ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: AUTOMATYKA z.118 Nr kol. 1338

Andrzej ADRABIŃSKI
Europejski Fundusz Leasingowy, Wrocław
Józef GRABOWSKI
Politechnika Wrocławska
Mieczysław WODECKI
Uniwersytet Wrocławski

SZEREGOWANIE ZADAŃ W PEWNYM ELASTYCZNYM SYSTEMIE PRODUKCYJNYM

Streszczenie. W referacie przedstawiamy przykłady zastosowania metod optymalizacji dyskretnej do rozwiązania problemu optymalnego wykorzystania automatów tokarskich, zwanych elastycznymi centrami tokarsko-frezarskimi. Optymalizacja pracy centrum obejmuje harmonogramowanie procesu obróbki oraz sterowanie wykonywaniem operacji na detalach. Z punktu widzenia teorii złożoności obliczeniowej rozpatrywane zagadnienie należy do klasy problemów *NP-trudnych.*

A SEQUENCING PROBLEM FOR JOB PROCESSING BY THE TURNING-LATHE

Summary. The purpose of this paper is to describe an optimization approach to a some turning-lathe machine. The paper includes two parts. The first one considers the sequencing problem of job processing by the machine. We prove that the problem is NP-hard. The second part of the paper considers the problem of the tool replacement by the turning-lathe. An algorithm for finding an optimal sequence of tool replacement intervals in the tool-store of the machine is presented.

1. Wprowadzenie

Ze względu na dużą kapitałochłonność automatów wymagany jest wysoki stopień ich wykorzystania. Jest to możliwe dzięki optymalnemu planowaniu procesu produkcyjnego. W trakcie produkcji detale poddawane są procesowi obróbki. Narzędzia potrzebne do wykonania operacji na detalach pobierane są automatycznie z odpowiednich magazynów wchodzących w skład centrum. Przed obróbką każdy detal mocowany jest na specjalnej palecie.

Optymalizacja pracy centrum obejmuje zasadniczo dwa problemy:

1) wyznaczanie optymalnej kolejności wykonywania detali (harmonogramowanie procesu załadunku na palety, obróbki i rozładunku detali),

2) sterowanie wykonywaniem operacji na detalach, w tym:

   (*i*) przydział i optymalizacja wykorzystania narzędzi,

   (*ii*) wyznaczenie terminów wymiany narzędzi w magazynach.

W dalszej części pokażemy, że nawet uproszczona wersja rozpatrywanego zagadnienia należy do klasy problemów *NP-trudnych*, tj. klasy problemów, dla których nie jest obecnie możliwe skonstruowanie algorytmów o wielomianowej złożoności obliczeniowej.

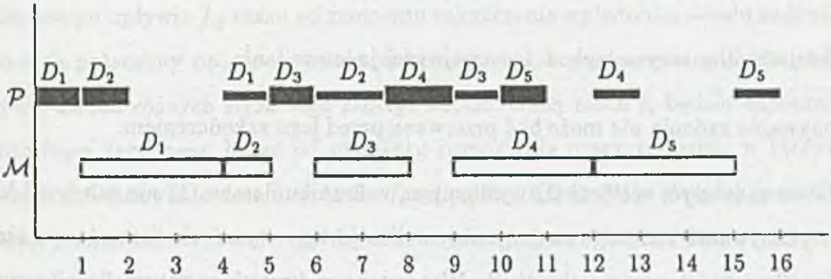
2. Sformułowanie problemu

Dany jest zbiór detali, które mają być wykonane przez centrum (w skrócie CTF). W skład centrum wchodzi maszyna (obrabiarka) wykonująca operacje: tokarskie, frezarskie i pomiarowe, magazyny narzędzi oraz palety, na których mocowane są obrabiane detale. W zależności od rodzaju obróbki maszyna pobiera automatycznie odpowiednie narzędzie z magazynu. Detal, przed wykonaniem na nim pierwszej operacji, musi być umocowany na palecie, a po wykonaniu ostatniej- zdjęty. Każde narzędzie może być użyte wielokrotnie, jednak łączny czas jego pracy nie może przekroczyć z góry zadanej żywotności. Centrum obsługiwane jest przez pracownika, który m.in. mocuje oraz zdejmuje detale z palet oraz dokonuje wymiany narzędzi w magazynach. Dany jest także porządek technologiczny określony na zbiorze detali, tj. kolejność wykonywania pewnych detali oraz dla każdego detalu, czasy i kolejność wykonywania operacji.

