

Eugeniusz Toczyłowski
Krzysztof Pieńkosz

Instytut Automatyki Politechniki Warszawskiej

MINIMALNOKOSZTOWE HARMONOGRAMOWANIE PRODUKCJI W ELASTYCZNYM SYSTEMIE PRODUKCYJNYM Z UWZGLĘDNIENIEM ROZDZIAŁU ZASOBÓW ŻUŻYWALNYCH¹

MINIMUM-COST PRODUCTION SCHEDULING AND RESOURCE ALLOCATION IN FLEXIBLE MANUFACTURING SYSTEMS

MINIMALKOSTENFÄHIGE HARMONOGRAMMBILDUNG IN DER FLEXIBLE PRODUKTION MIT DER TRANSPORT- UND LAGERBEGRENZUNGEN SOWIE VERSCHLEIßRESSOURCEN

Streszczenie: W pracy jest przedstawiony algorytm dla problemu szczegółowego harmonogramowania produkcji ESP, w którym czasy wykonywania niektórych operacji technologicznych mogą być zmieniane w wyniku rozdziału zasobów zużywalnych, natomiast zasoby odnawialne systemu są modelowane w sposób zunifikowany jako tzw. procesory uogólnione.

Summary: An algorithm is presented for the detailed production scheduling problem in a FMS cell, where processing rates of some operations may be reduced by allocations of some consumable resources, whereas the renewable resources (work-centers, buffers and material handling systems) are modelled in a unified way as *general processors*.

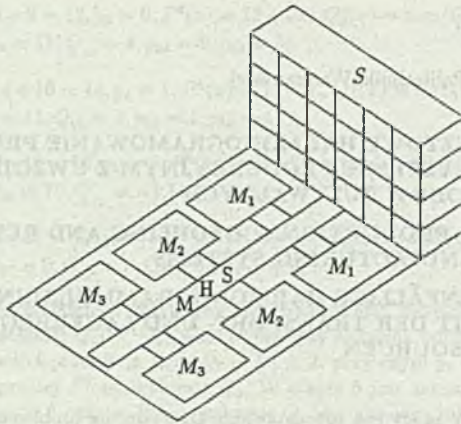
Zusammenfassung: Im Beitrag wird der zweigeschossige Algorithmus für die ausführliche Harmonogrammbildung in der flexiblen Produktion gezeigt. In diesem Produktionssystem die Geschwindigkeiten des Ausführens manchen Arbeitsgänge können im Resultat der Verteilung der Verschleißressourcen zu ändern sein. Die Erneuerungsressourcen des Systems modelliert man auf vereinheitlichte Weise als so genannte verallgemeinerte Prozessoren.

1. Wprowadzenie

Zautomatyzowany elastyczny system produkcyjny (ESP) jest to zespół sterowanych komputerowo maszynowych narzędzi pracy, zintegrowanych poprzez zautomatyzowany transport, magazynowanie oraz układ sterowania. Do najważniejszych elementów ESP zaliczymy podstawowe niezżywalne zasoby systemu, takie jak sterowane komputerowo maszyny lub zrobotyzowane stanowiska pracy oraz inne urządzenia produkcyjne, ponadto urządzenia transportowe i magazyny (centralny i bufory międzyoperacyjne). Do realizacji różnorodnych operacji mogą być wymagane dodatkowe zasoby typu odnawialnego (takie jak narzędzia, pomocnicze elementy, palety itp.) oraz mogą być wykorzystywane zasoby typu zużywalnego (takie jak energia, koszt używania narzędzi), których przydział umożliwia skrócenie niektórych operacji. System jest sterowany przez komputerowy układ sterowania, realizujący różnorodne funkcje sterowania operacyjnego oraz harmonogramowania produkcji. Na rys. 1 przedstawiono schematycznie przykładowe elastyczne gniazdo wytwórcze zawierające 3 pary zautomatyzowanych stanowisk technologicznych (centra obróbkowe), magazyn centralny oraz system transportu części.

Rozważamy produkcję jednostkową w ESP. Przyjmujemy, że w wyniku rozwiązania zadania harmonogramowania nadrzędnego jest zadany zestaw zadań produkcyjnych do wykonania w najbliższym okresie (tzw. wsad produkcyjny) oraz że ESP jest *uzbrojony* do wykonywania tych zadań, a więc jest dokonana

¹Praca częściowo finansowana przez Komitet Badań Naukowych



Rys. 1: Przykład elastycznego gniazda produkcyjnego (S – magazyn centralny, MHS – system transportu, M_1, M_2, M_3 – sterowane komputerowo stanowiska)

Fig. 1. An example of a Flexible Manufacturing Cell (S – central storage, MHS – transport system, M_1, M_2, M_3 – computer-controlled machines)

alokacja niezbędnych narzędzi i ewentualnie innych zasobów odnawialnych (palet, uchwytów) potrzebnych do realizacji wszystkich wymaganych operacji.

W pracy jest rozważany problem szczegółowego harmonogramowania produkcji ESP, w którym w najbliższym okresie należy wykonać zestaw różnorodnych zadań produkcyjnych. Każde z zadań jest złożone z ciągu operacji realizowanych na różnych maszynach według indywidualnych marszrut. Realizacja operacji technologicznych, transportowych lub operacji oczekiwania wymaga zasobów odnawialnych (takich jak centra obróbkowe, wózki transportowe, magazyny buforowe), przy czym prędkości wykonywania niektórych operacji technologicznych mogą być zmieniane w wyniku rozdziału zasobów zużywalnych (np. kosztów związanych z szybszą obróbką, a tym samym — szybszym zużywaniem narzędzi).

Problem znalezienia optymalnego harmonogramu szczegółowego w ESP można zaliczyć do najtrudniejszych zadań optymalizacji dyskretnej. Nawet w przypadku znacznie prostszych, klasycznych zagadnień szeregowania (takich jak model typu job-shop bez ograniczeń pojemności buforów międzyoperacyjnych i ograniczeń transportowych) do rozwiązywania zadań o realistycznych wymiarach najczęściej stosuje się algorytmy przybliżone [2, 1]. Uwzględnienie ograniczeń buforowych (i transportowych) w klasycznych modelach szeregowania napotyka na poważne problemy modelowe i obliczeniowe i nie wykroczyło dotychczas poza szczególne przypadki, np. problemu przepływowego.

