

Antoni KORCYL, Tadeusz SAWIK
Akademia Górniczo-Hutnicza

ALGORYTM TYPU TABU DLA ZADANIA RÓWNOWAŻENIA OBCIĄŻEŃ ELASTYCZNEGO SYSTEMU MONTAŻOWEGO ¹

Streszczenie: W pracy przedstawiono model matematyczny i nowy algorytm heurystyczny dla równoważenia łącznego czasu montażu i transportu w elastycznym systemie montażowym. Algorytm jest kombinacją heurystyki Tabu Search oraz algorytmu ROC (ang. Rank Order Clustering). Zamieszczono wyniki eksperymentów obliczeniowych, które potwierdzają korzystne własności zaproponowanego podejścia.

A TABU SEARCH ALGORITHM FOR BALANCING OF A FLEXIBLE ASSEMBLY SYSTEM

Summary: The paper presents an integer programming model and a new heuristic algorithm for balancing of a flexible assembly system. The algorithm is a combination of ROC (Rank Order Clustering) and Tabu Search heuristic. Results of computational experiments with the algorithm are reported.

EIN TABU SEARCH VERFAHREN FÜR DAS PROBLEM DES AUSGLEICHENS DES FAS

Zusammenfassung: In der Arbeit wird ein mathematisches Modell und ein neues heuristisches Verfahren zum Ausgleichen der gesamten Montage- und Transportzeit beschrieben. Das Verfahren stellt eine Kombination der Tabu Search Heuristik und des ROC (eng. Rank Order Clustering) Algorithmus dar. Es werden auch Ergebnisse der Berechnungsexperimente dargestellt, die gute Eigenschaften dieses Verfahrens bestätigen.

¹Praca była częściowo finansowana przez KBN, grant nr 8 S505 003 05.

1. Wstęp

Elastyczny system montażowy (FAS ang. Flexible Assembly System) można zdefiniować jako zintegrowany system produkcyjny składający się ze zautomatyzowanych stanowisk montażowych i urządzeń transportowych, przeznaczony do jednoczesnego montażu wielu różnych wyrobów w krótkich seriach [11]. Każde stanowisko (stacja) składa się z jednej lub kilku maszyn pracujących równolegle (zwykle robotów montażowych) oraz zmieniaczy palet, podajników części składowych, magazynków uchwytów i narzędzi. Ze względu na ograniczoną przestrzeń roboczą, jaką dysponuje stanowisko montażowe dla pomieszczenia podajników części, na każdej stacji możliwe jest wykonanie tylko ograniczonej liczby operacji montażowych. Problem równoważenia obciążeń stacji montażowych polega na przydziale części składowych do stacji, tak aby łączne czasy operacji montażowych przydzielonych do różnych stacji były zbliżone [2,10]. Jednak z drugiej strony często wzrasta liczba przemieszczeń montowanych wyrobów pomiędzy stacjami. Rozwiązanie problemu optymalizacji funkcjonowania systemu przy kryterium minimalizacji liczby przemieszczeń montowanych wyrobów może z kolei prowadzić do nierównomiernych obciążeń stacji. Najlepszym rozwiązaniem byłoby więc równoważenie obciążeń stacji montażowych wraz z minimalizacją przemieszczeń wyrobów pomiędzy nimi. Złożoność obliczeniowa takiego zadania sprawia, że praktyczne zastosowanie modelu programowania całkowitoliczbowego oraz rozwiązanie go za pomocą standardowego pakietu programowania całkowitoliczbowego wymaga bardzo dużych nakładów obliczeniowych, por. [12]. Ponadto każde z wymienionych zadań optymalizacji rozwiązywane oddzielnie jest dostatecznie złożone, aby usprawiedliwić stosowanie algorytmów heurystycznych (np. [1]).

W artykule przedstawiony zostanie nowy algorytm heurystyczny do rozwiązania problemu optymalizacji wykorzystania zasobów elastycznego systemu montażowego, w którym jako kryterium optymalności przyjęto zrównoważenie obciążeń stacji montażowych wraz z jednoczesną minimalizacją przemieszczeń wyrobów pomiędzy stacjami.

