ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1980
Seria: AUTOMATYKA z. 54 Nr kol. 650

Józef Grabowski
Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej

ZAGADNIENIA KOLEJNOŚCIOWE W OPTYMALIZACJI CIĄGÓW KRYTYCZNYCH W DYSKRETNYCH SYSTEMACH PRODUKCYJNYCH

Streszczenie. Praca zawiera sformułowanie i rozwiązanie zagadnienia kolejnościowego taśmowego z kryterium minimalizacji maksymalnej terminowości. Poza tym przedstawiono praktyczne przykłady zastosowań tego zagadnienia do optymalizacji tzw. "wąskich gardeł" (gniazd krytycznych), występujących w dyskretnych procesach produkcyjnych.

1. Wstęp

W trakcie projektowania wyposażenia zakładów dąży się do tego, aby w czasie trwania procesu produkcyjnego nie było tzw. "wąskich gardeł" (gniazd krytycznych), ograniczających wydajność całego ciągu produkcyjnego. W razie wystąpienia takiej sytuacji zwykle zwiększa się liczbę odpowiednich agregatów, likwidując gniazda krytyczne. Nie zawsze jednak jest to możliwe ze względów ekonomicznych lub technicznych. W trakcie projektowania zakłada się ewentualną możliwość wystąpienia "wąskiego gardła" na agregatach, których koszty zakupu i eksploatacji są szczególnie duże. Uzyskuje się to, zwiększając moc produkcyjną (przez zwiększenie liczby lub wydajności) pozostałych agregatów o stosunkowo niewielkich kosztach zakupu i eksploatacji. Szczególnie kosztowne są ciągi walcownicze, prasy kuzienne o dużym nacisku, ciężkie obrabiarki, urządzenia specjalistyczne itp. Piece grzewcze natomiast, chłodnie czy też stanowiska do obróbki ręcznej są agregatami o niewielkich kosztach, na których (przy racjonalnym projektowaniu) nie powinno wystąpić "wąskie gardło". Niemniej starzenie się urządzeń, zmiany asortymentu produkcji lub technologii wykonania powodują, że "wąskie gardło" może wystąpić w innych gniazdach. W takim przypadku służby techniczno-organizacyjne obsługi procesu mogą stosunkowo łatwo zidentyfikować takie gniazda.

Formułując zagadnienia kolejnościowe jako zadania optymalizacji przebiegu procesu możemy uznać - w świetle powyższych uwag - że na urządzeniach z gniazd niekrytycznych nie musimy uwzględniać kolejności wykonywania operacji. Jest to tym bardziej istotne, że w zdecydowanej większości praktycznych sytuacji występują zagadnienia kolejnościowe NP - zupełne, dla których, chcąc uzyskać dobre rozwiązanie, musimy stosować metody podziału (np. metodę podziału i ograniczeń, programowanie dynamiczne). Zmniejszenie liczby urządzeń, na których należy wyznaczać optymalną ko-

lejność, przyspiesza działanie algorytmów opartych na tych metodach. Z drugiej strony, tego rodzaju podejście implikuje nowe typy zagadnień kolejnościowych, dla których brak jest algorytmów rozwiązania.

W niniejszym referacie przedstawimy jedno z takich zagadnień związane z taśmowym systemem produkcyjnym. Dla tego zagadnienia zaprezentujemy:

- model matematyczny,
- algorytm optymalizacji,
- praktyczne przykłady zastosowań.

2. Model matematyczny zagadnienia kolejnościowego taśmowego

Załóżmy, że zadanie produkcyjne procesu jest dane w postaci zbioru zadań

$$J = \{J_1, J_2, \ldots, J_n\},$$

które mają być wykonane przy użyciu zbioru maszyn

$$\mathcal{M} = \{M_1, M_2, \ldots, M_m\}.$$

Każde zadanie $J_j \in J$, składa się z ciągu operacji

$$J_j = \langle O_{j1}, O_{j2}, \ldots, O_{jm} \rangle.$$

Operacje te mają być wykonane w zadanym porządku technologicznym, tzn. każda operacja O_{jk} ma być wykonana po zakończeniu operacji O_{jk-1}, a przed wykonaniem operacji O_{jk+1}.

