenia jest odpowiednio $O(n^3)$, $O(n^2\log n)$ lub $O(n^2)$ w zależności od zastosowanego sposobu wyznaczania wartości r_k, q_k, $k \in J$. Niewątpliwie interesującym zagadnieniem jest wykazanie wzajemnych zależności pomiędzy wszystkimi opisanymi dolnymi ograniczeniami.

W przypadku ogólnego problemu $1|prec,d_{ij},D_{ij}|C_{max}$ pojawia się NP-zupełny problem istnienia dowolnej kolejności dopuszczalnej. Przy rozstrzyganiu tego problemu pomocne są odpowiednio skonstruowane dolne ograniczenia. Dla każdej pary $(k,l)\in R$ takiej, że $D_{kl} < \infty$, wyznaczmy dolne ograniczenie C_* optymalnej wartości funkcji celu dla problemu $1|prec,r_j,q_j|C_{max}$ ze zbiorem zadań $C_k = \tilde{A}_k \cap \tilde{B}_k$, z relacją R ograniczoną do tego zbioru, oraz wartościami $r_j = d_{kj}$, $j\in A_k \cap C_k$, $r_j = 0$, $j\in (\tilde{A}_k\setminus A_k) \cap C_k$, $q_j = d_{jl}$, $j\in B_l \cap C_k$, $q_j = 0$, $j\in (\tilde{B}_l\setminus B_l) \cap C_k$. Jeżeli zachodzi $C_* > D_{kl}$, to nie istnieje żadna dopuszczalna kolejność wykonywania zadań ze zbioru J. Zagadnienie doboru dolnego ograniczenia dla problemu $1|prec,r_j,q_j|C_{max}$ przedyskutowano wcześniej.

5. Górne ograniczenie

Rozważmy problem $1|prec,d_{ij}|C_{max}$ z wartościami q_j, $j\in J$ wyznaczonymi w sposób opisany w poprzednim rozdziale. Następujący algorytm aproksymacyjny generuje przybliżoną kolejność π wykonywania zadań dla tego problemu.

Algorytm aproksymacyjny.

(i) podstaw $b_j := |B_j|$, $r_j := 0$, $j\in J$, QUEE $:= \emptyset$, $k := 0$, $H := 0$,

(ii) do QUEE wpisz wszystkie zadania $j\in J$, dla których $b_j = 0$ oraz uporządkuj QUEE wg niemalejących wartości r_j,

(iii) jeżeli QUEE $= \emptyset$, to stop,

(iv) weź pierwsze zadanie z QUEE i nazwij je l; jeśli $H < r_l$, to $H := r_l$,

(v) wyznacz zbiór $K = \{j\in \text{QUEE}: r_j \le H\}$ oraz zadanie $j^*\in K$ takie, że $q_{j^*} = \max_{j\in K} q_j$; podstaw QUEE $:=$ QUEE $\setminus \{j^*\}$, $k := k+1$, $\pi(k) := j^*$, $H := H+p_{j^*}$,

(vi) dla każdego $i\in A_{j^*}$ wykonaj krok (vii),

(vii) postaw $r_i := \max(r_i, H+d_{j^*i})$, $b_i := b_i-1$. Jeżeli $b_i = 0$, to zadanie i wpisz do QUEE przeporządkowując QUEE wg niemalejących wartości r_i,

(viii) przejdź do kroku (iii).

Złożoność powyższego algorytmu jest $O(n^2)$. Dla problemu $1|prec,d_{ij},D_{ij}|C_{max}$ proponuje się wprowadzić następujące modyfikacje: w kroku (i) dodać podstaw $R_j := \infty$, $j\in J$; w kroku (v) zadanie j^* wybierać wg warunku $R_{j^*} = \min_{j\in K} R_j$; w kroku (vii) dodać podstaw $R_i := \min(R_i, H+D_{j^*i})$. Oczywiście w kolejności π otrzymanej zarówno Algorytmem aproksymacyjnym, jak i jego zmodyfikowaną wersją mogą występować naruszenia ograniczeń związanych z D_{ij}. Aktualna wartość górnego ograniczenia UB używana w algorytmie podziału i ograniczeń jest aktualizowana w oparciu o wartość H otrzymaną powyższym algorytmem.

6. Algorytm

Konstrukcja algorytmu typu podziału i ograniczeń dla rozważanego problemu może być prowadzona w ramach jednego z następujących podejść:

- podejście oparte na budowie częściowego uszeregowania zadań, np. [14],
- podejście oparte na pojęciu ścieżki krytycznej i bloku zadań, np. [8],
- podejście oparte na analizie klik w grafie dysjunktywnym, np. [5].

Ocena, które z wymienionych podejść jest najbardziej odpowiednie do konstrukcji algorytmu rozwiązywania, wymaga zaprojektowania odpowiednich 3 algorytmów i wykonania komputerowych badań porównawczych. Poniżej podane zostaną tylko istotne elementy algorytmów zrealizowanych w ramach ww. podejść.

