ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1982

Seria: AUTOMATYKA z. 63

Ewa SKUBALSKA

Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej

ZAGADNIENIE OPTYMALIZACJI KOLEJNOŚCI OPERACJI Z PRZEZBROJENIAMI PRZY KRYTERIUM MINIMALIZACJI KOSZTÓW

Streszczenie. Rozważany jest problem ustalania kolejności realizacji n niezależnych operacji wykonywanych na jednej maszynie. Należy ustalić czasy rozpoczęcia i zakończenia realizacji operacji tak, by minimalizować sumaryczne koszty związane z przezbrojeniami maszyny oraz karami za przekroczenie normatywnych terminów wykonania operacji. Sformułowano model matematyczny zagadnienia oraz algorytm rozwiązania oparty na schemacie metody podziału i ograniczeń.

1. SFORMUŁOWANIE PROBLEMU

Dany jest zbiór n operacji $J = \{1,...,n\}$, które mogą być wykonane na jednej maszynie. Czas realizacji każdej operacji wynosi p_j, $j \in J$. Przy przejściu od realizacji jednej operacji do realizacji następnej, zachodzi konieczność dokonania przezbrojenia maszyny. Dla każdej pary operacji $i,j \in J$ określony jest czas trwania przezbrojenia - t_{ij} oraz koszt przezbrojenia a_{ij}. W szczególnym przypadku koszty przezbrojenia są proporcjonalne do czasu jego trwania (lub bezpośrednio mierzone czasem trwania), to znaczy $a_{ij} = \gamma \cdot t_{ij}$, $i,j \in J$, gdzie γ jest stałym współczynnikiem. Ponadto określone są czasy i koszty przezbrojenia, w przypadku gdy dana operacja ma być wykonywana jako pierwsza. Określają je wielkości t_{oj} oraz a_{oj}, $j \in J$. Analogicznie określone są czas i koszt trwania przezbrojenia po wykonaniu wszystkich operacji. Wielkości te, oznaczane odpowiednio przez $t_{j,n+1}$ oraz $a_{j,n+1}$, zależą od operacji, która była realizowana jako ostatnia.

Każdej operacji przyporządkowane są normatywne terminy:

- r_j - najwcześniejszy wymagany czas rozpoczęcia realizacji operacji $j \in J$;
- d_j - najpóźniejszy wymagany czas zakończenia realizacji operacji $j \in J$.

Wyprzedzenie terminu r_j, bądź opóźnienie względem terminu d_j, wiąże się z koniecznością poniesienie dodatkowych kosztów. Jeżeli moment rozpoczęcia realizacji operacji $j \in J$ oznaczymy przez s_j, a moment zakończenia realizacji przez c_j (oczywiście $c_j = s_j + p_j$), to kara związana z wyprzedzeniem normatywnego terminu r_j jest równa:

$$w_j \cdot E_j(s_j) = w_j \cdot \max\left\{0, r_j - s_j\right\};$$

natomiast kara związana z przekroczeniem terminu d_j jest równa:

$$v_j \cdot T_j(c_j) = v_j \cdot \max\left\{0, c_j - d_j\right\},$$

gdzie w_j oraz v_j są stałymi współczynnikami kar określonymi dla każdej operacji $j \in J$.

Zagadnienie optymalizacji polega na ustaleniu takiej kolejności realizacji operacji na maszynie $\pi(1), \pi(2), \ldots, \pi(n)$, $(\pi(i) \in J,\ i = 1,\ldots,n)$ oraz czasów rozpoczęcia (a tym samym i zakończenia) realizacji operacji, by minimalizować sumaryczne koszty związane z realizacją przezbrojeń oraz karami za przekroczenie normatywnych terminów wykonania operacji.

2. MODEL MATEMATYCZNY

Znaleźć ciąg $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ oraz wartości s_j $(j = 1,\ldots,n)$, które minimalizują:

$$\sum_{i=1}^{n+1} a_{\pi(i-1)\pi(i)} + \sum_{j=1}^{n} \left(w_j \cdot E_j(s_j) + v_j \cdot T_j(s_j + p_j)\right) \tag{1}$$

przy ograniczeniach:

$$\exists!\, i \in \left\{1,\ldots,n\right\}: \pi(i) = j,\ j \in J, \tag{2}$$

$$s_{\pi(i)} \geq s_{\pi(i-1)} + p_{\pi(i-1)} + t_{\pi(i-1)\pi(i)}; \quad i = 1,\ldots,n, \tag{3}$$

