ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1988
Seria:AUTOMATYKA z.95 Nr kol.971

Eugeniusz Toczyłowski
Instytut Automatyki Politechniki Warszawskiej

HIERARCHICZNY ALGORYTM HARMONOGRAMOWANIA PRODUKCJI WYROBÓW ZALEŻNYCH W GNIEŹDZIE PRODUKCYJNYM PRZY KRYTERIUM KOSZTOWYM [1]

Streszczenie. Przedstawiona została koncepcja systemu wspomagającego harmonogramowanie zadań w ogólnym gnieździe produkcyjnym, będącym ogniwem złożonego systemu produkcji. Algorytm harmonogramowania jest zgodny z systemem planowania produkcji, gdzie w kryterium działania istotną rolę pełnią koszty i zysk. Najważniejszym celem harmonogramowania jest spełnienie pożądanych terminów realizacji zadań. Zatem produkcja w gnieździe jest realizowana akurat na czas bądź też z pewnym wyprzedzeniem. Jako kryterium jakości sterowania produkcją przyjęto minimalizację łącznych kosztów przy spełnieniu ograniczeń. W kosztach produkcji uwzględnia się bezpośrednie koszty produkcji wraz z kosztami przezbrojeń, kary za wykonanie zadań z wyprzedzeniem oraz koszty związane z zamrażaniem środków obrotowych.

1. Wprowadzenie

W pracy rozważany jest problem harmonogramowania produkcji w gnieździe produkcyjnym, będącym ogniwem wielostopniowego systemu produkcyjnego o wytwarzaniu małoseryjnym, w którym realizowane są operacje obróbki i montażu.

Miarą jakości jest kryterium maksymaiizacji zysku, będącego różnicą między wartością sprzedaży a kosztami produkcji. Do kosztów produkcji włączane są bezpośrednie koszty produkcji wraz z kosztami przezbrojeń, koszty magazynowania oraz koszty związane z zamrażaniem środków obrotowych (tym niższe, im krótsze są czasy przejścia zadań produkcyjnych przez system).

Zadania realizowane w gnieździe służą dalszym stopniom produkcyjnym, np. liniom montażu finalnego. Opóźnienie dostawy jednej tylko części powoduje zatrzymanie produkcji w dalszych stopniach produkcyjnych. W najlepszym razie powoduje to wysoki koszt zamrożenia środków obrotowych w wykonanych terminowo częściach. Najczęściej opóźnienie łączy się z wysoką karą za nieterminowość ukończenia wyrobu finalnego. Dlatego też w modelu rozważanym w pracy przyjmuje się, że produkcja w gnieździe może być realizowana bądź akurat na czas,bądź też z pewnym wyprzedzeniem (kara za ewentualną nieterminowość jest bardzo wysoka).

Przyjmuje się, że zadania można pogrupować w zbiory zadań identycznych, wykonywanych według tych samych marszrut. Uwzględniane są koszty i czasy przezbrajania między zadaniami należącymi do różnych grup. Ze względu na te koszty, zadania grupowane są w porcje produkcyjne, których wielkości zależą od wielkości przezbrojeń, stopnia obciążenia stanowisk,i są zmiennymi decyzyjnymi zadania.

[1]Praca częściowo finansowana w ramach problemu R.P.I. 02 w temacie 5.3

Problem decyzyjny obejmuje jednoczesne wyznaczanie wielkości porcji, uszeregowanie operacji na wszystkich maszynach oraz wyznaczenie czasów rozpoczęcia wszystkich operacji. Kosztowa postać funkcji celu powoduje, że problem należy do szczególnie trudnej klasy zadań kolejnościowych (z nieregularnymi funkcjami celu), dla których harmonogramy optymalne nie należą do harmonogramów aktywnych.

W pracy przedstawiono hierarchiczny algorytm wykorzystujący agregację ogólnego zadania i projekcję na podproblemy rozwiązywane w poszczególnych warstwach za pomocą różnorodnych algorytmów optymalizacji służących do rozwiązywania następujących problemów:

- zagregowanego zadania wyboru wielkości porcji i zgrubnego uszeregowania otrzymanych zleceń produkcyjnych,
- przydziału konkretnych zadań do zleceń produkcyjnych oraz wstępnego uszeregowania wszystkich operacji na maszynach,
- uszeregowania wszystkich operacji na maszynach i wyznaczenia czasów rozpoczęcia wszystkich operacji.

Przedstawiono modele zadań optymalizacji dla każdego z podproblemów, omówiono algorytmy rozwiązujące te problemy oraz przeanalizowano problemy koordynacji rozwiązań między poszczególnymi warstwami.

2. Sformułowanie zadania

W gnieździe produkcyjnym znajduje sie określona ilość maszyn realizujących różnorodne operacje. Każde z zadań produkcyjnych polega na wykonaniu ciągu operacji realizowanych na określonych maszynach. Marszruty dla każdego z zadań mogą być różne. Zadania wprowadzane są do systemu w przedziale czasu $[0, T]$.