Podejście oparte na budowie częściowego uszeregowania zadań polega na badaniu uszeregowań, w których zadania na pewnej liczbie początkowych pozycji posiadają ustaloną kolejność wykonywania, zaś pozostałe zadania nie mają określonej kolejności wykonywania. Niech $U \subseteq J$ oznacza zbiór zadań uszeregowanych, σ kolejność ich wykonywania, zaś $N = J \setminus U$ zbiór zadań nieuszeregowanych. Otrzymanie nowego uszeregowania jest możliwe poprzez wybór pewnego zadania $j \in N$ oraz utworzenie kolejności σj, tj. umieszczenie zadania j na pozycji $|U|+1$ -szej. Zadanie j jest wybierane ze zbioru zadań szeregowalnych $E = \{i \in N: B_i \subseteq U\}$. Dodatkowo, możliwe jest zawężenie zbioru E poprzez wprowadzenie prostych testów opartych na odpowiednich dolnych ograniczeniach. Tak np. jeśli zachodzi $C_{max}(\sigma)+p_j+\max_{i \in A_j} d_{ji}+p_i \geq UB$, to $j \notin E$. Podobnie jeśli $\max_{i \in B_j} d_{ij}+\sum_{i \in N} p_i \geq UB$, to $j \notin E$. Zależności tego typu można podać więcej, także w oparciu o wartości r_k, q_k, $k \in J$, których sposób wyznaczania opisano w rozdz.4. Pracochłonność badania tak sformułowanych testów w odniesieniu do korzyści wynikających z ich zastosowania w algorytmie może być oceniana tylko na drodze eksperymentalnej. Podejście oparte na budowie częściowego uszeregowania zadań przy ustalaniu kolejności "od końca" jest w pełni analogiczne. Do zalet omawianego podejścia należy prostota opisu algorytmu oraz łatwość implementacji programowej. Korzystając z faktu, że w każdym węźle drzewa rozwiązań uszeregowanie częściowe może być opisane pewną relacją [7], to wyniki otrzymane w rozdz.4-5 mogą być bezpośrednio wykorzystane w algorytmie zaprojektowanym na bazie tego podejścia.

Podejście blokowe wykorzystuje fakt, że problem wyznaczania wartości S_j $j \in J$ dla ustalonej π oraz $C_{max}(\pi)$ posiada odpowiednią interpretację grafową (patrz rozdz.2). W związku z powyższym możemy zdefiniować pojęcie ścieżki krytycznej (w grafie G_n) oraz pojęcie bloku zadań (jako podzbioru kolejnych zadań ze ścieżki krytycznej wykonywanych bez przerw pomiędzy nimi). Dalej, można wykazać podstawowe twierdzenie eliminacyjne związane z blokami zadań oraz zbudować drzewo rozwiązań w typowy dla tego podejścia sposób, np. [7]. Metody wyznaczania dolnych i górnych ograniczeń dla tego algorytmu omówiono w rozdz.4-5.

Podejście oparte na analizie klik w grafie dysjunktywnym w zasadzie można uważać za podejście pośrednio związane z tworzeniem uszeregowań częściowych poprzez ustalanie zadań na początkowych albo końcowych wolnych pozycjach równocześnie. Podejście to wymaga określenia zbioru U zadań uszeregowanych na początkowych pozycjach, zbioru W zadań uszeregowanych na końcowych pozycjach, zbioru $N = J\setminus(U \cup W)$ zadań nieuszeregowanych, zbioru E zadań szeregowalnych na pierwszej wolnej pozycji oraz zbioru S zadań szeregowalnych na ostatniej wolnej pozycji. Jeżeli $|E| = 1$ lub $|S| = 1$, to kolejne zadanie do uszeregowania zostało jednoznacznie określone i może być uszeregowane. W przeciwnym przypadku dla zadań ze zbioru N zostają wprowadzone dodatkowe ograniczenia kolejnościowe, wynikające z ustalenia pewnego łuku dysjunktywnego. Konsekwencją tego jest zawężenie zbioru E lub S, odpowiednio. Podobnie jak pierwszym omawianym podejściu, możliwe jest także wprowadzenie dodatkowych testów zawężających zbiory E i S.

7. Zakończenie

Przykładem możliwości zastosowania przedstawionego problemu jest algorytm aproksymacyjny dla gniazdowego zagadnienia kolejnościowego opisany w pracy [1]. Podstawowa idea tego algorytmu polega na wyznaczaniu uszeregowania na pojedynczej maszynie, przy założeniu, że uszeregowania dla innych maszyn zostały już wcześniej ustalone. Autorzy pracy [1] zastosowali w tym celu odpowiednio zdefiniowany problem $1|r_j,q_j|C_{max}$. Bezpośrednim tego efektem jest generowanie rozwiązań niedopuszczalnych oraz możliwość "zapętlenia się" procedury aproksymacyjnej. Wszystkich tych kłopotów można uniknąć stosując problem $1|prec,d_{ij}|C_{max}$ zamiast $1|r_j,q_j|C_{max}$.