2. Model matematyczny zadania

Rozważmy elastyczny system montażowy składający się z M stacji montażowych $i = 1, \dots, M$. W systemie z N różnych typów części montowanych jest K różnych typów

wyrobów. Znane jest zapotrzebowanie d_k na wyrób typu k , $k = 1, \dots, K$. Wyprodukowanie każdego typu wyrobu wymaga montażu różnych części składowych w określonej kolejności. Niech $J_k \subset \{1, \dots, N\}$ oznacza zbiór typów części wchodzących w skład wyrobu typu k , a R_k zbiór par typów części (j, r) , takich że w wyrobie typu k montaż części typu j musi bezpośrednio poprzedzać montaż części typu r . Oznaczmy przez p_{jk} czas montażu części typu j w wyrobie typu k . Każda stacja montażowa i jest wyposażona w s_i podajników części. Czas transportu wyrobu pomiędzy stacjami i oraz l oznaczmy przez q_{il} .

W elastycznym systemie montażowym czasy montażu poszczególnych części składowych są stosunkowo krótkie i porównywalne z czasami transportu wyrobów pomiędzy stacjami. Jako kryterium optymalności należy więc przyjąć zrównoważenie obciążeń stacji montażowych z uwzględnieniem czasów przemieszczeń wyrobów pomiędzy nimi.

Model matematyczny zadania równoważenia łącznego czasu montażu i transportu w elastycznym systemie montażowym przedstawiono poniżej (por. [12]).

Jako zmienne decyzyjne przyjęto:

$$x_{ij} = \begin{cases} 1, & \text{jeżeli część typu } j \text{ jest przydzielona do podajnika przy stacji } i; \\ 0, & \text{inaczej} \end{cases}$$

$$y_{iljk} = \begin{cases} 1, & \text{jeżeli wyrób typu } k \text{ po wykonaniu operacji } j \text{ przechodzi ze stacji } i \\ & \text{do stacji } l; \\ 0, & \text{inaczej} \end{cases}$$

Zminimalizować

$$PQ_{max} \quad (1)$$

przy ograniczeniach

$$\sum_{k=1}^K \sum_{j \in J_k} d_k p_{jk} x_{ij} + \sum_{k=1}^K \sum_{l \neq i} \sum_{j \in J_k} d_k q_{il} y_{iljk} \leq PQ_{max}; \quad \forall i \quad (2)$$

$$\sum_{i=1}^M x_{ij} = 1; \quad \forall j \quad (3)$$

$$\sum_{j=1}^N x_{ij} \leq s_i; \quad \forall i \quad (4)$$

$$x_{ij} + x_{lr} - y_{iljk} \leq 1; \quad \forall k, \forall i, l, l \neq i, \forall (j, r) \in R_k \quad (5)$$

$$-x_{ij} - x_{lr} + 2y_{iljk} \leq 0; \quad \forall k, \quad \forall i, l, l \neq i, \quad \forall (j, r) \in R_k \quad (6)$$

$$x_{ij} \in \{0, 1\}; \quad \forall i, j \quad (7)$$

$$Y_{iljk} \in \{0, 1\}; \quad \forall i, l, k \quad (8)$$

Funkcja celu (1) reprezentuje sumę czasów montażu i transportu wyznaczonych dla stacji będącej wąskim gardłem w systemie i określonej przez ograniczenie (2). Pierwszy składnik w ograniczeniu (2) reprezentuje obciążenie stacji będącej wąskim gardłem, a drugi łączny czas transportu montowanych wyrobów do tej stacji. Ograniczenie (3) zapewnia przydział części każdego typu do tylko jednej stacji, zaś (4) uwzględnia ograniczoną liczbę podajników części przy każdej stacji. Warunki (5) i (6) zapewniają wybór dla każdego typu wyrobu marszruty przechodzącej przez te stacje, do których podajników przydzielono kolejno montowane typy części. Warunki te pośrednio uwzględniają również indywidualne ograniczenia kolejnościowe dla każdego typu wyrobu.

3. Algorytm heurystyczny ROC_Tabu_Search

Zaproponowana w dalszej części złożona heurystyka ROC_Tabu_Search jest algorytmem, równoważenia obciążeń stacji montażowych z jednoczesną minimalizacją przemieszczeń wyrobów pomiędzy stacjami.

