ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1978
Seria: AUTOMATYKA z. 46 Nr kol. 582

Andrzej Adrabiński, Mieczysław Wodecki
Uniwersytet Wrocławski
Józef Grabowski
Politechnika Wrocławska

ALGORYTM OPTYMALIZACJI ZAGADNIEŃ SEKWENCYJNYCH W DYSKRETNYCH PROCESACH PRODUKCYJNYCH

Streszczenie: W wielu gałęziach przymysłu procesy produkcyjne można przedstawić jako przepływy dyskretnych detali. Dla takich procesów sformułowano w referacie ogólne zagadnienie kolejnościowe oraz zaproponowano model matematyczny i algorytm optymalizacyjny w oparciu o teorię grafów dysjunktywnych.

1. Wstęp

W wielu gałęziach przemysłu (hutniczy, maszynowy, budowlany itd.) proces produkcyjny charakteryzuje się przepływem materiałów (w ciągu technologicznym) w postaci pojedynczych elementów lub ich partii. Elementy te są poddawane operacjom obróbki na kolejnych agregatach technologicznych (maszynach). Czasy poszczególnych operacji są określone i zwykle różne dla różnych elementów i agregatów. W związku z tym powstaje zagadnienie określenia kolejności wykonywania elementów na poszczególnych maszynach aby uzyskać minimalny czas wykonania wszystkich operacji.

Dla tak postawionego zagadnienia, zwanego ogólnym problemem kolejnościowym, przedstawiamy model matematyczny oraz algorytm rozwiązania w oparciu o elementy teorii grafów dysjunktywnych i metodę podziału i ograniczeń.

2. Model matematyczny ogólnego problemu kolejnościowego

W paragrafie tym przedstawimy matematyczny model ogólnego problemu kolejnościowego, podamy własności tego modelu oraz scharakteryzujemy graf dysjunktywny reprezentujący ten model.

Niech

$$N = \{1,2,\ldots,n\}$$

będzie zbiorem zadań (operacji), które mają być wykonane za pomocą maszyn ze zbioru

$$B = \{1,2,\ldots,m\}.$$

Oznaczmy przez

$$Q = \{1,2,\ldots,q\}$$

zbiór różnych typów (rodzajów) maszyn. Każdemu $k \in Q$ odpowiada $B_k \in B$, podzbiór maszyn tego samego rodzaju (tzn. o tych samych możliwościach funkcjonalnych), które jednak mogą mieć różne parametry techniczne (np. wydajność) oraz podzbiór operacji $N_k \in N$, które mają być wykonane przy użyciu maszyn typu k. Zakładamy, że zachodzą następujące zależności

$$\bigcup_{k\in Q} N_k = N, \quad N_k \cap N_l = \emptyset, \quad k,l \in Q,\ k \neq l, \qquad \bigcup_{k\in Q} B_k = B, \quad B_k \cap B_l = \emptyset,$$

$$k,l \in Q,\ k \neq l.$$

Naszym celem będzie dokonanie przydziału maszyn do wykonania operacji oraz wyznaczenie optymalnej kolejności wykonywania operacji na poszczególnych maszynach, przy dodatkowym założeniu, że na szukaną kolejność mogą być nałożone pewne ograniczenia wynikające z technologicznego porządku wykonywania operacji.

Niech

$$T \subset N \times N$$

będzie relacją częściowego porządku wyrażającą wymagania porządku technologicznego.

Ogólny problem kolejnościowy polega na określeniu przydziału maszyn operacjom i wyznaczeniu takiej kolejności wykonywania operacji, która

minimalizuje łączny czas ich wykonania oraz spełnia ograniczenia:
(i) każda operacja może być wykonywana tylko na jednej maszynie z określonego dla niej typu,
(ii) żadna maszyna nie może wykonywać jednocześnie więcej niż jedną operację,
(iii) zachowany musi być technologiczny porządek wykonywania operacji,
(iv) wykonywanie żadnej operacji nie może być przerwane.

Wprowadzimy teraz następujące oznaczenia:

t_0 – czas rozpoczęcia wszystkich operacji (bez straty ogólności możemy przyjąć $t_0 = 0$),

t_z – szukany czas wykonania wszystkich operacji,

t_j^x – czas rozpoczęcia operacji j ($j \in N$),

t_j^y – czas zakończenia operacji j ($j \in N$),

c_{ik} – czas wykonywania i-tej operacji za pomocą k-tej maszyny ($c_{ik} > 0$, $i \in N_j$, $k \in B_j$, $j \in Q$; w przeciwnym przypadku, tzn. gdy $i \notin N_j$ lub $k \notin B_j$ dla $j \in Q$ przyjmujemy $c_{ik} = \infty$),

x_{ik} – zmienna decyzyjna określona następująco:

$$x_{ik} = \begin{cases} 1, & \text{gdy i-ta operacja jest wykonywana na k-tej maszynie,} \\ 0, & \text{w przeciwnym razie.} \end{cases}$$

Stosując powyższe oznaczenia, ogólny problem kolejnościowy ma następującą postać:

(1) zminimalizować t_z,

przy oznaczeniach:

(2) $\sum_{k=1}^{m} x_{ik} = 1$, $t_i^y - t_i^x \geq \sum_{k=1}^{m} x_{ik} c_{ik}$, $i \in N$,

(3) $[(x_{ik} = 1) \wedge (x_{jk} = 1)] \Rightarrow [(t_i^x \geq t_j^y) \vee (t_j^x \geq t_j^y)]$, $i \neq j$, $i,j \in N$, $k \in B$,

(4) $t_j^x \geq t_i^y$, $(i,j) \in T$,

(5) $t_j^x \geq t_0$, $j \in N$,

(6) $t_z \geq t_i^y$, $i \in N$,

(7) t_0, t_z, t_j^x, $t_j^y \geq 0$, $j \in N$, $t_z < \infty$, $x_{ik} \in \{0,1\}$, $i \in N$, $k \in B$.