W pracy Toczyłowskiego [7] opracowano model matematyczny problemu harmonogramowania produkcji w ESP wprowadzając zunifikowany model tzw. *procesora ogólnego* pozwalającego na jednakowe traktowanie operacji technologicznych, transportowych i magazynowych. Model ten umożliwia reprezentację podstawowych ograniczeń zasobowych systemu (t j. maszyny, buforów i systemów transportowych) w postaci odpowiednio konstruowanej sieci zdarzeń, przy czym po uszeregowaniu zdarzeń podproblem wyznaczenia harmonogramu może być łatwo sprowadzony do sieciowego zadania programowania liniowego o specjalnej strukturze. Znacznie prostszy problem harmonogramowania produkcji realizowanej cyklicznie w systemach o strukturze potokowej był rozważany w [5, 6].

W niniejszej pracy jest rozważany problem harmonogramowania produkcji w ESP w przypadku ogólniejszym, gdy w systemie występują zarówno zasoby odnawialne, jak i zużywalne. Zapropo-

algorytm harmonogramowania ma strukturę dwupoziomową. Na poziomie nadrzędnym jest wyznaczane uszeregowanie oraz alokacja operacji za pomocą złożonego algorytmu poszukiwań (z wykorzystaniem heurystyk oraz sterowanego algorytmu symulowanego wyżarzania), natomiast na poziomie niższym, w wyniku rozwiązywania odpowiednich zadań programowania liniowego o specjalnej strukturze, są wyznaczone chwile rozpoczynania i kończenia operacji oraz rozdział zasobów zużywalnych pomiędzy operacje.

Rozwiązanie każdego zadania liniowego dostarcza istotnych dodatkowych informacji o aktualnym harmonogramie (współczynników wrażliwościowych oraz wartości zmiennych dopełniających niektórych ograniczeń), które mogą być wykorzystywane w algorytmie poszukiwań do znacznego ograniczenia przestrzeni poszukiwań w wyniku selekcji najbardziej atrakcyjnych punktów sąsiedztwa aktualnego harmonogramu. W rozdziale 4 zilustrowano na przykładzie sposób wykorzystania tych informacji wrażliwościowych w stosunkowo prostej metodzie poprawy uszeregowania operacji za pomocą lokalnej zamiany operacji (zdarzeń). W rozdziale 5 jest rozważany ogólny schemat szeregowania zdarzeń wykorzystujący zaawansowane techniki poszukiwań. Sformulowano tam ogólne uwagi dotyczące wykorzystania informacji wrażliwościowych do zwiększenia efektywności metod poszukiwań.

2. Model matematyczny problemu harmonogramowania

W elastycznym systemie produkcyjnym są realizowane następujące rodzaje operacji: *technologiczne*, *magazynowe* i *transportowe*. Do wykonania każdej operacji jest wymagany pewien odnawialny zasób podstawowy oraz (ewentualnie) dodatkowe odnawialne zasoby pomocnicze, które tutaj pominiemy, jako że zasoby te są na mocy założenia dostępne w dostatecznej liczbie w wyniku właściwego uzbrojenia systemu dla rozważanego wsadu produkcyjnego.

Operacje technologiczne są realizowane za pomocą maszyn technologicznych, operacje transportowe są realizowane za pomocą systemów transportu, natomiast operacje magazynowe (oczekiwania na operacje technologiczne i transportowe) są realizowane za pomocą buforów (magazynów). Zbiór wszystkich maszyn, czyli maszyn technologicznych, buforów i środków transportowych ESP, oznaczamy odpowiednio przez L^p, L^b, L^t , przy czym $L^p \cup L^b \cup L^t = L$. Niech $J = \{1, \dots, N\}$ oznacza zbiór zadań do wykonania w gniczdzie produkcyjnym. Niech $M = \{1, \dots, m\}$ będzie zbiorem wszystkich indywidualnych operacji zadań produkcyjnych (operacji technologicznych, magazynowania i transportowych), natomiast $M_j = \{m_{j-1} + 1, \dots, m_j\}$, $M_j \subset M$, zbiorem operacji zadania $j \in J$. Zadanie związane z operacją m oznaczmy przez j_m . Tam, gdzie nie prowadzi to do nieporozumień, będziemy opuszczać indeks maszyny l .

Przyjmujemy, że operacje zadania j są realizowane zgodnie z liniowym uporządkowaniem $(m_{j-1} + 1, \dots, m_j)$. Dla operacji $n \in M$ wprowadzamy zmienną decyzyjną t_n^+ określającą chwilę zdarzenia polegającego na *opuszczeniu* maszyny po zakończeniu wykonywania operacji n . Podobnie zmienna t_n^- określa chwilę zdarzenia polegającego na wprowadzeniu zadania do maszyny i *rozpoczęciu* wykonywania operacji n . Zmienne $t_n^+, n \in M$, umożliwiają jednoznaczne określenie harmonogramu realizacji wszystkich operacji. Relacje liniowego porządku poprzedzania operacji należących do tych samych zadań są reprezentowane przez zbiór $\mathcal{R} \subset M \times M$ uporządkowanych par operacji. Zachodzi $(m, n) \in \mathcal{R}$, jeżeli w zadaniu $j = j_m = j_n$ operacja m bezpośrednio poprzedza operację n , a więc \mathcal{R} reprezentuje warunki czasowe

$$t_n^- = t_m^+, \quad (m, n) \in \mathcal{R} \quad (1)$$

W odróżnieniu od klasycznego modelu szeregowania operacji technologicznych, w rozważanym modelu nie dopuszcza się wcześniejszego ukończenia operacji bezpośrednio poprzedzającej przed operacją następującą (operacje czekania w buforach są traktowane identycznie jak operacje technologiczne).

Omówimy teraz relacje czasowe między operacjami wynikające z ograniczeń zasobowych odnawialnych i zużywalnych. Modele podstawowych zasobów odnawialnych wykorzystywanych wspólnie (tzn.

równoległych grup maszyn technologicznych, buforów i systemów transportu) zapiszemy w postaci zuniifikowanego modelu ogólnego zasobu – tak zwanego procesora ogólnego [7]. Procesor $l \in L$ ma ograniczoną pojemność b_l , czyli maksymalną liczbę operacji, jakie mogą być w każdej chwili wykonywane współbieżnie za jego pomocą. Przyjmujemy, że w stanie początkowym napelnienie procesora wynosi η_l zadań. Operacja $n \in M$ wykonywana na procesorze l jest charakteryzowana przez trzy nieujemne czasy p_n, p_n^-, p_n^+ , przy czym p_n jest minimalnym czasem realizacji operacji zadania j_n (Czas wykonania operacji n na maszynie l jest równy p_n , lecz dla uproszczenia notacji indeks l pomijamy.) oraz p_n^-, p_n^+ są minimalnymi odstępami czasu, po których można wprowadzić/wyprowadzić inne zadanie (w szczególności czasy te mogą być zerowe lub bliskie zeru). Dla wielu operacji czas p_n może być stałym parametrem, natomiast w ogólności dla pewnych operacji technologicznych z podzbioru $M^p \subseteq M$ czas ten może być zmienny, będąc funkcją przydzielonej ilości zasobów zużywalnych. Niech R będzie zbiorem zasobów zużywalnych rozdzielanych między operacje technologiczne. Przyjmujemy, że w rozważanym okresie dostępność zasobu $r \in R$ wynosi Q_r jednostek zasobu. Dla operacji $n \in M^p$ o zmiennym czasie trwania przyjmujemy następujący model:

$$p_n = p_n^0 - \sum_{r \in R} q_{nr} z_{nr} \quad 0 \leq z_{nr} \leq \bar{z}_{nr}, r \in R, \quad (2)$$

przy czym p_n^0 jest nominalnym czasem wykonywania operacji n , natomiast q_{nr} jest współczynnikiem skrócenia czasu jej wykonywania w wyniku przydzielenia jednostki zasobu r dostępnego w przedziale $[0, \bar{z}_{nr}]$.