Optymalizacja procesu wykonywania detali polega na ustaleniu, dla każdej zmiany pracy centrum, kolejności ich obróbki i kolejności wykonania operacji dla każdego detalu, zgodnie z wymogami porządku technologicznego, oraz na wyznaczeniu zbiorów narzędzi i momentów wymiany narzędzi w magazynach tak, aby stosunek sumy czasów wykonania wszystkich operacji, tzw. czasu pracy pod wiórem, do całkowitego czasu obróbki wszystkich detali był maksymalny.

Na rys. 1 przedstawiamy diagram Gantta ilustrujący harmonogram czynności wykonywanych przez maszynę i pracownika przy obróbce pięciu detali D_1, D_2, D_3, D_4, D_5 w czasie 16-godzinnej pracy centrum przy założeniu, że mamy do dyspozycji dwie palety. Symbolem $\mathcal{P}$ oznaczono czynności wykonywane przez pracownika: mocowanie detalu na palecie (linie pogrubione) oraz zdejmowanie z palety (linie cieńsze). Natomiast $\mathcal{M}$ oznacza maszynę. Dla tego przykładu stosunek czasu pracy maszyny pod wiórem do całkowitego czasu pracy wynosi 4/5 (maszyna kończy pracę w 15 jednostce czasu).

W dalszej części pracy *zadaniem* nazywać będziemy zbiór operacji wykonywanych przez maszynę w czasie jednego zamocowania detalu na palecie. Detal może być obrabiany wielokrotnie, przy różnych zamocowaniach.



Rys. 1.

Załóżmy, że dany jest zbiór n zadań

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

które mają być wykonane przez maszynę wchodzącą w skład centrum. Przed rozpoczęciem wykonywania zadania detal z nim związany mocowany jest na palecie, a po wykonaniu zdejmowany. Dla każdego zadania $J_i \in J$ niech z_i, p_i oraz w_i będą odpowiednio: *czasem załadunku*, *czasem wykonywania* oraz *czasem wyładunku* detalu z palety. Zakładamy, że centrum pracuje w systemie *zmianowym*. Oznaczmy przez *dz - długość zmiany* liczoną w godzinach ($dz \leq 24$). Detal, w dniu obróbki, musi być załadowany, wykonany i zdjęty z palety. W dalszej części pracy będziemy także rozpatrywać przypadek, gdy centrum pracuje w systemie *ciągłym*, tj. bez wyłączania maszyny.

Rozpatrywany przez nas problem harmonogramowania procesu obróbki detali polega na wyznaczeniu w zbiorze J podzbiorów zadań, które będą wykonane w kolejnych dniach pracy maszyny, oraz kolejności ich wykonywania w każdym dniu tak, aby stosunek

$$W = \frac{C_w}{C_c}, \tag{1}$$

zwany *współczynnikiem wykorzystania maszyny*, osiągnął wartość maksymalną, gdzie:

C_w – jest *efektywnym* (pod wiórem) *czasem* pracy maszyny, czyli $C_w = \sum_{i=1}^{n} p_i$, a

C_c – jest *całkowitym czasem* wykonania zadań, tj. czasem pracy centrum, jaki upłynął od momentu rozpoczęcia mocowania pierwszego detalu na palecie do momentu zakończenia wykonywania ostatniego zadania przez maszynę.

Dodatkowo muszą być spełnione następujące ograniczenia:

a) w każdej chwili maszyna wykonuje co najwyżej jedno zadanie,

b) wykonywanie zadania nie może być przerwane przed jego zakończeniem.

Jak łatwo zauważyć, wielkość C_w występująca w liczniku ilorazu (1) nie zależy od kolejności wykonywania zadań, a zatem maksymalizacja tego wyrażenia jest równoważna minimalizacji wartości mianownika, tj. C_c. Wobec tego za kryterium optymalizacji przyjmujemy czas wykonania wszystkich zadań C_c, który należy minimalizować.