Ograniczenia (2) są matematycznym zapisem ograniczeń (i) oraz (iv), ograniczenia (3) odpowiadają ograniczeniom (ii), a (4) są ograniczeniami porządku technologicznego T. Ograniczenia (5) – (7) są oczywiste.

Oznaczmy przez P problem (1) – (7). Problem P jest problemem programowania liniowego, częściowo całkowitoliczbowego, które jak dotychczas nie doczekało się efektywnych metod obliczeniowych.

Ustalonemu problemowi P, a więc ustalonym zbiorom N, B, Q, $\{N_i\}$, $\{B_i\}$ oraz wartościom $\{c_{ik}\}$ przyporządkujemy teraz multidigraf dysjunktywny z obciążonymi łukami

$$G = \langle A, U; V', V \rangle,$$

gdzie A jest zbiorem wierzchołków, a U, V', V zbiorami łuków.

Zbiór wierzchołków A jest sumą rozłącznych zbiorów

$$A = X \cup Y \cup \{z_0\} \cup \{z_1\},$$

gdzie $X = \{x_i\}_{i \in N}$, $Y = \{y_i\}_{i \in N}$, i wirzchołki x_i, y_i, z_0, z_1 odpowiadają zdarzeniom t_i^x, t_i^y, t_0, t_z.

Podzbiór łuków U składa się z łuków odpowiadających ograniczeniom (4) – (6), a więc (a) dla każdej pary $(i,j) \in T$ mamy $(y_i, x_j) \in U$, (b) $(z_0, x_j) \in U$ dla każdego wierzchołka j, który nie ma żadnego poprzednika, (c) $(y_j, z_1) \in U$ dla każdego wierzchołka j, który nie ma żadnego następnika. Wszystkie łuki zbioru U obciążamy wagami 0.

Zbiór łuków V' odpowiada ograniczeniom (2), z których wynika, że każdej operacji $i \in N_1$ przyporządkowane jest dokładnie jedna maszyna z B_l. Dla każdej operacji $i \in N_1$ wprowadzamy $|B_l|$ łuków $(x_i, y_i)_k$, $k = 1, 2, \ldots, |B_l|$, o tych samych początkach i końcach, odpowiadających maszynom, na których może być wykonywana ta operacja. Tak zdefiniowany podzbiór łuków nazywamy <u>i-tym zbiorem dysjunktywnym</u> i oznaczamy przez V_i. Kolejne łuki z V_i obciążamy wagami c_{ik}, $k \in B_l$, gdzie l określone jest przez $i \in N_l$. V' zawiera podzbiory V_i' odpowiadające wszystkim operacjom, zatem

(8) $$V' = \{V_i'\}_{i \in N}.$$

Dowolny łuk dla ustalonego zbioru V_i' nazywamy <u>kuzynem</u> innych łuków z tego zbioru, a czynność zastępowania łuku którymś z jego kuzynów nazywamy <u>przełączaniem</u>.

Ograniczenia (2) powodują, że w dowolnym rozwiązaniu dopuszczalnym problemu P, dla każdej operacji $i \in N$ równość $x_{ik} = 1$ jest spełniona dla dokładnie jednego $k \in B$. W terminach teorii grafów oznacza to, że każdemu rozwiązaniu dopuszczalnemu odpowiada w zbiorze V' zbiór reprezentantów rodziny zbiorów $\{V_i'\}$.

Niech S' oznacza dowolny taki zbiór reprezentantów. S' nazywamy <u>reprezentacją zbiorów</u>.

Ograniczenia (3) powodują, że żadne dwie operacje nie mogą być jednocześnie wykonywane na tej samej maszynie z B_l, $l \in Q$, więc aby zapobiec wykonywaniu na jednej maszynie jednocześnie więcej niż jednej operacji, dla każdej pary operacji $i, j \in N_1$, $(i,j) \notin T$, $l \in Q$ wprowadzamy parę łuków $[(y_i, x_j), (y_j, x_i)]$, zwaną dalej parą łuków dysjunktywnych lub po prostu <u>parą dysjunktywną</u>. Dowolny łuk z takiej pary nazywamy <u>partnerem</u> drugiego łuku z tej pary, a czynność zastępowania łuku partnerem nazywamy <u>przełączaniem</u>. V jest zbiorem wszystkich par łuków dysjunktywnych, tj.

