ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992

Seria: AUTOMATYKA z. 109 Nr kol. 1175

Czesław Smutnicki
Politechnika Wrocławska

OGÓLNY GNIAZDOWY PROBLEM SZEREGOWANIA PRZY ŻĄDANYCH TERMINACH ZAKOŃCZENIA OPERACJI

GENERAL JOB-SHOP PROBLEM WITH DUE DATES

ОБЩАЯ ГНЕЗДОВАЯ ЗАДАЧА ЧЕРЕДОВАНИЯ С ТРЕБУЕМЫМИ МОМЕНТАМИ ОКОНЬЧЕНИЯ ОПЕРАЦИИ

Streszczenie: W pracy rozważany jest ogólny gniazdowy problem szeregowania z kryterium minimalizacji maksymalnej kary za nieterminową (zbyt wczesną lub zbyt późną) realizację czynności. Omówiono metody wyznaczania rozwiązań dokładnych i przybliżonych.

Summary: The paper deals with the general job-shop problem with criterion of minimization the maximum earliness and tardiness penalties. Exact and approximation algorithms are proposed.

Резюме: В статье представляется общая гнездовая задача чередования с критерием минимизации максимального штрафа за несрочное (ранное или опоздавшее) выполнение операции. Представлены методы нахождения точных и приближенных решений.

1. Wstęp

W literaturze przedstawiono dotychczas wiele algorytmów zarówno aproksymacyjnych, jak i dokładnych dla problemów szeregowania, przy założeniu różnych postaci funkcji celu oraz różnych ograniczeń dodatkowych, [5], [10], [14]. Nowy kierunek badań otwierają problemy z nieregularnymi funkcjami celu [2], [11], [12], [17], które umożliwiają m.in. modelowanie strategii szeregowania na czas (just in time) oraz modelowanie dialogowych systemów poszukiwania rozwiązania kompromisowego. Problemy te charakteryzują się brakiem spełnienia zasady maksymalnego wykorzystania maszyn (tj. harmonogram jest dosunięty maksymalnie w lewo na osi czasu), przez co rozwiązanie optymalne nie musi być aktywne (semiaktywne). W konsekwencji klasyczne podejścia stosowane do rozwiązywania problemów z regularnymi funkcjami celu są mało przydatne oraz zachodzi potrzeba wypracowania nowych podejść. W niniejszej pracy przedstawiono próbę zaadaptowania podejścia opartego na koncepcji ścieżki krytycznej do rozwiązywania jednego z takich problemów.

2. Sformułowanie problemu

Dany jest zbiór operacji $N=\{1,2,\dots,n\}$ przeznaczonych do wykonywania na maszynach ze zbioru $M=\{1,2,\dots,m\}$. Operacja j-ta odpowiada czynności realizowanej na maszynie μ_j w czasie $p_j>0$ z kosztem równym $h_j(S_j)$, gdzie $h_j(t)$ jest funkcją posiadającą jedno (niekoniecznie właściwe) minimum, zaś S_j jest terminem rozpoczęcia wykonywania tej operacji, $j\in N$. Dany jest także częściowy porządek $R^O\subseteq N\times N$ wykonywania operacji, implikujący ograniczenia postaci $S_i+p_i\le S_j$, $(i,j)\in R^O$. Zakłada się, że wykonywanie operacji na maszynie nie może być przerywane oraz że maszyna może wykonywać co najwyżej jedną operację w dowolnym momencie czasu. Poszukuje się terminów rozpoczęcia wykonywania operacji S_j^*, które minimalizują maksymalny koszt związany z zakończeniem wykonywania poszczególnych operacji, tzn. $\max_{j\in N} h_j(S_j)$. Problem ten będzie w dalszym ciągu nazywany problemem (P).

Znane w literaturze problemy szeregowania sklasyfikowane w [9] jako $\alpha|\beta|\gamma$, gdzie $\alpha\in\{1, F, J\}$, $\beta\in\{o, r_j, C_j\le d_j\}$ oraz $\gamma\in\{C_{max}, L_{max}, T_{max}, f_{max}\}$ mogą być traktowane jako szczególne przypadki problemu (P), otrzymane poprzez odpowiednie zdefiniowanie funkcji kosztu poszczególnych operacji. Tak np.:

(a) termin gotowości operacji r_j (ready time) może być wyrażony przez wprowadzenie do funkcji $h_j(t)$ składnika $M\cdot\max\{0,r_j-t\}$, gdzie M dostatecznie duża liczba dodatnia,

(b) żądanie zakończenia operacji przed terminami krytycznymi d_j (dead line) może być wyrażone przez wprowadzenie składnika $M\cdot\max\{0,t+p_j-d_j\}$,

(c) funkcje kryterialne $C_{max}, L_{max}, T_{max}, f_{max}$ są szczególnymi przypadkami funkcji $h_j(t)$, tzn. $h_j(t)=t+p_j$, $h_j(t)=t+p_j-d_j$, $h_j(t)=\max\{0,t+p_j-d_j\}$, $h_j(t)=f_j(t)$, odpowiednio.