Oznaczmy przez **P** wyżej opisany problem optymalizacji kolejności wykonywania zadań przez maszynę.

TWIERDZENIE 1. *Problem* **P** *optymalizacji kolejności wykonywania zadań przez maszynę należy do klasy problemów NP-trudnych.* ∎

Dowód twierdzenia zamieszczony jest w pracy [1].

Z powyższego twierdzenia wynika, że obecnie nie jest możliwa konstrukcja algorytmu, o wielomianowej złożoności obliczeniowej, wyznaczania optymalnego rozwiązania problemu **P**. Fakt ten usprawiedliwia stosowanie algorytmów heurystycznych, o wielomianowej złożoności obliczeniowej, do rozwiązywania tego typu zagadnień. Algorytmy takie przedstawiamy w dalszej części.

Załóżmy, że na zbiorze zadań J określona jest relacja częściowego porządku RT wyrażająca wymagania *porządku technologicznego*. Jeśli para $(J_i, J_j) \in RT$, to oznacza, że rozpoczęcie ładowania na paletę detalu zadania J_j może nastąpić dopiero po rozładunku detalu zadania J_i. Ma to miejsce między innymi wtedy, gdy detal obrabiany jest wielokrotnie przy różnych zamocowaniach. Relację tę można przedstawić w postaci grafu skierowanego $G = (V, E)$, zwanego *grafem pierwszeństwa*, którego zbiorem wierzchołków jest $V = \{1, 2, ..., n\}$, a E zbiorem łuków. Odpowiednikiem zadania $J_i \in J$ jest w grafie G wierzchołek i. Para różnych wierzchołków $(u, v) \in V$ tworzy łuk w G, tj. $(u, v) \in E$ wtedy i tylko wtedy, gdy $(J_u, J_v) \in RT$. W dalszej części pracy, wszędzie tam, gdzie nie prowadzi to do nieporozumień, będziemy zamiennie pisali wierzchołek i oraz zadanie J_i.

Wprowadzimy teraz pewne dodatkowe pojęcia i oznaczenia. *Wagą wierzchołka* $i \in V$ jest czas wykonywania p_i zadania J_i. Przez f_{ij} oznaczamy *wagę łuku* $(i, j) \in E$. Jeśli $f_{ij} > 0$, to znaczy, że rozpoczęcie ładowania detalu zadania J_j na paletę może nastąpić dopiero po upływie f_{ij} czasu od momentu zakończenia wyładunku detalu zadania J_i. Jest to czas potrzebny np. do wymiany zamocowania lub ostygnięcia, gdy oba zadania dotyczą dwóch różnych stron tego samego detalu. Dalej niech r_i będzie *najwcześniejszym możliwym terminem*, licząc od momentu rozpoczęcia pracy centrum, w którym można rozpocząć ładowanie detalu zadania J_i na paletę. Korzystając z wyżej przedstawionych pojęć i oznaczeń wprowadzamy dodatkowo następujące ograniczenia:

c) zachowany musi być porządek technologiczny wykonywania zadań oraz uwzględnione wagi wierzchołków i łuków grafu pierwszeństwa,

d) termin rozpoczęcia załadunku detalu nie może być mniejszy niż jego najwcześniejszy możliwy termin załadunku.

W dalszej części pracy rozważać będziemy problem **P** łącznie z ograniczeniami **c** i **d**.

3. Algorytmy optymalizacji procesu obróbki detali

Oznaczmy przez H_i całkowicie uporządkowany podzbiór zbioru zadań wykonywanych przez maszynę w i- tym dniu pracy, tj. zbiór, w którym ustalona jest kolejność wykonywania wszystkich zadań, uwzględniająca również relację porządku technologicznego. Z kolei niech t_i (*zajętość maszyny*) będzie terminem zakończenia wykonywania przez maszynę ostatniego zadania znajdującego się aktualnie w H_i oraz T_i (*zajętość pracownika*) terminem zakończenia przez pracownika załadunku (albo wyładunku) detalu ostatniego w i-tym dniu zadania. Zadania należące do zbiorów H_i nazywać będziemy *zadaniami ustalonymi*, a pozostałe z J *zadaniami wolnymi*. Dzień k-ty pracy maszyny jest *dopuszczalnym* dla zadania J_j, jeżeli w dniu tym zadanie to może być *całkowicie wykonane*, tj.:

