ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: AUTOMATYKA z. 125 Nr kol.1391

Antoni KORCYL, Piotr ŁEBKOWSKI
Akademia Górniczo – Hutnicza, Kraków

GENEROWANIE SEKWENCJI MONTAŻOWYCH W ELASTYCZNYM GNIEŹDZIE MONTAŻOWYM - ALGORYTM HEURYSTYCZNY TYPU TABU SEARCH

Streszczenie. W artykule przedstawiono dwupoziomowy system wspomagania podejmowania decyzji krótkookresowego planowania montażu w elastycznym gnieździe montażowym. Gniazdo składa się ze stacji połączonych zautomatyzowanym systemem transportowym, w którym kilka różnych typów wyrobów jest jednocześnie montowanych. Na pierwszym poziomie za pomocą algorytmu genetycznego generowane są alternatywne sekwencje operacji. Następnie za pomocą heurystyki tabu search na niższym poziomie dokonywany jest wybór sekwencji montażowej dla każdego wyrobu oraz rozdział operacji pomiędzy stacje w celu zrównoważenia ich obciążenia oraz zminimalizowania czasu transportu pomiędzy nimi.

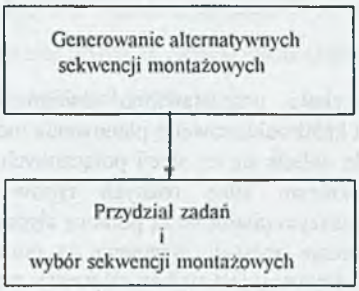
GENERATION OF ASSEMBLY SEQUENCES IN FLEXIBLE ASSEMBLY CELL - A TABU SEARCH APPROACH

Summary. The paper presents a two-level decision support for short term planning in a flexible asssembly cell. The system is made up of several assembly machines linked with an automated material handling, where several different product types can be ssimultaneousslu assembled. First, a genetic algorithm at the upper level generates alternative assembly sequences for a mix of products, and then a tabu search heuristic at the lower level selects best assembly sequence for each product and determines an allocation of assembly tasks among the machines so as to balance workloads and minimize total transportation time.

1. Wstęp

W krótkoterminowym planowaniu montażu realizowane są dwa cele. Po pierwsze, przydział operacji montażowych do stacji posiadających ograniczoną przestrzeń roboczą oraz wybór sekwencji montażowych. Kryterium optymalizacji przydziału operacji jest minimalizacja całkowitego czasu montażu oraz transportu detali w gnieździe.

W artykule zaproponowano dwupoziomowy system wspomagania podejmowania decyzji dla krótkookresowego planowania montażu (rys.1). Na poziomie wyższym za pomocą algorytmu genetycznego generowane są zbiory dopuszczalnych sekwencji i podsekwencji montażowych dla każdego montowanego wyrobu. Na poziomie niższym za pomocą heurystyki opartej na metodzie tabu search dokonywany jest wybór sekwencji montażowej dla każdego wyrobu oraz przydział operacji montażowych do poszczególnych stacji z jednoczesnym zrównoważeniem obciążeń stacji i minimalizacją czasu transportu pomiędzy stacjami.



Rys. 1. Dwupoziomowe krótkoterminowe planowanie montażu
Fig. 1. A two-level short-term assembly planning

2. Podstawowe elementy algorytmu genetycznego

Analiza [7] zbioru dopuszczalnych sekwencji montażowych rzeczywistych zespołów pozwoliła stwierdzić, że większość sekwencji danego produktu jest prawie identyczna. Różnice sprowadzają się między innymi do odwrócenia fragmentu sekwencji, zamiany miejscami dwóch części lub podzespołów, wymiany fragmentów sekwencji między różnymi sekwencjami tworząc w ten sposób nowe uporządkowanie. Właśnie te wymienione 3 typy różnic między sekwencjami i związane z nimi sposoby tworzenia nowych sekwencji montażowych nasunęły analogię z algorytmami genetycznymi. Cięcie i wymiana fragmentów sekwencji odpowiadać będzie operacji krzyżowania. Odwracanie fragmentów sekwencji czy wymiana dwóch składników to mutacje. W niniejszym rozdziale opisana jest procedura generowania sekwencji montażowych oparta na algorytmie genetycznym.