Zauważmy, że ograniczenia twarde (hard) dyskutowane w p. (a)-(b) zostały zastąpione o - graniczeniami miękkimi (soft) poprzez wprowadzenie odpowiedniej funkcji kary. W praktyce harmonogramowania i szeregowania zadań ograniczenia są zwykle formułowane jako miękkie, tzn. "warunkowo przekraczalne" lub "z karanym przekroczeniem". W sytuacji istnienia zbyt wielu ograniczeń nieprzekraczalnych (twardych) wyznaczenie jakiegokolwiek rozwiązania dopuszczalnego jest zwykle problemem NP-trudnym; w takim przypadku zmiana charakteru wybranych ograniczeń (z twardych na miękkie) pozwala wyznaczyć rozwiązanie satysfakcjonujące użytkownika.

Z kolei interpretacja ogólnych funkcji kosztu $h_j(t)$ jest związana z istnieniem dla każdej operacji momentu czasowego (ogólnie przedziału czasowego), w którym pożądane jest jej rozpoczęcie (lub symetrycznie jej zakończenie). W przypadku idealnym, gdy operacje są rozpoczynane w żądanych momentach czasowych (just in time), ponoszone koszty są zerowe. Nieterminowe, zbyt wczesne lub zbyt późne, rozpoczęcie operacji pociąga za sobą koszty zależne od wielkości nieterminowości. Odpowiednio do przedstawionej interpretacji funkcję $h_j(t)$ można przedstawić w innej postaci, wygodniejszej także do dalszych rozważań. Niech $h_j^0=\min_{-\infty<t<\infty} h_j(t)$. Oznaczmy przez a_j oraz b_j argumenty funkcji $h_j(t)$ takie, że $a_j=\min\{x:\ h_j(x)=h_j^0\}$, $b_j=\max\{x:\ h_j(x)=h_j^0\}$. Oczywiście zachodzi $a_j\le b_j$. Przyśpieszeniem operacji j, rozpoczynanej w chwili t, względem terminu a_j nazywamy wielkość $E_j(t)=\max\{0,a_j-t\}$, zaś spóźnieniem operacji j względem terminu b_j nazywamy wielkość $T_j(t)=\max\{0,t-b_j\}$. Funkcję $g_j(t)=h_j(a_j-t)-h_j^0$ określoną dla $t\ge0$ nazywamy funkcją kosztu przyśpieszenia, zaś funkcję $f_j(t)=h_j(t+b_j)-h_j^0$ określoną dla $t\ge0$ nazywamy funkcją kosztu spóźnienia. Zauważmy, że funkcje $g_j(t)$ oraz $f_j(t)$ są niemalejące względem swoich argumentów oraz $g_j(0)=0$, $f_j(0)=0$. Dla poprawnego zdefiniowania problemu wymaga się, aby tak otrzymane funkcje były ciągłe lub lewostronnie ciągłe. Ostatecznie funkcję $h_j(t)$ można zapisać jako

$$h_j(t) = h_j^0+\max\{g_j(E_j(t)),f_j(T_j(t))\} \tag{1}$$

Wszystkie rozpatrywane w literaturze funkcje kosztu wykonywania operacji zależne od jej terminu rozpoczęcia, zakończenia, przyśpieszenia lub spóźnienia są szczególnymi przypadkami funkcji (1).

3. Rozwiązanie problemu

Idea rozwiązania opiera się na dwóch przesłankach. Najpierw wprowadza się pojęcie kolejności wykonywania operacji realizowanych na tej samej maszynie w celu uwzględnienia ograniczeń na skończoną (jednostkową) przepustowość maszyn. Następnie problem (P) jest dekomponowany na dwa podproblemy: poszukiwanie optymalnej kolejności wykonywania operacji (poziom górny) oraz poszukiwanie terminów wykonywania operacji dla danej kolejności ich wykonywania (poziom dolny). Do rozwiązania stosuje się metodę dwupoziomową z algorytmem typu podziału i ograniczeń na poziomie górnym oraz specjalizowanym algorytmem wielomianowym na poziomie dolnym. Szczegółowy opis algorytmów przedstawiono w rozdziałach 3.1 i 3.2, odpowiednio.

