

Antoni Korcyl, Tadeusz Sawik
Akademia Górniczo-Hutnicza

ALGORYTM TYPU TABU DLA WYZNACZANIA PARTII PRODUKCYJNYCH I OBCIĄŻENIA MASZYN W ELASTYCZNYM SYSTEMIE PRODUKCYJNYM

A TABU SEARCH ALGORITHM FOR PART TYPE SELECTION AND MACHINE LOADING IN A FLEXIBLE MANUFACTURING SYSTEM

ALGORITHME DU TYPE TABOU POUR SÉLECTION DES PIÈCES ET ALLOCATION DES MACHINES DANS UN SYSTÈME DE FABRICATION FLEXIBLE

Streszczenie: Przedstawiono modele programowania całkowitoliczbowego i nowy algorytm heurystyczny dla wyznaczenia partii produkcyjnych i obciążenia maszyn w elastycznym systemie produkcyjnym. Algorytm stanowi kombinację procedury ECT (ang. Earliest Completion Time) i heurystyki Tabu Search. Zamieszczono wyniki eksperymentów obliczeniowych, które potwierdzają korzystne własności zaproponowanego podejścia.

Summary: This paper presents integer programming models and a new heuristic algorithm for part type selection and machine loading in a flexible manufacturing system. The algorithm is a combination of ECT (Earliest Completion Time) and Tabu Search heuristic. Results of computational experiments with the algorithm are reported, which prove its practical value.

Resume: Cet article décrit un nouvel algorithme du type tabou pour sélection des pièces et allocation des machines et tâches dans un système de fabrication flexible. Des tests numériques indiquent que l'algorithme proposé se compare avantageusement à d'autres méthodes.

1. Wstęp

Przygotowanie produkcji w elastycznym systemie produkcyjnym wymaga rozwiązania dwóch podstawowych zadań krótkookresowego planowania produkcji (por. Sawik [8,9,11], Stecke i Kim [14]):

1. Podział zlecenia produkcyjnego na partie typów części przeznaczonych do jednoczesnego wykonywania,
2. Obciążenie maszyn, tzn. przydział operacji wraz z narzędziami dla każdej partii do poszczególnych maszyn z magazynkami narzędzi o ograniczonych pojemnościach.

Wymienione zadania powinny być rozwiązywane jednocześnie. Podział na partie typów części i obciążenie maszyn dla każdej partii należałoby wyznaczyć tak, aby zapotrzebowania na wszystkie produkowane wyroby zaspokoić np. w najkrótszym czasie. Jednak złożoność obliczeniowa takiego zadania w praktyce wymaga jego dekompozycji i kolejnego rozwiązywania zadań składowych 1 i 2. Ponadto, każde z zadań składowych jest już samo dostatecznie złożone, aby usprawiedliwić stosowanie różnych algorytmów heurystycznych (np. Afentakis, i in. [1], Hwang [5], Rajagopalan [7]).

W artykule przedstawiony zostanie nowy algorytm heurystyczny dla jednoczesnego podziału zlecenia produkcyjnego na partie produkcyjne i obciążenia maszyn w ramach każdej partii. W algorytmie poszukiwanie rozwiązania zadania podziału na partie produkcyjne dokonywane jest za pomocą metody Tabu Search (Glover [2,3,4]) natomiast obciążenie maszyn wyznacza się za pomocą heurystyki ECT (ang. Earliest Completion Time), (np. Sawik [11]), stosowanej dla szeregowania operacji na maszynach równoległych, którą odpowiednio zmodyfikowano w celu uwzględnienia ograniczeń narzędziowych.

Tablica 1

		Notacja
		Indeksy
g	typ narzędzia $g \in G$	
i	maszyna $i \in I = \{1, \dots, m\}$	
j	operacja $j \in J = \{1, \dots, n\}$	
k	typ części $k \in K = \{1, \dots, q\}$	
		Parametry wejściowe
a_{gj}	1, jeśli wykonanie operacji j wymaga użycia narzędzia typu g 0, inaczej	
$J(i)$	zbiór operacji wykonywalnych na maszynie i	
J_k	zbiór operacji dla części typu k	
m	liczba maszyn	
n	liczba operacji	
n_k	liczba operacji dla części typu k ($n_k = J_k $)	
p_{ij}	czas wykonywania operacji j na maszynie i	
q	liczba typów części	
r_k	zapotrzebowanie na typ części k (liczba części do wykonania)	
s_g	liczba gniazd narzędziowych zajmowanych w magazynku narzędzi przez narzędzie typu g	
S_i	pojemność magazynka narzędzi maszyny i	
		Zmienne decyzyjne
C_{max}	maksymalne obciążenie maszyny	
e_{gi}	1, jeśli narzędzie typu g należy załadować do magazynku maszyny i 0, inaczej	
x_{ij}	liczba operacji j przydzielonych do maszyny i	
z_k	1, jeśli część typu k wchodzi w skład danej partii produkcyjnej 0, inaczej	