(9) $$V = \{[(y_i, x_j), (y_j, x_i)] : i, j \in N_1,\ (i,j) \notin T,\ l \in Q\}.$$

Zauważmy, że jeżeli istnieje $k \in B$ takie, że $x_{ik} = x_{jk} = 1$ dla $i \neq j$, $i, j \in N$, czyli jeżeli ta sama maszyna k została przydzielona dwóm zadaniom i oraz j, to ograniczenie (3) powoduje, że opracja i jest wykonywana przed operacją j lub odwrotnie. Łatwo pokazać, że w tym przypadku może być spełniona tylko jedna z nierówności $t_i^x \geq t_j^y$ albo $t_j^x \geq t_i^y$. W terminach teorii grafów oznacza to, że dowolnemu rozwiązaniu dopuszczalnemu problemu P odpowiada podzbiór łuków dysjunktywnych S, który zawiera co najwyżej jeden łuk z każdej pary łuków dysjunktywnych. S nazywamy <u>reprezentacją łuków</u>.

Ograniczenia (3) mogą teraz być napisane w następującej postaci: Jeżeli $x_{ik} = x_{jk} = 1$, $k \in B_l$, to $(y_j, x_i) \in S$ albo $(y_i, x_j) \in S$ dla $i, j \in N_1$, $l \in Q$, gdzie S jest reprezentacją łuków. Tę postać ograniczeń (3) nazywamy <u>warunkiem implikacyjnym</u>.

Niech

(10) $$\bar{D} = \{D : D = \langle A, U \cup S' \cup S \rangle\}$$

będzie rodziną digrafów częściowych multidigrafu dysjunktywnego G takich, że S' jest dowolną reprezentacją zbiorów, a S jest reprezentacją łuków, dla której D jest digrafem bezkonturowym i spełnia warunek implikacyjny. Innymi słowy, $\bar{D}$ jest rodziną digrafów odpowiadających rozwiązaniom dopuszczalnym problemu P. Zauważmy, że dla ustalonego rozwiązania dopuszczalnego, t_z jest długością drogi krytycznej (tj. najdłuższej) w digrafie odpowiadającym temu rozwiązaniu. Digraf D należący do rodziny $\bar{D}$, a odpowiadający ustalonym reprezentacjom S_r i S_p będziemy oznaczać przez $D_{rp} = \langle A, U \cup S_r \cup S_p \rangle$, a długość najdłuższej drogi w D_{rp} przez L_{rp}.

Drogę krytyczną o długości L_*, gdzie $L_* = \min_{D_{rp} \in \bar{D}} L_{rp}$, nazywamy <u>minimaksymalną</u> w multidigrafie dysjunktywnym G, a związane z nią reprezentacje S_* i S_* nazywamy optymalnymi.

Możemy teraz sformułować twierdzenie

TWIERDZENIE 1. Problem P jest równoważny problemowi znalezienia długości minimaksymalnej L_* w multidigrafie dysjunktywnym G. Optymalną wartością t_z jest długość L_* drogi krytycznej w digrafie D_{**}, a wartości pozostałych zmiennych t_j^x, t_j^y ($j \in N$), mogą być wyznaczone za pomocą etapu „wstecz" metody drogi krytycznej zastosowanej do digrafu D_{**}.

W ten sposób rozwiązanie zagadnienia P sprowadza się do znalezienia digrafu $D \in \bar{D}$, który ma najkrótszą drogę krytyczną.

3. Opis działania algorytmu

Załóżmy, że mamy reprezentację zbiorów

$$(11) \qquad S_0' = \{(x,y) \in V' : \exists\, i \in N,\ (x,y) \in V_i \wedge c(x,y) = \min_{(a,b) \in V_i'} c(a,b)\},$$

gdzie $c(x,y)$ jest obciążeniem łuku (x,y) oraz reprezentacją łuków S_1 taką, że graf $D_{10} = \langle A, U \cup S_1 \cup S_0' \rangle$ jest bezkonturowy. Wówczas łuk $(v,u) \in V'$ lub $(y,x) \in V$ będziemy nazywać zgodnym, jeżeli $(v,u) \in S_0'$ lub $(y,x) \in S_1$, a przeciwnym, jeżeli $(v,u) \in V \setminus S_0'$ lub $(y,x) \in V \setminus S_1$.

Reprezentacja łuków S_r grafu $D_{rp} \in \bar{D}$ może zawierać z każdej pary łuków $[(y,x),(u,v)]$ jedną z trzech możliwości:

1) łuk zgodny,
2) łuk przeciwny,
3) nie zawierać żadnego łuku z danej pary dysjunktywnej.

Możliwość 3 będziemy nazywać łukiem pustym i oznaczać przez $\overline{(y,x)}$ lub $\overline{(u,v)}$. Natomiast czynność zastępowania łuku zgodnego łukiem pustym (tzn. usuwania łuku zgodnego z grafu) będziemy nazywać wyłączaniem.

Drogę minimaksymalną spełniającą warunek implikacyjny w multidigrafie dysjunktywnym G uzyskujemy generując ciąg digrafów $\{D_{rp}\}$, $D_{rp} \in \bar{D}$, rozpoczynając od digrafu $D_{11} = \langle A, U \cup S_1 \cup S_1' \rangle$, gdzie S_1 oraz S_1' są dowolnymi reprezentacjami łuków oraz zbiorów takimi, że digraf $D_{11} \in \bar{D}$.

Dowolny digraf D_{sq} otrzymujemy z pewnego digrafu D_{rq} lub D_{sp} przez przełączenie albo wyłączenie jednego łuku zgodnego z reprezentacji S_r lub przełączenie łuku zgodnego z reprezentacji zbiorów S_p'.