Niech $J = \{1, \ldots, n\}$ oznacza zbiór zadań do wykonania w gnieździe produkcyjnym w rozpatrywanym przedziale czasu. Zbiór maszyn w gnieździe oznaczamy przez $L = \{1, \ldots, l\}$. Niech $M = \{1, \ldots, m\}$ będzie zbiorem wszystkich operacji wszystkich zadań, natomiast $M_j = \{m_{j-1} + 1, \ldots, m_j\}$, $M_j \subset M$, jest zbiorem operacji zadania j. Zadanie j realizowane jest zgodnie z uporządkowaniem $(m_{j-1} + 1, \ldots, m_j)$ jego operacji. Dla uproszczenia wyrażeń przyjmijmy dla każdego zadania $j, j \in J$ pozorną operację początkową $m = m_{j-1}+1$, odpowiadającą wprowadzeniu zadania do systemu produkcyjnego. Najwcześniejszy czas wprowadzenia zadania j do systemu jest równy a_j. Zbiór operacji pozornych oznaczmy przez M_0 , natomiast zbiór $\{m_j : j \in J\}$ operacji końcowych wszystkich zadań oznaczamy przez M_e. Zakładamy, że operacja $m, m \in M \setminus M_0$, musi być realizowana na ustalonej maszynie l_m. Stąd dla dowolnej operacji $m, m \in M \setminus M_0$, mamy ustalone zadanie j_m oraz maszynę l_m odpowiadające tej operacji. Czas wykonania operacji m jest równy p_m. Dla operacji m wprowadzamy zmienną decyzyjną t_m będącą jej czasem zakończenia. Zmienne $t_m, m \in M$ pozwalają na jednoznaczne określenie harmonogramu. Przyjmujemy, że po zakończeniu operacji m rozpoczęcie następnej operacji n na tej samej maszynie może wymagać niezerowego czasu przygotowawczego s_{mn} (czasu przezbrojenia). Relacje poprzedzania operacji tych samych zadań są reprezentowane przez zbiór $R \subset M \times M$ uporządkowanych par operacji. Zachodzi $(m,n) \in R$, jeżeli operacja n bezpośrednio poprzedza operację m, a więc R reprezentuje relację

$$t_n - t_m \geq p_n \qquad (m,n) \in R. \tag{2.1}$$

Niech E będzie zbiorem nieuporządkowanych par tych operacji, które są wykonywane na tej samej maszynie. Warunek, że maszyna w danej chwili wykonuje co najwyżej jedną operację, można wyrazić przez warunki dysjunktywne

$$\begin{aligned} t_n - t_m &\geq p_n + s_{mn} \\ &\textit{albo} \\ t_m - t_n &\geq p_m + s_{nm} \qquad \{m,n\} \in E \end{aligned} \tag{2.2}$$

Przyjmujemy, że zadania można pogrupować w zbiór K rodzin N_k zadań mających identyczne marszruty, zbliżone czasy operacji i współczynniki zużycia zasobów oraz koszty związane z zamrażaniem środków obrotowych i koszty magazynowania. Przyjmujemy też, że zadania z tej samej rodziny dzielą główne czasy przygotowawczo-zakończeniowe. Różnice pomiędzy zadaniami należącymi do tej samej grupy, np. różnice w indywidualnych czasach i kosztach operacji, powinny być pomijalne. Głównym czynnikiem różnicującym zadania tej samej grupy są różne terminy gotowości zadań oraz pożądane terminy zakończenia zadań. Również operacje $m, m \in M$ można pogrupować w zbiory operacji podobnych $M^{\mu}, \mu \in \mathcal{M}$ wymagających tych samych zasobów, mających podobne czasy wykonania i podobne przezbrojenia. Dla uproszczenia zakładamy, że wszystkie operacje typu μ mają identyczne czasy i koszty

$$p_m = P_{\mu} \quad \text{dla } m \in M^{\mu}, \mu \in \mathcal{M}, \tag{2.3}$$
$$q_m = c_{\mu} \quad \text{dla } m \in M^{\mu} \setminus M_e, \mu \in \mathcal{M}, \tag{2.4}$$
$$h_m = h_k^{+} \quad \text{dla } m \in M_j \cap M_e, j \in N_k \tag{2.5}$$

gdzie: P_{μ}, c_{μ} jest czasem i kosztem dla operacji typu μ, h_k^{+} jest karą za wcześniejsze wykonanie zadania typu k o jednostkę czasu. Zadanie z grupy k składa się z ciągu n_k operacji następujących typów $(\mu_{k1}, \mu_{k2}, \ldots, \mu_{kn_k})$.