Dla procesora $l \in L$ niech E_l^- będzie zbiorem nieuporządkowanych par zdarzeń $\{m, n\}$ polegających na wprowadzeniu par zadań j_m, j_n do tego procesora. Analogicznie, niech E_l^+ będzie zbiorem nieuporządkowanych par zdarzeń $\{m, n\}$ polegających na wyprowadzeniu odpowiednich zadań z procesora l . Relację porządku oraz niejednoczesności wprowadzania i wyprowadzania dwóch zadań z procesora można wyrazić przez warunki dyzjunktywne. Przypuścimy, że mamy ustalone selekcje zupełne S_l^-, S_l^+ luków dyzjunktywnych należących do zbiorów E_l^-, E_l^+ , tak że jest określone jednoznaczne uszeregowanie wprowadzania zadań do i wyprowadzania zadań z procesora l . Muszą być spełnione warunki:

- następstwa wprowadzania zadań do procesora

$$t_m^- + p_m^- \leq t_n^- \quad (m, n) \in S_l^- \quad (3)$$

- następstwa wyprowadzania zadań z procesora

$$t_m^+ + p_m^+ \leq t_n^+ \quad (m, n) \in S_l^+ \quad (4)$$

Rozważmy teraz warunki gwarantujące realizowalność operacji wykonywanych za pomocą procesora, a w szczególności – nieprzepełnianie procesora. Wykorzystamy podstawową obserwację, że wprowadzenie kolejnego zadania do procesora l o pojemności b_l jest możliwe dopiero wówczas, gdy z procesora zostanie wyprowadzone zadanie znajdujące się o b_l pozycji wcześniej w uszeregowaniu wyjściowym S_l^+ tego procesora. Dla procesora l oraz dla ustalonych uszeregowień S_l^-, S_l^+ niech s_m^-, s_n^+ oznaczają pozycje (kolejność) wprowadzania i wyprowadzania zadania j_n określone w ciągach S_l^- i S_l^+ . Dalej, oznaczmy przez $B_l(S_l)$ zbiór par operacji $\{(m, n) : m \in S_l^+, n \in S_l^-\}$ związanych przez ograniczoną pojemność procesora, czyli par, które spełniają warunek

$$s_m^+ + b_l - \eta_l = s_n^- \quad (5)$$

Oznacza to, że jeśli rozpoczyna się pewna operacja n , znajdująca się na pozycji $s_n^- = s_m^+ + b_l - \eta_l$ i dotycząca zadania j_n wchodzącej do procesora l , to zadanie j_m , której odpowiada operacja m , musi ustąpić miejsca zadaniu j_n w tym procesorze, czyli musi zachodzić warunek

$$t_m^+ \leq t_n^-, \quad (m, n) \in B_l(S_l). \quad (6)$$

Dodatkowo musi być spełniony warunek minimalnych czasów realizacji operacji

$$t_n^- + p_n \leq t_n^+ \quad n \in M \setminus M^P \quad (7)$$

$$t_n^- + p_n^o - \sum_{r \in R} q_{nr} z_{nr} \leq t_n^+ \quad n \in M^P \quad (8)$$

Zauważmy, iż w szczególnym przypadku modelu elementarnego procesora, gdy $b_l = 1, \eta_l = 0$, zachodzi zależność $B_l(S_l) = S_l$, przy czym $S_l = S_l^- = S_l^+$. Stąd model pojedynczej maszyny technologicznej może być szczególnym przypadkiem modelu procesora ogólnego.

3. Harmonogramowanie szczegółowe

Rozważamy dwupoziomowy algorytm harmonogramowania, w którym zmienne decyzyjne dzielimy na dwie grupy. W pierwszej grupie znajdują się wszystkie 'trudne' zmienne decyzyjne, związane z alokacją operacji do procesorów oraz uszeregowaniem wszystkich zdarzeń, natomiast w grupie drugiej znajdują się zmienne określające chwile zdarzeń oraz rozdział zasobów zużywalnych pomiędzy operacje, których wartości można wyznaczyć w wyniku rozwiązywania odpowiedniego zadania liniowego. Dla ustalonych wartości zmiennych decyzyjnych z pierwszej grupy określających dopuszczalną alokację A i uszeregowanie zdarzeń S , w celu wyznaczenia harmonogramu szczegółowego należy określić przydział (z_{nr}) zasobów zużywalnych do operacji oraz chwile t_n^-, t_n^+ rozpoczęcia i ukończenia wszystkich operacji $n \in M$, tak aby były spełnione wszystkie relacje poprzedzania, ograniczenia zasobów odnawialnych, a zarazem aby sumaryczne koszty związane z realizacją harmonogramu były minimalne. W przypadku gdy zadana alokacja A i uszeregowanie zdarzeń S są niedopuszczalne, harmonogram szczegółowy nie może być zrealizowany. Warunkiem koniecznym dopuszczalności zmiennych alokacyjnych oraz selekcji $S \subset E$ jest oczywiście acykliczność grafu $G_S = (M, \mathcal{R} \cup S)$. Ogólne kryterium kosztowe jakości harmonogramu można przedstawić w postaci pewnej funkcji $F(t^-, t^+, z)$ chwil realizacji wszystkich zdarzeń oraz rozdziału zużytych zasobów. Bez dużej utraty ogólności można przyjąć, że funkcja F jest liniowa, jako że koszty wykonywania operacji, zużycia zasobów zużywalnych oraz koszty magazynowania i ewentualne koszty zaległości są zazwyczaj przyjmowane jako funkcje liniowe (por. [6], s. 153). Stąd zadanie wyznaczenia harmonogramu realizacji operacji można sformułować w postaci zadania programowania liniowego, w którym rozważa się chwile zdarzeń rozpoczęcia i ukończenia wszystkich operacji technologicznych, magazynowych i transportowych oraz relacje czasowe, jakie muszą być spełnione w odpowiedniej sieci wszystkich zdarzeń.