2. Model matematyczny zadania

Rozważmy elastyczny system produkcyjny składający się z m maszyn różnego typu. Zbiór maszyn $I = \{1, \dots, m\}$ obejmuje obrabiarki, maszyny do czyszczenia oraz stanowiska załadunkowe i wyładunkowe. W systemie wytwarza się różne typy części. Do wyprodukowania części typu $k \in K$ konieczne jest wykonanie ciągu operacji ze zbioru $J_k \in J = \{1, \dots, n\}$. Oznaczmy przez p_{ij} czas wykonywania operacji j na maszynie i . Każda maszyna wyposażona jest w lokalny magazynek narzędzi o pojemności S_i gniazd narzędziowych. Wykonanie operacji j wymaga użycia odpowiednich typów narzędzi, przy czym narzędzie typu $g \in G$ zajmuje w magazynku s_g gniazd narzędziowych. Dla każdego typu części $k \in K$ zadane jest zapotrzebowanie r_k (wymagana liczba części typu k). Oznacza to, że każda operację $j \in J_k$ należy wykonać r_k razy.

Organizacja produkcji w ESP przeważnie wymaga, aby wszystkie narzędzia konieczne do wykonania wybranego zestawu typów części znalazły się w magazynkach narzędzi odpowiednich obrabiarek przed rozpoczęciem produkcji. Z drugiej strony ograniczona pojemność magazynków narzędzi uniemożliwia jednoczesną produkcję wszystkich typów części, na które istnieje zapotrzebowanie. Stąd pojawia się następujące zadanie, któremu poświęcono ten artykuł: ze zbioru K wszystkich typów części należy wybrać największy podzbiór, zapewniający najlepsze wypełnienie magazynków narzędzi i jednocześnie rozdzielić wymagane operacje wraz z narzędziami pomiędzy maszyny tak, aby zrównoważyć obciążenia wszystkich maszyn, gdzie obciążenie maszyny oznacza łączny czas wykonywania wszystkich przydzielonych operacji. Zadanie to powtarzamy, dopóki w zbiorze K znajdują się typy części, na które istnieje zapotrzebowanie.

Model PTS&ML (Part Type Selection & Machine Loading) zadania podziału zlecenia na partie pro-

dukcyjne z jednoczesnym obciążeniem maszyn przedstawiono poniżej:

$$\max \sum_{k \in K} \left(\sum_{g \in G} \sum_{j \in J_k} a_{gj} s_g \right) z_k \quad (1)$$

przy ograniczeniach

$$\sum_{i \in I} x_{ij} = r_k z_k \quad j \in J_k, k \in K \quad (2)$$

$$\sum_{j \in J} p_{ij} x_{ij} \leq C_{max} \quad i \in I \quad (3)$$

$$\sum_{g \in G} s_g c_{gi} \leq S_i \quad i \in I \quad (4)$$

$$\sum_{j \in J} a_{gj} x_{ij} \leq \left(\sum_{k \in K} n_k r_k \right) e_{gi} \quad j \in G, i \in I \quad (5)$$

$$c_{gi} \in \{0, 1\} \quad g \in G, i \in I \quad (6)$$

$$x_{ij} \geq 0, \text{ całkowite } i \in I, j \in J \quad (7)$$

$$z_k \in \{0, 1\} \quad k \in K \quad (8)$$

$$C_{max} \geq 0 \quad (9)$$

Funkcja celu (1) zapewnia wybór w skład partii produkcyjnej największej liczby typów części, których wykonanie wymaga narzędzi najlepiej łącznie wypełniających magazynki narzędzi.

Ograniczenie (2) zapewnia rozdział pomiędzy maszyny wymaganej liczby operacji dla każdego typu części wybranego w skład partii.

Ograniczenie (3) zapewnia zrównoważenie obciążeń wszystkich maszyn tak, aby wszystkie maszyny równocześnie kończyły wykonywanie przydzielonych operacji. Umożliwia to równoczesne rozpoczęcie wymiany narzędzi dla kolejnej partii, w magazynkach wszystkich maszyn.

Zauważmy, że C_{max} oznacza łączny czas wykonywania wszystkich operacji przydzielonych do maszyny stanowiącej "wąskie gardło" systemu. Wielkość ta może być podstawą dolnego oszacowania długości harmonogramu produkcji odpowiadającego wybranym przydziałom operacji do maszyn. Nie uwzględnia ona jednak przestoju maszyn związanych z oczekiwaniem na kolejne części do obróbki.

Ograniczenie (4) zabezpiecza przed przekroczeniem pojemności magazynków narzędzi, a (5) wiąże przydział operacji do maszyny z jednoczesnym przydziałem odpowiednich narzędzi.

Aby obniżyć złożoność obliczeniową zadania przeprowadza się dekompozycję modelu PTS&ML na następujące dwa kolejno rozwiązywane problemy:

1. Wyznaczanie partii produkcyjnej Model PTS (Part Type Selection):

$$\max \sum_{k \in K} \left(\sum_{g \in G} \sum_{j \in J_k} a_{gj} s_g \right) z_k \quad (10)$$

przy ograniczeniach

$$a_{gj} z_k \leq c_{gi}, \quad g \in G, i \in I, j \in J(i) \cap J_k \quad (11)$$

oraz (4),(6),(8) jak w modelu PTS&ML

2. Obciążenie maszyn Model ML (Machine Loading):

$$\min C_{max} \quad (12)$$

przy ograniczeniach

(3),(4),(5),(6),(7),(9) jak w modelu PTS&ML

Dekompozycja modelu PTS&ML pozwala na dwuetapowe rozwiązywanie zadania podziału zlecenia na partie produkcyjne i obciążenie maszyn. Wyniki rozwiązania modelu PTS stanowią dane wejściowe dla modelu ML w drugiej fazie rozwiązywania problemu.