(i) od momentu rozpoczęcia wykonywania przez centrum zadań z J do momentu T_k, w którym (w k-tym dniu) pracownik może rozpocząć ładowanie detalu zadania J_j, upłynęło co najmniej r_j czasu,

(ii) załadunek, wykonanie i rozładunek zadania J_j nastąpi w k-tym dniu,

(iii) od momentu zakończenia rozładunku każdego zadania J_i będącego poprzednikiem J_j, (tj. $(i,j) \in E$) do chwili T_k upłynęło co najmniej f_{ij} czasu.

Oznaczmy przez A_i oraz B_i zbiory *następników* oraz *poprzedników* wierzchołka i w grafie G. Jeżeli z wierzchołka i do wierzchołka j istnieje droga w G, to $i \in B_j$ oraz $j \in A_i$. Dla zadania $J_i \in J$ niech

$$q_i = \max\{\sum_{j \in A_i} p_j,\ \max\{d_{ij} : j \in A_i\}\},$$

gdzie d_{ij} jest długością drogi w grafie G z wierzchołka i do j, tj. sumą wag wierzchołków i łuków tworzących tę drogę. Jeśli $A_i = \emptyset$, to przyjmujemy, że $q_i = 0$. Ponadto niech

$$b_i = r_i + p_i + q_i. \tag{2}$$

Jak łatwo zauważyć, wielkość b_i jest najwcześniejszym terminem zakończenia wykonywania zadania J_i.

Wyznaczając kolejność wykonywania zadań, spośród zadań wolnych, których wszystkie poprzedniki w grafie pierwszeństwa G są już ustalone, wybieramy zadanie o maksymalnej wartości wyrażenia (2). Dla ustalenia uwagi załóżmy, że wybraliśmy zadanie J_k. Następnie spośród dni dopuszczalnych dla tego zadania wybieramy pierwszy. W dniu tym zadanie J_k będzie wykonywane. Jeśli wybraliśmy dzień l, to:

- *ustalamy* zadanie J_k w zbiorze H_l, tj. $H_l \leftarrow H_l \cup \{J_k\}$, przy czym zadanie to będzie wykonywane jako ostatnie,
- uaktualniamy czas zajętości pracownika i maszyny uwzględniając fakt, że zadanie J_k będzie wykonywane w l-tym dniu,
- modyfikujemy zbiory poprzedników zadania J_k: $\quad \forall i \in A_k, \quad B_i \leftarrow B_i \setminus \{k\}$,
- uaktualniamy najwcześniejsze możliwe terminy rozpoczęcia wykonywania zadań uwzględniając termin wykonania J_k, tj.

$$\forall j \in A_k, \quad r_j := \max\{r_j,\ (l-1)dz + t_l + w_l + f_{kj}\}.$$

Po wykonaniu wyżej opisanych modyfikacji wszystkie zbiory poprzedników zawierają jedynie zadania wolne, tj. zadania, które nie zostały jeszcze ustalone.

Dokładny opis algorytmu jest zamieszczony w pracy [1], a jego złożoność obliczeniowa jest równa $O(n^2)$. Strategia, którą stosujemy wyznaczając numer zadania i odpowiedni dzień, jest oparta na idei algorytmu *najlepszego dopasowania uporządkowanych zadań* (ang. first-fit decreasing) wyznaczającego rozwiązanie przybliżone dla problemu pakowania (zobacz [3]).

TWIERDZENIE 2. *Jeżeli L_{OP} jest minimalną liczbą dni potrzebnych do wykonania zadań ze zbioru J, a L_{AHZ} liczbą dni wyznaczoną przez wyżej opisany algorytm, to wówczas*

$$L_{\mathrm{AHZ}} \leq \frac{11}{9} L_{\mathrm{OP}} + 4.$$

∎

Powyższe twierdzenie zostało udowodnione w pracy [3].