Proces generowania digrafów można przedstawić w postaci drzewa H rozwiązań dopuszczalnych. Każdy wierzchołek w H odpowiada pewnemu digrafowi D_{rp}, a łuk - pewnej parze digrafów D_{rp}, D_{sp} lub D_{rp}, D_{rq} takiej, że odpowiednio D_{sp} lub D_{rq} został otrzymany przez przełączenie albo wyłączenie jednego łuku dysjunktywnego $(y,x) \in S_r$ na łuk przeciwny $(u,v) \in S_s$ lub pusty $\overline{(u,v)}$, albo przez przełączenie łuku $(x,y) \in S_p'$ na łuk przeciwny $(v,u) \in S_q'$. Innymi słowy, łuk (D_{rp}, D_{sp}) w H będzie odpowiadał łukowi $(u,v) \in S_s$ albo łukowi $\overline{(u,v)}$, a łuk (D_{rp}, D_{rq}) łukowi $(v,u) \in S_q'$. Wygenerowanie nowego digrafu (tj. wierzchołka w H) jest związane z wyborem pewnego łuku zgodnego do przełączenia albo wyłączenia z S_r lub przełączenia w S_p'. Czynność tę będziemy nazywać operacją wyboru.

Dla każdego nowo wygenerowanego digrafu wykonujemy następujące czynności:

a) testowanie drogi krytycznej, polegające na sprawdzeniu, czy dalsze przełączanie (albo wyłączanie) łuków zgodnych może doprowadzić do wygenerowania digrafu, w którym droga krytyczna będzie krótsza od najkrótszej dotychczas znalezionej,
b) testowanie warunku implikacyjnego, polegające na sprawdzeniu, czy nowo wygenerowany digraf spełnia warunek implikacyjny.

Jeżeli wynik testowania (a) jest negatywny, to wówczas odrzucamy nowo wygenerowany digraf i cofamy się do jego poprzednika w drzewie H.

Przy generowaniu nowego digrafu D_{sp} lub D_{rq} z D_{rp} będziemy ustalać pewne łuki odpowiednio w reprezentacji S_s lub S_q'. Jeżeli digraf D_{sp}

został otrzymany z D_{rp} w wyniku przełączenia albo wyłączenia łuku zgodnego $(y,x)\in S_r$, to w D_{sp} ustalamy chwilowo odpowiadający łukowi (y,x) łuk przeciwny $(u,v)\in S_s$ albo łuk pusty $(\overline{u,v})$. Łuki ustalone chwilowo nie są brane pod uwagę w dalszym procesie generowania następników względem wierzchołka drzewa H, dla którego zostały ustalone.

Jeżeli cofamy się do D_{rp} i to po raz pierwszy po łuku przeciwnym lub pustym względem pewnego łuku zgodnego, to ustalamy tymczasowo ten łuk przeciwny lub pusty. Natomiast jeżeli cofanie odbywa się już po raz drugi, to łuk zgodny zostaje ustalony na trwałe. Łuk zgodny, dla którego został ustalony tymczasowo łuk przeciwny lub pusty, może ulec zmianie, ale tylko wtedy, jeżeli zachodzi potrzeba wykonania drugiej, jeszcze nie wykonanej czynności przełączenia albo wyłączenia. Bardziej formalnie, dla każdego digrafu D_{rp} określamy zbiór łuków dysjunktywnych ustalonych chwilowo lub trwale $E_r\subset S_r$ oraz zbiór łuków dysjunktywnych F_r ustalonych tymczasowo. Łuki ustalone chwilowo w zbiorze E_r należą do drogi od korzenia drzewa H do wierzchołka D_{rp}, przy czym w E_r są to łuki przeciwne względem łuków zgodnych. Każdy łuk ustalony na trwałe w E_r jest łukiem zgodnym i odpowiada dwóm łukom, przeciwnemu i pustemu - odrzuconym w H przy cofaniu, które uprzednio były wprowadzone do H przez przełączenie i wyłączenie tego łuku zgodnego. Żaden łuk ze zbioru E_r nie ulega zmianie w dowolnym następniku D_{tu} względem D_{rp}. Każdy łuk ustalony tymczasowo w F_r jest łukiem przeciwnym lub pustym odrzuconym z H przy cofaniu. Zbiór F_r zawiera wszystkie takie łuki leżące na drodze od korzenia do węzła D_{rp} w H.