3. Opis metody rozwiązania

Zaproponowana w dalszym ciągu heurystyka ECT_Tabu_Search jest algorytmem, który jednocześnie dokonuje podziału zlecenia produkcyjnego na partie produkcyjne, a w ramach każdej partii zrównoważa obciążenia maszyn. Algorytm dla rozwiązania zadania obciążenia maszyn z uwzględnieniem ograniczeń narzędziowych zbudowano na podstawie heurystyki ECT. W algorytmie tym każda kolejna operacja, wybrana spośród operacji oczekujących na wykonanie, jest przydzielana do takiej maszyny, na której zostanie ona najwcześniej ukończona. Natomiast podział zlecenia produkcyjnego na partie produkcyjne jest dokonywany na bazie metody Tabu Search. Metoda ta w sposób iteracyjny dąży do takiego podziału zlecenia, dla którego liczba typów części w każdej partii będzie możliwie największa oraz będą spełnione ograniczenia wynikające z rozwiązania w każdej iteracji zadania obciążenia maszyn przy ograniczonych pojemnościach magazynków narzędzi.

3.1. Metoda Tabu Search

Tabu Search jest nowoczesną metodą heurystyczną stosowaną do rozwiązywania problemów optymalizacji dyskretnej. Została ona skonstruowana w celu uniknięcia pułapki lokalnego optimum podczas poszukiwania rozwiązań za pomocą innych metod. W wyniku zastosowania metody Tabu Search otrzymuje się rozwiązania optymalne lub bardzo bliskie optymalnemu, co potwierdzają liczne zastosowania tego podejścia w różnych dziedzinach [4]. Metoda wykorzystuje elastyczne struktury pamięci, co pozwala na bardziej szczegółowe przeszukiwanie obszaru rozwiązań dopuszczalnych niż przy zastosowaniu systemów o sztywnych strukturach pamięci lub systemów bez pamięci. Ponadto wykorzystuje warunki dla strategicznego dodawania ograniczeń i usuwania ich w trakcie poszukiwania rozwiązania, oraz funkcje pamięci służące do intensyfikacji i wzbogacania procesu poszukiwania rozwiązania poprzez wypuklanie cech historycznie uznanych za dobre podczas procesu poszukiwania rozwiązania oraz kierowanie poszukiwań w nowe regiony. Metoda Tabu Search może być łączona z innymi metodami, np. metodą podziału i ograniczeń, oraz umożliwiała rozpoczęcie poszukiwania rozwiązania za pomocą uproszczonej procedury, która w trakcie procesu obliczeń może być zmieniana poprzez wprowadzanie bardziej zaawansowanych i specjalistycznych elementów wymaganych w danym zastosowaniu. Metodę Tabu Search można stosować do sterowania procesem poszukiwania, który wykorzystuje zbiór posunięć przekształcających jedno rozwiązanie (lub zespół rozwiązań) w inne i jednocześnie pozwala na oszacowanie atrakcyjności takiego posunięcia. Pod pojęciem atrakcyjności posunięcia należy rozumieć stopień spełnienia ograniczeń przez wartości zmiennych wybrane w danym rozwiązaniu.

Metoda oparta jest na elementach sztucznej inteligencji. Głównymi jej składnikami są pamięć krótkoterminowa, pamięć długoterminowa i funkcja kryterium kwalifikacji.

Najważniejszym elementem algorytmu Tabu Search jest *lista tabu* zwana też *pamięcią krótkoterminową*, składająca się ze zmiennych, których wejście w skład rozwiązania w danym kroku nie jest dozwolone. Są to zmienne, które w poprzednich kilku krokach były wybierane do rozwiązania. Zapobiega to powstawaniu cykli przy wyborze zmiennych do rozwiązania oraz pozwala na unikanie pułapek lokalnych rozwiązań optymalnych. *Lista tabu* może posiadać ściśle określoną wielkość, lub też może ulegać zmianom w trakcie uruchamiania algorytmu. Jest aktualizowana po każdym kroku (określeniu zestawu zmiennych wchodzących do rozwiązania), tzn. wprowadzenie każdej nowej zmiennej do rozwiązania ($z_k = 1$) powoduje usunięcie innej zmiennej znajdującej się na liście najdłuższej ($z_l = 0$), jeżeli jest to dopuszczalne. Z *listą tabu* ściśle związana jest inna lista pozwalająca określić status zmiennej, czyli określić, od ilu iteracji dana zmienna jest wybierana do rozwiązania i znajduje się na *liście tabu*.

Jeżeli na *liście tabu* znajdują się tylko takie zmienne, które nie mogą być z niej usunięte, ponieważ nie były odpowiednią liczbę razy wybierane do rozwiązania, wtedy stosuje się *funkcję kryterium kwalifikacji*, która pozwala po spełnieniu odpowiednich, z góry narzuconych warunków, na zmianę statusu zmiennej

znajdującej się na liście tabu i usunięcie jej z listy oraz wprowadzenie w jej miejsce nowej zmiennej do rozwiązania oraz na listę.