W celu uproszczenia opisu algorytmu przyjęliśmy, że liczba godzin pracy centrum w każdym dniu jest stała. W ogólnym przypadku długości zmian mogą być różne.

System pracy ciągłej, tj. bez zatrzymywania maszyny, można traktować jako pracę zmianową, w której długość zmiany dz jest równa nieskończoności. Ze względu jednak na specyfikę tego problemu skonstruowaliśmy oddzielny, bardziej efektywny algorytm. Wyznaczając kolejności wykonywania zadań korzystamy ze znanej w literaturze *zasady Jacksona* stosowanej w wielu algorytmach dla problemów szeregowania zadań. Dokładnie jest ona przedstawiona w pracy [2]. Poniżej zamieszczamy jedynie krótki opis algorytmu, a sam algorytm jest przedstawiony w pracy [1].

Na początku przyjmujemy, że zajętość pracownika $T=0$, maszyny $t = 0$ oraz zbiór zadań ustalonych $H = \emptyset$. Niech

$$K = \{J_i \in J : J_i \notin H, B_i = \emptyset, r_i \leq t\},$$

będzie zbiorem zadań wolnych, których wykonywanie może się rozpocząć w chwili t. Jeżeli zbiór $K \neq \emptyset$, to wybieramy do wykonywania zadanie z K, dla którego wartość wyrażenia (2) jest minimalna. Gdy $K = \emptyset$, wtedy wyznaczamy zbiór

$$K' = \{J_i \in J : J_i \notin H, B_i = \emptyset\},$$

i do wykonywania wybieramy zadanie z K' o najmniejszym możliwym terminie rozpoczęcia. Łatwo wykazać, że jeżeli zbiór $K = \emptyset$, to $K' \neq \emptyset$. Po wyznaczeniu zadania do wykonywania modyfikujemy zajętości pracownika i maszyny oraz zbiory poprzedników i najwcześniejsze możliwe terminy rozpoczęcia wykonywania zadań tak, jak w algorytmie dla pracy zmianowej. Złożoność obliczeniowa opisanego algorytmu jest równa $O(n^2)$.

Wyniki obliczeniowe przedstawionych algorytmów, dla wielu losowo generowanych przykładów, jak również dla dużej liczby przykładów wziętych z praktyki, w pełni potwierdzają słuszność założeń dotyczących modelu procesu obróbki detali, jak i wyboru algorytmów.

4. Współdziałanie obrabiarki z magazynami narzędzi

Wchodząca w skład centrum obrabiarka wyposażona jest w magazyny narzędzi. Znajdujące się w nich narzędzia mogą nie wystarczyć do wykonania wszystkich operacji przewidzianych na dany dzień. Dlatego w trakcie obróbki operator dokonuje wymian.

Każde zadanie $J_i \in J$ jest ciągiem m_i ponumerowanych *operacji*:

$$J_i = (O_i^1, O_i^2, ..., O_i^{m_i}),$$

które należy wykonać w zadanym z góry *porządku technologicznym*, tzn. operacja O_i^k ma być wykonana bezpośrednio po operacji O_i^{k-1}, a przed O_i^{k+1}. Operacja może być: *tokarska,* *frezarska* lub *pomiarowa*. Niech p_i^k będzie czasem wykonywania, n_i^k numerem typu narzędzia, a z_i^k zmienną, której wartością jest numer otworu w magazynie, z którego maszyna pobiera narzędzie do wykonywania operacji O_i^k. Dalej niech s_j będzie numerem typu, a c_j *dopuszczalnym* (maksymalnym) *czasem pracy* narzędzia znajdującego się w j-tym otworze. Podobnie jak w przypadku zadań zakładamy, że:

a') w każdej chwili maszyna wykonuje co najwyżej jedną operację,

b') wykonywanie żadnej operacji nie może być przerwane,

c') każdą operację należy wykonać jednym narzędziem.