Wszystkie zadania mają być wykonane kolejno na maszynach $M_1, M_2, \ldots, M_m$, co oznacza, że operacja O_{jk} ma być wykonana na maszynie M_k.

Wprowadzimy jeszcze następujące oznaczenia:

S_{jk} - termin rozpoczęcia wykonywania zadania J_j na maszynie M_k,

C_{jk} - termin zakończenia wykonywania zadania J_j na maszynie M_k,

P_{jk} - czas trwania wykonywania zadania J_j na maszynie M_k (tzn. operacji O_{jk}).

Na proces wykonywania zadań nałożone są następujące wymagania:

1) Wykonanie każdego zadania $J_j \in J$ nie może się rozpocząć wcześniej niż zadany termin r_j, tzn. $S_{j1} \geqslant r_j$,

2) Wykonanie każdego zadania nie powinno się zakończyć później niż zadany termin d_j.

W praktyce, zadane terminy r_j mogą być terminami dostawy surowców i półfabrykatów służących do realizacji danego zadania, natomiast terminy d_j mogą być terminami określonymi przez zleceniodawcę.

Zagadnienie optymalizacji procesu polega na ustaleniu takiej kolejności wykonywania zadań na poszczególnych maszynach, aby minimalizować maksymalną terminowość

$$L_{max} = \max_j L_j = \max_j [C_{jm} - d_j].$$

Tego rodzaju zagadnienie oznaczamy symbolem $n|m|F, r_j \geqslant 0|L_{max}$. Jeżeli dodatkowo wprowadzimy wymaganie, aby kolejność wykonywania zadań była jednakowa, to otrzymamy permutacyjne zagadnienie taśmowe oznaczane symbolem $n|m|P, r_j \geqslant 0|L_{max}$.

Funkcję celu możemy przedstawić w inny sposób. Jeżeli przyjmiemy $\mathcal{K} = \max_i d_i$, $q_i = \mathcal{K} - d_i$ oraz $L_i' = C_{im} + q_i$, to wtedy mamy $L_{max} = \max_i L_i = L'_{max} + \mathcal{K}$ a zagadnienie $n|m|F, r_i \geqslant 0|L_{max}$ jest równoważne zagadnieniu $n|m|F, r_i \geqslant 0|L'_{max}$. Zagadnienia te możemy przedstawić nieco inaczej. Niech M_0 oraz M_{m+1} będą dodatkowymi maszynami o nieograniczonej przepustowości, tzn. każda z tych maszyn może wykonywać równocześnie n zadań. Teraz każde zadanie J_j składać się będzie z ciągu operacji $\langle O_{j0}, O_{j1}, \ldots, O_{jm+1} \rangle$, które mają być wykonane na maszynach $M_0, M_1, \ldots, M_{m+1}$. Liczbę r_j możemy interpretować jako czas trwania operacji O_{j0} na maszynie M_0, natomiast końcówka q_j jako czas trwania O_{jm+1} na M_{m+1}. Przyjmujemy, że funkcja celu będzie L_{max}, która może być interpretowana jako termin zakończenia wszystkich zadań C_{max} na ostatniej (tzn. M_{m+1}) maszynie (rys.1). Zagadnienie nasze możemy teraz oznaczać przez $n|m+2|F, M_0, M_{m+1}$ - bez ogr.$|C_{max}$ lub przez $n|M|F, r_i \geqslant 0, q_i \geqslant 0|C_{max}$. Zagadnienie $n|M|F, r_i \geqslant 0|L_{max}$ jest NP-zupełne co usprawiedliwia stosowanie metody podziału i ograniczeń do jego rozwiązania.