W dalszej analizie będziemy posługiwać się modelem grafowym problemu (P). Z każdą operacją skojarzymy wierzchołek grafu z danymi p_j, μ_j, $g_j(t)$, $f_j(t)$ oraz zmienną S_j; zmienne $E_j(S_j)$, $T_j(S_j)$ są związane z S_j. Z każdą parą operacji (i,j), dla których określono kolejność wykonywania i j, skojarzymy łuk (i,j) grafu. Oznaczmy przez $N_k=\{j\in N:\mu_j=k\}$ zbiór operacji wykonywanych na maszynie k-tej oraz $n_k=|N_k|$, $k\in M$. Kolejność wykonywania operacji na maszynie k jest określona permutacją $\pi_k=(\pi_k(1),\dots,\pi_k(n_k))$ elementów ze zbioru N_k. Oznaczmy przez Π_k zbiór wszystkich permutacji na N_k. Kolejność wykonywania operacji na wszystkich maszynach jest określona przez $\pi=(\pi_1,\dots,\pi_m)$, gdzie $\pi\in\Pi=\Pi_1\times\Pi_2\times\dots\times\Pi_m$. Rozważmy graf $G(\pi)=(N,R(\pi))$, gdzie

$$R(\pi) = R^0 \cup \bigcup_{k=1}^{m}\bigcup_{i=1}^{n_k-1}\{(\pi_k(i),\pi_k(i+1))\}. \tag{2}$$

Zauważmy, że $\pi\in\Pi$ reprezentuje dopuszczalną kolejność wykonywania operacji na wszystkich maszynach wtedy i tylko wtedy, gdy graf $G(\pi)$ jest acykliczny. Zdefiniujmy zatem zbiór kolejności dopuszczalnych $\Pi=\{\pi\in\Pi:\ G(\pi)\text{-acykliczny}\}$. Zatem zgodnie z przedstawioną ideą problem (P) może być zapisany jako: wyznacz $\pi^*\in\Pi$ takie, że

$$H(\pi^*) = \min_{\pi\in\Pi} H(\pi) \tag{3}$$

gdzie

$$H(\pi) = \min_{S\in S(\pi)} (h_j^0+\max\{g_j(E_j(S_j)),f_j(T_j(S_j))\}) \tag{4}$$

przy czym

$$S(\pi) = \{S=(S_1,\dots,S_n):\ S_i+p_i\le S_j,\ (i,j)\in R(\pi)\}. \tag{5}$$

Zauważmy, że poszukiwane w problemie (P) optymalne terminy rozpoczęcia operacji S_j^*, $j\in N$, są wyznaczone przez rozwiązanie problemu (4)-(5) dla π^*.

3.1. Wyznaczanie terminów rozpoczęcia operacji

Problem (4)-(5) jest w ogólnym przypadku problemem optymalizacji z nieliniową funkcją celu i liniowymi ograniczeniami. W dalszym ciągu przedstawimy specjalizowany wielomianowy algorytm do rozwiązania tego problemu. Opis metody podamy przy założeniu, że $h_j(t)$ są ciągłe oraz $h_j^0=0$, $j\in N$ (możliwe jest ominięcie tych wymagań).

Niech π będzie pewną kolejnością wykonywania operacji. Analizę zawartą w tym rozdziale będziemy prowadzić w oparciu o graf $G(\pi)$. Dla wygody zapisu zbiór krawędzi $R(\pi)$ grafu będziemy przedstawiać alternatywnie za pomocą zbiorów bezpośrednich poprzedników i następników wierzchołków, odpowiednio

$$A_j(\pi) = \{i\in N: (j,i)\in R(\pi)\}, \quad B_j(\pi) = \{i\in N: (i,j)\in R(\pi)\}, \quad j\in N. \tag{6}$$

Ciąg wierzchołków grafu $v=(v_1,v_2,\dots,v_{r_v})$ taki, że $v_k\in N$, $k=1,\dots,r_v$, $(v_k,v_{k+1})\in R(\pi)$, $k=1,\dots,r_v-1$, będziemy nazywać drogą, zaś

$$L(v) = \sum_{k=1}^{r_v-1} p_{v_k} \tag{7}$$

długością tej drogi. Oznaczmy przez

$$D_{ij}(\pi) = \{v:\ v_1=i \text{ oraz } v_{r_v}=j\} \tag{8}$$

zbiór dróg w $G(\pi)$ o początkach i oraz końcach j, zaś przez

$$V_{ij}(\pi) = \bigcup_{v\in D_{ij}(\pi)} \bigcup_{k=1}^{r_v} \{v_k\} \tag{9}$$

zbiór wierzchołków należących do wszystkich dróg z $D_{ij}(\pi)$. Niech

$$Q(\pi) = \{(i,j):\ D_{ij}(\pi)\neq\emptyset\} \tag{10}$$

będzie zbiorem par wierzchołków, między którymi istnieje droga. Drogę $u_{ij}\in D_{ij}(\pi)\neq\emptyset$, $u_{ij}=(u_{ij1},u_{ij2},\dots,u_{ijr_{u_{ij}}})$ taką, że

$$L(u_{ij}) = \max_{v\in D_{ij}(\pi)} L(v) \tag{11}$$

będziemy nazywać najdłuższą drogą z i do j. Niech

$$D(\pi) = \bigcup_{i=1}^{n} \bigcup_{j=1}^{n} D_{ij}(\pi) \tag{12}$$

będzie zbiorem wszystkich dróg grafu $G(\pi)$. Formalnie drogi u_{ij} zależą od π, jednakże dla uproszczenia notacji nie będziemy tego uwzględniać w zapisie.