Natomiast, jeżeli nowy digraf D_{rq} został otrzymany z D_{rp} w wyniku przełączenia łuku zgodnego $(x,y)\in S_p'$, to w D_{rq} ustalamy chwilowo łuk przeciwny $(v,u)\in S_q'$, na który nastąpiło przełączenie. Łuk (v,u) nie może być przełączony w żadnym następniku D_{rq}. Jeżeli cofamy się do D_{rp} po łuku przeciwnym względem pewnego łuku zgodnego, to ustalamy tymczasowo ten łuk przeciwny. Przy czym, jeżeli cofanie odbywa się po raz r_j ($r_j = |V_j|-1$, $j\in Q$), to łuk zgodny zostaje ustalony na trwałe. W przypadku, gdy $r_j = 0$, to taki łuk zgodny nie będzie podlegał przełączeniu i od początku będzie ustalony na trwałe. Dla każdego digrafu D_{rp} określamy zbiór łuków dysjunktywnych $E_p'\subset S_p'$ ustalonych chwilowo lub na trwałe oraz zbiór łuków dysjunktywnych F_p ustalonych tymczasowo. Łuki ustalone chwilowo ze zbioru E_p' należą do drogi od korzenia do wierzchołka D_{rp} w H i są to łuki przeciwne. Każdy łuk ustalony na trwałe w E_p' jest łukiem zgodnym i odpowiada wszystkim łukom przeciwnym odrzuconym w H przy cofaniu, które poprzednio były wprowadzone do H przez przełączanie tego łuku zgodnego. Każdy łuk ustalony tymczasowo w F_p jest łukiem przeciwnym, który został odrzucony w H przy cofaniu. Pozostałe uwagi dotyczące tych zbiorów są takie same, jak dla zbiorów E_r i F_r. Łatwo zauważyć, że w kompletnym drzewie H istnieje wierzchołek D_{rp}, w którym $E_p' = S_p'$. Wierzchołek ten będzie punktem startowym algorytmu. Pozwoli to na wyłączenie maksymalnej liczby łuków z reprezentacji S_r w pierwszych iteracjach algorytmu i na uzyskanie dobrego rozwiązania początkowego.

Przejdźmy teraz do szczegółowego opisania wprowadzonych operacji.

Operacja testowania drogi krytycznej

Celem tej operacji jest obliczenie dolnego ograniczenia długości dróg krytycznych w digrafach $D_{sq}\in D$, które mogą być wygenerowane z D_{rp}. Łukami, które nie ulegają zmianie przy generowaniu następników z D_{rp} są łuki ze zbiorów E_r i E_p'. Utwórzmy więc digraf $D(E_r\cup S_p') = \langle A, U\cup E_r\cup S_p'\rangle$ i niech $L(E_r\cup S_p')$ oznacza długość drogi krytycznej w $D(E_r\cup S_p')$. Niech L^* będzie

najkrótszą drogą krytyczną w wygenerowanych do tej pory digrafach w drzewie H.

Jeżeli $L(E_r \cup S'_p) \geq L^*$, to digraf D_{rp} zostaje odrzucony wraz z ewentualnymi następnikami.

Operacja wyboru

Dla ustalonego digrafu D_{rp} operacja wyboru polega na wyznaczeniu w D_{rp} łuku zgodnego, który ma zostać przełączony lub wyłączony. Przełączamy lub wyłączamy jedynie łuki zgodne leżące na drodze krytycznej, bo tylko takie mogą spowodować zmniejszenie długości drogi [2,6]. Jeżeli przez C_{rp} oznaczymy zbiór łuków drogi krytycznej, to będą to łuki ze zbiorów odpowiednio

(12) $$W_r = (S_r \setminus E_r) \cap C_{rp} \quad \text{lub}$$

(13) $$W'_p = (S'_p \setminus E'_p) \cap C_{rp}.$$

Niech K_r będzie zbiorem łuków przeciwnych i pustych względem wszystkich łuków zgodnych ze zbioru W_r. Oczywiście $|K_r| = 2|W_r|$. Zbiór K_r nazywamy zbiorem kandydatów. Bezpośrednim następnikiem digrafu D_{rp} w H może być digraf otrzymany z D_{rp} w wyniku przełączenia lub wyłączenia łuku $(y,x) \in W_r$, czyli w wyniku zamiany łuku (y,x) odpowiednio na łuk przeciwny $(u,v) \in K_r$ lub na łuk pusty $(\overline{u,v}) \in K_r$.

Podobnie, niech K'_p będzie zbiorem łuków przeciwnych względem łuków zgodnych ze zbioru W'_p. Zbiór ten także nazywamy zbiorem kandydatów.

Będziemy starali się znaleźć taki łuk zgodny w W_r lub W'_p, którego przełączenie albo wyłączenie wygeneruje bezpośredni następnik o możliwie najkrótszej drodze krytycznej. Do oceny wyboru łuków ze zbiorów kandydatów posłużą nam liczby $\Delta_{rp}[(y,x),(u,v)]$, $\Delta_{rp}[(y,x),(\overline{u,v})]$ i $\Delta_{rp}[(x,y),(v,u)]$ dokładnie zdefiniowane w [2], które wyrażają oszacowanie zmiany długości dróg krytycznych przy przełączeniu lub wyłączeniu łuku (y,x) albo przełączeniu łuku (x,y).

Testowanie warunku implikacyjnego.

Operacja ta polega na sprawdzaniu, czy nowo wygenerowany digraf spełnia warunek implikacyjny. Niespełnienie tego warunku może nastąpić tylko przy operacji wyłączenia łuku dysjunktywnego ze zbioru W_r. Każdy łuk $(y_j,x_i) \in W_r$ ma dwa łuki przyległe (x_j,y_j) oraz (x_i,y_i), które należą do zbioru S'_p. W procesie generowania ciągu digrafów D_{rp} łuki przyległe ulegają zmianie, co powoduje zmianę treści warunku implikacyjnego. W związku z tym sprawdzenie tego warunku jak i możliwość wyłączenia łuku $(y_j,x_i) \in W_r$ może nastąpić w przypadku, gdy jego łuki przyległe będą ustalone chwilowo lub na trwałe. Niech R_r będzie podzbiorem zbioru kandydatów K_r zawierającym łuki puste, a

(14) $$R_{rp} = \{(y_j,x_i) \in R_r : (x_j,y_j) \in F'_p,\ (x_i,y_i) \in F'_p\}$$

zbiorem łuków pustych, których łuki przyległe (x_i,y_i), (x_j,y_j) są ustalone chwilowo lub trwale.