Zadanie TS :

$$\min F(t^+, t^-, z) \quad (9)$$

przy spełnieniu warunku nieujemności zmiennych oraz następujących warunków:

- następstwa operacji przyległych należących do tych samych zadań

$$t_m^+ = t_n^-, \quad (m, n) \in \mathcal{R} \quad (10)$$

- ograniczeń wynikających z ograniczonych możliwości procesorów

$$t_m^+ \leq t_n^-, \quad (m, n) \in B_l(S_l), l \in L \quad (11)$$

$$t_m^- + p_m^- \leq t_n^- \quad (m, n) \in S_l^-, l \in L \quad (12)$$

$$t_m^+ + p_m^+ \leq t_n^+ \quad (m, n) \in S_l^+, l \in L \quad (13)$$

$$t_n^- + p_n \leq t_n^+ \quad n \in M \setminus M^P \quad (14)$$

$$t_n^- + p_n^o - \sum_{r \in R} q_{nr} z_{nr} \leq t_n^+ \quad n \in M^P \quad (15)$$

$$\sum_{n \in M^p} z_{nr} \leq Q_r \quad r \in R \quad (16)$$

$$0 \leq z_{nr} \leq \bar{z}_{nr} \quad n \in M, r \in R \quad (17)$$

Ilorazowa sieć zdarzeń. Zadanie TS zawiera dla każdej operacji n dwie zmienne czasowe t_n^- , t_n^+ , a zatem w porównaniu do klasycznych modeli zadań szeregowania liczba wszystkich zmiennych jest dwukrotnie wyższa. Z drugiej strony jednak jest dość łatwo przekształcić to zadanie do problemu równoważnego, w którym z równań (10) eliminujemy połowę zmiennych.

Dla ustalonych alokacji i uszeregowania operacji (A, S) następstwo zdarzeń rozpoczynania i kończenia operacji można przedstawić w postaci odpowiedniej sieci zdarzeń. Oznaczmy przez \mathcal{E}_m^- zdarzenie polegające na rozpoczęciu operacji m w chwili t_m^- oraz przez \mathcal{E}_m^+ zdarzenie polegające na ukończeniu operacji m w chwili t_m^+ . W sieci zdarzeń zdarzenia te będą reprezentowane przez wierzchołki. Niektóre zdarzenia muszą przebiegać równocześnie, ze względu na warunek następstwa operacji przyległych (10), przy czym warunek ten jest spełniony zawsze, a więc mamy do czynienia z jednoczesnością nieprzypadkową a strukturalną. Łatwo sprawdzić, że relacja równoczesności jest relacją równoważności. Stąd wprowadzamy klasy równoważności $v_i \in V$, $v_i = [\mathcal{E}_m^-]$ zdarzeń ze względu na ich równoczesność:

$$\mathcal{E}_m^+, \mathcal{E}_n^- \in v_i \Leftrightarrow t_m^+ = t_n^- \quad (18)$$

Przestrzeń ilorazowa $V = \{v_i\}$ jest zbiorem wszystkich klas zdarzeń jednoczesnych. Każdej klasie v_i odpowiada pojedyncza zmienna t_i , określająca chwilę zaistnienia zdarzeń jednoczesnych. Problem zredukowany otrzymujemy z zadania TS w wyniku eliminacji równań (10) oraz zastąpienia w pozostałych ograniczeniach zadania TS zmiennych t_m^-, t_n^+ przez zmienne t_i odpowiadające klasom zdarzeń jednoczesnych.

4. Przykład ilustracyjny

Sposób wykorzystania informacji wrażliwościowych do poprawiania uszeregowania zdarzeń zaprezentujemy na przykładzie elastycznego gniazda wytwórczego z rys. 1. Gniazdo to składa się z:

- trzech centrów obróbkowych M_1 , M_2 , M_3 , przy czym każde z centrów ma dwa stanowiska robocze,
- systemu transportowego MHS w postaci jednego wózka AGV,
- magazynu centralnego S , w którym są przechowywane zarówno wyroby dopiero wchodzące do produkcji, jak również wyroby ukończone, a także wyroby oczekujące między operacjami na dalszą obróbkę. Przy centrach obróbkowych nie ma magazynów buforowych – funkcję tę pełni magazyn centralny.

Do realizacji są cztery zadania A , B , C i D , przy czym zadania A i B wymagają wykonania operacji kolejno na procesorach:

$$S \rightarrow MHS \rightarrow M_1 \rightarrow MHS \rightarrow S \rightarrow MHS \rightarrow M_2 \rightarrow MHS \rightarrow S,$$

a zadania C i D na procesorach:

$$S \rightarrow MHS \rightarrow M_2 \rightarrow MHS \rightarrow S \rightarrow MHS \rightarrow M_1 \rightarrow MHS \rightarrow S \rightarrow MHS \rightarrow M_3 \rightarrow MHS \rightarrow S.$$

Nominalne czasy wykonywania p_n^0 poszczególnych operacji n podano w tabeli 1^(a). Czasy te mogą ulec skróceniu, jeżeli zostanie przydzielona pewna ilość zasobu zużywalnego. Występuje jeden zasób zużywalny w ilości 10 jednostek ($Q_r = 10$). Współczynniki skrócenia czasu wykonywania operacji q_{nr} przy wykorzystaniu tego zasobu mają wartości takie jak w tabeli 1^(b). Każdej z operacji n można przydzielić co najwyżej 5 jednostek zasobu zużywalnego, tj. $\bar{z}_n = 5$. Poszukuje się uszeregowania o minimalnej długości.

a)

Procesor Zadanie	M_1	M_2	M_3	MHS
<i>A</i>	40	60	-	5
<i>B</i>	50	40	-	5
<i>C</i>	30	20	40	5
<i>D</i>	20	25	25	5

b)

Procesor Zadanie	M_1	M_2	M_3	MHS
<i>A, B</i>	2	1	-	0
<i>C, D</i>	3	2	1	0

Tabela 1. a) Nominalne czasy wykonywania operacji - p_n^0 ; b) Współczynniki skrócenia czasu wykonywania operacji - q_{nr} przy użyciu zasobu zużywalnego

Table 1. a) Nominal processing times p_n^0 ; b) Coefficients q_{nr} of the processing time reduction per one unit of the usable resource

Przy zadanych uszeregowaniach wejściowych S_l^- i wyjściowych S_l^+ na poszczególnych procesorach $l \in \{M_1, M_2, M_3, MHS, S\}$ pełny model TS w postaci ograniczeń (10)-(17) zawiera 44 zmienne t_n^- (bo jest 10 operacji obróbkowych, 20 operacji transportowych i 14 operacji magazynowych), 44 zmienne t_n^+ i 10 zmiennych zasobowych z_n (jest 10 operacji, których czas wykonywania zależy od zasobu zużywalnego, patrz tabela 1^(b)). W modelu występuje 40 ograniczeń typu (10) (tyle jest operacji przyległych w obrębie zadań *A, B, C, D*), 23 ograniczenia (11), po 39 ograniczeń typu (12) i (13), 34 ograniczenia (14), 10 ograniczeń (15) i jedno ograniczenie (16).