Zdefiniujmy dla $(i,j)\in Q(\pi)$ pomocniczy (zrelaksowany) problem

$$H_{ij}(\pi) = \min_{S\in S(\pi)} \max\{g_i(E_i(S_i)), f_j(T_j(S_j))\}. \tag{13}$$

Problem ten polega na wyznaczeniu terminów rozpoczęcia operacji przy założeniu, że tylko dwie funkcje kosztu $g_i(t)$, $f_j(t)$ są niezerowe. Pokażemy dalej, jak wyznaczyć rozwiązanie problemu (13). Dla $(i,j)\in Q(\pi)$, korzystając z definicji $E_j(t)$, $T_j(t)$ oraz $S(\pi)$, mamy

$$\begin{aligned} E_i(S_i)+T_j(S_j) &\geq \max\{0,a_i-S_i\}+\max\{0,S_j-b_j\} \geq \max\{0,a_i-b_j+S_j-S_i\} \\ &\geq \max\{0,a_i-b_j+L(u_{ij})\} = \Delta_{ij}(\pi). \end{aligned} \tag{14}$$

Ponieważ z definicji zachodzi $T_j(S_j)\geq 0$, zatem z (14) mamy

$$T_j(S_j) \geq \max\{0, \Delta_{ij}(\pi) - E_i(S_i)\}. \tag{15}$$

Funkcja $f_j(t)$ jest niemalejąca, zatem z (13) oraz (15) otrzymujemy

$$H_{ij}(\pi) \geq \min_{S \in S(\pi)} \max\{g_i(E_i(S_i)), f_j(\max\{0, \Delta_{ij}(\pi) - E_i(S_i)\})\}$$

$$= \min_{S_i} \max\{g_i(E_i(S_i)), f_j(\max\{0, \Delta_{ij}(\pi) - E_i(S_i)\})\}. \tag{16}$$

Zauważmy, że minimalizację po nieograniczonej zmiennej S_i możemy zastąpić minimalizacją po zmiennej $x \in X$, gdzie X jest zbiorem wartości przyśpieszeń dla wszystkich wartości S_i, czyli $X = \{x\colon x \geq 0\}$. Wprowadźmy dalej funkcję

$$F_{ij}(u) = \min_{x \geq 0} \max\{g_i(x)), f_j(\max\{0, u - x\})\} \tag{17}$$

określoną dla $u \geq 0$. Jak łatwo sprawdzić, $F_{ij}(u)$ jest funkcją niemalejącą oraz $F_{ij}(0) = 0$. Z definicji $g_i(t)$ dla $x > u$ zachodzi

$$\max\{g_i(x)), f_j(0)\} \geq \max\{g_i(u)), f_j(0)\}. \tag{18}$$

W konsekwencji z (17) i (18) otrzymujemy prostszą postać na funkcję $F_{ij}(u)$

$$F_{ij}(u) = \min\{\min_{0 \leq x \leq u} \max\{g_i(x)), f_j(u-x)\}, \min_{x > u} \max\{g_i(x)), f_j(0)\}\}$$

$$= \min_{0 \leq x \leq u} \max\{g_i(x)), f_j(u-x)\}. \tag{19}$$

Ostatecznie z (16), (17) oraz (19) mamy

$$H_{ij}(\pi) \geq F_{ij}(\Delta_{ij}(\pi)) \tag{20}$$

Problem (19) jest problemem minimalizacji funkcji jednej zmiennej x w przedziale $[0,u]$. W przypadku gdy funkcje $g_j(t), f_j(t)$ są ciągłe oraz $g_j(0) = 0 = f_j(0)$, problem ten może być sprowadzony do rozwiązania równania $g_i(x) = f_j(u-x)$. Dla konkretnych postaci funkcji $g_j(t), f_j(t)$, (np. liniowe) możliwe jest otrzymanie rozwiązania w postaci analitycznej. Niech x' będzie rozwiązaniem problemu (19) dla $u = \Delta_{ij}(\pi)$. Pokażemy, jak skonstruować rozwiązanie $S' \in S(\pi)$ problemu (13) takie, że $H_{ij}(\pi) = F_{ij}(\Delta_{ij}(\pi))$. W pierwszej kolejności wyznaczamy terminy rozpoczęcia operacji należących do $V_{ij}(\pi)$. Przyjmujemy

$$S'_i = a_i - x', \qquad S'_k = \max_{j \in B_k(\pi) \cap V_{ij}(\pi)} (S'_k + p_k), \quad k \in V_{ij}(\pi) \setminus \{i\}. \tag{21), (22}$$