Heurystyka stanowi połączenie algorytmu ROC dla metody grupowania [6] oraz algorytmu Tabu Search ([3,4,5]). Różne zastosowania metody Tabu Search obejmują m.in. problemy: plecakowy, komiwojażera, pakowania, kolorowania grafu, harmonogramowania, wyznaczania partii produkcyjnych itp. (por. [5,7,8,9]).

Metoda grupowania ROC (ang. Rank Order Clustering) bazuje na sortowaniu wierszy i kolumn w macierzy incydencji maszyna-typ części w celu wyodrębnienia oddzielnych grup technologicznych. Dla każdej kolumny oraz wiersza przydzielamy binarną wagę i obliczamy odpowiadającą im równowartość dziesiętną. Wagi dla kolumny j oraz wiersza i obliczamy wg wzorów:

$$\text{wiersz } i \quad \sum_{k=1}^n a_{ik} 2^{n-k}$$

$$\text{kolumna } j \quad \sum_{k=1}^m a_{kj} 2^{n-k}$$

Następnie dokonujemy sortowania poszczególnych kolumn i wierszy macierzy incydencji w kolejności nierosnących wag. W rezultacie uzyskujemy odrębne grupy technologiczne.

Metoda Tabu Search jest nowoczesną metodą heurystyczną stosowaną do rozwiązywania problemów optymalizacji dyskretnej, zwiększającą prawdopodobieństwo uniknięcia pułapki lokalnego optimum podczas poszukiwań rozwiązania optymalnego.

Głównymi składnikami tej metody są *pamięć krótkoterminowa*, zwana *listą tabu*, *pamięć długoterminowa* oraz *funkcja kryterium kwalifikacji*. *Lista tabu* jest zbiorem zmiennych, których stan w danej iteracji nie może ulec zmianie. Z *listą tabu* jest ściśle związany parametr pozwalający określić, od ilu iteracji dana zmienna znajduje się na *liście tabu* i czy posiada *status tabu*, tzn. niemożliwa jest zmiana jej stanu w danej iteracji.

Funkcja kryterium kwalifikacji pozwala po spełnieniu odpowiednich, z góry narzuconych, warunków na zmianę statusu zmiennej znajdującej się na *liście tabu* oraz zmianę jej stanu.

Pamięć długoterminowa pozwala na urozmaicenie oraz intensyfikację poszukiwań poprzez odpowiednie zdefiniowanie tej funkcji. Funkcja ta jest uaktywniana w momencie, gdy nie jest możliwa zmiana stanu zmiennych oraz gdy użycie *funkcji kryterium kwalifikacji* nie powoduje żadnej zmiany na *liście tabu*, a nie wszystkie regiony obszaru rozwiązań dopuszczalnych zostały zbadane.

Algorytm ROC_Tabu_Search składa się z dwóch głównych kroków: wstępnego przydziału części do poszczególnych stacji oraz wyznaczenia rozwiązania optymalnego w oparciu o metodę Tabu Search. Wstępny przydział części do stacji jest dokonywany w oparciu o zmodyfikowany algorytm ROC [6], w którym macierz incydencji maszyna-typ części zastąpiono przez macierz incydencji typ wyrobu-typ części. Na podstawie wyników podziału na odrębne grupy technologiczne dokonujemy wstępnego przydziału części do poszczególnych stacji oraz obliczamy wartość funkcji celu, którą jest suma czasów montażu i transportu wyznaczonych dla stacji będącej wąskim gardłem w systemie.

Kolejny krok to ustalenie wielkości wejściowych dla algorytmu Tabu Search. Rozmiar *listy tabu* jest stały, ponieważ nie rozważamy problemu maksymalizacji liczby montowanych wyrobów (np. [7,8]). Wartość *Best_Solution* jest równa wartości funkcji celu obliczonej w pierwszym kroku algorytmu, za pomocą procedury ROC. Ustalana jest wielkość *Status_Tabu*. W proponowanym algorytmie *Status_Tabu* posiadają takie części, którym w