$$\pi(i) \in J,\ i = 1,\ldots,n; \tag{4}$$

$$s_{\pi(i)} \geq 0,\ i = 1,\ldots,n; \tag{5}$$

przy czym dla uproszczenia zapisu przyjęto:

$$\pi(0) = 0,\ \pi(n+1) = n+1,\ s_0 = 0,\ p_0 = 0.$$

Ograniczenie (2) oznacza, że ciąg π jest permutacją liczb $\left\{1,\ldots,n\right\}$ odpowiadających kolejnym operacjom. Ograniczenie (3) oznacza, że czas rozpoczęcia wykonania operacji $\pi(i)$ jest nie mniejszy niż suma czasu zakończenia realizacji poprzedzającej operacji – $c_{\pi(i-1)} = s_{\pi(i-1)} + p_{\pi(i-1)}$

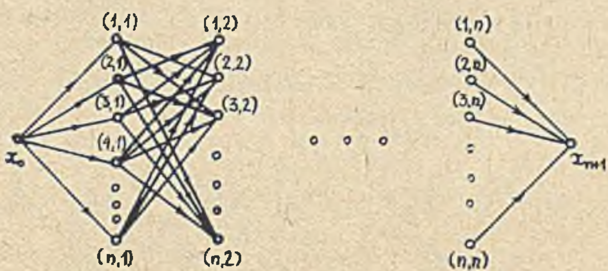
oraz czasu trwania przezbrojenia maszyny - $t_{\pi(i-1)\pi(i)}$. Oznaczmy zbiór wszystkich permutacji π przez $\hat{\Pi}$.

3. DOLNE OSZACOWANIE

Jeżeli odrzucimy w modelu (1) - (5) ograniczenie (2) i zastąpimy je słabszym ograniczeniem postaci

$$\pi(i-1) \neq \pi(i), \quad i = 2,\dots,n \qquad (2')$$

wtedy otrzymamy problem, który można sformułować jako zagadnienie wyznaczania najtańszej (ze względu na funkcję celu postaci (1)) drogi w specjalnie skonstruowanym grafie $G = \langle N,A \rangle$, którego strukturę przedstawia rys. 1.



Rys. 1. Struktura grafu $G = \langle N,A \rangle$

W grafie tym zbiór wierzchołków N zawiera fikcyjne węzły x_o i x_{n+1} oraz n^2 wierzchołków (j,i), odpowiadających umieszczeniu odpowiednio operacji $j \in J$ na i-tej pozycji w sekwencji operacji $(i = 1,\dots,n)$. Zbiór łuków A zawiera wszystkie możliwe ustawienia par operacji na sąsiednich pozycjach w sekwencji. Graf o takiej strukturze zastosowano w pracy [5] przy wyznaczaniu dolnego oszacowania problemu komiwojażera. Tak więc każdy ciąg $\pi = (\pi(1),\dots,\pi(n))$ spełniający warunek (2') jednoznacznie określa w grafie $G = \langle N,A \rangle$ drogę między wierzchołkami x_o i x_{n+1}. Oznaczmy zbiór wszystkich takich ciągów przez Π. Jest oczywiste, że $\hat{\Pi} \subset \Pi$

Problem postaci (1), (2'), (3) - (5) jest nadal zagadnieniem NP-zupełnym. Aby otrzymać zagadnienie o złożoności wielomianowej dokonano relaksacji funkcji celu (1) zastępując ją funkcję postaci:

$$\bar{F}(\pi) = \sum_{i=1}^{n+1} J^{i}_{\pi(i-1)\pi(i)}, \qquad (1')$$

gdzie $\mathcal{T}^i_{jl}$ oznacza dolne oszacowanie przyrostu wartości funkcji celu (1), jeśli $l \in J$ operacja jest wykonywana jako i-ta w kolejności bezpośrednio za operacją $j \in J$.

Wartości $\mathcal{T}^i_{jl}$ wyznaczane są iteracyjnie w następujący sposób:

$$\mathcal{T}^i_{jl} = a_{jl} + \min_{\theta,\theta_o} \Big(\delta^{i-1}_j . \max\left\{0, \bar{c}^{i-1}_j - \theta_o\right\} + w_l . \max\left\{0, r_l + p_l - \theta\right\} +$$

$$+ v_l . \max\left\{0, d_l - \theta\right\}\Big), \tag{6}$$

przy ograniczeniach $\theta_o \geq \hat{c}^{i-1}_j$, $\theta = \theta_o + t_{jl} + p_l$.