3. Zagregowane zadanie harmonogramowania

W celu konstrukcji zagregowanego zadania harmonogramowania przyjmujemy, że horyzont harmonogramowania $[0,T]$ został podzielony na T okresów o długości $\Delta_1, \Delta_2, \ldots, \Delta_T$, gdzie $T = \sum_{t=1}^{T} \Delta_t$. Zakładamy, że pożądane terminy zakończenia zadań odpowiadają chwilom zakończenia tych okresów. Zadanie $j, j \in N_k$ ukończone w okresie t wymaga ciągu operacji; niektóre realizowane są w okresie t, niektóre w okresach poprzedzających. Operacje przyczyniają się do produkcyjnego czasu przejścia zadania w systemie przez dodanie czasów wykonania operacji, czasów przygotowawczo-zakończeniowych, czasu transportu między maszynami i czasów oczekiwania w kolejkach. Najbardziej zmiennym składnikiem tych czasów są okresy oczekiwania w kolejkach, zależące od takich czynników, jak obciążenia maszyn, reguły szeregowania zadań w kolejkach, itp. Czynniki te są trudne do uwzględnienia na szczeblu harmonogramowania zagregowanego. Można je uwzględnić w sposób zagregowany przez dodanie średnich zagregowanych czasów oczekiwania w kolejkach. Czasy te są zmienne, zależne od chwilowych obciążeń maszyn.

W teorii kolejek obserwuje się [2,8,13,14], że średnie czasy oczekiwania w kolejkach zależą od proporcji między czasem zajętości i czasem postoju maszyn. Jeżeli maszyny są obciążone w pewnych okresach na 100%, odpowiednie czasy oczekiwania w kolejkach są długie. Na odwrót, jeżeli maszyna jest planowo obciążona na powiedzmy 75%, to w rezultacie średni czas oczekiwania w kolejkach jest stosunkowo krótki i bardziej stabilny do oszacowania. Tym samym można spodziewać się, że średnie czasy oczekiwań w kolejkach mogą być szacowane z dopuszczalną dokładnością tylko wtedy, gdy maszyny są właściwie obciążane w rezultacie rozwiązania odpowiedniego zadania harmonogramowania, uwzględniającego obciążenia maszyn i inne zasoby w rozpatrywanym horyzoncie. Takim zadaniem jest rozpatrywany na najwyższym poziomie problem zagregowanego harmonogramowania. Z drugiej strony oszacowanie czasów przejścia zadań w systemie jest wymagane do konstrukcji poprawnego modelu zagregowanego. Mówiąc precyzyjniej, w modelu zagregowanym należy jedynie oszacować zużycie zasobów w poszczególnych okresach czasu, co jest zadaniem łatwiejszym, gdyż uśrednionym. Tak więc do prawidłowego działania hierarchicznego algorytmu harmonogramowania wymagane jest zanurzenie algorytmów realizujących zadania każdej warstwy w strukturę hybrydowego systemu wspomagania decyzji, w którym różnorodne problemy decyzyjne rozwiązywane są za pomocą metod optymalizacji dyskretnej, technik złożonych systemów oraz reguł inteligentnego wnioskowania, połączone w hierarchiczny system wspomagania decyzji.

Niech τ_{kt} będzie czasem przejścia w okresie t dla zadań $j, j \in N_k$, oszacowanym dla założonych, maksymalnych obciążeń maszyn. Tak więc zadania $j, j \in N_k$ kończone w okresie t mają operacje wykonywane w okresach $t - \tau_{kt} + 1, t + \tau_{kt} + 2, \ldots, t$, a tym samym w tych okresach zużywa się pewne zasoby.

Dzięki wyborowi odpowiednio długich okresów Δ_t, czas przejścia zadań przez system może być rzędu $1 - 3$ okresów. Niech

$$\tau_k = max_t\{t | t - \tau_{kt} < 0\}$$

oznacza czas przejścia tych zadań grupy k, które muszą być wprowadzone do systemu przed pierwszym okresem harmonogramowania, aby mogły być ukończone przed okresem τ_k.

W zadaniu harmonogramowania zagregowanego często nie jest możliwe uwzględnienie wszystkich szczegółowych ograniczeń, takich jak dostępność pojedynczej maszyny w ustalonym okresie. Możliwe jest jednak wprowadzenie ograniczeń zagregowanych, reprezentujących grupy funkcjonalnie podobnych zasobów, takich jak grupy maszyn, narzędzi, palet, siły roboczej, itp. Przyjmiemy, że liczba zasobów uwzględnionych w modelu zagregowanym jest równa R. Celem harmonogramowania na szczeblu nadrzędnym jest pogrupowanie zadań podobnych, należących do tych samych grup, w porcje zadań wykonywanych razem oraz ustalenie terminów wprowadzenia porcji zadań do systemu dzięki odpowiedniemu przegrupowaniu zadań należących do tych samych grup w okresach $1,\ldots,T$. Zadanie programowania dyskretnego odpowiadające powyższemu problemowi jest postaci