Wzór (22) określa, że operacje $k \in V_{ij}(\pi) \setminus \{i\}$ rozpoczynają się najwcześniej jak mogą, biorąc pod uwagę tylko operacje ze zbioru $V_{ij}(\pi)$. Zatem zachodzi $S'_k = S'_i + L(u_{ik})$ $k \in V_{ij}(\pi) \setminus \{i\}$ oraz w szczególności $S'_j = S'_i + L(u_{ij})$. Terminy rozpoczęcia pozostałych operacji, tzn. $k \in N \setminus V_{ij}(\pi)$, można wybrać dowolnie (dopuszczalnie), bowiem nie mają one wpływu na wartość funkcji celu (13). Jak łatwo sprawdzić, minimalna wartość funkcji celu (13) jest równa

$$H_{ij}(\pi) = \max\{g_i(E_i(S'_i)), f_j(T_j(S'_i + L(u_{ij}))\} = \max\{g_i(x'), f_j(\Delta_{ij}(\pi) - x')\}$$

$$= F_{ij}(\Delta_{ij}(\pi)) \tag{23}$$

Pokażemy dalej, jak w oparciu o rozwiązania problemów (13) dla $(i,j)\in Q(\pi)$ skonstruować rozwiązanie $S''\in S(\pi)$ problemu (4)-(5) takie, że

$$H(\pi) = H^*, \text{ gdzie } H^* = \max_{(i,j)\in Q(\pi)} H_{ij}(\pi). \qquad (24), (25)$$

Oznaczmy przez

$$x''_j=\max\{t:\ g_j(t)=H^*\},\ j\in N. \qquad (26)$$

Rozwiązanie S'' jest określone następującym wzorem rekurencyjnym

$$S''_j = \max\{a_j-x''_j, \max_{i\in B_j(\pi)}(S''_i+p_i)\},\ j\in N, \qquad (27)$$

gdzie $\max_{j\in B_i}(S''_j+p_j)=-\infty$ jeśli $B_i=\emptyset$. W dalszym ciągu pokażemy krótkie uzasadnienie tego faktu. Ponieważ każdy problem postaci (13) jest relaksacją problemu (P), zatem oczywiście $H(\pi)\ge H^*$. Pokażemy dalej, że zachodzi

$$\max_{j\in N} \max\{g_j(E_j(S''_j)), f_j(T_j(S''_j))\} \le H^*. \qquad (28)$$

Z określenia (27) i (26) mamy

$$\max_{j\in N} g_j(E_j(S''_j)) \le \max_{j\in N} g_j(x''_j) = H^*. \qquad (29)$$

Dalszą część dowodu poprowadzimy przez zaprzeczenie. Przypuśćmy, że istnieje operacja $l\in N$ taka, że $f_l(T_l(S''_l))>H^*$. Wyznaczmy wówczas operację o najmniejszym indeksie $k\in N\setminus\{l\}$ taką, że $S''_k=a_k-x''_k > \max_{j\in B_k(\pi)}(S''_j+p_j)$ oraz $S''_l=S''_k+L(u_{kl})$; operacja taka zawsze istnieje. Zgodnie z definicją x''_k zachodzi $g_k(x)>H^*$ dla $x>x''_k$, $f_l(T_l(S''_l)) = f_l(S''_k+L(u_{kl})) = f_l(\Delta_{kl}(\pi)-x''_k) > H^*$ oraz

$$\begin{aligned} H_{kl}(\pi) &= \min\{\min_{0\le x\le x''_{kl}} \max\{g_k(x), f_l(\Delta_{kl}(\pi)-x)\},\\ &\qquad \min_{x''_{kl}<x\le\Delta_{kl}(\pi)} \max\{g_k(x), f_l(\Delta_{kl}(\pi)-x)\}\}\\ &\ge \min\{\min_{0\le x\le x''} f_l(\Delta_{kl}(\pi)-x), \min_{x''<x\le\Delta_{kl}(\pi)} g_k(x)\}\\ &= \min\{f_l(\Delta_{kl}(\pi)-x''_{kl}), \min_{x''_{kl}<x\le\Delta_{kl}(\pi)} g_k(x)\}\\ &> H^* \ge H_{kl}(\pi) \end{aligned} \qquad (30)$$

co prowadzi do sprzeczności, że H^* było wartością największą. Zatem mamy $f_j(T_j(S''_j))\le H^*$, $j\in N$, co w połączeniu z (29) daje (28).

Ostatecznie pełny algorytm wyznaczania terminów rozpoczęcia wykonywania operacji dla danej kolejności ich wykonywania ma następującą postać.

Algorytm (dolny poziom)

(1) wyznacz zbiór $Q(\pi)$ oraz długości najdłuższych dróg między każdą parą wezłów $L(u_{ij})$, $(i,j)\in Q(\pi)$,

(2) wyznacz $F_{ij}(\Delta_{ij}(\pi))$ zgodnie z (19) dla $(i,j)\in Q(\pi)$ oraz wyznacz H^* zgodnie z (25),

(3) wyznacz wartości maksymalnych przyśpieszeń x''_j wg (26),

(4) wyznacz terminy rozpoczęcia wykonywania operacji S''_j ze wzorów (27).