W algorytmie genetycznym realizowany jest proces reprodukcji. Najlepsze osobniki poddawane są działaniu operatorów krzyżowania. W celu rozszerzenia zbioru badanych rozwiązań proces ten jest zakłócany operatorem mutacji. Operator krzyżowania obcina i szczepi gałęzie grafu sekwencji montażowej, operator mutacji zmienia drzewo montażowe w bliskie mu drzewo różniące się dwoma genami. Tak utworzoną nową populację poddaje się ocenie

określającej stopień przystosowania do środowiska. Oczywiście sekwencje o większej wartości funkcji dopasowania mają większą szansę na wybór do następnych operacji krzyżowania i mutacji. Wybór chromosomu z najlepszymi wartościami funkcji dopasowania może prowadzić, szczególnie w początkowej fazie działania algorytmu, do generowania kolejnych sekwencji w otoczeniu ekstremów lokalnych. Z tego powodu selekcja chromosomów jest procesem losowym. Spełnienie przez algorytm warunków zatrzymania procesu reprodukcji powoduje wyprowadzenie najlepszego chromosomu.

Sekwencje montażowe w proponowanym algorytmie są chromosomami - indywiduami populacji zbioru możliwych sekwencji, geny tworzące chromosom to pojedyncze części montowanego wyrobu. Konstrukcja każdego algorytmu genetycznego wymaga określenia następujących składników: genetycznej reprezentacji potencjalnych rozwiązań, metody generowania populacji rozwiązań początkowych, funkcji dopasowania, służącej ocenie potencjalnych rozwiązań, operatorów genetycznych, warunków zakończenia generowania kolejnych populacji oraz innych stałych parametrów (a wśród nich rozmiaru populacji, prawdopodobieństwa zastosowania operatorów genetycznych).

Kod chromosomów - reprezentacja potencjalnych rozwiązań. W literaturze poświęconej zagadnieniom algorytmów genetycznych i ewolucyjnych (m.in.: [3,8]) można odnaleźć szereg wskazówek dotyczących sposobów reprezentowania drzew rozwiązań. Dla potrzeb utworzonego systemu przyjęto następujący sposób reprezentacji sekwencji montażowych. Pojedyncze sekwencje montażowe stanowią chromosomy poszczególnych osobników populacji możliwych rozwiązań. Składają się one z genów oznaczających poszczególne części montowanego wyrobu. Sekwencja montażowa jest określana przez kolejność pojawiania się kolejnych genów w chromosomie. W celu wyodrębnienia w strukturze wyrobu poszczególnych podzespołów wprowadza się do chromosomów symbole kontrolne w postaci nawiasów ograniczających z obu stron sekwencje podzespołów. Interpretacja zapisanych za ich pomocą sekwencji jest zgodna z klasyczną kolejnością wykonywania działań arytmetycznych. Na przykład chromosom o następującej strukturze: (A(BC)(((DE)F))G) odpowiada następującej sekwencji operacji: pobierz część D, połącz z nią część E, do tak powstałego podzespołu dodaj część F; równocześnie z tą podsekwencją połącz część B i C; uchwyć część A i kolejno domontuj podzespół BC, DEF i część G. Taki sposób kodowania zapewnia możliwość zapisu sekwencji montażowych w postaci łańcuchów znakowych o skończonej długości i przy określonym, skończonym zbiorze symboli.

Operatory genetyczne. W prezentowanym systemie, podobnie jak we wszystkich klasycznych programach ewolucyjnych, zastosowano operatory genetyczne dwojakiego rodzaju:

- krzyżowania - istotny w przeszukiwaniu obiecujących obszarów przestrzeni rozwiązań i odpowiedzialny za zbieżność obliczeń,
- operator mutacji - wprowadzający do populacji pewien czynnik losowej zmienności.