Urozmaicenie oraz intensyfikację poszukiwań otrzymuje się poprzez wprowadzenie odpowiednio zdefiniowanej funkcji pamięci długoterminowej. Pozwala to na skierowanie poszukiwań w rejony dotychczas nie rozpatrywane lub na włączenie innych ograniczeń do rozważań. Pamięć długoterminowa jest uaktywniana, jeżeli nie jest możliwe wprowadzenie nowej zmiennej do rozwiązania, oraz gdy użycie funkcji kryterium kwalifikacji nie powoduje żadnej zmiany na liście tabu, a jednocześnie nie wszystkie regiony obszaru rozwiązań dopuszczalnych zostały zbadane.

Algorytm Tabu Search jest przerywany, gdy osiągnięte rozwiązanie jest wystarczająco bliskie oszacowania rozwiązania optymalnego, lub gdy w ciągu określonej liczby iteracji otrzymane rozwiązanie nie jest poprawiane. Maksymalna liczba iteracji jest funkcją rozmiaru problemu.

4. Heurystyka złożona ECT_Tabu_Search

Zadanie obciążenia maszyn z uwzględnieniem ograniczeń narzędziowych rozwiązywane jest za pomocą algorytmu ECT. W algorytmie tym spośród wszystkich operacji oczekujących na wykonanie każdą kolejno wybraną operację przydziela się do takiej maszyny, na której zostanie ona najwcześniej ukończona. Niech T_i będzie łącznym czasem wykonywania operacji już przydzielonych do maszyny i , a J oznacza zbiór operacji nie przydzielonych jeszcze wymagana liczbę razy.

Zmodyfikowany algorytm ECT jest następujący:

Krok 1. Przyjmij $T_i = 0$, $i = 1, \dots, m$; $J = \{1, \dots, n\}$.

Krok 2. Dla wszystkich niewykonanych operacji dla wszystkich typów części znajdujących się w zleceniu produkcyjnym wyznacz $t_i = \min_i \{T_i + p_{ij}\}$ i uporządkuj t_i w kolejności niemalejących wartości. Sprawdź, czy są operacje wykonywalne tylko na jednej maszynie (Unique.Operation.Assignment). Jeżeli tak, to sprawdź, czy wymagane narzędzia zmieszczą się w magazynku narzędzi tej maszyny, jeżeli wcześniej nie zostały już przydzielone i przydziel takie operację do maszyny r_j razy.

Podstaw $J = J \setminus \{j\}$, $r_j = 0$ $T_i = T_i + r_j \cdot p_{ij}$

Jeśli nie, to usuń tę operację ze zlecenia i odpowiednio zaktualizuj S oraz T_i . Następnie wybierz takie p_{ij} , dla którego t_i jest najmniejsze.

Krok 3. Dla wybranej operacji, która ma być przydzielona do maszyny i , sprawdź, czy wymagane narzędzia zmieszczą się w jej magazynku, jeżeli wcześniej nie zostały przydzielone. Jeżeli narzędzia nie zmieszczą się, to wybierz maszynę, której odpowiada następna z kolei wartość t_i i dokonaj ponownego sprawdzenia. Powtarzaj tę procedurę aż do uzyskania wyniku pozytywnego.

Podstaw $T_i = T_i + p_{ij}$ oraz $r_j = r_j - 1$. Jeżeli $r_j = 0$, to przejdź do *Kroku 4*. Jeżeli dla danej operacji j nie jest możliwe przydzielenie narzędzi do żadnej z maszyn, to usuń operację j oraz wszystkie operacje dla danego typu części k i odpowiednio zaktualizuj wartości T_i . Wróć do *Kroku 2*.

Krok 4. Podstaw $J = J \setminus \{j\}$. Jeżeli $j > n$, to zakończ obliczenia. Inaczej wróć do *Kroku 2*.

Procedura ta jest uaktywniana w momencie sprawdzania, czy zmienna z_k wprowadzana do rozwiązania ($z_k = 1$) spełnia wszystkie ograniczenia, lub podczas usuwania zmiennej z_k z rozwiązania ($z_k = 0$).

Przypomnijmy, że wybór $z_k = 1$ oznacza wejście części typu k w skład partii produkcyjnej, zaś $z_k = 0$ usunięcie części typu k z partii.

Poszukiwanie rozwiązania optymalnego dla zadania podziału zlecenia na partie produkcyjne w zaproponowanym algorytmie bazuje na algorytmie Tabu Search.

Algorytm rozpoczyna się od sprawdzenia, czy możliwe jest wykonanie całego zlecenia w jednej partii produkcyjnej i obciążenie maszyn zgodnie z powyżej opisanym algorytmem ECT. Jeśli nie jest to możliwe, wtedy następuje określenie wielkości początkowej listy tabu Tabu_Size_Start, zależnej od liczby typów części znajdujących się w zleceniu. Ponieważ kryterium optymalności problemu podziału zlecenia produkcyjnego na partie produkcyjne dąży do maksymalizacji liczby typów części w partii, wielkość listy tabu jest powiększana w trakcie uruchamiania algorytmu, jeżeli zaistnieją takie warunki. Lista tabu jest powiększana o stałą wartość Tabu_Size_Increment. Parametr Tabu_Size_Increment może przyjmować różne wartości w

zależności od liczby typów części w zleceniu. Istotnym parametrem jest liczba iteracji Max.Iteration, po osiągnięciu której następuje zatrzymanie algorytmu, jeśli wcześniej nie wyznaczono rozwiązania optymalnego. W zaproponowanym algorytmie $ECT_TS \text{ Max.Iteration} = 3q$ (gdzie q jest liczbą typów części w zleceniu).