Wartość $\hat{c}^{i-1}_j$ oznacza dolne oszacowanie czasu realizacji sekwencji i-1 operacji, z których ostatnia to operacja $j \in J$, natomiast $\delta^{i-1}_j = \min_{k \neq j, k \in J} \delta^{i-1}_{kj}$ oznacza dolne oszacowanie przyrostu funkcji celu (1), jeżeli przyspieszymy termin zakończenia realizacji operacji $j \in J$ względem wyznaczonego w poprzedniej iteracji terminu $\bar{c}^{i-1}_j$.

Oznaczmy optymalną wartość θ w wyrażeniu (6) przez $\bar{c}^i_{jl}$, wtedy aktualną wartość δ^i_{jl} określa nierówność:

$$\delta^i_{jl} . \max\left\{0, \bar{c}^i_{jl} - \hat{\theta}\right\} \leq \delta^{i-1}_j . \max\left\{0, \bar{c}^{i-1}_j - (\hat{\theta} - p_l - t_{jl})\right\} +$$

$$+ w_l . \max\left\{0, r_l + p_l - \hat{\theta}\right\} + v_l . \max\left\{0, \hat{\theta} - d_l\right\},$$

przy czym $\hat{\theta} \leq \bar{c}^i_{jl}$ jest przyspieszonym terminem realizacji operacji $l \in J$ wykonywanej jako i-ta w kolejności.

Wartości $\hat{c}^i_l$ oraz $\bar{c}^i_l$ wyznaczane są odpowiednio jako minimalne spośród otrzymanych do tej pory wartości $\hat{c}^i_{jl}$ oraz $\bar{c}^i_{jl}$, tzn.:

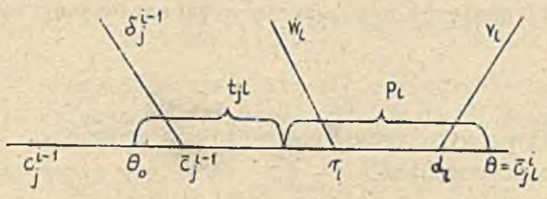
$$\hat{c}^i_l = \min_{j \neq l, j \in J} (\hat{c}^{i-1}_j + t_{jl} + p_l),$$

$$\bar{c}^i_l = \min_{j \neq l, j \in J} \bar{c}^i_{jl}.$$

Jako wartości początkowe przyjmuje się $\hat{c}^o_o = \bar{c}^o_o = 0$, $\delta^o_o = \infty$.

Tak więc wartość $\mathcal{T}^i_{jl}$ zawiera trzy składniki:

- koszt przezbrojenia a_{jl};
- dolne oszacowanie kosztów przekroczenia terminów realizacji operacji l-tej realizowanej jako i-ta w sekwencji (bezpośrednio po operacji $j \in J$)
- dolne oszacowanie przyrostu funkcji celu (1) związane z ewnetualnym przyspieszeniem terminów wykonania wszystkich i-1 operacji poprzedzających operację $l \in J$.



Rys. 2. Wyznaczanie wartości $\mathcal{T}_{jl}^{i}$

Sposób wyznaczania wartości $\mathcal{T}_{jl}^{i}$ ilustruje rysunek 2.

Istnieje możliwość wzmocnienia dolnego oszacowania otrzymanego przez rozwiązanie problemu (1'), (2'), (4), poprzez wprowadzenie kar za przekroczenie ograniczenia (2). Przeprowadza się to, dodając do funkcji celu dodatkowy człon $\sum_{i=1}^{n} u_{\pi(i)}$, w którym u_j ($j \in J$) oznacza karę za przekroczenie ograniczenia (2) ze względu na operację $j \in J$. Kara ta jest: - dodatnia jeśli w ciągu $\pi \in \Pi$ operacja $j \in J$ występuje więcej niż jeden raz; - ujemna, jeśli nie występuje w ogóle; - równa zeru, jeśli jest uwzględniona dokładnie raz.