Problem sterowania pracą magazynów polega na wyznaczeniu:
- narzędzia do wykonywania każdej operacji $O_i^k \in J_i$, $(J_i \in J)$,
- momentów wymiany narzędzi w magazynach,
- zbiorów narzędzi wyjmowanych i wstawianych w trakcie wymiany,
- liczby narzędzi każdego typu potrzebnych do obróbki detali na każdy dzień pracy centrum

tak, aby liczba potrzebnych narzędzi oraz liczba wymian była jak najmniejsza.

Poniżej przedstawimy opis algorytmu sterowania pracą magazynów zakładając, że centrum pracuje w systemie zmianowym.

Załóżmy, że rozwiązaliśmy problem wyjściowy P harmonogramowania procesu obróbki, tj. wyznaczyliśmy uporządkowane zbiory zadań na każdy dzień. Ponieważ kolejność wykonywania operacji każdego zadania jest z góry określona (porządek technologiczny), mamy więc na każdy dzień pracy wyznaczoną kolejność wykonywania operacji.

Przypuśćmy, że przystępujemy do rozpatrywania operacji O_i^k, a wszystkie operacje, które mają być przed nią wykonane, zostały już rozpatrzone. Dla ustalenia uwagi przyjmijmy, że jest to operacja tokarska. Omówimy teraz pewne czynności związane z ustaleniem numeru otworu, z którego maszyna pobiera narzędzie do wykonania tej operacji. Jeśli w magazynie narzędzi brak jest takiego, którym można wykonać rozpatrywaną operację, a są jeszcze wolne otwory, to wykonujemy czynność *wstawiania* narzędzia do magazynu. Natomiast gdy wszystkie otwory są zajęte, to wykonujemy czynność *wymiany* polegającą na zwolnieniu pewnego otworu i następnie wstawieniu odpowiedniego narzędzia. Poniżej opiszemy obie czynności.

Wstawianie narzędzia do magazynu

Jeżeli w magazynie brak jest narzędzia, którym należy wykonać operację O_l^k, a są jeszcze wolne otwory, to wstawiamy nowe narzędzie. Czynność ta polega więc na wyznaczeniu wolnego otworu w magazynie i umieszczeniu w nim narzędzia, którym będzie wykonywana rozpatrywana operacja.

Wymiana narzędzi w magazynie

Załóżmy, że wśród narzędzi znajdujących się w magazynie nie ma takiego, którym można wykonać operację O_l^k, i wszystkie otwory są zajęte. Czynność wymiany polega na wyznaczeniu numeru pewnego otworu i usunięciu znajdującego się w nim narzędzia, a następnie wykonaniu czynności wstawiania. W pierwszej kolejności usuwać będziemy narzędzie, które nie będzie już używane do wykonywania pozostałych na dany dzień operacji. Jeśli nie ma takiego, to usuwamy narzędzie należące do najliczniejszego typu i mające najmniejszy dopuszczalny czas pracy.

Powróćmy obecnie do problemu wyznaczania numeru otworu w magazynie, z którego należy pobrać narzędzie do wykonywania operacji. Jeśli w magazynie jest wiele narzędzi typu n_l^k, którymi można wykonać operację O_l^k, to wybieramy spośród nich takie, dla którego różnica dopuszczalnego czasu pracy i czasu wykonania operacji jest nieujemna i minimalna, tj. wybieramy narzędzie z otworu q, dla którego:

$$c_q - p_l^k = \min\{j\colon j\text{- jest numerem otworu oraz } c_j - p_l^k \geq 0 \text{ i } s_j = n_l^k\}.$$

Następnie zmiennej z_l^k nadajemy wartość q (numer otworu, w którym znajduje się wybrane narzędzie) oraz modyfikujemy dopuszczalny czas pracy przydzielonego narzędzia c_q przyjmując $c_q - p_l^k$. Opisana metoda minimalizacji liczby narzędzi ustalonego typu, potrzebnych do wykonania wszystkich (tego typu) operacji, sprowadza się do znanego w literaturze problemu pakowania. Konstruując algorytm dla tego problemu będziemy stosowali strategię *najlepszego dopasowania* (ang. best-fit), z której korzystaliśmy przy wyznaczaniu rozwiązania przybliżonego dla problemu **P**. Dokładny jej opis zamieszczony jest w pracy [3], a oszacowanie uzyskanego rozwiązania daje poniższe twierdzenie.