Spśród wszystkich zmiennych z_k , z których może być wyznaczone rozwiązanie, wybierane jest n zmiennych zgodnie z rozmiarem listy tabu. Rozwiązanie składające się z tych zmiennych jest przyjmowane jako najlepsze - Best.Solution. Zmienne wchodzące do tego rozwiązania zostają umieszczone na liście tabu, tzn. każda zmienna wprowadzana do rozwiązania zmienia swoją wartość z zera ($z_k = 0$) na jeden ($z_k = 1$). Jednocześnie wraz z listą tabu jest uaktualniana druga lista ściśle z nią powiązana - Tabu.Status, która podaje, jak długo zmienna znajduje się na liście tabu. Maksymalny czas przebywania zmiennej na liście tabu jest określony parametrem Tabu, także zależnym od liczby typów części w zleceniu. Jeśli Tabu.Status zmiennej osiągnie wartość Tabu, wtedy jest możliwe usunięcie zmiennej z rozwiązania oraz z listy tabu ($z_k = 0$), a w jej miejsce wprowadzenie innej zmiennej do rozwiązania.

Wybór nowej zmiennej do rozwiązania jest jednym z ważniejszych elementów algorytmu. W pierwszym kroku tego etapu ustalamy listę wszystkich zmiennych, które mogą być wprowadzone do rozwiązania. Polega to na zbudowaniu ciągu $[k_1, k_2, \dots, k_q]$ uporządkowanych nierosnąco wag $\rho_1 \geq \rho_2 \geq \dots \geq \rho_q$ (gdzie waga $\rho_k = \sum_{g \in G} \sum_{j \in J_k} a_{gj} s_{gj}$, oznacza łączną liczbę gniazd narzędziowych niezbędnych do załadowania narzędzi koniecznych do wykonania wszystkich operacji dla części typu k). Do rozwiązania wybierane są w pierwszej kolejności te typy części, które wymagają największej liczby gniazd dla załadowania wszystkich narzędzi. Wybór kolejnego typu części do partii produkcyjnej następuje poprzez przeglądnięcie wektora k rozpoczynając od k_1 do k_q . Polega to na sprawdzeniu, czy jest możliwe wykonanie wszystkich operacji dla wybranego typu części łącząc z pozostałymi operacjami dla wszystkich typów części wchodzących w skład partii (poprzez rozwiązanie zadania obciążenia maszyn - Machine.Loading). Dodatkowo sprawdza się, czy wartość funkcji celu New.Solution jest większa od dotychczasowej wartości funkcji celu, Best.Solution.

Znaleziona zmienna otrzymuje status tabu i jest wprowadzana na listę tabu. Następnie przystępujemy do poszukiwania zmiennej, która zostanie usunięta z rozwiązania, gdyż na tym etapie poszukiwań lista tabu ma początkowy rozmiar. Poszukiwanie zmiennej usuwanej przeprowadza się poprzez przeglądnięcie wektora k , rozpoczynając od k_q aż do k_1 . Jeśli znaleziona zmienna znajduje się najdłużej na liście tabu, jej status tabu przekroczył wartość Tabu (liczba iteracji, którą wybrany w skład rozwiązania typ części powinien znajdować się na liście tabu), a jej odrzucenie nie spowoduje pogorszenia wartości funkcji celu otrzymanej po dodaniu poprzedniej zmiennej, to zmienna ta zostaje usunięta z rozwiązania, a wszystkie związane z nią parametry i wskaźniki zostają uaktualnione oraz przeprowadzane jest ponowne obciążenie maszyn dla nowej listy zmiennych (Machine.Loading).

W przypadku gdy na liście tabu znajdują się tylko zmienne, które zostały do niej dodane w ostatnich iteracjach, wtedy pomocna staje się funkcja kryterium kwalifikacji pozwalająca na modyfikację listy tabu. Zmianę statusu tabu zmiennej niedostępnej w danym kroku można uzyskać, gdy istnieje możliwość polepszenia wartości funkcji celu lub włączenia do poszukiwań rozwiązań do tej pory nie sprawdzanych.

Procedura poszukiwania zmiennych wchodzących do rozwiązania trwa do momentu, gdy nie jest możliwe ani dodanie nowej zmiennej, ani też usunięcie zmiennej pomimo zastosowania kryterium kwalifikacji.

Intensyfikacja oraz urozmaicenia poszukiwań dokonuje się poprzez zastosowanie określonej funkcji pamięci długoterminowej. Ta procedura algorytmu jest uaktywniana po kilku pierwszych krokach, gdy nie następuje zmiana wartości funkcji celu oraz nie jest możliwe wprowadzenie nowych zmiennych do rozwiązania.