W ten sposób otrzymujemy po przekształceniu problem

$$F(u) = \min_{\pi\in\Pi} \sum_{i=1}^{n+1} (\mathcal{T}_{\pi(i-1)\pi(i)}^{i} + u_{\pi(i)}) - \sum_{l=1}^{n} u_l,$$

który jest nadal problemem postaci (1'), (2'), (4).

Dla dowolnych kar u_l ($l = 1,\ldots,n$), przy łatwym do spełnienia warunku $\sum_{l=1}^{n} u_l = 0$, zachodzi nierówność:

$$\min_{\pi\in\Pi}\Big(\sum_{l=1}^{n+1} \mathcal{T}_{\pi(i-1)\pi(i)}^{1}\Big) = \min_{\pi\in\Pi} \sum_{i=1}^{n+1} (\mathcal{T}_{\pi(i-1)\pi(i)}^{1} + u_{\pi(i)})$$

$$\geq \min_{\pi\in\Pi} \sum_{i=1}^{n+1} (\mathcal{T}_{\pi(i-1)\pi(i)}^{i} + u_{\pi(i)}) = F(u).$$

Wynika stad, że dla dowolnych kar u_l ($l = 1,\ldots,n$) wartość $F(u)$ jest dolnym oszacowaniem wartości rozwiązania problemu (1) - (5). Najlepsze kary ze względu na wartość dolnego oszacowania można otrzymać rozwiązując problem

$$\max_{u=(u_1,\ldots,u_n)} \left\{ F(u) \,\Big|\, \sum_{l=1}^{n} u_l = 0 \right\}.$$

Problem ten można rozwiązać przy użyciu subgradientowej metody optymalizacji [1, 3].

4. WYZNACZANIE OPTYMALNYCH TERMINÓW REALIZACJI OPERACJI PRZY USTALONEJ SEKWENCJI π

Oznaczmy przez $F_c(\pi)$ najmniejszy koszt realizacji n operacji przy ustalonej sekwencji ich wykonania $\pi \in \hat{\Pi}$. Otrzymujemy wtedy, że

$$F_c(\pi) = \min_{\bar{c} \in C(\pi)} F(\pi, \bar{c}) = \min_{c \in C(\pi)} \left\{ \sum_{i=1}^{n+1} a_{\pi(i-1)\pi(i)} + \sum_{i=1}^{n} (w_i \cdot E_i(c_i - p_i) + v_i \cdot T_i(c_i)) \right\},$$

gdzie $\bar{c} = (c_{\pi(1)}, \ldots, c_{\pi(n)})$ jest wektorem terminów zakończenia realizacji operacji $\pi(1), \ldots, \pi(n)$, natomiast $C(\pi)$ oznacza zbiór dopuszczalnych wektorów $\bar{c}$, tzn. takich, które spełniają ograniczenia (3), (5).

Niech π^k oznacza częściową sekwencję w π złożoną z k elementów, $\pi^k = (\pi(1), \ldots, \pi(k))$. Bez straty ogólności możemy przyjąć, że $\pi = (1, \ldots, n)$. Załóżmy, że $c_1^{k-1}, \ldots, c_{k-1}^{k-1}$ jest rozwiązaniem optymalnym problemu:

$$\min_{(c_1, \ldots, c_{k-1}) \in C(\pi^{k-1})} \sum_{i=1}^{k-1} (w_i \cdot E_i(c_i - p_i) + v_i \cdot T_i(c_i)). \tag{7}$$

Wtedy rozwiązaniem optymalnym problemu (7) dla sekwencji operacji $\pi^k = (1, \ldots, k)$ jest wektor czasów $(c_1^k, \ldots, c_k^k)$ wyznaczany według następującego algorytmu:

A. Jeżeli wartość $\tilde{c}_k^k$ minimalizuje wyrażenie $w_k \cdot E_k(c_k - p_k) + v_k \cdot T_k(c_k)$ oraz termin ten nie jest sprzeczny z ustalonymi uprzednio wartościami $(c_1^{k-1}, \ldots, c_{k-1}^{k-1})$, czyli $\tilde{c}_k^k \geq c_{k-1}^{k-1} + t_{k-1,k} + p_k$, wtedy $c_i^k = c_i^{k-1}$, $i = 1, \ldots, k-1$, $c_k^k = \tilde{c}_k^k$.