3. Pewne własności zagadnienia $n|m|P, r_j \geqslant 0, q_j \geqslant 0|C_{max}$

Niech $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ będzie dowolną permutacją liczb $\{1, 2, \ldots, n\}$. Niech $\Pi = \{\pi\}$ będzie zbiorem wszystkich tych permutacji. Każda permutacja $\pi \in \Pi$ określa kolejność wykonywania zadań $J_{\pi(1)}, J_{\pi(2)}, \ldots, J_{\pi(n)}$ na każdej maszynie. Zagadnienie optymalizacji polega na znalezieniu takiej permutacji π^*, dla której mamy

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

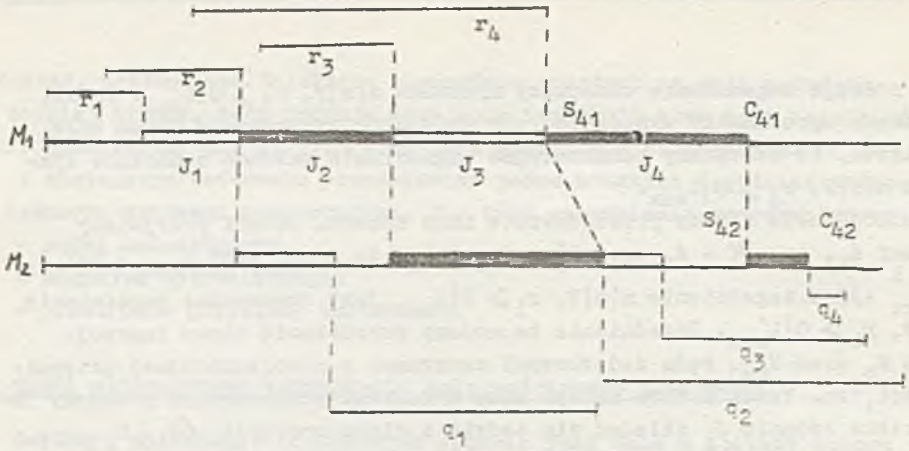
gdzie

$$C_{max}(\pi) = \max_{1 \leqslant j \leqslant n} \left[C_{\pi(j)m} + q_{\pi(j)m} \right] \tag{1}$$

Twierdzenie 1

Niech $\pi \in \Pi$ będzie dowolną permutacją określającą kolejność wykonywania zadań na maszynach, wtedy otrzymujemy

$$C_{max}(\pi) = \max_{1 \leqslant i_1 \leqslant i_2 \leqslant \ldots \leqslant i_{m+1} \leqslant n} \Big[r_{\pi(i_1)} + \sum_{i=i_1}^{i_2} p_{\pi(i)1} + \sum_{i=i_2}^{i_3} p_{\pi(i)2} + \ldots + \sum_{i=i_m}^{i_{m+1}} p_{\pi(i)m} + q_{\pi(i_{m+1})} \Big] \tag{2}$$



Rys.1.

Każdy ciąg liczb całkowitych $\langle i_1, i_2, \ldots, i_{m+1} \rangle$ spełniających

$$1 \leqslant i_1 \leqslant i_2 \leqslant \ldots \leqslant i_{m+1} \leqslant n$$

będziemy nazywać <u>drogą</u> w π. Drogę $\langle u_1^\pi, u_2^\pi, \ldots, u_{m+1}^\pi \rangle$ taką, dla której zachodzi

$$C_{max}(\pi) = r_{\pi(u_1^\pi)} + \sum_{i=u_1^\pi}^{u_2^\pi} p_{\pi(i)1} + \sum_{i=u_2^\pi}^{u_3^\pi} p_{\pi(i)2} + \ldots + \sum_{i=u_m^\pi}^{u_{m+1}^\pi} p_{\pi(im)} + q_{\pi(u_{m+1}^\pi)} \tag{3}$$

będziemy nazywać <u>drogą krytyczną</u>.

Wprowadzimy teraz kilka pojęć i oznaczeń dla każdej permutacji π.