Działanie *operatora krzyżowania* polega na połączeniu cech dwóch osobników rodzicielskich. Realizowane jest to poprzez wycinanie fragmentu kodu jednego z osobników i umieszczanie go w kodzie drugiego z rodziców. Warunkami krzyżowania dwóch chromosomów są: występowanie w dwóch rozważanych chromosomach (drzewach montażu) identycznych podzespołów, zróżnicowanie poddrzew tych podzespołów, zróżnicowanie pozostałych części chromosomów. Dla przykładu, zakładając istnienie dwóch osobników rodzicielskich o kodach:

(A(BC)(((DE)F))G) i (C(BA)(((DF)E))G),

za pomocą operatora krzyżowania można otrzymać dwa różne osobniki pochodne o kodach:

(A(BC)(((DF)E))G) i (C(BA)(((DE)F))G).

Punkt krzyżowania się chromosomów jest określony przez wspólny podzespół. Powstałe w wyniku krzyżowania osobniki potomne spełniają narzucone na proces montażowy ograniczenia, oczywiście przy założeniu, że ich sekwencje rodzicielskie też je spełniały. Wynika to z własności systemów kolejkowych, jakimi są sekwencje montażowe. Inna sytuacja powstaje po użyciu operatora mutacji.

Działanie operatora mutacji można zdefiniować jako lokalną zamianę miejsc dwóch genów reprezentujących pojedyncze części lub podzespoły montowanego wyrobu. Dla przykładu, z chromosomu (**A**(BC)(((DF)E))G) można otrzymać osobnika pochodnego o kodzie: (**B**(AC)(((DE)F))G).

Punkt mutacji kodu jest wybierany losowo na podstawie współczynnika prawdopodobieństwa miejscowej mutacji. Efekt działania operatora mutacji może być bardzo odległy od kodu istniejącego przed zastosowaniem tego operatora i niekoniecznie musi reprezentować poprawną sekwencję montażu. Dlatego po każdym użyciu operatora mutacji wynik jego działania jest sprawdzany pod względem dopuszczalności sekwencji operacji na podstawie grafu dopuszczalnych połączeń podzespołów w montowanym wyrobie. W przypadku stwierdzenia niezgodności kodu wynikowego z informacją zawartą w tym zbiorze cała sekwencja - powstający nowy chromosom jest usuwany.

Funkcja dopasowania. Zadaniem funkcji dopasowania jest ukierunkowanie poszukiwań przez program rozwiązania w stronę sekwencji montażowych najtańszych pod względem kosztów wykonania. Funkcja oceny rozwiązań dopuszczalnych, obliczająca dla występujących w populacji osobników wartości ich przystosowania do środowiska, może bazować na jednym lub jednocześnie kilku z wymienionych poniżej przykładowych kryteriów:

- kryterium stabilności aktualnie montowanych podzespołów ,
- kryterium liczby podzespołów niezbędnych do dokonania montażu,
- kryterium liczby i rodzaju środków transportu niezbędnych do dokonania montażu,
- kryterium złożoności ruchu koniecznego do dokonania montażu itp.

Populacja początkowa. Początkowa populacja w proponowanym programie generowania sekwencji montażowej jest złożona ze zbioru kilku sekwencji otrzymanych heurystyczną metodą omówioną w [6]. Wykorzystywane tam reguły preferują sekwencje łatwe do wykonania - sekwencje składające się z małej liczby części i podzespołów. Populację początkową można także otrzymać wykorzystując doświadczenia ekspertów, którzy zaproponują kilka dobrych, ich zdaniem, uporządkowań operacji montażowych.

Stałe parametry algorytmu. Jednym z najważniejszych parametrów programu genetycznego jest liczebność populacji. W przyjętym rozwiązaniu zastosowano zmienną liczebność populacji. Rozwiązanie to cechuje specyficzny mechanizm selekcji, wykorzystujący pojęcie *czasu życia* danego chromosomu, co jest równoważne liczbie pokoleń, przez które chromosom pozostaje „żywy". *Czas życia* chromosomu zastępuje pojęcie selekcji i ponieważ zależy on od przystosowania danego osobnika do środowiska, wpływa na liczebność populacji w każdym pokoleniu. Programy ewolucyjne ze zmienną liczebnością populacji samoczynnie dostrajają liczebność populacji do etapu procesu reprodukcji. Podczas rekombinacji tworzy się pewna liczba osobników potomnych, dołączających się do populacji. Liczba ta jest wprost proporcjonalna do początkowej liczebności populacji i wynosi: *liczebność* * *r* (gdzie: *r* jest prawdopodobieństwem krzyżowania). Każdy osobnik z populacji jest wybierany do reprodukcji z jednakowym prawdopodobieństwem, niezależnym od jego przystosowania. Oczekiwana liczba potomków danego osobnika jest więc wprost proporcjonalna do czasu jego życia. Wynika stąd, że osobniki o współczynnikach dopasowania przekraczających aktualną średnią powinny być obdarzone większymi wartościami czasu życia, by mogły dać początek większej liczbie rozwiązań pochodnych.