B. Jeżeli wyznaczona uprzednio wartość c_k^k jest sprzeczna z terminami $(c_1^{k-1}, \ldots, c_{k-1}^{k-1})$, wtedy należy przyjąć $c_k^{k-1} = c_{k-1}^{k-1} + t_{k-1,k} + p_k$ oraz wyznaczyć zbiór $J^k = \{l, l+1, \ldots, k\}$, który zawiera wszystkie kolejne operacje wykonywane przy aktualnie wyznaczonych czasach c_i^{k-1} bezpośrednio jedna po drugiej (bez przerw), czyli $c_i^{k-1} = c_{i-1}^{k-1} + t_{i-1,i} + p_i$, $i = l+1, \ldots, k$.

a) Jeśli $l=0$, zbiór J^k zawiera wszystkie operacje z sekwencji $(1, \ldots, k)$ oraz nie istnieje możliwość przyspieszenia terminów wykonania operacji. Wartości $c_i^k = c_i^{k-1}$, $i = 1,\ldots,k$ są rozwiązaniem optymalnym problemu.

b) Jeżeli $l > 0$, wtedy istnieje możliwość przyspieszenia terminów realizacji operacji należących do zbioru $\mathcal{J}^k$. Należy sprawdzić, czy spowoduje to zmniejszenie sumy kar: $\sum_{i=l+1}^{k} (w_i'.E_i(c_i-p_i) + v_i.T_i(c_i))$. Nastąpi to tylko wtedy, gdy spełniona jest nierówność:

$$\sum_{i=l+1}^{k} \left[w_i.x_i(c_i^{k-1}) + v_i.y_i(c_i^{k-1})\right] < 0, \tag{8}$$

gdzie:

$$x_i(c) = \begin{cases} 1, & \text{gdy } c \leqslant r_i + p_i, \\ 0, & \text{w przeciwnym przypadku,} \end{cases}$$

oraz

$$y_i(c) = \begin{cases} 1, & \text{gdy } c > d_i, \\ 0, & \text{w przeciwnym przypadku.} \end{cases}$$

Jeżeli nierówność (8) nie jest spełniona, wtedy zmiana terminów realizacji operacji $(1,\ldots,k)$ nie spowoduje zmniejszenia kar za przekroczenie normatywnych terminów r_i oraz d_i. Wartości rozwiązania optymalnego problemu są wyznaczane tak, jak w przypadku (a). Jeżeli natomiast nierówność (8) jest spełniona, zmniejsza się wartość wszystkich terminów c_i^{k-1}, $i = 1,\ldots,k$, która nie spowoduje zmiany nierówności (8) oraz nie koliduje z terminem zakończenia realizacji operacji $l-1$. Następnie aktualizuje się zawartość zbioru $\mathcal{J}^k$ i ponownie sprawdza warunki opisane w punktach (a) i (b).

Wyznaczone wartości optymalne $(c_1^n,\ldots,c_n^n)$ są rozwiązaniem optymalnym problemu wyjściowego postaci $\min\limits_{\bar{c} \in C(\pi)} F(\pi,\bar{c})$.

Algorytm powyższy ma złożoność $O(n^3)$ [6].

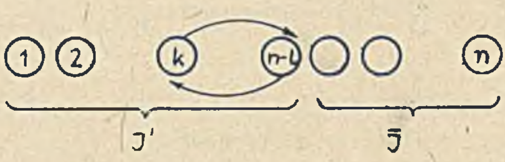
5. ALGORYTM ROZWIĄZYWANIA ZAGADNIENIA

Algorytm rozwiązania zagadnienia (1) - (5) oparty jest na schemacie metody podziału i ograniczeń z mieszaną strategią podziału [2].

Rozpoczynając od pewnej dopuszczalnej sekwencji operacji $\pi_0 \in \hat{\Pi}$ generuje się ciąg permutacji $\pi_r \in \hat{\Pi}$. Dla każdej permutacji π_r wyznacza się jej koszt równy $F_c(\pi_r)$. Nową permutację $\pi_q \in \hat{\Pi}$ otrzymuje się poprzez ustalenie nowej pozycji w sekwencji dla pewnej operacji, której pozycja do tej pory nie była ustalona. W trakcie działania algorytmu zbiór operacji J jest podzielony na dwa rozłączne podzbiory $\bar{J}_r$ i $J_r' = J - \bar{J}_r$.

Zbiór $\bar{J}_r$ jest zbiorem takich operacji, których pozycja w permutacji $\pi_r \in \hat{\Pi}$ została ustalona.