Ciąg zadań $\langle J_{\pi(g)}, J_{\pi(g+1)}, \ldots, J_{\pi(h)} \rangle$, w którym $g = u_k^\pi$ oraz $h = u_{k+1}^\pi$, będziemy nazywać <u>k-tym blokiem</u> w π, $k = 1,2,\ldots,m$. Zadania $J_{\pi(u_k^\pi)}$ oraz $J_{\pi(u_{k+1}^\pi)}$, będziemy nazywać odpowiednio <u>zadaniem pierwszym</u> oraz <u>ostatnim</u> w k-tym bloku w π. Z określeń tych wynika, że zadanie $J_{\pi(u_{k+1}^\pi)}$ jest jednocześnie pierwszym zadaniem w k+1-tym bloku oraz ostatnim w k-tym bloku. Łatwo zauważyć, że w czasie wykonywania zadań k-tego bloku na maszynie M_k, maszyna ta wykonuje te zadania bez przestoju. Dalej niech

$$P_k^\pi = \left\{ \pi(j) \,\middle|\, u_k^\pi \leqslant j < u_{k+1}^\pi \right\}$$

będzie zbiorem wskaźników zadań k-tego bloku w π.

W dalszych rozważaniach, dla uproszczenia oznaczeń, będziemy opuszczać wskaźnik π w oznaczeniach u_k^π, P_k^π. Każda permutacja π ma następującą postać

$$\pi = (\pi(1), \ldots, \pi(u_1), \ldots, \underbrace{\pi(u_k), \overbrace{\pi(u_k+1), \ldots, \pi(u_{k+1}-1)}^{\text{k-ty blok wewnętrzny}}}_{\text{k-ty blok}}, \overbrace{\pi(u_{k+1}), \underbrace{\ldots, \pi(u_{k+2})}_{\text{k+1-ty blok wewnętrzny}}}^{\text{k+1-ty blok}}, \ldots$$

$$\ldots, \pi(n)).$$

Dla przykładu przedstawionego na rys.1 mamy: $u_1 = 1$, $u_2 = 2$, $u_3 = 2$. Ciągi zadań $\langle J_1, J_2 \rangle$ oraz $\langle J_2 \rangle$ są odpowiednio pierwszym i drugim blokiem.

Twierdzenie 2.

Niech $\pi \in \Pi$ będzie dowolną permutacją określającą kolejność wykonywania zadań z drogą krytyczną $\langle u_1, u_2, \ldots, u_{m+1} \rangle$. Jeżeli istnieje permutacja $\beta \in \Pi$ taka, że $C_{max}(\beta) < C_{max}(\pi)$, wtedy w β przynajmniej jedno zadanie dla przynajmniej jednego bloku z π poprzedza pierwsze zadanie (lub występuje za ostatnim zadaniem) tego bloku.

Z twierdzenia 2 otrzymujemy warunek konieczny dla otrzymania permutacji β takiej, że $C_{max}(\beta) < C_{max}(\pi)$. Z twierdzenia tego wynika następujący wniosek:

Wniosek 1.

Dla każdej permutacji $\pi \in \Pi$, jeżeli istnieje permutacja $\beta \in \Pi$ taka, że $C_{max}(\beta) < C_{max}(\pi)$, wtedy permutacja β została otrzymana przez przesunięcie w π przynajmniej jednego zadania z przynajmniej jednego bloku przed pierwsze (lub za ostatnie) zadanie z tego bloku.

4. Schemat przebiegu algorytmu

Zasada działania algorytmu jest oparta na metodzie podziału i ograniczeń z mieszaną strategią podziału. Opis działania przedstawiono poniżej.