Problem AH:

$$\text{minimalizuj } \sum_{t=1}^{T}\Big(\sum_{k=1}^{K}(h_{kt}^{+}I_{kt}^{+}) + \sum_{\mu=1}^{M}(s_{\mu}v_{\mu t} + c_{\mu}o_{\mu t})\Big) \tag{3.1}$$

przy ograniczeniach

$$I_{k,t-1}^{+} + x_{kt} - I_{kt}^{+} = d_{kt} \qquad k = 1,\ldots,K; t = 1,\ldots,T \tag{3.2}$$

$$o_{\mu t} = \sum_{(k,\tau)\in KT_{\mu t}} b_{\mu k\tau}^{t} x_{k,t+\tau} \qquad \mu = 1,\ldots,M; t = 1,\ldots,T \tag{3.3}$$

$$\sum_{\mu=1}^{M}(E_{\mu r}v_{\mu t} + P_{\mu r}o_{\mu t}) \le B_{rt} \qquad r = 1,\ldots,R; t = 1,\ldots,T \tag{3.4}$$

$$I_{k0}^{+} = I_{kT}^{+} = 0 \qquad k = 1,\ldots,K \tag{3.5}$$

$$I_{kt}^{+}, x_{kt} \ge 0,\ \text{całkowite} \qquad \forall k,t \tag{3.6}$$

$$0 \le o_{\mu t} \le M_{\mu t}v_{\mu t},\quad v_{\mu t} \in \{0,1\} \qquad \forall \mu,t \tag{3.7}$$

W zadaniu występują następujące zmienne:

- x_{kt} - liczba zadań typu k ukończonych pod koniec okresu $t, t = \tau_k, \tau_{k+1},\ldots,T$, gdzie τ_k jest czasem przejścia początkowych zadań typu k,
- I_{kt}^{+} - liczba zrealizowanych z wyprzedzeniem zadań typu k w okresie t (mających pożądany termin realizacji późniejszy niż t i ukończonych najpóźniej w okresie t),
- $o_{\mu t}$ - liczba operacji typu μ wykonywanych w okresie t,
- $v_{\mu t}$ - zmienna binarna odpowiadająca wznowieniu operacji typu μ w okresie t równa 1 wtedy i tylko wtedy gdy $o_{\mu t} > 0$.

Ponieważ zadania wprowadzone do systemu przed chwilą zerową muszą być znane, zakłada się, że $x_{kt}, t = 1,\ldots,\tau_k - 1$ są znane. Pozostałe parametry zadania (wszystkie nieujemne):

- M - liczba typów operacji,
- h_{kt}^{+} - kara za wcześniejsze o okres wykonanie zadania typu k w okresie t, $h_{kt}^{+} = h_k^{+}\Delta_{t+1}$,
- d_{kt} - liczba zadań typu k mających pożądany termin ukończenia w okresie t,
- s_{μ}, c_{μ} - koszt wznowienia i jednostkowy koszt wykonania operacji μ,
- $b_{\mu k\tau}^{t}$ - liczba operacji typu μ wymaganych w okresie t w celu ukończenia zadania typu k pod koniec okresu $t + \tau$,

$KT_{\mu t}$ - $\{ (k,r)$: operacja μ jest wymagana w okresie t, aby ukończyć zadanie typu k w okresie $t+r\}$,

$M_{\mu t}$ - liczba nie mniejsza niż maksymalna liczba operacji typu μ w okresie t,

B_{rt} - łączna dostępność zasobu r w okresie t,

$E_{\mu r}, P_{\mu r}$ - liczba jednostek zasobu r wymagana przez jedno przezbrojenie i wykonanie jednej operacji typu μ,

Najczęściej operacje różnych zadań są różnych typów, a zatem macierze $B_r = [b_{\mu k r}], \mu \in M$ mają co najwyżej jedną niezerową pozycję w każdym wierszu. Jeżeli w pewnym wierszu występuje więcej elementów niezerowych, oznacza to, że wymagane są podobne operacje dla zadań różnych typów. Jeżeli porcje różnych zadań nie mogą być wykonywane bezpośrednio po sobie, należy w modelu wprowadzić dwa różne typy operacji podobnych.

Jeżeli czasy przygotowawcze $E_{\mu r t}$, koszty $s_{\mu t}$ oraz warunki całkowitoliczbowości na zmienne x_{kt}, I_{kt} są pomijalne, to zadanie AH jest łatwym zadaniem programowania liniowego. Metody rozwiązania zadania AH dla przypadku dyskretnego, oparte o relaksacje Lagrange'a oraz transformacje sieciowe, opisano w[16].