Rozważane zostaną dwie wersje algorytmu. W pierwszej, zbiorowi $\bar{J}_r = \{j_{n-l+1},\dots,j_n\}$, $l = |\bar{J}_r|$ odpowiadać będzie rozwiązanie częściowe $(j_{n-l+1},\dots,j_n)$, w którym ustalonych jest l ostatnich pozycji w sekwencji operacji. W wersji drugiej, zbiór $\bar{J}_r = \{j_1,\dots,j_l\}$ odpowiada permutacji częściowej $(j_1,\dots,j_l)$ ustalonej od początku. Przy generowaniu permutacji π_q z permutacji $\pi_r \in \hat{\Pi}$, pewna operacja $j \in J'_r$ zostanie umieszczona na $n-l$-tej pozycji (lub $l+1$ pozycji) w π_q, natomiast operacje należące do zbioru $\bar{J}_r$ będą zajmowały te same pozycje w permutacji π_q i π_r. Pozostałe operacje zostaną uporządkowane w π_q przy użyciu dowolnej reguły heurystycznej. Na rys. 3 przedstawiono najprostszy sposób generowania nowej permutacji, który polega na zmianie pozycji tylko dwu operacji.



Rys. 3. Generowanie nowej permutacji

Proces generowania permutacji prowadzi się aż do chwili, gdy wystąpi jedna z dwu następujących sytuacji:

- można wykazać, że z danej permutacji π_r nie można wygenerować (bezpośrednio lub pośrednio) permutacji, która dałaby mniejszą wartość funkcji celu (1) niż najlepsze do tej pory znalezione rozwiązanie;
- zbiór $\bar{J}_r = J$, czyli wszystkie operacje mają ustalone pozycje.

Wtedy należy odrzucić rozpatrywaną sekwencję π_r i cofnąć się do sekwencji π_s, z której π_r została bezpośrednio wygenerowana, aktualizując odpowiednio zawartości zbiorów $\bar{J}_s$ i J'_s. Proces generowania permutacji $\pi_r \in \hat{\Pi}$ można przedstawić w postaci drzewa H, w którym każdy wierzchołek będzie odpowiadał pewnej permutacji $\pi_r \in \hat{\Pi}$, a łuk będzie przedstawiał parę $\langle \pi_r, \pi_q \rangle$, gdzie permutacja π_q została wygenerowana bezpośrednio z π_r. Oznaczmy przez $\Pi_1(j_k,\dots,j_n)$ zbiór wszystkich ciągów $\pi \in \Pi$, w których ustalony jest fragment sekwencji począwszy od k-tej pozycji, to znaczy:

$$\Pi_1(j_k,\dots,j_n) = \left\{ \pi \in \Pi \,\middle|\, \pi(k) = j_k,\dots,\pi(n) = j_n \right\}.$$

Zbiór wszystkich permutacji $\pi_q \in \hat{\Pi}$, które można w danej sytuacji wygenerować z permutacji $\pi_r = (\pi_r(1),\dots,\pi_r(k-1), j_k,\dots,j_n)$ jest równy $\hat{\Pi}_1(j_k,\dots,j_n) = \Pi_1(j_k,\dots,j_n) \cap \hat{\Pi}$.

Analogicznie można zdefiniować zbiory $\Pi_2(j_1,\dots,j_k)$ oraz $\hat{\Pi}_2(j_1,\dots,j_k)$ w drugiej wersji algorytmu.

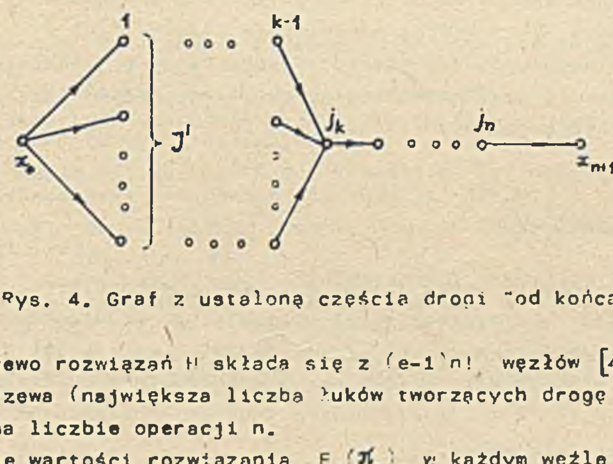
Podział zbioru $\hat{\Pi}_1(j_k,\dots,j_n)$ przeprowadzany jest w następujący sposób:

$$\hat{\Pi}_1(j_k,\dots,j_n) = \bigcup_{j\in J-\{j_k,\dots,j_n\}} \hat{\Pi}_1(j,j_k,\dots,j_n).$$