Krok (1) Algorytmu można zrealizować przez n-krotne powtórzenie algorytmu Bellmana. Niech σ będzie uporządkowaniem wierzchołków grafu spełniającym warunek $B_{\sigma(j)}(\pi)\subseteq\{\sigma(1),\ldots,\sigma(j-1)\}$, $j\in N$. Uporządkowanie takie można wyznaczyć w $O(n+|R(\pi)|)$ krokach. Długości $L(u_{ij})$ można wtedy wyznaczyć, realizując dla wszystkich $1\le i<j\le n$ następujący wzór rekurencyjny

$$L(u_{\sigma(i)\sigma(j)}) = \max_{k\in B_{\sigma(j)}(\pi)\cap\{\sigma(i),\ldots,\sigma(j-1)\}} \{L(u_{\sigma(i)k})+p_k\} \tag{31}$$

oraz

$$L(u_{\sigma(i)\sigma(j)}) = \infty \quad \text{jeśli } B_{\sigma(j)}\cap\{\sigma(i),\ldots,\sigma(j-1)\}=\emptyset. \tag{32}$$

Realizacja wzorów (31), (32) dla ustalonej pary (i,j) wymaga $O(n+|R(\pi)|)$ kroków. Podobnie wzór (27) przyjmuje postać

$$S''_{\sigma(i)} = \max\{a_{\sigma(i)}-x''_{\sigma(i)}, \max_{j\in B_{\sigma(i)}(\pi)}(S''_j+p_j)\},\quad i=1,\ldots,n. \tag{33}$$

Ostatecznie Algorytm dolnego poziomu ma złożoność $O(n(n+|R(\pi)|)s)$, gdzie s jest liczbą iteracji potrzebnych do wyznaczenia $\dot{F}_{ij}(u)$ dla danego u (s jest stałą w przypadku analitycznego wyznaczenia rozwiązania dla (19)).

3.2. Wyznaczanie optymalnej kolejności wykonywania operacji

Z analizy przedstawionej w rozdziale 3.1 wynika, że wartość $H(\pi)$ możemy zapisać jako

$$H(\pi) = \max_{v\in D(\pi)} F_{v_1 v_{r_v}}(\max\{0, a_{v_1}-b_{v_{r_v}}+L(v)\}). \tag{34}$$

Drogę $u\in D(\pi)$, która maksymalizuje prawą stronę wzoru (34), będziemy nazywać drogą krytyczną. Dla tak określonej drogi mają zastosowanie wszystkie rezultaty zawarte w pracy [8], w szczególności pojęcie bloku operacji oraz eliminacyjne własności bloku. W związku z tym do wyznaczania optymalnej kolejności wykonywania operacji proponuje się zastosować algorytm podziału i ograniczeń wykorzystujący specyfikę podejścia blokowego [8]. Ponieważ szczegóły dotyczące sposobu generacji drzewa rozwiązań, metody podziału, strategii przeglądania oraz dowodów zbieżności metody zostały już opisane szczegółowo w w.w. pracy, zatem w dalszym ciągu ograniczamy się tylko do elementów istotnie nowych, tj. do sposobu wyznaczania wartości funkcji celu (dla danej kolejności wykonywania operacji) oraz sposobu wyznaczania dolnych i górnych ograniczeń dla metody podziału i ograniczeń.

3.3. Przypadki szczególne

Szereg przypadków szczególnych było analizowanych w pracach [1], [6], [13], [18]. Ograniczały się one głównie do problemów jednomaszynowych lub przepływowych z jednakowymi funkcjami kosztu.

Problem (P1). Rozpatrzmy problem (P) przy założeniu, że wszystkie funnkcje kosztu są jednakowe, tzn. $g_j(t)=g(t)$, $f_j(t)=f(t)$, $j\in N$. Zatem z (34) otrzymujemy

$$H(\pi) = \max_{v \in D(\pi)} F(\max\{0, a_{v_1} - b_{v_r} + L(v)\})$$

$$= F(\max_{v \in D(\pi)} \max\{0, a_{v_1} - b_{v_r} + L(v)\}$$

$$= F(\max\{0, \max_{v \in D(\pi)} (a_{v_1} - b_{v_{r_v}} + L(v))\}), \tag{35}$$

gdzie

$$F(u) = \min_{0 \le x \le u} \max\{g(x)), f(u-x)\}. \tag{36}$$

Wartość $\max_{v \in D(\pi)} (a_{v_1} - b_{v_{r_v}} + L(v))$ jest maksymalnym spóźnieniem w problemie, w którym określono nieprzekraczalne terminy gotowości a_j, zaś spóźnienie jest mierzone względem terminów b_j, $j \in N$. Problem tego typu jest klasyfikowany jako $\alpha|r_j,\beta|L_{max}$ [9]. Problemy z funkcją celu L_{max} są stosunkowo dobrze zbadane w literaturze, [10], [14], zaś do ich rozwiązywania można stosować techniki podobne jak do rozwiązywania problemów z minimalizacją terminu zakończenia wszystkich operacji.