4. Dezagregacja harmonogramu zagregowanego

Głównym wynikiem zadania zagregowanego jest zbiór porcji $x_{kt}, k = 1,\ldots,K; t = 1,\ldots,T$. Oznacza to, że dla grupy zadań podobnych określone są tylko liczby zadań w porcjach realizowanych w różnych okresach. W celu uzyskania przydziału konkretnych zadań do porcji produkcyjnych oraz uszeregowania operacji na wszystkich maszynach, wielkości te należy zdezagregować, a następnie wyznaczyć początkowy wybór zbioru łuków dysjunktywnych $S, S \subset E$, określający pełne uszeregowanie wszystkich detalooperacji.

W szczegółowym harmonogramie każde zadanie może być rozpatrywane jako ciąg bloków operacji wykonywanych na kolejnych maszynach. Operacje przyległe wykonywane na tej samej maszynie tworzą bloki sąsiadujących ze sobą operacji porcji produkcyjnych odpowiadających zadaniom tych samych typów. W ogólności porcje produkcyjne i transportowe na różnych maszynach mogą być różne – przetwarzanie bloku operacji na pewnej maszynie może się rozpocząć po wykonaniu części operacji bloku (na przykład odpowiadającej jednej palecie) na poprzedniej maszynie. Początkowe uszeregowanie szczegółowe otrzymane w wyniku dezagregacji harmonogramu zagregowanego daje porcje produkcyjne identyczne dla wszystkich maszyn wykonujących operacje zadań danego typu. Algorytm harmonogramowania szczegółowego pozwala jednak na rozbijanie bloków zadań, w rezultacie w dalszych krokach można otrzymać zupełnie dowolny harmonogram szczegółowy. Dezagregacja harmonogramu zagregowanego jest realizowana w trzech krokach:

(i) określenie dla wszystkich zadań ustalonej grupy $k, k = 1,\ldots,K$, które zadania są przydzielone do której porcji produkcyjnej,

(ii) uszeregowanie zadań jednego typu przyporządkowanych do tej samej porcji produkcyjnej,

(iii) uszeregowanie porcji produkcyjnych w każdym okresie $\Delta_t, t = 1, \ldots, T$.

Przydział zadań do porcji produkcyjnych. Rozważmy rodzinę zadań N_k. Zadania te muszą być przyporządkowane do porcji $x_{kt}, t \in T_k$, gdzie $T_k = \{t | x_{kt} \neq 0, 1 \leq t \leq T\}$. Przypuśćmy, że zadania $j, j \in N_k$, mają identyczne współczynniki kosztowe i produkcyjne oraz że uszeregowania terminów gotowości zadań do wykonania i pożądanych terminów zakończenia są sobie równe. Głównym czynnikiem rozróżniającym zadania są zatem pożądane terminy zakończenia zadań $d_j, j \in N_k$, niekoniecznie odpowiadające chwilom zakończenia zagregowanych okresów. W powyższym przypadku optymalny przydział zadań produkcyjnych do porcji produkcyjnych jest trywialny — należy uporządkować zadania zgodnie z porządkiem oczekiwanych terminów zakończenia zadań.

W ogólnym przypadku współczynniki kosztowe i produkcyjne zadań tej samej grupy są różne. Ponadto można też uwzględnić przypadki, gdy funkcje kosztów mogą być nieliniowe. Nawet w ogólnym przypadku nie jest trudne oszacowanie dla ustalonych j, t łącznych kosztów związanych z wykonywaniem zadania j w sytuacji, gdy zadanie j jest przydzielone do wykonania w okresie t, włączając do tego bezpośrednie koszty wykonywania wszystkich operacji zadania j na maszynach. Niech b_{jt} oznacza łączny koszt związany z zadaniem j ukończonym w okresie t. Wprowadźmy zmienne binarne $v_{jt}, j \in N_k, t \in T_k$, v_{jt} równe 1 wtedy i tylko wtedy, gdy zadanie j jest przydzielone do porcji wykonanej w okresie t. Zadanie j nie może być przyporządkowane do porcji x_{kt}, jeśli najwcześniejszy termin gotowości zadania a_j nie pozwala na ukończenie zadania j w okresie t

$$v_{jt} = 0 \ \text{ if } \ t - \tau_{kt} \leq a_j \quad j \in N_k, t \in T_k. \tag{4.1}$$

Zadanie dezagregacji dla k-tej grupy zadań jest problemem transportowym postaci

$$\text{minimalizuj} \ \sum_{t \in T_k} \sum_{j \in N_k} b_{jk} v_{jk} \tag{4.2}$$

przy warunku, że każde zadanie jest przydzielone do jednej porcji produkcyjnej

$$\sum_{t \in T_k} v_{jt} = 1 \qquad j \in N_k, \tag{4.3}$$

oraz każda porcja ma określoną liczbę zadań

$$\sum_{j \in N_k} v_{jt} = x_{kt} \quad t \in T_k. \tag{4.4}$$

Zadania przydzielone do porcji produkcyjnych mogą być uszeregowane przybliżoną metodą opisaną w rozdziale 5 wykorzystującą priorytety kosztów marginalnych. Podobnie mogą być uszeregowane porcje produkcyjne przydzielone do realizacji w każdym okresie.