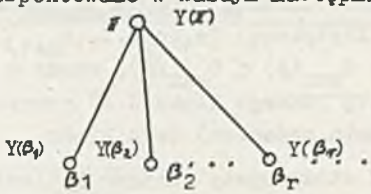
Rozpoczynając od permutacji początkowej $\alpha \in \Pi$ generujemy ciąg permutacji $\pi \in \Pi$. Dla każdej permutacji z ciągu obliczamy wartość $C_{max}(\pi)$, wyznaczamy drogę krytyczną $\langle u_1, u_2, \ldots, u_m \rangle$ oraz określamy bloki zadań (jeżeli permutacja π ma więcej niż jedną drogę krytyczną, to do rozważań wybieramy dowolną z nich). Każdą nową permutację $\beta \in \Pi$ otrzymujemy z pewnej permutacji z ciągu, przez przesunięcie jednego zadania J_j z k-tego bloku w permutacji π. Zadanie J_j z k-tego bloku wewnętrznego jest przesuwane w π na pozycję u_k (lub u_{k+1}), tzn. zadanie to w permutacji β jest wykonywane jako pierwsze (lub ostatnie) z wszystkich zadań k-tego bloku. Pierwsze (lub ostatnie) zadanie k-tego bloku jest przesuwane na u_{k+1} (lub u_k) pozycję w π. Jeżeli dla permutacji β będziemy mogli wykazać, że przez przesuwanie prac w β nie otrzymamy poprawy wyniku C_{max}, wtedy odrzucamy rozpatrywaną permutację β i cofamy się do permutacji π, z której wygenerowaliśmy β. Proces generowania permutacji przedstawimy w postaci drzewa $H = \langle T, R \rangle$. Każdy węzeł w H będzie przedstawiał pewną permutację π, a łuk w H będzie odpowiadał pewnej parze permutacji $\langle \pi, \beta \rangle$, gdzie β została otrzymana przez przesunięcie jednego zadania w π.

Wygenerowanie nowej permutacji (węzła H) jest związane z wyborem pewnego zadania do przesunięcia w π. Wybór ten jest związany z zasadą po-

działu węzłów w H. Jeżeli nową permutację β otrzymaliśmy przez przesunięcie w π zadania J_j z k-tego bloku, to wtedy ustalamy wymagania częściowego porządku

$$W_k^j = \{J_j < J_h | h \in P_k - \{j\}\} \text{ lub } Z_k^j = \{J_h < J_j | h \in P_k - \{j\}\}.$$

Wymagania te muszą być respektowane w każdym następniku względem β w H.



Rys.2.

Generowanie bezpośrednich następników β_r (rys.2) względem π powoduje, że zbiór rozwiązań odpowiadający węzłowi π zostaje rozbity na podzbiory $Y(\beta)_r$ (niekoniecznie rozłączne). Z każdym z tych podzbiorów jest związane zdanie logiczne wskazujące, że pewne zadanie J_j (którego przesunięcie w π wygenerowało β_r) ma być wykonywane jako pierwsze (lub ostatnie) z wszystkich zadań k-tego bloku z π. Zdanie to jest realizowane poprzez ustalenie wymagań W_k^j (lub Z_k^j). Dla ustalenia uwagi, niech β_1 będzie pierwszym następnikiem względem π wygenerowanym poprzez przesunięcie J_j na pierwszą pozycję k-tego bloku z π. Po dokonaniu cofania z β_1 do π otrzymujemy zdanie logiczne dopełniające, że J_j nie może być wykonywana jako pierwsza z rozważnego bloku. Dla zapewnienia rozłączności zbiorów rozwiązań $Y(\beta)_r$ (r = 1,2,...), zdanie to powinno być przestrzegane w innych następnikach generowanych z π, tzn. we wszystkich β_r, $r \geqslant 2$. Jednak jest ono trudne do zrealizowania przy użyciu wymagań częściowego porządkowania. Stąd też jesteśmy zmuszeni zrezygnować z rozłączności podzbiorów rozwiązań $Y(\beta)_r$. Dalej, jeżeli β_2 został wygenerowany poprzez przesunięcie J_j na ostatnią pozycję k-tego bloku, wtedy po cofaniu otrzymujemy kolejne zdanie logiczne, że zadanie J_j nie może być wykonywane jako ostatnie z tego bloku. Łącząc to zdanie z poprzednim otrzymujemy zdanie, że zadanie J_j nie może być wykonywane ani jako pierwsze ani jako ostatnie z k-tego bloku. Powinno ono być przestrzegane we wszystkich β_r, $r \geqslant 3$. Jeżeli wszystkie zadania z k-tego bloku były przesuwane, wtedy otrzymujemy zdanie logiczne dopełniające, że żadne zadanie z k-tego bloku wewnętrznego nie może być wykonywane ani jako pierwsze ani jako ostatnie w k-tym bloku, natomiast zadanie pierwsze tego bloku nie może być wykonywane jako ostatnie, a zadanie ostatnie jako pierwsze. Łącząc logicznie te zdania otrzymujemy, że zadanie pierwsze musi być wykonywane jako pierwsze, a ostatnie jako ostatnie w k-tym bloku. Stąd też możemy ustalić wymagania częściowego porządku