W każdej iteracji algorytmu, dla każdej nowo wygenerowanej permutacji $\pi_r = (\pi_r(1),\dots,\pi_r(k-1),j_k,\dots,j_n)$ wyznacza się dolne oszacowanie wartości rozwiązania optymalnego w zbiorze $\hat{\Pi}_1(j_k,\dots,j_n)$ poprzez rozwiązanie problemu

$$\min_{\pi\in\Pi_1(j_k,\dots,j_n)} \sum_{i=1}^{n+1} \left(\mathcal{T}_{\pi}^{i}(i-1)\pi(i) + u_{\pi(i)}\right).$$

Zagadnieniu temu odpowiada graf przedstawiony na rys. 4.



Rys. 4. Graf z ustaloną częścią drogi "od końca"

Pełne drzewo rozwiązań H składa się z $(e-1)n!$ węzłów [4]. Maksymalna głębokość drzewa (największa liczba łuków tworzących drogę skierowaną w H) jest równa liczbie operacji n.

Obliczenie wartości rozwiązania $F_c(\pi_r)$ w każdym węźle drzewa pozwala na połączenie metody podziału i ograniczeń z dowolną metodą heurystyczną. Algorytm rozwiązania problemu można sformułować następująco (pierwsza wersja uwzględniająca ustalanie sekwencji operacji "od końca"):

Algorytm

Niech π_0 będzie korzeniem drzewa H oraz niech $\mathcal{J}_0 = \emptyset$, $k^* = \infty$, gdzie k^* oznacza wartość najlepszego znanego do tej pory rozwiązania.

Niech π_r i $\mathcal{J}_r = \{j_k,\dots,j_n\}$ będą permutacją i zbiorem operacji o ustalonej pozycji w r-tej iteracji algorytmu.

Krok 1. Wyznaczyć dolne oszacowanie dla zbioru $\hat{\Pi}_1(j_k,\ldots,j_n)$. Jeżeli wartość dolnego oszacowania jest nie mniejsza niż k^*, przejść do kroku 4.

Krok 2. Obliczyć wartość rozwiązania $F_c(\pi_r)$. Jeśli uzyskano poprawę, czyli $F_c(\pi_r) < k^*$, to przyjąć $k^* = F_c(\pi_r)$ i zapamiętać permutację π_r.

Krok 3. Jeżeli $J_r' \neq \emptyset$ wybrać $j^* \in J_r'$, które minimalizuje wartość dolnego oszacowania:

$$\min_{j \in J_r'} \min_{\pi \in \Pi_1(j,j_k,\ldots,j_n)} \sum_{i=1}^{n+1} (J^1_{\pi(i-1)\pi(i)} + u_{\pi(i)}).$$

Usunąć ze zbioru J_r', te operacje, dla których dolne oszacowanie jest większe od k^*. Wygenerować nową permutację $\pi_q = (\pi_q(1),\ldots,\pi_q(k,2),j^*,j_k,\ldots,j_n)$. Usunąć operację j^* ze zbioru J_r' i umieścić w zbiorze $\bar{J}_r$. Utworzyć w drzewie H nowy węzeł π_q oraz nowy łuk $<\pi_r,\pi_q>$. Przyjąć $\bar{J}_q = \bar{J}_r$ oraz $J_q' = J-\bar{J}_q$. Przejść do kroku 1. Jeśli $J_r' = \emptyset$, przejść do kroku 4.

Krok 4. Cofnąć się do bezpośredniego poprzednika permutacji π_r w drzewie H, to znaczy π_s, której odpowiadają zbiory J_s' i $\bar{J}_s$. Przejść do kroku 3. Jeśli należy cofnąć się od korzenia drzewa H, to algorytm kończy działanie. Permutacja π^* związana z aktualną wartością k jest rozwiązaniem optymalnym problemu.