Łuk zgodny ze zbioru W_r może być wyłączony, jeżeli jego łuk pusty należy do zbioru R_{rp} i wyłączenie to nie powoduje niespełnienia warunku implikacyjnego. Jeżeli natomiast wyłączenie tego łuku powoduje niespełnienie warunku implikacyjnego, wówczas łuk pusty ustalamy tymczasowo.

Jeżeli dla digrafu D_{rp} wynik operacji testowania drogi krytycznej jest negatywny, albo zbiory kandydatów K_r i K'_p są puste, to wówczas należy cofnąć się do bezpośredniego poprzednika D_{sp} lub D_{rq} digrafu D_{rp}. Można udowodnić twierdzenie, że jeżeli cofanie z digrafu D_{sp} do D_{rp} odbywa się po łuku pustym $(\overline{y,x})$ (względem łuku zgodnego $(y,x) \in S_r$), to należy wtedy cofnąć się do bezpośredniego poprzednika D_{lp} lub D_{rk} digrafu D_{rp} (cofanie o dwa poziomy) [2].

W przypadku, gdy digraf D_{rp} ma w drzewie H bezpośredniego poprzednika D_{sp} lub D_{rq}, wówczas operacja cofania polega na odrzuceniu wszystkich informacji dotyczących digrafu D_{rp}, uaktualnieniu zbiorów E_s, F_s, S_s lub E'_q, F'_q, S'_q oraz usunięciu ze zbioru kandydatów K_s lub K'_q łuku, po którym następuje cofanie. W przeciwnym przypadku, tj. gdy D_{rp} nie ma w drzewie H poprzedników, należy wygenerować nowy wierzchołek w H będący poprzednikiem digrafu D_{rp}.

Jeżeli zbiór E'_p jest niepusty (gdy jest pusty, wówczas reprezentacje S_* i S'_* związane z aktualną wartością L^* są optymalne), wyznaczamy zbiór łuków

(15) $$P_p = E'_p \cap C_{rp}.$$

Gdy $P_p = \emptyset$, to przyjmujemy

(16) $$P_p = E'_p.$$

Następnie znajdujemy łuk zgodny $(x,y) \in P_p$ oraz jego kuzyna $(v,u) \notin S'_p$ takiego, że

(17) $$\Delta'_{rp}[(x,y),(v,u)] = \min \{\Delta'_{rp}[(a,b),(c,d)]\},$$

gdzie minimum jest brane po łukach zgodnych $(a,b) \in P_p$ oraz ich wszystkich kuzynach $(c,d) \notin S_p$.

Digraf D_{rq}, będący poprzednikiem D_{rp}, tworzymy przełączając łuk $(x,y) \in P_p$ (tj. biorąc $E'_q = E'_p \setminus \{(x,y)\}$) oraz uaktualniając reprezentację zbiorów, tj. przyjmując

$$S'_q = S'_p \setminus \{(x,y)\} \cup \{(v,u)\},$$

gdzie łuki (x,y) i (v,u) zostały wyznaczone w (17).

4. Algorytm

W tym paragrafie opiszemy szczegółowo poszczególne kroki algorytmu rozwiązywania ogólnego zagadnienia kolejnościowego.

Niech korzeniem drzewa H będzie digraf $D_{11} = \langle A, U \cup S_1 \cup S'_1 \rangle$, oczywiście D_{11} musi należeć do $\bar{D}$. Reprezentacja S_1 zawiera dokładnie po jednym łuku z każdej pary łuków dysjunktywnych, a S'_1 jest dowolną reprezentacją zbiorów. Przyjmujemy $E'_1 := S'_1$, $E_1 := F_1 := F'_1 := \emptyset$, $L^* := \infty$. Dalej, niech $D_{rp} = \langle A, U \cup S_r \cup S'_p \rangle$, E_r, E'_p, F_r, F'_p będą odpowiednio digrafem, zbiorami łuków ustalonych chwilowo i trwale oraz tymczasowo w r+p-tej iteracji algorytmu.

Krok 1. Operacja testowania

Obliczyć $L(E_r \cup S'_p)$, tj. długość drogi krytycznej w digrafie $D(E_r \cup S'_p)$, która jest dolnym ograniczeniem długości dróg krytycznych we wszystkich następnikach $D_{sq} \in \bar{D}$, które mogą być wygenerowane z digrafu D_{rp}.

Jeżeli $L(E_r \cup S'_p) \geq L^*$, to przejść do kroku 4. W przeciwnym przypadku przejść do kroku 2.

Krok 2. Operacja obliczania

Obliczyć długość drogi krytycznej L_{rp}. Jeżeli $L_{rp} < L^*$, to $L^* := L_{rp}$ oraz wyznaczyć zbiory łuków W_r i W'_p, zgodnie z (12) i (13).