trakcie kolejnych iteracji zmieniono przydział do stacji. *Status Tabu* (czyli zakaz zmiany przydziału części do innej stacji) utrzymuje się w ciągu kilku następnych iteracji, których liczba zależy od rozmiaru rozwiązywanego problemu. Wybór części, która zostanie przydzielona do innej stacji, polega na znalezieniu stacji będącej w danej iteracji wąskim gardłem. Następnie spośród wszystkich części przydzielonych do tej stacji, które nie mają *statusu tabu* (możliwy jest ich przydział do innych stacji), wybierana jest ta, która po przydziale do innej stacji spowoduje, że nowa wartość funkcji celu (1) *New_Solution* będzie mniejsza niż wartość *Best_Solution* w danej iteracji. Następnie aktualizuje się wszystkie wielkości związane z *listą tabu* (*Status_Tabu*, liczba iteracji itd). Procedura ta jest powtarzana, dopóki nie będzie możliwe poprawienie wartości funkcji celu *Best_Solution* lub nie zostanie przekroczona liczba iteracji. *Funkcja kryterium kwalifikacji* pozwala na zmianę statusu części posiadającej *Status_Tabu*, w przypadku gdy zmiana przydziału tej części spowoduje polepszenie wartości *Best_Solution*.

W rozważanym algorytmie jest stosowana tzw. *pamięć długoterminowa* jako element intensyfikacji i urozmaicenia poszukiwań. W algorytmie *ROC_Tabu_Search* rolę *pamięci długoterminowej* pełni częstość przydzielania części składowych do poszczególnych stacji. Na podstawie tej informacji następuje zmiana przydziału części do stacji i powrót do głównej linii algorytmu. Procedura ta jest uruchamiana w algorytmie, w momencie gdy nie istnieje możliwość poprawienia wartości *Best_Solution*, a liczba iteracji nie osiągnęła jeszcze wartości *Max_Iteration*.

Algorytm jest wykonywany dopóki nie zostanie osiągnięte rozwiązanie optymalne lub znalezione rozwiązanie nie ulega poprawie w ciągu liczby iteracji równej *Max_Iteration*. Uproszczony opis programu *ROC_Tabu_Search* przedstawiono na r ys.1.

4. Przykład liczbowy

Dla ilustracji zaproponowanego algorytmu rozważmy przykład elastycznego systemu montażowego składającego się z $M = 4$ stacji montażowych. Każda stacja dysponuje jednakową liczbą $s_i = 4$ podajników części. W systemie z $N = 10$ typów części montowanych jest $K = 5$ typów wyrobów. Wykonanie wyrobu typu k wymaga montażu części różnego typu j w następującej kolejności: dla $k = 1: j = 1,2,3,4,5$; dla $k = 2: j = 2,1,4,3,5$;

dla $k = 3$: $j = 3,4,5,6,7$; dla $k = 4$: $j = 1,2,6,7,8,9,10$; dla $k = 5$: $j = 4,5,8,9,10$.

Należy wyprodukować jednakowe ilości wyrobów wynoszące $d_1 = d_2 = d_3 = d_4 = d_5 = 40$ sztuk.

Program ROC.Tabu_Search;

Declarations;

begin

 Data;

 Make_Group_Technology (ROC);

 Station_Loading;

 Count_Solution;

 Tabu_Status := Initialize;

 Best_Solution := Solution;

 Number_of_Iteration := 0;

 repeat

 repeat

 Number_of_Iteration := Number_of_Iteration + 1;

 Make_List_Components_to_Move;

 Find_Component_to_Move;

 Count_New_Solution;

 If (Requirements Completed) and (New_Solution) Best_Solution)

 then

 begin

 Tabu_Status := Update_Tabu_Status;

 Station_Loading := Update_Station_Loading;

 Tabu_List_Frequency := Update_Tabu_List_Frequency;

 Best_Solution := New_Solution;

 end;

 Long_Term_Memory;

 until New_Solution = Best_Solution;

 until Number_of_Iteration = Max_Iteration;

 end;

 Print_Results;

end.

Rys. 1. Algorytm ROC.Tabu_Search

Fig. 1. Algorithm ROC.Tabu_Search

Tab. 1. Przydziały typów części do stacji (x_{ij}^*/x_{ij}^H)

Stacja <i>i</i>	Typ części <i>j</i>									
	1	2	3	4	5	6	7	8	9	10
1						1	1		1	
2					1			1		
3	0	0	1	1						
4	1	1	0							1

Czasy p_{jk} montażu części dla poszczególnych typów wyrobów są jednakowe i wynoszą $p_{jk} = 2$. Czas transportu wyrobu pomiędzy sąsiednimi stacjami wynosi 2 jednostki czasu, a stacje rozmieszczone są tak, że wartości q_{il} czasów przemieszczeń pomiędzy stacjami wynoszą: $q_{12} = q_{14} = q_{21} = q_{23} = q_{32} = q_{34} = q_{43} = q_{41} = 2$, oraz $q_{13} = q_{24} = q_{31} = q_{42} = 4$.