$$\left\{J_{\pi(u_k)} \prec J_h \mid h \in P_k - \{\pi(u_k)\}\right\} \quad \text{oraz} \quad \left\{J_h \prec J_{\pi(u_{k+1})} \mid h \in P_k - \{\pi(u_{k+1})\}\right\}$$

W algorytmach, jeżeli β jest otrzymywana przez przesunięcie zadania z k-tego bloku w π, wtedy dla β ustalamy arbitralnie wymagania częściowego porządku zadań dla pierwszych k-1 bloków. Zapewnia to rozłączność następników (w sensie podzbiorów rozwiązań) generowanych poprzez przesuwanie zadań w różnych blokach. Z powyższych rozważań wynika, że dla każdej permutacji π mamy ustalony zbiór F_π wymagań częściowego porządku. Zbiór ten możemy alternatywnie przedstawić poprzez zbiory wskaźników zadań

$$A_\pi^j = \left\{h \mid (J_j \prec J_h) \in F_\pi\right\} \text{ i } B_\pi^j = \left\{h \mid (J_h \prec J_j) \in F_\pi\right\}, \; j = 1,2,\ldots,n.$$

Zbiór A_π^j (B_π^j) wskazuje, które zadania mają być wykonywane przed (po) zadaniem J_j. Dla zapewnienia rozłączności podzbiorów rozwiązań zadanie J_j nie może być przesuwane, jeżeli $B_\pi^j \cap P_k = \emptyset$ (lub $A_\pi^j \cap P_k = \emptyset$).

Węzeł π w H możemy odrzucić, jeżeli:

(a) dolne ograniczenie ma wartość $C_{max}(\beta)$ dla wszystkich następników β wygenerowanych z π jest nie mniejsze, niż aktualne górne ograniczenie;

(b) żaden nie zawiera więcej, niż jedno zadanie (tzn. $u_1 = u_2 = \ldots = u_{m+1}$);

(c) żadne z zadań nie może być przesuwane na pierwszą lub ostatnią pozycję swojego bloku bez naruszenia ustalonych wymagań częściowego porządku.-Dwa ostatnie przypadki wynikają z twierdzenia 2.

5. Algorytm

Niech korzeniem drzewa rozwiązań H będzie permutacja α, oraz niech $F_\alpha = \emptyset$, $C^* = \infty$.

Niech π oraz F_π, będą odpowiednio permutacją oraz zbiorem wymagań częściowego porządku w π-tej iteracji algorytmu.

Krok 1. Obliczyć dolne ograniczenie LB. Jeżeli LB $\geqslant C^*$, to przejść do kroku 4. W przeciwnym przypadku przejść do kroku 2.

Krok 2. Obliczyć $C_{max}(\pi)$. Jeżeli $C_{max}(\pi) < C^*$, to przyjąć $C^* := C_{max}(\pi)$. Ustalić drogę krytyczną $\langle u_1, u_2, \ldots, u_{m+1}\rangle$. Jeżeli $u_1 = u_2 = \ldots = u_{m+1}$, to przejść do kroku 4. W przeciwnym przypadku przejść do kroku 3.

Krok 3. Wybrać dowolne zadanie J_j i wygenerować nową permutację przez przesunięcie zadania J_j na pozycję u_k (lub u_{k+1}) w π, oraz ustalić odpowiednie wymagania częściowego porządku. Jednocześnie utworzyć w drzewie H nowy węzeł β oraz nowy łuk $\langle\pi,\beta\rangle$. Następnie przejść do kroku 1.