W algorytmie ECT_Tabu_Search rolę funkcji pamięci długoterminowej pełni częstość występowania każdej zmiennej w czasie poszukiwania rozwiązań na liście tabu. Związany jest z nią parametr List.Tabu.Frequency, który informuje, ile razy dana zmienna była wybierana i usuwana z rozwiązania optymalnego. Częstość pojawiania się zmiennej w rozwiązaniu jest wykorzystywana przy budowie nowej listy tabu, w której skład wchodzi w początkowej fazie zmienne najczęściej pojawiające się dotąd na liście tabu. Powoduje to skierowanie poszukiwań w regiony do tej pory nie rozpatrywane i intensyfikację poszukiwań rozwiązania poprzez rozszerzenie rozmiaru listy tabu.

Algorytm jest wykonywany, dopóki nie zostanie osiągnięte rozwiązanie optymalne lub znalezione rozwiązanie nie ulega polepszeniu w ciągu określonej liczby iteracji Max.Iteration.

Uproszczony opis programu ECT_Tabu_Search przedstawiono na rys. 1.

```
Program ECT.Tabu_Search;
Declarations;
begin
  Data;
  Unique_Operation_Assignments;
  Machine>Loading ( ECT );
  if (Requirements not Completed)
  then
  begin
    Weight_Part_Type;
    Part_Type_Sort;
    Make_Set_List;
    Tabu_Size := Tabu_Size.Start;
    Tabu_List := Tabu_List.Initialize;
    Solution := 0;
    Best_Solution := Solution;
    Number_of_Iteration := 0;
    Unique_Operation_Assignments;
    Machine>Loading (ECT);
    repeat
      repeat
        Number_of_Iteration := Number_of_Iteration + 1;
        Add_Part_Type;
        Unique_Operation_Assignments;
        Machine>Loading (ECT);
        If (Requirements Completed) and (New_Solution < Best_Solution)
        then
          begin
            Tabu_List := Update_Tabu_List;
            Tabu_Status := Update_Tabu_Status;
            Tabu_List_Frequency := Update_Tabu_List_Frequency;
            Best_Solution := New_Solution;
          end;
        Remove_Part_Type;
        Unique_Operation_Assignments;
        Machine>Loading (ECT);
      until New_Solution = Best_Solution;
      Tabu_Size := Tabu_Size + Tabu_Size.Increment;
    until Number_of_Iteration = Max_Iteration;
  end;
  Print_Results;
end.
```

Rysunek 1. Algorytm ECT.Tabu_Search
Figure 1. Algorithm ECT.Tabu_Search

5. Przykład liczbowy

Dla ilustracji rozważmy przykład elastycznego systemu produkcyjnego składającego się z $m = 5$ maszyn różnego typu. Zlecenie produkcyjne obejmuje 4 typy części. Należy wytworzyć odpowiednio $r_1 = 5$ części typu 1, $r_2 = 4$ części typu 2, $r_3 = 4$ części typu 3 oraz $r_4 = 2$ części typu 4. Czasy wykonywania operacji dla każdego typu części na różnych maszynach oraz wymagane narzędzia wraz z liczbami wymaganych gniazd narzędziowych w magazynkach maszyn podano w Tabelcy 2.

Tabelca 2
Czasy wykonywania operacji oraz wymagane typy narzędzi

Typ części k Operacja j	1			2				3			4			S _i
	1	2	3	4	5	6	7	8	9	10	11	12	13	
i = 1	4	2		1	1	8	7	6	7	8	8	5	4	12
2	3	7		4	1	8	3			5	4	5	4	12
3	4	4	3	7		9		3	3			1		12
4	2	8	8		7	1		9	9	3	3	6		12
5	1		8	5	7	8	8	6	1	8	2	5	8	12
Wymagane typy narzędzi	9	11	13	3	3	10	1	3	3	4	3	4	1	
		14		7	4		5	7	10	7	13	6	4	
				10			7		13	13	14	8	11	
				14			11		15	15	15	10		
$\sum_{i=1}^5 r_i$	1	6	3	6	2	1	6	2	8	8	10	4	5	

W Tabelcy 3 zamieszczono przydziały operacji do maszyn (zmienne x_{ij}) wyznaczone poprzez rozwiązanie zadań programowania całkowitoliczbowego (a) = PTS&ML ($x_{ij}^{(a)}$) i (b) = PTS/ML ($x_{ij}^{(b)}$) za pomocą pakietu programowania całkowitoliczbowego LINGO [12] oraz za pomocą algorytmu (c) = ECT_Tabu_Search ($x_{ij}^{(c)}$).

Tabelca 3

Przydziały operacji $x_{ij}^{(a)}/x_{ij}^{(b)}/x_{ij}^{(c)}$

Typ części k Operacja j	1			2				3			4		
	1	2	3	4	5	6	7	8	9	10	11	12	13
i = 1		5/5/5		-3/-	2/4/2			1/1/-					-1/1
2	-3/-				2/-/2		4/3/4			-1/-			2/1/1
3			4/3/4	-1/-				3/3/3	-1/1			1/2/2	
4	-1/-		-1/-			4/4/4				4/3/4	-1/1	1/-	
5	5/2/4		1/1/1				-1/-	-1/1	4/4/3			2/1/1	

W wyniku zastosowania modelu PTS&ML całe zlecenie zgrupowano w jednej partii produkcyjnej, dla której maksymalne obciążenie maszyn wynosi $C_{max}^{PTS\&ML} = 22$. Czas obliczeń wyniósł $CPU^{PTS\&ML} = 8$ godz.27min.37sek.