Jeżeli $W_r = \emptyset$ i $W'_p = \emptyset$, to przejść do kroku 4. W przeciwnym przypadku określić zbiory łuków-kandydatów K_r i K'_p oraz wykonać

$$K_r := K_r \setminus F_r,$$

$$K'_p := K'_p \setminus F'_p.$$

Jeżeli $K_r = \emptyset$ i $K'_p = \emptyset$, to przejść do kroku 4. W przeciwnym przypadku obliczyć $\Delta_{rp}[(y,x),(u,v)]$, $\Delta_{rp}[(y,x),(\overline{u,v})]$, $\Delta'_{rp}[(x,y),(v,u)]$ dla odpowiednich łuków $(u,v)\in K_r$, $(\overline{u,v})\in K_r$, $(\overline{v,u})\in K'_p$ i przejść do kroku 3.

Krok 3. Operacja wyboru

Jeżeli $K_r = \emptyset$ i $K'_p = \emptyset$, to przejść do kroku 4. W przeciwnym przypadku postąpić w następujący sposób.

Jeżeli $R_{rp} \neq \emptyset$, to wybrać łuk pusty $(\overline{u,v})\in R_{rp}$ taki, że

$$\Delta_{rp}[(y,x),(\overline{u,v})] = \min_{(\overline{c,d})\in R_{rp}} \{\Delta_{rp}[(a,b),(\overline{c,d})]\},$$

oraz sprawdzić warunek implikacyjny postępując następująco. Jeżeli $(y,x) = (y_j,x_i)$ i operacje i oraz j są wykonywane na tej samej maszynie (tzn. $x_{ik} = x_{jk}$), to łuku zgodnego nie można przełączyć (wyłączyć) na łuk pusty $(\overline{u,v})$. Wtedy należy przyjąć

$$K_r := K_r \setminus \{(\overline{u,v})\},$$

$$E_r := E_r \cup \{(\overline{u,v})\}.$$

i przejść do wykonywania kroku 3 od początku. W przeciwnym przypadku, tzn. gdy operacje i oraz j są wykonywane na różnych maszynach, należy wygenerować nowy digraf D_{sp} przez wyłączenie łuku dysjunktywnego (y,x) oraz przyjąć

$$S_s := S_r \setminus \{(y,x)\},$$

$$E_s := E_r, \quad F_s := F_r.$$

Następnie do drzewa H dołączyć nowy wierzchołek (digraf) D_{sp} oraz łuk (D_{rp},D_{sp}) odpowiadający łukowi pustemu $(\overline{u,v})$ i przejść do kroku 1.

Jeżeli $R_{rp} = \emptyset$ i $K'_p \neq \emptyset$, to wyznaczyć łuk $(v,u)\in K'_p$ taki, że

$$\Delta'_{rp}[(x,y),(v,u)] = \min_{(c,d)\in K'_p} \{\Delta'_{rp}[(a,b),(c,d)]\}$$

i wygenerować nowy digraf D_{rq} przez przełączenie łuku zgodnego (x,y) na łuk przeciwny (v,u) oraz chwilowe ustalenie tego łuku przeciwnego, czyli przyjąć

$$S'_q := S'_p \setminus \{(x,y)\} \cup \{(v,u)\},$$

$$E'_q := E'_p \cup \{(v,u)\}, \quad F'_q := F'_p.$$

Do drzewa H dołączyć nowy wierzchołek (digraf) D_{rq} i łuk (D_{rp},D_{rq}) odpowiadający łukowi przeciwnemu (v,u) oraz przejść do wykonywania kroku 1.

Jeżeli $R_{rp} = \emptyset$ i $K'_p = \emptyset$, to wyznaczyć łuk przeciwny $(u,v)\in K_r \setminus R_{rp}$ taki, że

$$\Delta_{rp}[(y,x),(u,v)] = \min_{(c,d)\in K_r \setminus R_{rp}} \{\Delta_{rp}[(a,b),(c,d)]\}.$$

Przełączając łuk zgodny (y,x) na łuk przeciwny (u,v) wygenerować digraf D_{sp} oraz przyjąć

$$S_s := S_r \setminus \{(y,x)\} \cup \{(u,v)\},$$

$$E_s := E_r \cup \{(u,v)\}, \quad F_s := F_r.$$

Następnie do drzewa H dołączyć nowy wierzchołek (digraf) D_{sp} oraz łuk (D_{rp},D_{sp}) odpowiadający łukowi przeciwnemu (u,v) i przejść do kroku 1.

Krok 4. Operacja cofania

Cofnąć się do bezpośredniego poprzednika digrafu D_{rp} w drzewie H. Jeżeli D_{rp} nie ma bezpośredniego poprzednika, to wyznaczyć zbiór łuków F_p, zgodnie z (15) i (16).

Jeżeli $P_p = \emptyset$, to wówczas reprezentacje S_* i S'_* związane z aktualną wartością L^* są optymalne, a droga krytyczna C_{**} w digrafie $D_{**} = \langle A, U \cup S_* \cup S'_* \rangle$ jest drogą minimaksymalną w multidigrafie dysjunktywnym G.