Opracowany system pozwala na swobodny wybór między trzema następującymi strategiami obliczania czasu życia osobnika [8]: przyporządkowaniem proporcjonalnym, przyporządkowaniem liniowym, przyporządkowaniem biliniowym.

Przedstawiony ogólnie w tym rozdziale program generowania sekwencji montażowej badany był eksperymentalnie dla wielu wyrobów montowanych w rzeczywistych procesach wytwórczych. Między innymi generowano sekwencje czterech zespołów, składających się łącznie z 15 części. Wyniki tego eksperymentu (tabl.2) - końcowe sekwencje operacji dla każdego zespołu - stanowiły podstawę zastosowania metody Tabu Search, równoważącej obciążenie elastycznego systemu montażowego. Istotne znaczenie w genetycznym tworzeniu zbioru dopuszczalnych sekwencji montażowych ma uzyskanie dobrych populacji początkowych. Testy przeprowadzone dla losowo wybranej populacji pierwotnej okazywały się kilkakrotnie wolniejsze od testów startujących z dobrych sekwencji. Przy dużej liczbie części - długim chromosomie - ta duża różnica prędkości zanika. Zatem algorytm genetyczny dobrze radzi sobie także z populacją długich chromosomów.

3. Algorytm heurystyczny dla wyboru sekwencji montażowych i zrównoważenia obciążenia systemu

Elastyczny system montażowy składa się z M stacji montażowych i (i = 1, ..., M), połączonych zautomatyzowanym systemem transportowym. W każdej stacji i ze względu na limitowaną powierzchnię roboczą tylko b_i podajników może być umieszczonych, co pozwala na wykonanie ograniczonej liczby operacji montażowych . W systemie montowanych jest K różnych typów wyrobów z N typów części składowych. Znany jest podzbiór typów części składowych koniecznych do montażu każdego wyrobu oraz zbiór dopuszczalnych sekwencji montażowych. Zbiór ten reprezentuje pary typów części takich, że w sekwencji montażowej dla danego wyrobu pierwszy element pary bezpośrednio poprzedza w operacji montażu drugi element pary. Dany jest czas montażu części j w wyrobie k oraz czas transportu konieczny do przemieszczenia wyrobu z jednej stacji do innej. Określone jest zapotrzebowanie na wyrób k.

Zaproponowany tu algorytm heurystyczny oparty został na metodzie Tabu Search. W algorytmie *ruchem* jest zmiana przydziału zadania montażu z jednej stacji do innej. Zmiana przydziału powoduje otrzymanie nowej wartości funkcji celu Q_{max}, reprezentującej sumę obciążeń stacji montażowej oraz czasu transportu dla stacji, będącej wąskim gardłem w

systemie [9, 10, 11]. Dla każdej zmiany przydziału części do stacji określana jest ze zbioru dopuszczalnych sekwencji montażowych dla każdego wyrobu sekwencja o najkrótszym czasie wykonania.

Lista_Tabu zawiera zadania, którym zmieniono przydział do stacji w kilku ostatnich iteracjach.

Status_Tabu jest utrzymywany przez kilka iteracji, w zależności od rozmiaru rozwiązywanego problemu.

Long_Term_Memory opiera się na częstości przydzielania zadań do różnych stacji.