5. Szczegółowe harmonogramowanie

Zadanie harmonogramowania szczegółowego, w którym rozważa się terminy ukończenia wszystkich detalooperacji, można zapisać następująco:

Zadanie HS

$$\text{maksymalizuj } Q = \sum_{m \in M_e} (\bar{h}_m t_m) + \sum_{m \in M \setminus M_e} q_m t_m \tag{5.1}$$

przy spełnieniu warunków

$$t_n - t_m \geq p_n \qquad (m,n) \in R \tag{5.2}$$

$$\begin{aligned} t_n - t_m &\geq p_n + s_{mn} \quad \text{albo} \\ t_m - t_n &\geq p_m + s_{nm} \quad \{m,n\} \in E \end{aligned} \tag{5.3}$$

$$t_m \geq 0 \qquad m \in M \setminus M_0 \tag{5.4}$$

$$t_m \geq a_{j_m} \qquad m \in M_0 \tag{5.5}$$

gdzie: q_m jest kosztem zamrożenia w jednostce czasu wartości związanej z wykonaniem operacji m na maszynie l_m, h_j jest kosztem zamrożenia i magazynowania gotowego wyrobu j, natomiast $\bar{h}_m = h_j + q_m$ dla $m \in M_e, j = j_m$. Zadanie *HS* najwygodniej jest przedstawiać na grafie dysjunktywnym $G = (M, R, E)$, gdzie M jest zbiorem wierzchołków, przy czym każdy wierzchołek odpowiada operacji; R jest zbiorem skierowanych łuków odpowiadających relacjom między opracjami tych samych zadań; E jest zbiorem nieskierowanych łuków, przy czym każdy odpowiada parze operacji wykonywanych na tej samej maszynie. Graf $G_0 = (M, R)$, otrzymany z G przez pominięcie łuków nieskierowanych, ma strukturę rozłącznych ścieżek odpowiadających zadaniom ze zbioru J.

Rozwiązywanie powyższego zadania wymaga w szczególności zorientowania łuków nieskierowanych. Graf G wygodnie jest reprezentować w postaci grafu dyzjunktywnego (patrz [15,1]), w którym każdy nieskierowany łuk $\{m,n\} \in E$ jest zastąpiony przez dysjunktywną parę (m,n) i (n,m), z której musi być wybrany tylko jeden łuk. Zbiór E zawiera zatem pary skierowanych łuków (m,n) i (n,m).

Selekcja jednego łuku z każdej pary $\{(n,m),(m,n)\} \subset E$ jest nazywana *zupełną* selekcją łuków dysjunktywnych. Dla dowolnej selekcji $S, S \subset E$ takiej, że graf $G_S = (M, R \cup S)$ jest acykliczny, zadanie *HS* jest redukowane do zadania programowania liniowego.

Zadanie LHS:

$$\text{maksymalizuj } Q = \sum_{m \in M_e} (\bar{h}_m t_m) + \sum_{m \in M \setminus M_e} q_m t_m \tag{5.6}$$

przy spełnieniu warunków

$$t_n - t_m \geq p_n \qquad (m,n) \in R \tag{5.7}$$

$$t_n - t_m \geq p_n + s_{mn} \qquad m,n \in S \tag{5.8}$$

$$t_m \geq 0 \qquad m \in M \setminus M_0 \tag{5.9}$$

$$t_m \geq a_{j_m} \qquad m \in M_0 \tag{5.10}$$

Zadanie *LHS* ma wygodną strukturę (zadanie dualne jest zadaniem sieciowym). Dla takich zadań można wykorzystać efektywne techniki programowania sieciowego [9].

Dla ustalonej selekcji zupełnej S wyznaczenie harmonogramu jest zatem stosunkowo proste. Problem znalezienia najlepszego uszeregowania $S, S \subset E$ należy do najtrudniejszych zadań optymalizacji dyskretnej. Nawet w przypadku prostszych zadań szeregowania z regularnymi funkcjami celu, do rozwiązywania zadań o realistycznych wymiarach najczęściej stosuje się algorytmy przybliżone [5,4]. Dla zadania z kryterium kosztowym w chwili obecnej realne jest poszukiwanie możliwie najlepszego algorytmu heurystycznego. W tej pracy proponujemy metodę wykorzystującą ideę priorytetów kosztowych z algorytmu *Sched-Star* [12]. Podstawowe różnice polegają na tym, że: (i) rozważany jest inny model zadania harmonogramowania; (ii) poszukiwane jest jedynie uszeregowanie, a nie harmonogram szczegółowy; (iii)rozwiązywanie zadania liniowego *LHS* pozwala na eliminację dużej części heurystycznych obliczeń, np. obliczanie współczynników wrażliwości; (iv) początkowe uszeregowanie operacji jest otrzymywane w wyniku dezagregacji rozwiązania zadania zagregowanego.