Zastosowanie modelu zdekomponowanego PTS/ML doprowadziło do podziału zlecenia na cztery partie produkcyjne {3}, {2}, {1}, {4} o maksymalnych obciążeniach maszyn wynoszących odpowiednio $C_{max}^I = 9$, $C_{max}^{II} = 9$, $C_{max}^{III} = 10$, $C_{max}^{IV} = 4$ i łącznym obciążeniu maksymalnym równym $C_{max}^{PTS/ML} = 27$. Łączny czas obliczeń wyniósł $CPU^{PTS/ML} = 13$ min.25sek.

W wyniku zastosowania heurystyki ECT_Tabu_Search zlecenie podzielone zostało na dwie partie produkcyjne {1,2,3}, {4}, dla których wartości C_{max} wynoszą odpowiednio 24 i 4, natomiast łączne obciążenie maksymalne $C_{max}^{ECT-TS} = 26$. Czas wykonywania obliczeń łącznie dla obu partii produkcyjnych wynosi $CPU^{ECT-TS} = 22.41$ sek.

Wszystkie obliczenia przeprowadzono na mikrokomputerze typu PC 386/33MHz z koprocesorem arytmetycznym.

Wyniki symulacji komputerowych

Tablica 4

Liczba maszyn m	Liczba typów części q	Liczba operacji n	$\epsilon_{C_{max}}$ %		ϵ_{CPU} %	
			ECT_TS	PTS/ML	ECT_TS PTS&ML	PTS/ML PTS&ML
3	5	15	6	10	9.07	20.4
	10	30	12	12	73.2	46.30
	10	50	5	12	80.13	92.12
5	5	15	8	23	26.14	86.06
	10	30	16	21	26.95	112.30
	10	50	18	20	35.45	106.08
10	5	15	7	12	11.35	133.70
	10	30	20	18	10.42	101.21
	10	50	17	24	84.72	98.76

6. Wyniki symulacji komputerowych

W celu sprawdzenia poprawności działania zaproponowanego algorytmu zostało przeprowadzonych szereg symulacji komputerowych. Dane wejściowe do obliczeń były generowane losowo za pomocą specjalnie zbudowanego generatora. Liczba maszyn m zmieniała się od 3 do 10. Dla każdego zestawu maszyn i ustalonej liczby typów części generowane były ciągi operacji dla każdego typu części, czasy wykonywania operacji na poszczególnych maszynach oraz liczba i typy narzędzi koniecznych do ich wykonania. Całkowita liczba typów narzędzi zmieniała się od 15 do 50, a wymagana liczba gniazd narzędziowych dla pojedynczego narzędzia s_p od 1 do 3. Generowano również wartości S_i pojemności magazynków narzędzi.

Wyniki obliczeń oceniono za pomocą następujących współczynników:

1. Współczynniki względnego przyrostu obciążenia maszyn

$$\epsilon_{C_{max}}^{(c)} = \frac{C_{max}^{ECT_TS} - C_{max}^{PTS\&ML}}{C_{max}^{PTS\&ML}} 100\%$$

$$\epsilon_{C_{max}}^{(b)} = \frac{C_{max}^{PTS/ML} - C_{max}^{PTS\&ML}}{C_{max}^{PTS\&ML}} 100\%$$

2. Współczynniki względnego skrócenia czasu obliczeń

$$\epsilon_{CPU}^{c/a} = \frac{CPU^{ECT_TS}}{CPU^{PTS\&ML}} 100\%$$

$$\epsilon_{CPU}^{b/a} = \frac{CPU^{PTS/ML}}{CPU^{PTS\&ML}} 100\%$$

W podanych wyżej współczynnikach $\epsilon_{C_{max}}$ maksymalne obciążenie maszyn $C_{max}^{PTS\&ML}$ dla modelu PTS&ML stanowi wartość optymalną C_{max} . Względne przyrosty $C_{max}^{ECT_TS}$ oraz $C_{max}^{PTS/ML}$ w stosunku do tej wartości pozwalają na ocenę dokładności, odpowiednio algorytmu ECT_Tabu_Search oraz modelu zdekomponowanego PTS/ML.

Jednocześnie współczynniki ϵ_{CPU} umożliwiają ocenę oszczędności czasu obliczeń związanych z dwoma ostatnimi podejściami, w stosunku do czasu wymaganego dla rozwiązania dokładnego modelu PTS&ML.

Wszystkie obliczenia wykonano na mikrokomputerze PC 386/33MHz z koprocesorem arytmetycznym, przy czym program komputerowy dla heurystyki ECT_Tabu_Search napisano w języku Turbo Pascal.

Zbiórce wyników symulacji komputerowych zostały zamieszczone w Tablicy 4.