6. PRZYKŁAD ILUSTRACYJNY

Dany jest problem z 5 operacjami o czasach wykonania: $p_1=2$, $p_2=p_4=p_5=1$, $p_3=3$. Normatywne terminy realizacji operacji są odpowiednio równe: $r_1=15$, $d_1=20$, $r_2=2$, $d_2=17$, $r_3=5$, $d_3=10$, $r_4=3$, $d_4=16$, $r_5=4$, $d_5=18$. Wszystkie jednostkowe kary za przekroczenie normatywnych terminów realizacji operacji są równe jedności $(w_i=v_i=1)$. Macierz kosztów przezbrojeń $[a_{ij}] = [t_{ij}]$ przedstawiono na rysunku 5.

Jako rozwiązanie początkowe π_o przyjęto permutację $\pi_o = (3,2,4,5,1)$ otrzymaną przy użyciu algorytmu heurystycznego [6], którego idea polega na ustaleniu kolejnych operacji w permutacji. Jeżeli ostatnio ustalona została w permutacji operacja k-ta, to jako następną w kolejności przyjmuje się operację i-tą, która minimalizuje wyrażenie:

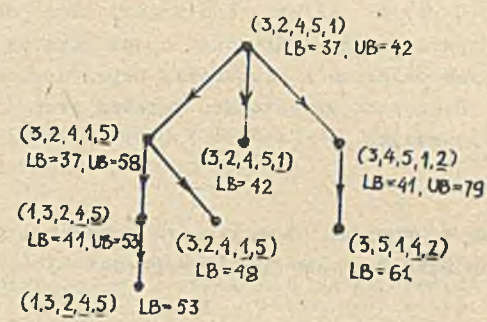
$$H_k(i) = 2a_{ki} + w_i r_i + \frac{d_i}{v_i} - p_i - a_{i,n+1}.$$

Drzewo rozwiązań wygenerowane przez algorytm przedstawia rys. 6. Rozwiązaniem optymalnym jest permutacja π_o z terminami realizacji $c_1=31$,

$c_2 = 14$, $c_3 = 7$, $c_4 = 19$, $c_5 = 23$. Wartość optymalna funkcji celu jest równa 42.

i \ j	1	2	3	4	5	0
0	0	6	7	8	6	X
1	X	6	7	8	6	0
2	6	X	3	4	6	6
3	7	3	X	7	8	7
4	8	4	7	X	3	8
5	6	6	8	3	X	6

Rys. 5. Wartości kosztów przezbrojeń



Rys. 6. Drzewo przezbrojeń

LITERATURA

[1] Geoffrion A.M.: Lagrangian relaxation and its use in integer programming. Mathematical Programming Study 2, 1974, s. 82-114.

[2] Grabowski J.: Uogólnione zagadnienia optymalizacji kolejności operacji w dyskretnych systemach produkcyjnych. Prace Nauk. ICT P. Wr., Ser.: Monografie 50, Wrocław 1979.

[3] Held M., Wolfe P., Crowder H.O.: Validation of subgradient optimization. Mathematical Programming 6, 1974, s. 62-88.

[4] Lenstra J.K.: Sequencing by enummerative methods.Mathematical Centrum, Amsterdam 1977.

[5] Picard C., Queyranne M.: The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. Operations Research, vol. 26, 1978, nr 1, s. 86-110.

[6] Skubalska E.: Zagadnienia optymalizacji minimalnokosztowej z funkcją kary w systemach transportowych. Praca doktorska, Wyd. Pol. Wr., Wrocław 1980.

Recenzent: prof. dr inż. Henryk KOWALOWSKI

Wpłynęło do Redakcji 15.05.82 r.

ЗАДАЧА ОПТИМАЛИЗАЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАЦИЙ С ПЕРЕОБОРУДОВАНИЕМ И МИНИМАЛЬНОСТНЫМ КРИТЕРИЕМ

Р е з ю м е

В работе представлено задачу последовательности операций на одной машине с критерием минимализации суммы штрафов из-за невыполнения сроков изготовления операций и стоимостей переоборудования машины.

Представлено алгоритм решения этой проблемы на базе метода ветьвей и ограничений.

THE SCHEDULING PROBLEM WITH SETUP COSTS AND WITH MINIMUM COST CRITERION

S u m m a r y

We consider a single-machine scheduling problem where penalties occur for operations that either commence before their target start date or are completed after their due date. The total cost is composed of earliness and tardiness penalties and setup costs s_{jk} occuring when operation k is followed by operation j.

A branch and bound algorithm is given. Subgradient optimization is used as a method of maximizing the resulting lower bounds for imbedding into tree search algorithm.