W algorytmie szeregowania operacji wykorzystywane są informacje otrzymywane w wyniku rozwiązywania zadania *LHS*. Dwie operacje wykonywane na tej samej maszynie bezpośrednio po sobie bez luzu czasowego nazywane są operacjami *przyległymi*. Maksymalne grupy operacji przyległych nazywane są blokami. W ogólności operacje jednego bloku należą do wielu zadań. Porcja produkcyjna przetwarzana na maszynie zawiera operacje przyległe należące do tej samej grupy zadań. Stąd każdy blok można traktować jako ciąg przyległych porcji produkcyjnych na jednej maszynie.

Właściwości bloków są wykorzystywane w algorytmach szeregowania zagadnień przepływowych i ogólnych gniazdowych z regularnymi funkcjami celu [7]. W zadaniach szeregowania z kryteriami kosztowymi analogiczne właściwości bloków nie zachodzą. W heurystycznym algorytmie szeregowania wyznaczenie bloków pozwala na obliczenie odpowiednich priorytetów operacji. Dla ustalonej selekcji $S, S \subset E$ w pewnym lokalnym otoczeniu rozwiązania S poszukiwana jest nowa selekcja S^* dająca poprawę wartości funkcji celu. Algorytm poszukiwań najlepszego rozwiązania może być oparty o klasyczny schemat poszukiwania minimum lokalnego, metodę symulacyjnego wyżarzania [11] lub poszukiwanie technikami sztucznej inteligencji [3,7]. Rozwiązanie S^* orzymywane jest z S za pomocą:

- zamiany dwóch przyległych operacji należących do tych samych typów zadań,
- zamiany przylegających porcji produkcyjnych.

W przypadku wielokrotnej zamiany operacji i porcji z jednoczesnym rozwiązywaniem nowych zadań *LHS* można uzyskać harmonogram różniący się istotnie w stosunku do harmonogramu początkowego. W szczególności zmieniają się bloki, porcje zadań podobnych na różnych maszynach i mogą być wprowadzane dodatkowe przezbrojenia w okresach postoju maszyn.

Oznaczmy przez $U_m(t_m)$ łączny marginalny koszt opóźnienia wykonywania operacji m w chwili t_m. Koszty $U_m, m \in M$ można obliczyć z rozwiązania zadania *LHS* jako współczynniki wrażliwości funkcji celu względem czasu operacji $p_m, m \in M$. Jeżeli

m, n są przyległymi operacjami tej samej porcji, wtedy zysk E_{mn} z zamiany operacji jest równy

$$E_{mn} = p_m U_n(t_n) - p_n U_m(t_m) = p_m p_n \left(\frac{U_n(t_n)}{p_n} - \frac{U_m(t_m)}{p_m} \right) \tag{5.11}$$

Ze wzoru (5.11) wynika, że operacje można uszeregować zgodnie z priorytetem marginalnego zysku

$$T_m(t_m) = \frac{U_m(t_m)}{p_m} \tag{5.12}$$

Analogicznie można uszeregować porcje operacji na pojedynczej maszynie odpowiadające zadaniom tej samej grupy . Niech porcja λ wymaga realizacji zbioru M^λ operacji na pewnej maszynie. Po zsumowaniu czasów wykonania

$$P^\lambda = \sum_{m \in M^\lambda} p_m,$$

oraz kosztów marginalnych

$$U^\lambda = \sum_{m \in M^\lambda} U_m$$

każda porcja może być traktowana jako operacja zagregowana, dla której priorytet marginalnego zysku wynosi

$$T^\lambda = \frac{U^\lambda}{P^\lambda}. \tag{5.13}$$

6. Adaptacja

W zadaniu harmonogramowania na górnym poziomie ograniczenia zasobowe oraz relacje poprzedzania mogą być reprezentowane jedynie w sposób zagregowany. W zadaniu tym należy przewidzieć dla dowolnej grupy zadań $j, j \in N_k$ kończonych w okresie t, które operacje wykorzystują pewne zagregowane zasoby w okresach $t - \tau_{kt} + 1, t + \tau_{kt} + 2, \ldots, t$. Jeżeli predykcja nie jest dokładna, otrzymany harmonogram może być niedopuszczalny. Rozwiązanie zadania szczegółowego harmonogramowania pozwala jednak na uzyskanie informacji, która może być wykorzystana na górnym poziomie do adaptacji czasów przejścia zadań w systemie oraz współczynników zużycia zasobów w modelu zagregowanym.