Powyższy model można jednak znacznie uprościć, jeżeli: (i) ograniczymy się do przestrzeni ilorazowej - zniknie wtedy połowa zmiennych t_n^- , t_n^+ oraz ograniczenia (10); (ii) zauważymy, że uszeregowania na procesorach M_1, M_2, M_3, S są jednoznacznie określone przez uszeregowanie na procesorze MHS . Można więc wyeliminować ograniczenia (12) i (13) dla procesorów M_1, M_2, M_3, S oraz ograniczenia (14) dla początkowych i końcowych operacji magazynowania; (iii) uwzględnimy fakt, że pojemność procesora MHS jest równa jeden, a zatem $S_{MHS}^- = S_{MHS}^+$. Pozwala to wyeliminować ograniczenia (12) i (13) dla procesora MHS . W rezultacie model harmonogramowania szczegółowego TS dla rozważanego przykładowego problemu można zapisać w postaci 23 ograniczeń (11), 26 ograniczeń (14), 10 ograniczeń (15) i jednego ograniczenia (16) używając 20 zmiennych $t_{i,MHS}^-$, $i = 1, \dots, 20$ określających chwile rozpoczęcia poszczególnych operacji na procesorze MHS , 20 zmiennych $t_{i,MHS}^+$, $i = 1, \dots, 20$ określających chwile zakończenia oraz 10 zmiennych zasobowych.

Załóżmy teraz, że w trakcie poszukiwania jak najlepszego harmonogramu otrzymaliśmy pewne dopuszczalne uszeregowanie na procesorach M_1, M_2, M_3, MHS, S , które chcielibyśmy poprawić. Jak wspomniano, uszeregowania na procesorach M_1, M_2, M_3, S są jednoznacznie określone przez uszeregowanie na procesorze MHS , a zatem wystarczy skupić się na analizie sposobu modyfikacji tylko tego uszeregowania. W celu jego poprawy spróbujemy wykorzystać informacje wrażliwościowe z zadania TS .

Określmy dla każdej pary a, b przyległych operacji na procesorze MHS następujący współczynnik $\alpha_{a,b}$.

$$\alpha_{a,b} = \lambda_{a_1} - \lambda_{a_2} - \lambda_{b_1} + \lambda_{b_2} \quad (19)$$

gdzie:

a_1, a_2 - operacje należące do tego samego zadania co operacja a , przy czym a_1 bezpośrednio poprzedza a , natomiast operacja a_2 występuje bezpośrednio za operacją a , czyli $(a_1, a) \in \mathcal{R}$ i $(a, a_2) \in \mathcal{R}$,

b_1, b_2 - operacje spełniające analogiczny warunek jak wyżej, tzn. $(b_1, b) \in \mathcal{R}$ i $(b, b_2) \in \mathcal{R}$,

$\lambda_{a_1}, \lambda_{a_2}, \lambda_{b_1}, \lambda_{b_2}$ - mnożniki Lagrange'a dla ograniczeń typu (14) i (15) odpowiadających poszczególnym operacjom a_1, a_2, b_1, b_2 .

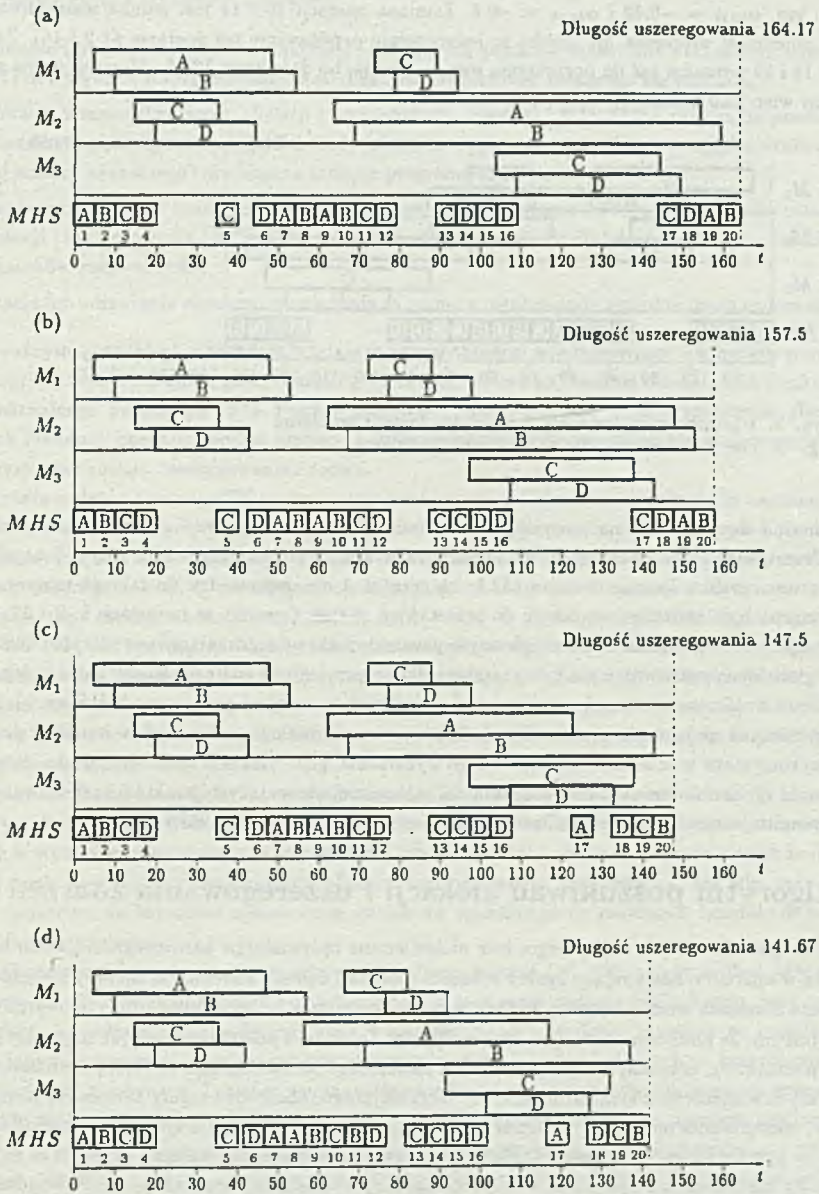
Wielkość $\alpha_{a,b}$ będziemy traktować jako miarę poprawy uszeregowania po zamianie kolejności wykonywania operacji a i b na procesorze MHS . W oparciu o miarę α można zaproponować dla rozważanego przykładowego problemu następujący prosty algorytm poprawiania uszeregowania zdarzeń.