Problem (P2). Rozpatrzmy problem (P) przy założeniu, że funkcje kosztu przyśpieszeń są jednakowe, tzn. $g_j(t)=g(t)$, $j \in N$. Zatem z (34) otrzymujemy

$$H(\pi) = \max_{v \in D(\pi)} F_{v_{r_v}} (\max\{0, a_{v_1} - b_{v_{r_v}} + L(v)\})$$

$$= \max_{j \in N} \max_{i:(i,j) \in Q(\pi)} F_j(\max\{0, a_i - b_j + L(v)\})$$

$$= \max_{j \in N} F_j(\max_{i:(i,j) \in Q(\pi)} \max_{v \in D_{ij}(\pi)} \max\{0, a_i - b_j + L(v)\}$$

$$= \max_{j \in N} F_j(\max\{0, \max_{i:(i,j) \in Q(\pi)} \max_{v \in D_{ij}(\pi)} (a_i + L(v)) - b_j\}), \tag{37}$$

gdzie

$$F_j(u) = \min_{0 \le x \le u} \max\{g(x)), f_j(u-x)\}. \tag{38}$$

Wartość $\max_{i:(i,j) \in Q(\pi)} \max_{v \in D_{ij}(\pi)} (a_i + L(v))$ jest terminem rozpoczęcia operacji j w problemie, w którym określono nieprzekraczalne terminy gotowości a_j, $j \in N$. Wyrażenie (37) może być interpretowane jako maksymalny koszt zależny od terminu zakończenia (rozpoczęcia) operacji. Problem tego typu jest klasyfikowany jako $\alpha|r_j,\beta|f_{max}$ [9]. Problemy z funkcją celu f_{max} były również przedmiotem rozważań [7].

3.4. Dolne ograniczenia

Dolne ograniczenie dla problemu (P) otrzymano stosując relaksacje prowadzące do otrzymania przypadków szczególnych (P1) i (P2) opisanych w rozdziale 3.3. I tak relaksacja funkcji kosztu ma postać: wszystkie funkcje kosztu operacji są jednakowe i równe odpowiednio $g(t)=\min_{j \in N} g_j(t)$, $f(t)=\min_{j \in N} f_j(t)$, $t \ge 0$. W rezultacie wszystkie metody budowania dolnych ograniczeń, opisane dla w/w. przypadków szczególnych, stają się przydatne do wyznaczania dolnych ograniczeń dla problemu (P). W szczególności dolne ograniczenia 1-maszynowe [15], [16] mogą być z powodzeniem użyte do wyznaczania dolnego ograniczenia dla problemu $\alpha|r_j,\beta|L_{max}$.

Pewną poprawę wartości dolnego ograniczenia otrzymanego powyżej można otrzymać poprzez zastosowanie relaksacji liczby zadań przed relaksacją funkcji kosztu. Ogólnie, idea ta zakłada wyznaczenie wartości dolnego ograniczenia dla pewnego podzbioru zadań $K \subset N$ (lub pewnego ciągu podzbiorów $K_1 \subset K_2 \subset \ldots \subset K_s \subseteq N$). Przyjmuje się, że ciąg podzbiorów jest konstruowany następująco: $s=n$, $K_n=N$, $K_{i-1}=K_i \setminus \{k\}$, gdzie $k \in K_i$ jest zadaniem takim, że $g_k(x'')=g(x'')=\min_{j \in K_i} g_j(x'')$, $g(t)=\min_{j \in K_i} g_j(t)$, $f(t)=\min_{j \in K_i} f_j(t)$, Δ_{LB} jest dolnym organiczeniem wartości T_{max} ze zbiorem zadań K_i, zaś x'' jest rozwiązaniem problemu (19) dla $u=\Delta_{LB}$, $i=n,\ldots,1$. Technika ta polega na "usuwaniu" kolejno zadań determinujących minimalną funkcję kosztu przyśpieszenia. Przez symetrię można zbudować analogiczny ciąg usuwając kolejno zadania determinujące minimalną funkcję kosztu spóźnienia.

3.5. Górne ograniczenie

Istotnym elementem metody podziału i ograniczeń, bazującej na podejściu blokowym, jest sposób generowania rozwiązania przybliżonego w każdym węźle drzewa rozwiązań. Rozwiązanie to stanowi podstawę przeprowadzenia podziału drzewa (w oparciu o bloki na ścieżce krytycznej) oraz pozwala na modyfikację bieżącej wartości górnego ograniczenia wykorzystywanej do zamykania węzłów drzewa. Niezależnie od tego rozwiązanie przybliżone może stanowić alternatywną metodę rozwiązywania problemu (P).