Krok 4. Cofnąć się do poprzednika γ permutacji β w H. Jeżeli należy cofnąć się od korzenia drzewa H, to algorytm kończy się, a permutacja π^* związana z aktualną wartością C^* jest optymalna. W przeciwnym przypadku odrzucić węzeł π. Następnie przejść do kroku 3.

6. Zastosowania

W dyskretnych procesach produkcyjnych zagadnienie przedstawione w poprzednich punktach pracy może wystąpić w procesach walcowania [1], wytopu i odlewu stali [2], produkcji detali oraz w taśmowej produkcji różnych wyrobów.

W niniejszej pracy przedstawimy zastosowanie rozważanego zagadnienia do optymalizacji procesu walcowania w Hucie im. B.Bieruta w Częstochowie [4]. W tym przypadku zadaniami są kęsiska, które mogą być walcowane w jednym z dwóch ciągów technologicznych (marszrut) walcowania. Ciągi te obejmują:

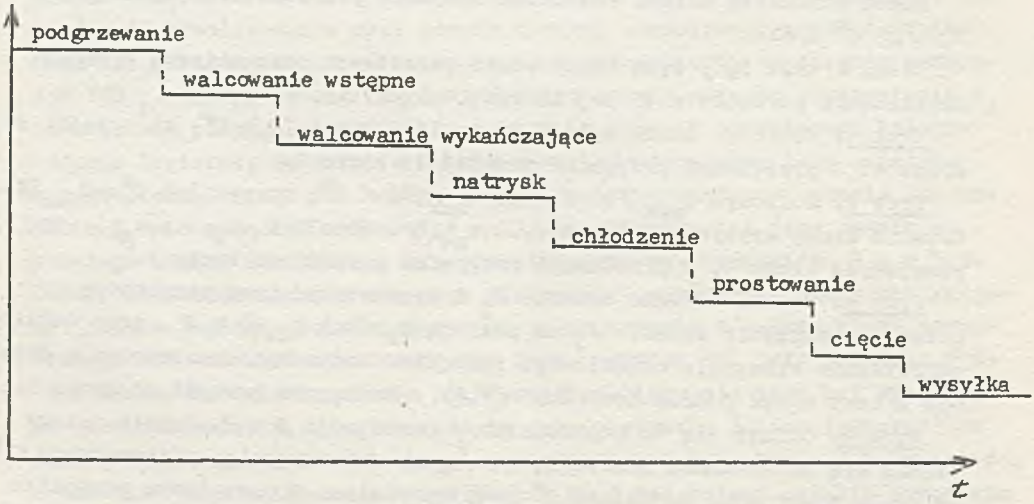
I - podgrzewanie kęsisk, walcowanie wstępne, walcowanie wykańczające, natrysk, chłodzenie, prostowanie I, cięcie, ekspedycja;

II - podgrzewanie kęsisk, walcowanie wstępne, walcowanie wykańczające, natrysk, normalizacja, prostowanie II, trawienie, cięcie, ekspedycja.

Ciągi te można traktować jako podciągi pewnego ogólnego ciągu operacji wykonywanych przy użyciu agregatów: podgrzewanie kęsisk - M_1, walcowanie wstępne - M_2, walcowanie wykańczające - M_3, natrysk - M_4, chłodzenie - - M_5, normalizacja - M_6, prostowanie I - M_7, prostowanie II - M_8, trawienie - M_9, cięcie - M_{10}, ekspedycja - M_{11}.

Zatem, dla każdego kęsiska J_j możemy przyporządkować jednakowy ciąg agregatów, przy czym jeżeli kęsisko J_j nie jest wykonywane przy użyciu agregatu M_k, wtedy należy przyjąć, że dla operacji O_{jk} czas jej wykonywania równy jest zeru.

Na rys.3 przedstawiono przykładowy cyklogram I marszruty.