Algorytm A

1. Dla każdej pary operacji przyległych a i b na procesorze MHS wyliczamy wartości współczynników $\alpha_{a,b}$.
2. Spośród par operacji a, b , dla których zamiana jest dopuszczalna, wybieramy parę a^*, b^* z najmniejszym współczynnikiem α .
3. Jeżeli wartość współczynnika α_{a^*,b^*} jest nieujemna, to KONIEC; aktualne uszeregowanie traktujemy jako najlepsze. Jeżeli $\alpha_{a^*,b^*} < 0$, to zamieniamy kolejność wykonywania operacji a^* i b^* na procesorze MHS .
4. Rozwiązujemy zadanie TS dla nowego uszeregowania. Jeżeli uzyskaliśmy poprawę (ewentualnie taką samą długość uszeregowania jak poprzednio), to przechodzimy do następnej iteracji realizując kroki 1-4. Jeżeli nowe uszeregowanie jest gorsze, to wracamy do poprzedniego uszeregowania, uznajemy zamianę pary a^*, b^* za niedopuszczalną i cofamy się do kroku 2.

W kroku 2 niedopuszczalność zamiany operacji może wynikać np. z przekroczenia pojemności procesorów, nieatrakcyjności zamiany (patrz krok 4) lub zabraniającego wykonywania zamian tych par operacji, które były wybierane w kilku poprzednich iteracjach (np. stosując metodę "tabu search").

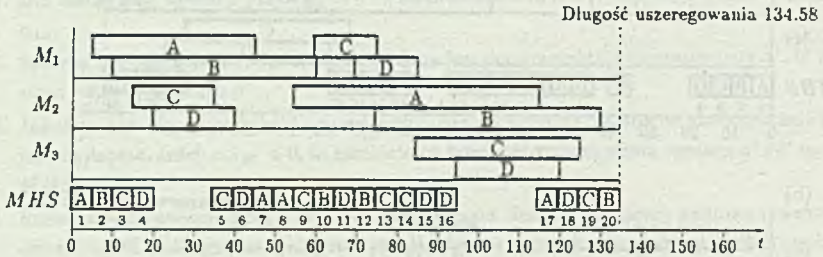
Działanie algorytmu zilustrujemy dla przypadku, gdy początkowe uszeregowanie na procesorze MHS w rozważanym przykładowym problemie jest postaci: $A, B, C, D, C, D, A, B, A, B, C, D, C, D, C, D, C, D, A, B$. Rozwiązując zadanie TS dla tego uszeregowania otrzymujemy harmonogram przedstawiony na rys. 2^(a) o minimalnym czasie 164.17. Mnożniki Lagrange'a $\lambda_{j,l}$ dla ograniczeń typu (15) odpowiadających operacjom z zadań $j = A, B, C, D$ wykonywanych na procesorach $l = M_1, M_2, M_3$ są następujące: $\lambda_{A,M_1} = 0, \lambda_{A,M_2} = 0, \lambda_{B,M_1} = -0.5, \lambda_{B,M_2} = 0, \lambda_{C,M_1} = -0.67, \lambda_{C,M_2} = 0, \lambda_{C,M_3} = -1, \lambda_{D,M_1} = -0.33, \lambda_{D,M_2} = -0.5, \lambda_{D,M_3} = 0$. Dla operacji magazynowych na procesorze S przyjmujemy, że mnożniki te są równe zero. Współczynnik $\alpha_{a,b}$ osiąga najmniejszą wartość równą -1.33 dla pary operacji 14 i 15. Zamieniamy więc kolejność wykonywania tych operacji na procesorze MHS i otrzymujemy w drugiej iteracji harmonogram o długości uszeregowania 157.5 przedstawiony na rys. 2^(b). W trzeciej, czwartej i piątej iteracji na podobnej zasadzie dokonujemy zamian operacji 17 i 18, potem 18 i 19, a następnie 17 i 18 uzyskując uszeregowania o długościach odpowiednio 152.5, 152.5 oraz 147.5. Harmonogram uzyskany w piątej iteracji jest przedstawiony na rys. 2^(c). Mnożniki Lagrange'a odpowiadające poszczególnym ograniczeniom zadania TS są następujące: $\lambda_{A,M_1} = 0, \lambda_{A,M_2} = 0, \lambda_{B,M_1} = -0.5, \lambda_{B,M_2} = 0, \lambda_{C,M_1} = -0.67, \lambda_{C,M_2} = 0, \lambda_{C,M_3} = 0, \lambda_{D,M_1} = -0.33, \lambda_{D,M_2} = -0.5, \lambda_{D,M_3} = -1$. Współczynniki $\alpha_{a,b}$ przyjmują najmniejsze wartości dla par 18 i 19 oraz 10 i 11, przy czym $\alpha_{18,19} = -1, \alpha_{10,11} = -0.67$. Po dokonaniu zamiany pary 18,19 i rozwiązaniu zadania TS stwierdzamy w szóstej iteracji, że nowe uszeregowanie jest dłuższe, bo równe 152.5. Wracamy więc do poprzedniego uszeregowania i zamieniamy operacje 10 i 11, co daje w siódmej iteracji uszeregowanie o długości 144.17. Następnie w ósmej iteracji zamieniamy kolejność wykonywania operacji 8 i 9 i otrzymujemy harmonogram przedstawiony na rys. 2^(d) z czasem 141.67. Współczynnik $\alpha_{a,b}$ osiąga najmniejszą wartość równą -1 znowu dla pary operacji 18 i 19, przy czym zamiana tych operacji podobnie jak w iteracji szóstej nie prowadzi do poprawy uszeregowania. Wymieniane są więc operacje 11 i 12 z następnym co do wielkości współczynnikiem $\alpha_{11,12} = -0.5$, co prowadzi do harmonogramu z czasem końcowym 140. W jedenastej iteracji zamieniamy operacje 9 i 10, dla których $\alpha_{9,10} = -0.78$ i uzyskujemy harmonogram przedstawiony na rys. 3. Długość uszeregowania wynosi 134.58. Wartości mnożników Lagrange'a dla tego przypadku są następujące: $\lambda_{A,M_1} = -0.25, \lambda_{A,M_2} = -0.5, \lambda_{B,M_1} = -0.25, \lambda_{B,M_2} = 0, \lambda_{C,M_1} = -0.17, \lambda_{C,M_2} = -0.25, \lambda_{C,M_3} = 0, \lambda_{D,M_1} = -0.17,$