Algorytm heurystyczny dla wyboru sekwencji montażowych i zrównoważenia obciążeń elastycznego systemu montażowego jest następujący.

$$Q_{max} = \max_{station}(station\ workload + transfer\ time)$$

Krok 1 Przydziel wszystkie typy części do stacji montażowych.
Define: *Aspiration Level, Long Term memory*
Fix: *Tabu_List, Status_Tabu, Max_Iteration, Aspiration_Level_Iteration, Long_Term_Iteration*

Krok 2 Oblicz Q_i dla każdej stacji i. Znajdź stację będącą wąskim gardłem, która spełnia warunek: $Q_{max} = \max(Q_i)$
Set: *New_Solution* = Q_{max}

Krok 3 Znajdź typ części j nie posiadający *Status_Tabu* i przydzielaj go do pozostałych stacji i, posiadających wolne podajniki.
Dla każdej zmiany przydziału typu części do stacji znajdź dla każdego wyrobu k sekwencję montażową o najkrótszym czasie wykonania.
Jeżeli nie ma takiego typu części, zastosuj funkcję *Aspiration Level*.

Krok 4 Uaktualnij wszystkie zmienne *Status_Tabu, Number_Of_Iteration.*
If *New_Solution* > *Best_Solution*
and *Number_Of_Iteration* = *Aspiration_Level_Iteration*
and *Number_Of_Iteration* < *Max_Iteration*
then *Aspiration_Level* function
goto Krok 2

Krok 5 Uaktualnij wszystkie zmienne *Status_Tabu, Number_Of_Iteration*
If *Best_Solution* > *New_Solution*
and *Number_Of_Iteration* < *Max_Iteration*
set *Best_Solution* = *New_Solution*
goto Krok 2

Krok 6 **If** *Number_Of_Iteration* < *Max_Iteration*
and *Number_Of_Iteration* = *Long_Term_Iteration*
and

New_Solution > *Best_Solution*
then apply
Long Term Memory
goto Krok 2
otherwise
TERMINATE

4. Przykład liczbowy

Dla zilustrowania zaproponowanego algorytmu rozważmy elastyczny system montażowy, składający się z $M = 6$ stacji montażowych. W systemie spośród $N = 15$ typów części montowane są $K = 4$ wyroby. Każda stacja montażowa jest wyposażona w $b_i = 6$ podajników części. Zbiory części wymagane do montażu każdego wyrobu J_k przedstawiono w tablicy 2. Dodatkowo w tablicy tej przedstawiono dopuszczalne sekwencje montażowe dla każdego wyrobu. Zapotrzebowania na każdy wyrób wynoszą: $d_1 = 20$, $d_2 = 20$, $d_3 = 20$, $d_4 = 20$. Czasy montażu dla każdego typu części przedstawiono w tablicy 1. Tablica 3 przedstawia czasy transportu pomiędzy poszczególnymi stacjami.

Tablica 1

Czasy montażu p_{ij}

Stacja	*Typ części j*														
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	4	4	4	5	4	3	3	2	1	1	3	5	4	2
2	2	4	4	4	5	4	3	3	2	1	1	3	5	4	2
3	2	3	2	2	5	2	4	4	2	3	2	2	2	4	3
4	2	3	2	2	5	2	4	4	2	3	2	2	2	4	3
5	2	3	2	3	4	5	5	4	2	3	4	2	2	3	3
6	2	3	2	3	4	5	5	4	2	3	4	2	2	3	3

Tablica 2

Części składowe oraz sekwencje montażowe

Wyrób k	*Części składowe J_k*	*Sekwencje montażowe*
1	1,2,3,4,5,6,7	3,2,7,6,4,1,5
		2,7,3,6,4,1,5
		6,3,2,7,4,1,5
		6,2,7,3,4,1,5
2	2,4,5,7,8,10,12,14	2,10,12,14,7,4,5,8
		2,7,4,10,12,14,5,8
		7,4,2,10,12,14,5,8
		10,12,14,2,7,4,5,8
3	1,3,4,5,6,9,11,13,15	15,11,13,3,6,1,4,5,8
		15,1,4,3,6,11,13,5,9
		15,3,6,1,4,11,13,5,9
		15,3,6,11,13,1,4,5,9
4	2,4,6,7,8,10,11,15	7,15,2,6,10,11,4,8
		7,15,10,11,2,6,4,8
		15,7,2,6,10,11,4,8
		15,7,10,11,2,6,4,8

Tablica 3

Czasy transportu q_{il}

Stacja i	1	2	3	4	5	6
1	0	0	2	2	2	2
2	0	0	2	2	2	2
3	2	2	0	0	2	2
4	2	2	0	0	2	2
5	2	2	2	2	0	0
6	2	2	2	2	0	0

W tablicy 4 przedstawiono przydziały typów części do poszczególnych stacji montażowych $x_{ij}^{(IP)}$, wyznaczone poprzez rozwiązanie zadania programowania całkowitoliczbowego, przy użyciu standardowego pakietu optymalizacji dyskretnej LINGO [8] oraz $x_{ij}^{(H)}$ - przy zastosowaniu algorytmu heurystycznego.