TWIERDZENIE 3. *Jeżeli L_{opt} jest rozwiązaniem optymalnym, a L_{bf} rozwiązaniem otrzymanym w wyniku stosowania strategii najlepszego dopasowania dla problemu pakowania, to*

$$L_{bf} \leq \frac{17}{10} L_{opt} + 2.$$

■

Dowód powyższego twierdzenia zamieszczony jest w pracy [3].

5. Opis algorytmu sterowania magazynami maszyny

Zakładamy, że centrum pracuje w systemie zmianowym. W wyniku działania odpowiedniego algorytmu wyznaczone zostały uporządkowane zbiory zadań H_l, ($l = 1,2,\ldots,l_{max}$) do wykonania w poszczególnych dniach, gdzie l_{max} jest numerem ostatniego dnia pracy centrum. Ponadto niech d będzie liczbą otworów w każdym magazynie.

Poniżej zamieszczamy opis procedury wyznaczania dla operacji O_v^j numeru otworu, z którego pobierane jest narzędzie do wykonywania ustalonej operacji. W celu uproszczenia zapisu będziemy korzystali z elementów języka Pascal.

```
min := maxint; z_v^j:=0;
repeat
    for i := 1 to d do
    if (s_i = n_v^j) and (c_i ≥ p_v^j) then
    if min > c_i - p_v^j then
    begin                       {Wybieramy narzędzie z i - tego otworu}
        min := c_i - p_v^j;  z_v^j := i
    end;
    if  min = maxint then             {Brak odpowiedniego narzędzia}
    if (są wolne otwory) then wstaw narzędzie else wymiana
    until z_v^j ≠ 0
```

Opis czynności wstawiania i wymiany narzędzi w magazynie, występujących w procedurze, zamieściliśmy w poprzednim rozdziale.

Algorytm sterowania magazynami narzędzi podczas procesu obróbki detali sprowadza się do wielokrotnego stosowania wyżej opisanej procedury, kolejno dla każdej operacji. Dokładny jego opis zamieszczony jest w pracy [1]. Jego złożoność obliczeniowa jest równa $O(d * lop)$, gdzie lop jest liczbą wszystkich operacji, a d liczbą otworów w magazynie.

Czynności wstawiania, wymiany i dublowania narzędzi mogą być zrealizowane przez algorytmy, których złożoność obliczeniowa jest stała, tj. $O(1)$ i zależy tylko od liczby otworów w magazynie.

6. Eksperymenty obliczeniowe

Algorytmy harmonogramowania procesu obróbki detali dla pracy zmianowej i ciągłej, przedstawione w tej pracy, zostały zaprogramowane w języku Turbo Pascal i przetestowane na przykładach wziętych z praktyki. Obliczony dla tych przykładów współczynnik wykorzystania maszyny $\mathcal{W}$ był dwukrotnie większy niż dla harmonogramów układanych ręcznie. Algorytmy te były także testowane na wielu losowo wygenerowanych przykładach. Dla każdego zadania wszystkie jego parametry były realizacją zmiennej losowej o rozkładzie jednostajnym. Przedziały, z których losowano wartości zmiennych, wyznaczono tak, aby wartości rzeczywistych danych, dla których prowadziliśmy obliczenia, należały do tych przedziałów. Każdy przykład był liczony dla jedno- i dwuzmianowej oraz ciągłej pracy centrum. Oprócz współczynnika wykorzystania maszyny $\mathcal{W}$ liczono także wartość ilorazu:

$$\mathcal{W}_{OS} = \frac{\mathcal{F}_{LB}}{\mathcal{F}_{AH}},$$

gdzie $\mathcal{F}_{AH}$ jest terminem zakończenia obróbki detali ze zbioru J wyznaczonym przez algorytm odpowiednio dla pracy ciągłej lub zmianowej, natomiast wartość wyrażenia $\mathcal{F}_{LB}$ jest dolnym ograniczeniem czasu zakończenia wykonywania zadań ze zbioru J (dokładny opis obliczania tej wartości przedstawiony jest w pracy [1]).