7. Podsumowanie

Wyniki eksperymentów obliczeniowych wskazują na korzystne własności zaproponowanego algorytmu ECT_Tabu_Search. Otrzymywane wartości funkcji celu, jakim była maksymalizacja liczby typów części w partii produkcyjnej, były podobne bądź takie same jak w przypadku zastosowania modeli programowania całkowitoliczbowego i standardowego pakietu optymalizacji dyskretnej. Natomiast maksymalne obciążenie maszyn C_{max} otrzymane w wyniku zastosowania algorytmu ECT_Tabu_Search nie różniło się od rozwiązań optymalnych więcej niż o 20%. Jednocześnie czas wyznaczania rozwiązania za pomocą algorytmu ECT_Tabu_Search był od 10 do 400 razy krótszy od czasu obliczeń dla modelu PTS&ML oraz od 10 do 100 razy krótszy niż w przypadku modelu PTS/ML. Wystarczająca dla praktyki produkcyjnej dokładność wyznaczanych rozwiązań oraz krótkie czasy obliczeń wskazują na możliwości zastosowania algorytmu ECT_Tabu_Search w ESP. Algorytm ten może być ponadto rozbudowywany tak, aby uwzględnić dodatkowe elementy takie jak ograniczona liczba palet i uchwytów obróbkowych, ograniczona żywotność i liczba narzędzi każdego typu, a także czasy transportu i liczba wózków transportowych w systemie.

Literatura

- [1] Afentakis P., Solomon M.M., Millen R.A.: The part-type selection problem. W: K.E.Stecke, R.Suri (red.), *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, Elsevier, Amsterdam, 1989, s.141-146.
- [2] Glover F.: Tabu Search, Part I. *ORSA Journal on Computing*, vol. 1, no. 3, 1989, s.190-206.
- [3] Glover F.: Tabu Search, Part II. *ORSA Journal on Computing*, vol. 2, no. 1, 1990, s.4-32.
- [4] Glover F.: Tabu Search: A Tutorial. *Interfaces*, vol. 20, no. 4, 1990, s.74-90.
- [5] Hwang S.: Part selection problems in flexible manufacturing systems planning stage. W: K.E.Stecke, R.Suri (red.), *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*. Elsevier, Amsterdam, 1986, s.297-309.
- [6] Korcyl A.: Optymalizacja rozdziału zadań i zasobów w elastycznym systemie produkcyjnym *Rozprawa doktorska*, AGH, Kraków, 1992, praca nie publikowana.
- [7] Rajagopalan S.: Formulation and heuristic solutions for parts grouping and tool loading in flexible manufacturing systems. W: K.E. Stecke, R.Suri (red.), *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, Elsevier, Amsterdam, 1986, s.311-320.
- [8] Sawik T.: Optimal machine loading and part routing in FMS by integer programming. *Zeszyty Naukowe AGH, Automatyka*, z. 49, 1989, s.287-295.
- [9] Sawik T.: Modele zadań krótkookresowego planowania produkcji w elastycznym systemie produkcyjnym. *Zeszyty Naukowe Politechniki Śląskiej, Automatyka*, z. 101, 1990, s.249-261.
- [10] Sawik T., Korcyl A.: Optymalizacja rozdziału zadań i zasobów w elastycznym systemie produkcyjnym. *Materiały Konferencyjne XI Krajowej Konferencji Automatyki*, Białystok, 1991 s.147-154.
- [11] Sawik T.: *Optymalizacja dyskretna w elastycznych systemach produkcyjnych*, WNT, Warszawa, 1992.
- [12] Schrage L., Cunningham K.: *LINGO, Optimization Modeling Language*, LINDO Systems Inc., Chicago, 1991.
- [13] Srivastava B., Wun-Hwa Chen: Part Type Selection Problem in Flexible Manufacturing Systems: Tabu Search Algorithms. *Annals of Operations Research*, zgłoszone do druku, 1992.
- [14] Stecke K.E., Kim I.: A study of FMS part type selection approaches for short-term production planning. *International Journal of Flexible Manufacturing Systems*, vol. 1, 1988, s.7-30.

Recenzent: Prof.dr inż. Henryk Kowalowski

Wpłynęło do Redakcji do 30.04.1992 r.

Abstract:

The paper presents integer programming formulations and a new tabu search algorithm for simultaneous part type selection and machine loading in a flexible manufacturing system. The purpose of the part type selection and machine loading is to determine a subset of part types for simultaneous processing and to allocate the operations and required tools of those selected part types among the machines with limited capacity tool magazines.

The combined part type selection and machine loading problem is formulated as an integer program *PTSEML*. In addition, 2-level program *PTS/ML* is presented for part type selection *PTS* at the upper level and machine loading *ML* at the lower level, which has been obtained by decomposition of the *PTSEML* model.

A complex heuristic called *ECT.Tabu.Search* is presented for solving the *PTSEML* in order to balance total processing time in the system. The heuristic is a combination of the tabu search algorithm for part type selection and the ECT (Earliest Completion Time) heuristic for machine loading. The latter procedure is modified appropriately to account for the tool magazine capacity constraints.

An extensive computational experiment has been performed to evaluate the performance of the *ECT.Tabu.Search* algorithm and to compare the solution results with those obtained by solving the two integer programs: 1-level *PTSEML* and 2-level *PTS/ML*, by using the latest version of the LINGO package for discrete optimization.

The results obtained have indicated that the *ECT.Tabu.Search* algorithm yields good solutions in short CPU run time.