Tablica 4

Przydział części składowych $x_{ij}^{(IP)} / x_{ij}^{(H)}$

Stacja k	Typ części j														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		1/1	1/0								0/1			0/1	
2							1/1			1/1		1/1		1/0	0/1
3	1/1			1/1					0/1		1/0				
4					1/1			0/1	1/0				1/0		
5			0/1			1/0							0/1		
6						0/1		1/0							1/0

Otrzymano następujące sekwencje montażowe dla każdego wyrobu:

- rozwiązując zadania programowania całkowitoliczbowego: $J_1 = \{3,2,7,6,4,1,5\}$, $J_2 = \{7,4,2,10,12,14,5,8\}$, $J_3 = \{15,3,6,11,13,1,4,5,9\}$, $J_4 = \{15,7,10,11,2,6,4,8\}$.
- stosując algorytm heurystyczny : $J_1 = \{2,7,3,6,4,1,5\}$, $J_2 = \{2,10,12,14,7,4,5,8\}$, $J_3 = \{15,11,13,3,6,1,4,5,9\}$, $J_4 = \{7,15,10,11,2,6,4,8\}$.

Rozwiązanie otrzymane w wyniku rozwiązania zadania programowania całkowitoliczbowego wynosiło $Q_{max} = 400$ dla stacji będących „wąskim gardłem" $i = 1,2,3,6$ oraz dla pozostałych stacji $i = 4,5$: $Q_{max} = 380$. Czas obliczeń wynosił 176 minut na komputerze klasy P200.

Zastosowanie algorytmu heurystycznego doprowadziło do uzyskania łącznego czasu obciążenia oraz transportu dla stacji będących „wąskim gardłem" w systemie $i = 4,6$ wartości $Q_{max} = 460$ oraz dla pozostałych stacji odpowiednio: $Q_{max} = 440$ dla $i = 1,2,3$, $Q_{max} = 200$ dla $i = 5$. Czas obliczeń wyniósł dla algorytmu tabu 3 minuty 27 sekund.

Dla oceny jakości zaproponowanego algorytmu przeprowadzono 100 testów z parametrami wejściowymi przedstawionymi w tablicy 5. Czasy montażu były generowane losowo z przedziału [1,10], a czasy transportu pomiędzy sąsiednimi stacjami były jednakowe i wynosiły 2 jednostki. Wyniki oceniono za pomocą dwóch współczynników:

1. Względnego przyrostu łącznego czasu montażu i transportu dla stacji będącej wąskim gardłem Q_{max}

$$\varepsilon_{Q_{max}} = \frac{Q^H_{max} - Q^P_{max}}{Q^P_{max}}$$

2. Względnego skrócenia czasu obliczeń:

$$\varepsilon_{CPU} = \frac{CPU^H}{CPU^P},$$

gdzie H oznacza algorytm heurystyczny a IP rozwiązanie optymalne.

Tablica 5 dodatkowo przedstawia średnie wartości obu współczynników ε_Q i ε_{CPU}, wyznaczone dla wszystkich testowanych problemów. Eksperymenty obliczeniowe wykazały, że średni względny przyrost wartości funkcji celu nie był większy niż 19%, z jednoczesnym średnim skróceniem czasu obliczeń 14 krotnym.