Rys. 2. Harmonogramy uzyskiwane w kolejnych iteracjach algorytmu A:
 a) 1 iteracja; b) 2 iteracja; c) 5 iteracja; d) 8 iteracja

Fig. 2. Schedules resulted from successive iterations of the algorithm A:
 a) iteration 1; b) iteration 2; c) iteration 5; d) iteration 8

$\lambda_{D,M_2} = -0.25$, $\lambda_{D,M_3} = -0.5$. Współczynniki $\alpha_{a,b}$ przyjmują ujemne wartości tylko dla dwóch par operacji, tzn. $\alpha_{10,11} = -0.42$ i $\alpha_{18,19} = -0.5$. Zamiana operacji 10 i 11 jest jednak niemożliwa, gdyż narusza pojemność procesora M_1 (chyba że jednocześnie przestawimy też operacje nr 2 i 16). Zamiana operacji 18 i 19 prowadzi zaś do pogorszenia uszeregowania, bo daje koszt 137.5. Uszeregowanie z rys. 3 traktujemy więc jako końcowe.



Rys. 3. Harmonogram końcowy dla przykładowego problemu
Fig. 3: The final schedule for the example problem

Jak można się przekonać na powyższym przykładzie, algorytm A poprawia uszeregowanie tylko lokalnie. Przeszawiając np. operację 20 na miejsce 18 w końcowym uszeregowaniu z rys. 3 uzyskujemy lepszy harmonogram o koszcie równym 132.5. Algorytm A nie doprowadził do takiego uszeregowania, chociaż można było zauważyć tendencje do przestawień w tym kierunku w iteracjach 6, 9 i 12. Widać więc, że algorytm A powinien być wzbogacony o pewne dodatkowe mechanizmy pozwalające dokonywać bardziej globalnych przestawień niż tylko zamiana dwóch przyległych operacji. Ruchy takie powinny być uwzględniane zwłaszcza wtedy, gdy zamiana pewnych dwóch operacji na procesorze MHS jest blokowana przez ograniczoną pojemność procesorów. Informacje wrażliwościowe mogą być w bardziej skuteczny sposób wykorzystane w schemacie symulowanego wyzarczenia, gdzie miara α może służyć jako instrument ograniczania otoczenia bieżącego rozwiązania do najbardziej obiecujących punktów sąsiedztwa. Wybór nowego punktu jest wtedy określany losowo z uwzględnieniem oceny ich jakości [3].

5. Algorytm poszukiwań alokacji i uszeregowania zdarzeń

Metoda poszukiwań jak najlepszego, lecz niekoniecznie optymalnego harmonogramu może być zrealizowana w oparciu o następujący ogólny schemat (por. [3]) będący kombinacją metody symulowanego wyzarczenia z metodą wrażliwościowej analizy perturbacyjnej oraz heurystykami priorytetowymi.

Przyjmijmy, że punktem startowym do iteracyjnego algorytmu poszukiwań jest jak najlepszy harmonogram początkowy, uzyskany np. po wyborze jak najlepszego harmonogramu ze zbioru harmonogramów uzyskiwanych wariantowo z symulatora ESP wykorzystującego różnorodne reguły heurystyczne, np. priorytetowe, szeregowania operacji w systemie [1, 2]. Z każdego otrzymanego z symulatora harmonogramu wybieramy jedynie alokację i uszeregowanie zdarzeń, natomiast pozostałe zmienne określające najlepszy, szczegółowy harmonogram wykonania wszystkich operacji wyznaczamy rozwiązując liniowe zadanie harmonogramowania TS minimalizując kosztowe kryterium jakości. (W rezultacie otrzymujemy początkową alokację A^0 operacji do procesorów oraz początkowe uszeregowanie zdarzeń S^0 .)

Na początku k -tej iteracji algorytmu poszukiwania alokacji i uszeregowania jest znany przydział operacji do maszyn A^k , uszeregowanie operacji S^k , harmonogram zdarzeń T^k oraz rozdział zasobów zużywalnych Z^k .

A oto skrócony zapis k -tej iteracji algorytmu:

1. generowanie i przybliżona ocena najbardziej obiecujących punktów sąsiedztwa aktualnego punktu (A^k, S^k) dokonywana z wykorzystaniem analizy 'wrażliwościowej' bieżącego rozwiązania,
2. wylosowanie z sąsiedztwa nowej alokacji i uszeregowania operacji (A^{k+1}, S^{k+1}) , przy czym prawdopodobieństwo wyboru punktu zależy od oceny jakości nowego punktu uzyskanej w wyniku wrażliwościowej analizy aktualnego rozwiązania zadania programowania liniowego TS^k ,
3. obliczenie szczegółowego harmonogramu poprzez rozwiązanie zadania TS^{k+1} , w którym dla ustalonej alokacji i uszeregowania (A^{k+1}, S^{k+1}) wyznacza się optymalne chwile zdarzeń T^{k+1} oraz rozdział Z^{k+1} zasobów zużywalnych.
4. akceptacja lub odrzucenia obranego rozwiązania za pomocą reguł metody symulowanego wyżarzania.

Bardzo ważnym elementem algorytmu, mającym istotny wpływ na efektywność algorytmu poszukiwań, jest zaprojektowanie sposobu generowania ograniczonego zbioru najbardziej obiecujących punktów sąsiedztwa aktualnego punktu (A^k, S^k) . Punkty sąsiedztwa można uzyskać przez wyznaczenie zbioru wariantowych realokacji operacji poprzez przemieszczenie niektórych krytycznych operacji na mniej krytyczne maszyny oraz zmiany uszeregowania zdarzeń.

Z rozwiązania zadania liniowego TS można uzyskać informacje wrażliwościowe (mnożniki Lagrange'a ograniczeń oraz wartości zmiennych dopełniających tych ograniczeń) pozwalające na mniej lub bardziej dokładne oszacowanie kosztu lub zysku z przesunięcia każdego ze zdarzeń. Przykładowo, rozważmy oszacowanie kosztu przesunięcia chwili t_n^- dla zdarzenia polegającego na rozpoczęciu operacji n . Jeżeli operacja m jest operacją tego samego zadania przyległą do n , t.j. $(m, n) \in R$, to mnożnik Lagrange'a ograniczenia (14) lub (15) dla operacji m określa jednostkowy zysk ze skrócenia czasu ukończenia tej operacji o jednostkę czasu. Jednocześnie wartość zmiennej dopełniającej tego ograniczenia określa dopuszczalny przedział tego przesunięcia bez konieczności przesunięcia zdarzeń poprzedzających. Z właściwości rozwiązania optymalnego wynika, że tylko jedna ze zmiennych sprzężonych (t.j. mnożnika Lagrange'a i zmiennej dopełniającej) może być dodatnia.