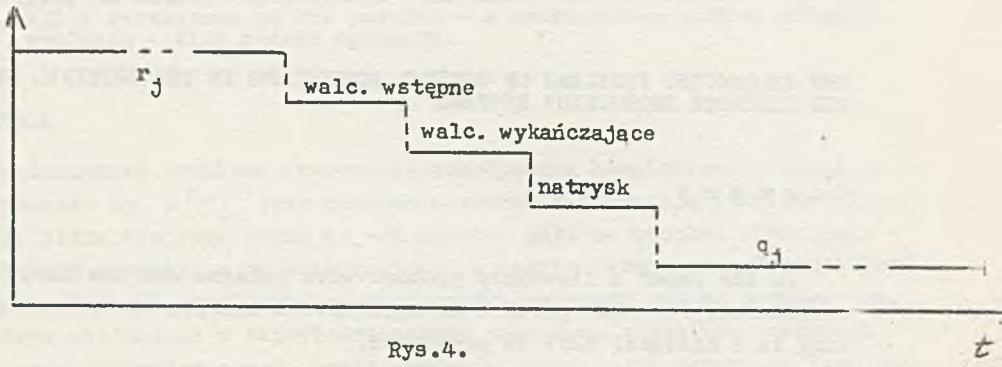


Rys.3.

Czasy wykonywania poszczególnych kęsisk na poszczególnych agregatach są różne dla różnych agregatów. Zależą one od gatunku stali, wymiarów

kęsisk, grubości walcowanej blachy itp.

W świetle wcześniejszych uwag, możemy przyjąć, że gniazdem krytycznym będzie ciąg walcowniczy, zawierający operacje walcowania wstępnego, wykończającego oraz operacje natrysku. Natomiast pozostałych urządzeń nie musimy uwzględniać przy wyznaczaniu kolejności operacji. Czas podgrzewania każdego kęsiska możemy traktować jako najwcześniejszy możliwy termin rozpoczęcia operacji walcowania wstępnego. Natomiast łączny czas trwania wszystkich operacji od chłodzenia aż do ekspedycji możemy traktować jako końcówkę q_j operacji natrysku (rys.4).



Rys.4.

Czyli:

$$r_j = p_{j1}$$

$$q_j = \sum_{k=5}^{11} p_{jk}.$$

Dla rozwiązania przedstawionego zagadnienia użyto algorytmu z poprzedniego punktu pracy. Dla 20 kęsisk czas obliczeń na maszynie cyfrowej ODRA-1325 wynosił 7 sekund, a liczba iteracji - 10.

7. LITERATURA

[1] Bargielski M.: Algorytm zarządzania wydziałem walcowni. Podstawy Sterowania. Tom 1, Z.1, 1971.

[2] Cieśliński J., Gościński A.: Zastosowanie przybliżonego planowania kalendarzowego do szeregowania operacji na odcinku stalownia-walcownia, Archiwum Automatyki i Telemechaniki. Nr 3, 1977.

[3] Grabowski J.; Uogólnione zagadnienia optymalizacji kolejności operacji w dyskretnych systemach produkcyjnych. Prace naukowe Inst.Cybern.Techn. PWr, Monografie nr 50, Wrocław 1979.

[4] Otrembnik B., Gąska A.: Określenie i zebranie danych o przepływie metalu przez Walcownie Blach Grubych. Sprawozdanie Nr 330 Instytutu Metalurgii Żelaza, Gliwice 1971.

ЗАДАЧА ОЧЕРЕДНОСТИ В УПРАВЛЕНИИ КРИТИЧЕСКИХ УЧАСТКОВ В ДИСКРЕТНОЙ СИСТЕМЕ ПРОИЗВОДСТВА

Р е з ю м е

Рассматривается задача очередности связаном с ленточной системой. Даются примеры применения в дискретной производственной системе.

THE SEQUENCING PROBLEMS OF OPTIMAL SCHEDULING IN THE CRITICAL SHOP IN THE DISCRETE PRODUCTION SYSTEMS

S u m m a r y

In the paper a flow-shop problem with release and due dates is considered. For this problem an application example of optimal scheduling in a critical shop is presented.