Zagadnienie $1|prec,d_{ij}|C_{max}$ może być również stosowane jako alternatywna metoda wyznaczania dolnego ograniczenia dla problemu gniazdowego, pośrednia (w sensie pracochłonności) pomiędzy proponowanym w pracy [6] dolnym ograniczeniem 1-maszynowym postaci $1|r_j,q_j|C_{max}$ oraz 2-maszynowym postaci $J2|prec,r_j,q_j|C_{max}$.

Przewiduje się także zastosowanie zagadnienia $1|prec,d_{ij},D_{ij}|C_{max}$ do wyznaczania dolnego ograniczenia oraz jako algorytm pomocniczy w algorytmach aproksymacyjnych, w realizowanym systemie wspomagania decyzji dla szeregowania zadań [2].

LITERATURA

[1] Adams J., Balas E.: The shifting bottleneck procedure for job shop scheduling, Management Science Report No MSRR-525, Graduate School of Industrial and Administration, Carnegie Mellon University, Pitsburgh, Pennsylwania, 1986.
[2] Anthonisse J.M., van Hee K.M., Lenstra J.K.: Resource Constrained Project Scheduling: an International Exercise in DSS Developement, Draft, IIASA, 1987.
[3] Baker K.R., Lawler E.L., Lenstra J.K., Rinnooy kan A.H.G.: Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints, Report 8028/0, Erasmus University Rotterdam, The Netherlands, 1981.
[4] Carlier J.: The one machine sequencing problem, European J. Oper. Res. 11, 1982, 42-47.
[5] Carlier J.:Problemes d'ordonnancement des contraintes de ressources, Algorithmes et complexité, Méthodologie et architecture des systèmes informatiques, N-40, Universite P. et M. Curie, Paris 1984.
[6] Grabowski J., Nowicki E., Smutnicki C.: Nowe metody wyznaczania dolnych ograniczeń dla zagadnienia kolejnościowego gniazdowego, Zeszyty Naukowe AGH 1064, ser. Automatyka z.39, 1985, 171-175.
[7] Grabowski J., Smutnicki C.: Zagadnienia kolejnościowe taśmowe, Cz.I, Przegląd Statystyczny XXX, z.3/4, 1983, 275-288, Cz.II, Przegląd Statystyczny XXXI, z.1/2, 1984, 31-45.
[8] Grabowski J., Nowicki E., Zdrzałka S.: A block approach for single machine scheduling with release dates and due dates, European J. Oper. Res. 26, 1986, 278-285.
[9] Graham R.L. et al.: Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 5, 1979, 287-326.
[10] Gupta S.K.,Kyparisis J.: Single machine scheduling research, OMEGA Int. J. of Mgmt Sci. 15(3), 1987, 207-227.
[11] Lawler E.L.: Optimal sequencing of a single machine subject to precedence constraints, Mgmt Sci 19, 1973, 544-546.
[12] Lenstra J.K., Rinnooy Kan A.H.G., Brucker P.: Complexity of machine scheduling problems, Ann. Discrete Math. 1, 1977, 343-362.
[13] Nowicki E., Smutnicki C.: On Bounds on the Minimum Maximum Lateness on One Machine Subject to Release Dates, OPSEARCH 24, 1987, 106-110.
[14] Rinnooy Kan A.H.G.: Machine Scheduling Problems: Classification, Complexity and Computations, Nijhoff The Hague 1976.
[15] Smutnicki C.: Dolne i górne ograniczenia dla problemu $F|r_j,prec|f_{max}$, Zeszyty Naukowe AGH 1176, ser. Automatyka z.42, 1987, 169-180.

Recenzent:Doc.dr hab.inż.J.Klamka

Wpłynęło do Redakcji do 1988-04-30.

ЗАДАЧА ЧЕРЕДОВАНИЯ С ОГРАНИЧЕНИЯМИ ВРЕМЕНИ ОЖИДАНИЯ

Р е з ю м е

В статье представляется одномашинная проблема чередования с критерием минимизации максимального срока окончания выполнения всех задач. В этой проблеме принимается также , что существуют ограничения очередности выполнения задач а скаждым ограничением этого типа связано нижнее и верхнее ограничение времени ожидания между выполняемыми задачами. Для этой проблемы даётся анализ вычислительной сложности а также предлагается алгоритм решения.

A SCHEDULING PROBLEM WITH WAITING TIME CONSTRAINTS

S u m m a r y

The paper deals with one-machine scheduling problem with minimum maximum completion time criterion. It is also assumed that there exist preceding constraints, and for each pair of jobs linked by these constraints the lower and upper bounds for waiting time between processing the jobs are given. For this problem the computational complexity analysis is carried out and solution algorithm is proposed.