Mając obliczone funkcje szacujące koszty przesunięcia w lewo chwil t_n^- oraz w prawo chwil t_n^+ , istnieje możliwość wyselekcjonowania tych potencjalnie korzystnych punktów sąsiedztwa aktualnego rozwiązania, otrzymanych w wyniku zmiany uszeregowania zdarzeń, dla których taka zamiana prowadzi do jak najkorzystniejszej oceny poprawy (wykorzystującej obliczone funkcje kosztów przesunięcia). Tak więc opisana metoda jest sposobem na świadome ograniczenie wariantów poszukiwań do rozwiązań ocenianych jako stosunkowo najkorzystniejsze.

Podczas generowania ograniczonego sąsiedztwa aktualnego punktu (A^k, S^k) należy zadbać, aby generowane pary (A, S) były dopuszczalne (t.j. umożliwiały rozwiązanie zadania TS). Należy przy tym uwzględnić fakt, iż uszeregowania wejściowe oraz wyjściowe S_i^-, S_i^+ ustalonej maszyny są częściowo powiązane z innymi zmiennymi określającymi uszeregowanie operacji technologicznych na innych, odpowiednio powiązanych maszynach. Tak więc zamiana dwóch zdarzeń, np. rozpoczynania pewnych operacji na tym samym procesorze, może implikować potrzebę zamiany zdarzeń innych, np. zamianę kolejności kończenia tych operacji. W szczególnych przypadkach pewne uszeregowania na jednym procesorze mogą nawet wynikać całkowicie z innych uszeregowani na innym procesorze dominującym. Przykładowo, jeżeli rozważamy bufor l , który jest buforem wejściowym tylko *jednej* maszyny, to selekcja S_i^+ jest określona jednoznacznie przez uszeregowanie wejściowe operacji na tej maszynie. Analogicznie, selekcja S_i^- jest określana jednoznacznie dla bufora wyjściowego *jednej* maszyny. Z drugiej strony, jeżeli pewna ilość maszyn zasila tylko jeden magazyn buforowy, to uszeregowanie S_i^- określa jednoznacznie uszeregowanie wyjściowe operacji na tych maszynach. Podobnie S_i^+ określa jednoznacznie uszeregowanie wejściowe operacji na wszystkich maszynach odbierających zadania wyłącznie z bufora l .

Uwagi końcowe. Rozważane w pracy podejście do harmonogramowania procesów produkcji w ESP w oparciu o zunifikowany model zasobów maszynowych, buforowych i transportowych z jednoczesnym uwzględnieniem alokacji zasobów zużywalnych prowadzi do dwupoziomowego algorytmu harmonogramowania, w którym na poziomie nadrzędnym można stosować złożone algorytmy poszukiwań jak najlepszego uszeregowania zdarzeń, natomiast na poziomie niższym jest rozwiązywane zadanie liniowe o specjalnej strukturze dostarczające istotnych informacji wrażliwościowych dla algorytmu nadrzędnego. Podkreślić tutaj trzeba, że sprawą otwartą jest dalsze rozwijanie efektywnych metod alokacji procesorów i szeregowania operacji w ESP w przypadku rozdziału zasobów zużywalnych – problem ten wymaga dalszych żmudnych badań, ze względu na specyfikę zagadnień szeregowania z alokacją zasobów zużywalnych [4], powodującą powstawanie wielu ścieżek krytycznych, co osłabia moc wielu reguł szeregowania.

Rozważane w tej pracy podejście algorytmiczne można również traktować jako potencjalną alternatywę w stosunku do klasycznych metod symulacji procesów produkcji w ESP, umożliwiającą realizację 'efektywnej' symulacji połączonej ze stosunkowo efektywnymi metodami szeregowania zdarzeń. W stosunku do metod symulacyjnych aparat ten umożliwia optymalizację wartości ważnych zmiennych decyzyjnych (chwile zdarzeń i alokację zasobów zużywalnych) oraz dostarcza informacji wrażliwościowych wykorzystywanych w nadrzędnym algorytmie poszukiwań.

Literatura

- [1] French S.: *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Horwood, Chichester, 1982.
- [2] Gere W.S.: Heuristics in Job Shop Scheduling. *Management Science* 13, 164-180, 1966.
- [3] Jagiello S., Toczyłowski E.: Szczegółowe minimalnokosztowe harmonogramowanie produkcji w gnieździe produkcyjnym z maszynami równoległymi. *Zesz. Nauk. Pol. Śl., ser. Automatyka*, z.102, 61-71, 1990.
- [4] Janiak A.: Dokładne i przybliżone algorytmy szeregowania zadań i rozdziału zasobów w dyskretnych procesach przemysłowych. *Prace Nauk. Inst. Cyb. Techn. Pol. Wroc.*, Nr. 87, 1991.
- [5] Toczyłowski E.: Dwupoziomowe harmonogramowanie produkcji średnioseryjnej w elastycznej linii potokowej. *Prace Nauk. Inst. Cyb. Techn. Pol. Wroc.*, Nr. 78, 149-155, 1988.
- [6] Toczyłowski E.: *Niektóre metody strukturalne optymalizacji do sterowania w dyskretnych systemach wytwarzania*. WNT, Warszawa, 1989.
- [7] Toczyłowski E.: Szczegółowe modele zasobów buforowych i transportowych zautomatyzowanych gniazd wytwarzania. *Prace Nauk. Inst. Cyb. Techn. Pol. Wroc.* Nr. 82, 267-272, 1990.

Recenzent: Prof.dr h.inż. Konrad Wala
Wpłynęło do Redakcji do 30.04.1992 r.

Abstract:

An algorithm is presented for the detailed production scheduling problem in a FMS cell, where processing rates of some operations may be reduced by allocations of some consumable resources, and the renewable resources (work-centers, buffers and material handling systems) are modelled in a unified way as *general processors*. This allows us to represent the limitations of the renewable resources and precedence relations on a network of events. This network can be easily transformed into a structured linear programming problem, which is a subproblem of a general scheduling problem in FMSs.

The algorithm has a two-level structure. At the higher level a composite search scheme is considered, which uses the simulated annealing algorithm with some additional rules incorporated for the reduction of the search space. These rules take into account sensitivity analysis of the LP solution of the lower-level problem.