Rozwiązanie przybliżone otrzymano stosując typową technikę: wyznacz przybliżoną kolejność wykonywania operacji π^A pewnym algorytmem aproksymacyjnym A, a następnie wyznacz $H(\pi^A)$. Do wyznaczania π^A proponuje się zastosować dowolny algorytm aproksymacyjny stosowany do problemu z funcją celu typu L_{max} lub f_{max}, patrz także rozdz. 3.3.

4. Uwagi końcowe

Wydaje się, że możliwe jest skonstruowanie pewnych dodatkowych własności eliminacyjnych, analogicznych do przedstawionych w pracach [3], [4] dla problemu z funkcją celu C_{max}, które można z pełnym powodzeniem zastosować w proponowanym algorytmie. Zatem podejście blokowe pozwala na rozwiązywanie stosunkowo szerokiej klasy problemów szeregowania, jednakże tylko przy założeniu minimaksowego charakteru kryterium.

LITERATURA

[1] Achuthan N.R., Grabowski J., Sidney J.B.: Optimal flow-shop scheduling with earliness tardiness penalties. Opsearch 1981, t. 18, s. 117-138.

[2] Baker K.R., Scudder G.D.: Sequencing with earliness and tardiness penalties: A review, Operations Research 18(1), 1990, 22-36

[3] Brucker P., Jurish B., Sieviers B.: A Fast Branch & Bound Algorithm for the Job-Shop Scheduling Problem. 1990, praca niepublikowana.

[4] Carlier J., Pinson E.: An algorithm for solving the job-shop problem. Management Science 35, 1989, 164-176.

[5] French S.: Sequencing and Scheduling. An Introduction to the Mathematics of Job-shop. Chichester 1982.

[6] Grabowski J., Smutnicki C.: Zagadnienia szeregowania z minimaksową funkcją kary. Archiwum Automatyki i Telemechaniki 1986 t. XXXI, s. 21-37.

[7] Grabowski J., Nowicki E., Smutnicki C.: Minimalizacja maksymalnego kosztu w gniazdowych zagadnieniach kolejnościowych taśmowych. Archiwum Automatyki i Telemechaniki 1988, t. XXXIII, s.389-402.

[8] Grabowski J., Nowicki E., Smutnicki C.: Algorytm blokowy szeregowania operacji w systemie gniazdowym. Przegląd Statystyczny r. XXXV, z.1, 1988. 67-80.

[9] Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics 1979, t. 5, s. 287-326.

[10] Gupta S.K., Kyparysis J.: Single Machine Scheduling Research. OMEGA International Journal of Mamagement Science 1987, t. 15, s. 207-227.

[11] Hall N.G., Posner M.E.: Earliness-tardiness scheduling problems. I: Weighted deviation of completion times about a common due date, Operations Research 39(5), 1991, 836-846.

[12] Hall N.G., Kubiak W., Sthi S.P.: Earliness-tardiness scheduling problems, II: Deviation of completion times about a restrictive due date. Operations Research 39(5), 1991, 847-856.

[13] Lakshiminarayan I., Lakshanan R., Papineau R., Rochete R.: Optimal single-machine scheduling with earliness and tardiness penalties. Operations Research 1978, t. 26, s. 1079-1082.

[14] Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B.: Sequencing and Scheduling: Algorithms and Complexity, Report BS-R8909, 1989, Centre for Mathematics and Computer Science. Amsterdam, The Netherlands.

[15] Nowicki E., Smutnicki C.: On lower bounds on the maximum lateness on one machine subject to release dates, OPSEARCH 1987, t. 24, s. 106-110.

[16] Nowicki E., Smutnicki C.: An approximation algorithm for single machine scheduling problem with release times and delivery times, Discrete Applied Mathemetics 1992 (w druku).

[17] Mosheiov G.: V-shaped policies for scheduling deteriorating jobs, Operatons Research 40(6), 1991, 979-991.

[18] Sidney J.B.: Optimal single-machine scheduling with earliness and tardiness penalties, Operations Research 1977, t. 25, s. 62-69.

Recenzent: Prof.dr h.inz. Jan Węglarz

Wpłynęło do Redakcji do 30.04.1992 r.

Abstract:

The paper deals with the general job-shop problem with penalty function defined for each task. The penalty depends on task starting time and can also be considered as earliness and tardiness penalties. The criterion is formulated as maximum among penalties and have to be minimized. The model is more general than classical flow- and job- shop problems with min-max criterion considered so far in literature. Since the optimization criteria is not regular then the task schedule need not be active. The proposed solution method decomposes the problem into two subproblems: finding starting times for a given task procesing order and finding the optimal task processing order. A speciali sed polynomial algorithm and the branch-and-bound algorithm based on block approach are proposed for the first and second subproblem, respectively. Some special cases, lower bounds and approximation algorithms are also discussed.