W tablicy 1 przykładowo zamieszczone są wyniki obliczeniowe dotyczące przykładów, dla których najwcześniejsze terminy rozpoczęcia załadunku zadań są zerami oraz zbiór łuków grafu pierwszeństwa jest pusty.

Liczba zmian równa 3 oznacza pracę ciągłą.

Podobnie jak wyżej, generując losowo przykłady, testowaliśmy algorytm sterowania magazynami maszyny. Wyniki są podobne do tych, ktore zamieściliśmy. Czas obliczeń

Tablica 1

liczba zadań	liczba zmian	$\mathcal{W}$			Oszacowanie $\mathcal{W}_{OS}$		
		min.	*mediana*	*max.*	*min.*	*mediana*	*max.*
	1	.87	.90	.92	.93	.95	97
10	2	.93	.94	.95	.96	.97	.98
	3	.98	.98	.98	.99	.99	.99
	1	.90	.90	.90	.95	.95	.96
30	2	.95	.95	.96	.97	.98	.99
	3	.99	.99	.99	.99	.99	.99
	1	.91	.91	.93	.96	.96	.96
60	2	.95	.96	.96	.98	.98	.99
	3	.99	.99	.99	.99	.99	.99

każdego przykładu był nie większy niż 1 sekunda.

Otrzymane wyniki w pełni potwierdzają słuszność wybranej metody konstrukcji algorytmów rozwiązania problemu optymalnej pracy centrum.

Przedstawione w pracy algorytmy zostały zaprogramowane w języku Turbo Pascal i stanowią pakiet oprogramowania, który może być wykorzystywany do sterowania procesem produkcji w dyskretnych systemach produkcyjnych.

LITERATURA

1. Adrabiński A., Grabowski J., Wodecki M.: Optymalizacja pracy elastycznego centrum obróbczego tokarsko-frezarskiego. Mat. Stos., XXXVII, 1994, str. 16-37.

2. Jackson R.J.: Scheduling a production line to minimize maximum tardiness. Research Raport 43, Manag. Sci. Res. Proj., University of California, Los Angeles, 1955.

3. Johnson D.S., Demers A., Ullman J.D., Garey M.R., Graham R.L.: Worst-case performance bound for simple one-dimensional packing algorithms. SIAM, J.Comput., 3,4(1974).

Recenzent: Dr hab. inż. Mirosław Zaborowski, prof. Pol.Śl.

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

This paper is dealing with a computer-aided system for a turning-lathe machine. It includes two parts. The first one considers the sequencing problem of job processing by the machine. The problem is defined and a mathematical model along with an algorithm for optimizing the processing time is also presented. We also prove that the problem is NP-hard. The discussed job scheduling problem may be briefly stated as follow. We are given:

1. a set $J = \{J_1, J_2, ..., J_n\}$ of jobs,
2. a partial order RT on J,
3. a turning lathe machine and an operator.

Moreover each job J_i consists of a given sequence of operations $J_i = \langle O_i^1, O_i^2, ..., O_i^{m_i}\rangle$; each operation requires a specific tool for its processing. A machine can process only one job at a time and preemption of a job is not permitted. An operation time is given for each processing operation and a processing time p_i for a job J_i is equal to $\sum_{j=1}^{m_i} p_i^j$. Furthermore each job has a loading and an unloading time on a tray (platform). The loading and unloading is done by the operator who also exchanges tools in tools-store. The problem is to sequence the processing of the jobs on a machine to optimize some performance criterion related to the completion time. The second part of the paper considers the problem of the tool replacement by the turning-lathe. An algorithm for finding an optimal sequence of tool replacement intervals in the tool-store of the machine is presented. Similarly to the previous we assume that:

1. a machine can process only one operation at a time,
2. preemption of an operation is not permitted,
3. each operation requires only one specific tool.

Moreover the tool-store has a limited capacity. The problem is to find an optimal assignment of tools to operations so that the number of used up tools and tool exchanges is minimal. Extensive computational experience with a set of test problems ranging in size up to sixty jobs show a high efficiency of the system. A discussion of the performance of the implemented system is also presented.