Dla uniknięcia niestabilności iteracyjnego algorytmu adaptacji, współczynniki nowego modelu mogą być wyznaczone jako wartości z poprzedniej iteracji plus wygładzona korekta współczynników wynikająca z porównania starych i nowych wartości adaptowanych parametrów. Jakkolwiek trudno tutaj o jakiekolwiek analityczne gwarancje zbieżności, doświadczenia z podobnym schematem iteracyjnym do adaptacji harmonogramów [12] pozwalały na uzyskanie znacznie poprawionego rozwiązania w stosunkowo niewielkiej liczbie iteracji.

Literatura

[1] Balas,E., 1979, Disjunctive Programming ,*Annals of Discrete Mathematics* 5,3-51.

[2] Cox, D.R. and Smith, W.L., *Queues*, Chapman and Hall, 1971.

[3] Fox, M.S., S. Smith, 1984, ISIS - a Knowledge-Based System for Factory Scheduling', *Expert Systems* Vol.1, No. 1, 25 - 49.

[4] French, S., 1982, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Horwood, Chichester.

[5] Gere, W.S., 1966, Heuristics in Job Shop Scheduling , *Management Science*, 13, 164-180.

[6] Grabowski,J. *Uogólnione zagadnienia optymalizacji kolejnościowej w dyskretnych systemach produkcyjnych* Monografie ICT PWr, Monografie 9, Wyd. Pol. Wroc., Wrocław 1979

[7] Grant, T.Y., 1986, Lessons for O.R. from A.I.: A Scheduling Case Study , *J.Opl.Res.Soc.*, vol.37, 41-58.

[8] Hillier, F. and Yu, P., *Queueing Tables and Graphs*. (North Holland, Amsterdam 1981).

[9] Kennington, J.F., R.V. Helgason, *Algorithms for Network Programming*, (John Willey & Sons, New York 1981).

[10] Kelley, J.E., The Cutting Plane Method for Solving Convex Programs , *SIAM Journal*, 8, 703-712.

[11] Kirpatrick, S., C.D. Gellat, Jr, M.P. Vecchi, Optimization by Simulated Annealing , *Science* 220 (13 May 1983), 671-680.

[12] Morton, T.E., S.R.Lawrence, S. Rajagopalan, S. Kekre, SCHED -STAR: A Price Based Schop Scheduling Module , Working Paper, July 1987, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh (to appear in Journal of Manufacturing and Operations Management).

[13] Page, E., Tables of waiting times for M/M/n, M/D/n and D/M/n, and their use to give approximate waiting times in more general queues. *Journal of the Operational Research Society*, 33, 453-473, 1982.

[14] Page, E., Queueing theory in OR. *Operational Research Series*, ed. K. Haley, 1972.

[15] Roy, B. and B. Sussman, 1964, Les problemes dórdonnancement avec contraintes disjonctives , SEMA, Note DS 9 bis.

[16] E. Toczyłowski, Two-level approach to solving large scale dynamic job-shop scheduling problems with groups of similar jobs. Proc. IXth International Conference on Production Research, Cincinnati, OH., 17-20 August, 1987, Mital (ed.), North Holland , pp. 2746-2752.

[17] Zangwill, W.I., 1969, A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot Size Production System — a Network Approach , *Management Science*, 15, 506-527.

Recenzent: Doc.dr hab.inż.M.Zaborowski

Wpłynęło do Redakcji do 1988-04-30.

ИЕРАРХИЧЕСКИЙ АЛГОРИТМ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА ЗАВИСИМЫХ ИЗДЕЛИЙ В ПРОИЗВОДСТВЕННОМ ГНЕЗДЕ СО СТОИМОСТНЫМ КРИТЕРИЕМ

Р е з ю м е

Дана концепция вспомагающей системы календарного планирования задач для общего производственного гнезда , входящего в состав сложной производственной системы. В качестве критерия качества управления производство принята минимизация общей стоимости с учётом ограничений. В стоимости производства учитываются непосредственная стоимость вместе со стоимостью переналадок , штрафы из-за опережений выполнения операций и другие.

HIERARCHICAL OPTIMISATION ALGORITHM FOR JOB-SHOP SCHEDULING WITH COST CRITERION AND INFINITE TARDINESS PENALTIES

Summary

We present the algorithms of the hierarchical decision support system for scheduling in a general job-shop cell which is a a part of a multistage production system. The scheduling model considered in the paper is compatible with the Master Planning System in which a profit-based planning objective is taken into account. The primary objective of scheduling is to meet due dates, therefore jobs processed in the cell must be completed just-in-time, or earlier. The secondary objective is to minimise total production costs, subject to resource and precedence constraints. The total costs include direct production costs, set-up costs, earliness penalties and holding costs of the work-in-process.