Tablica 5

Wyniki eksperymentów obliczeniowych

Liczba stacji montażowych M	*Liczba wyrobów* K	*Liczba części składowych* N	ε_{Q}	ε_{CPU}
3	3	10	13	6.52
	4	15	17	12.55
	5	20	18	18.54
4	3	10	11	5.66
	4	15	14	13.46
	5	20	19	16.01
5	3	10	13	4.68
	4	15	15	9.67
	5	20	19	14.43

5. Podsumowanie

Oceny zaproponowanego algorytmu heurystycznego dla krótkoterminowego planowania montażu w elastycznym systemie montażowym dokonano przeprowadzając szereg eksperymentów obliczeniowych dla danych generowanych losowo. Wyniki porównywano z wynikami otrzymanymi po rozwiązaniu zadań programowania całkowitoliczbowego. Rezultaty eksperymentów obliczeniowych potwierdzają, że można otrzymywać wyniki porównywalne z rozwiązaniami optymalnymi w wielokrotnie krótszym czasie. Dwupoziomowe podejście do

generowania sekwencji montażowych za pomocą algorytmu genetycznego oraz wybór sekwencji montażowych i zrównoważenie obciążeń w elastycznym systemie montażowym wydaje się być przydatne w praktycznych zastosowaniach w krótkoterminowym planowaniu montażu.

LITERATURA

1. Ben-Arieh, D.: A methodology for analysis of assembly operations' difficulty. *International Journal of Production Research*, 32/8, 1994, pp. 1879-1895.
2. Bonneville, F., Perrard, C., Heuriourd, J.M.: A Genetic Algorithm to Generate and Evaluate Assembly Plans. In: *Proceedings INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, Paris, 1995, pp. 231-239.
3. Goldberg, D.E. *Algorytmy genetyczne i ich zastosowania*. WNT, Warszawa 1996.
4. Korcyl, A., Łebkowski, P., Sawik, T.: Selection of Assembly Sequences and Balancing Workloads in FAL. In: *Proceedings INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, Paris, 1995, pp. 349-359.
5. Kroll, E.: Modelling and Reasoning for Computer - Based Assembly Planning, Concurent Engineering (Kusiak, A. (Ed)), J.Wiley, New York, 1993.
6. Łebkowski, P.: Algorytmy generowania rysunku rozstrzelonego i sekwencji montażowej dla FAS. *ZN AGH, Automatyka*, Kraków, z 64, 1993, s. 501-519.
7. Łebkowski, P.: A two-level procedure for generating assembly sequences using a genetic algorithm. *Proceedings of International Conference on Industrial Engineering and Production Management*, Lyon, Vol. 2, 1997, pp. 129-138.
8. Michalewicz, Z.: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT, Warszawa, 1996.
9. Sawik, T.: Integer programming models for the design and balancing of flexible assembly systems, In: *Mathematical and Computer Modelling*, Vol. 21, No. 4, 1995, pp. 1-12.
10. Sawik, T.: Simultaneous loading, routing and assembly plan selection in a flexible assembly system. *Proceedings of the 10th Meeting of the European Chapter on Combinatorial Optimization*, Tenerife 1997.
11. Sawik, T.: An LP-based approach for loading and routing in a flexible assembly system. In: *Proceedings of International Conference on Industrial Engineering and Production Management*, Lyon, Vol. 2, 1997, pp. 216-225.
12. Schrage, L. and Cunningham, K.: *LINGO, Optimization Modelling Language*. LINDO Systems Inc., Chicago 1991.

Recenzent: Prof.dr inż. Tadeusz Puchałka

Abstract

The two main short-term planning objectives for a flexible assembly cell are to assign tasks to assembly machines with limited workin space and to select assembly sequences for a mix of products in order to equalize machine workloads and to minimize machine-to machine movements subject to precedence relations among the tasks.

The paper presents a two level decision support for the flexible assembly cell short-term planning. First, a genetic algorithm at the upper level generates a set of alternative assembly sequences for each product, and then a tabu search heuristic at the lower level selects one assembly sequence for each product and determines an allocation of assembly tasks among the machines so as to balance machine workloads and minimize total transportation time.

In order to evauate performance of the two-level approach proposed for the flexible assembly cell short-term planning several test problems have been solved.

The final solution results have been compared with those obtained by solving to optimality the 0-1 programming model. The computational experiments have indicated that good solution results can be obtained in a reasonable computation time.

The two-level approach with for the generating assembly sequences and tabu search for sequence selection and machine loadin seems to be aplicable in practice such a short-term allocation of the flexible assembly cell resources.