W przypadku, gdy $P_p \neq \emptyset$, wyznaczyć łuk zgodny $(x,y) \in P_p$ i jego kuzyna $(v,u) \in S'_p$ oraz skonstruować na podstawie digrafu D_{rp} digraf D_{r1} i przyjąć

$$S'_1 := S'_p \setminus \{(x,y)\} \cup \{(v,u)\},$$

$$E'_1 := E'_p \setminus \{(x,y)\}, \quad F'_1 := F'_p \cup \{(x,y)\},$$

przy czym jeżeli $(x,y) \in V'_k$, $k \in Q$ i $r_k = 1$, to przyjąć

$$E'_1 := E'_1 \cup \{(v,u)\}.$$

Następnie przejść do wykonywania kroku 2.

Jeżeli natomiast digraf D_{rp} ma bezpośredniego poprzednika D_{kp} lub D_{r1} w drzewie H, to wówczas odrzucić wszystkie informacje dotyczące digrafu D_{rp} i uaktualnić informacje dotyczące D_{kp} lub D_{r1} w niżej opisany sposób.

Jeżeli D_{rp} został wygenerowany z digrafu D_{kp} przez przełączenie łuku zgodnego $(y,x) \in W_k$ na łuk przeciwny (u,v), wówczas wykonać

$$K_k := K_k \setminus \{(u,v)\},$$

$$F_k := F_k \cup \{(u,v)\},$$

$$S_k := S_r \setminus \{(u,v)\} \cup \{(y,x)\}$$

przy czym, jeżeli $\overline{(u,v)} \in F_k$, to

$$E_k := E_k \cup \{(y,x)\},$$

$$F_k := F_k \setminus \{(u,v), \overline{(u,v)}\}$$

oraz przejść do kroku 3.

Z kolei, jeżeli D_{rp} został wygenerowany z D_{kp} w wyniku wyłączenia łuku zgodnego $(y,x) \in W_k$, wówczas wykonać

$$K_k := K_k \setminus \{\overline{(y,x)}\},$$

$$E_k := E_k \cup \{(y,x)\},$$

$$S_k := S_r \cup \{(y,x)\},$$

przy czym jeżeli $(u,v) \in F_k$, gdzie (u,v) jest partnerem łuku (y,x), to przyjąć

$$F_k := F_k \setminus \{(y,x)\}$$

oraz przejść do wykonywania od początku kroku 4 dla digrafu D_{kp}.

Natomiast, jeżeli D_{rp} został wygenerowany z digrafu D_{r1} w wyniku przełączenia łuku zgodnego $(x,y) \in K'_p$ na łuk przeciwny (v,u), wówczas wykonać

$$K'_1 := K'_1 \setminus \{(v,u)\},$$

$$F'_1 := F'_1 \cup \{(v,u)\},$$

$$S'_1 := S'_p \setminus \{(v,u)\} \cup \{(x,y)\}$$

przy czym, jeżeli cofanie odbywa się po raz r_j ze względu na łuk $(x,y) \in V'_j$, $j \in K$, to

$$E'_1 := E'_1 \cup \{(x,y)\},$$

$$F'_1 := F'_1 \setminus V'_j.$$

i przejść do kroku 3.

W pracy [2] zawarty jest dowód twierdzenia, które możemy sformułować w następujący sposób:

TWIERDZENIE 2. Podany wyżej algorytm wyznacza drogę minimaksymalną w multidigrafie dysjunktywnym $G = \langle A,U;V;V' \rangle$.

5. Literatura

[1] Adrabiński A., Wodecki M.: An algorithm for solving a machine sequencing problem with parallel machines, Zastosow. Matem. (w druku).

[2] Adrabiński A., Grabowski J., Wodecki M.: Algorytm rozwiązywania ogólnego problemu kolejnościowego, Raport Nr N-28, Institute of Computer Science, Wrocław University, 1977.

[3] Balas E.: Discrete programming by the filter method, Opns. Res. 15 (1967), 915-967.

[4] –, Machine sequencing via disjunctive graphs: an implicit enumeration algorithm, Opns.Res. 17 (1969), 941-957.

[5] Gorenstein, An algorithm for project (job) sequencing with resource constraints, Opns. Res. 20 (1972), 835-850.

[6] Grabowski J.: A new formulation and solution of the sequencing problem: Mathematical model, Zastosow. Matem. 15 (1976), 325-343.

[7] – A new formulation and solution of the sequencing problem: Algorithm, Zastosow. Matem. 15 (1976), 463-474.

[8] – Formulation and solution of the sequencing problem with parallel machines, Lecture Notes in Control and Information Sciences 7 – Optimization Techniques, Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5-9, 1977 (PART 2), Springer-Verlag 1978, 400-410.

АЛГОРИТМ ОПТИМАЛИЗАЦИИ ПОСЛЕДОВАТЕЛЬНЫХ ЗАДАЧ В ДИСКРЕТНЫХ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССАХ

Р е з ю м е

В работе рассматривается проблемма последовательности изготовления деталей на имеющемся оборудовании так, чтобы получить минимальное время выполнения операций.

Даётся математическая модель и алгоритм решения на базе элементов теории дизъюнктивных графов и метода раздела и ограничений.

ALGORITHM FOR SEQUENTIAL PROBLEMS OPTIMIZATION IN DISCRETE PRODUCTION PROCESSES

S u m m a r y

In many branches of industry the production processes may be characterised with flux of discrete details. The general sequencing problem for such processes is stated in the paper, then a mathematical model is presented, and an algorithm depending on the theory of disjunctive graphs theory described.