W tabelicy 1 podano przydziały części do stacji (zmienne x_{ij}) wyznaczone poprzez rozwiązanie zadania programowania całkowitoliczbowego przy użyciu pakietu LINGO [13] (x_{ij}^*) oraz przy zastosowaniu algorytmu ROC.Tabu.Search (x_{ij}^H).

Zmienne y_{ijk} wyznaczają dla każdego typu wyrobu marszruty w postaci ciągu kolejno odwiedzanych stacji. Dla modelu programowania całkowitoliczbowego i pakietu LINGO otrzymano: dla $k = 1$: $i = 4,3,2$; dla $k = 2$: $i = 4,3,2$; dla $k = 3$: $i = 3,2,1$; dla $k = 4$: $i = 4,1,2,1,4$; dla $k = 5$: $i = 3,2,1,4$.

Natomiast stosując algorytm ROC.Tabu.Search otrzymano: dla $k = 1$: $i = 3,4,3,2$; dla $k = 2$: $i = 3,4,2$; dla $k = 3$: $i = 4,3,2,1$; dla $k = 4$: $i = 3,1,2,1,4$; dla $k = 5$: $i = 3,2,1,4$.

Łączny czas montażu i transportu dla stacji $i = 2$, będącej wąskim gardłem w systemie wynosi $PQ_{max} = 800$ jednostek. Dla stacji $i = 4$ otrzymano ten sam rezultat. Dla pozostałych stacji łączne czasy montażu i transportu wynoszą: 640 dla stacji $i = 1$ oraz 720 dla stacji $i = 3$. Czas obliczeń CPU = 1godz.12min.

Zastosowanie heurystyki ROC.Tabu.Search doprowadziło do uzyskania łącznego czasu montażu i transportu dla stacji $i = 2$ będącej wąskim gardłem w systemie, $PQ_{max} = 880$. Dla pozostałych stacji uzyskano jednakowe łączne czasy montażu i transportu równe 720. Czas obliczeń CPU = 2min.15sek.

Obliczenia przeprowadzono na mikrokomputerze typu PC 3863MHz z koprocесorem arytmetycznym.

5. Podsumowanie

W celu sprawdzenia jakości zaproponowanego algorytmu przeprowadzono szereg symulacji komputerowych. Dane wejściowe były generowane losowo.

Wyniki oceniano za pomocą dwóch współczynników:

Współczynnika względnego przyrostu łącznego czasu montażu i transportu dla stacji będącej wąskim gardłem:

$$\epsilon_{PQ} = \frac{PQ_{max}^H - PQ_{max}^*}{PQ_{max}^*}$$

gdzie: PQ_{max}^* = wartość optymalna funkcji celu (1) wyznaczona za pomocą pakietu LINGO,

PQ_{max}^H = wartość funkcji celu (1) wyznaczona za pomocą heurystyki ROC.Tabu.Search.

Współczynnika względnego skrócenia czasu obliczeń:

$$\epsilon_{CPU} = \frac{CPU^H}{CPU^*}$$

gdzie: CPU^* = czas obliczeń dla pakietu LINGO,

CPU^H = czas obliczeń dla heurystyki ROC.Tabu.Search.

Wartości funkcji celu otrzymane za pomocą algorytmu ROC.Tabu.Search były bliskie otrzymanym dla modelu programowania całkowitoliczbowego i pakietu optymalizacji dyskretnej LINGO. Odchyłki nie przekroczyły 12%. Natomiast czasy obliczeń dla algorytmu ROC.Tabu.Search były od 30 do 150 razy krótsze.

Wystarczająca dla praktyki dokładność oraz krótkie czasy obliczeń wskazują na możliwość zastosowania algorytmu ROC.Tabu.Search. Dalsze prace są prowadzone w celu poprawy dotychczasowych rezultatów. Ponadto algorytm zmodyfikowano tak, aby uwzględnić liczbę wózków transportowych oraz występujący w elastycznych liniach montażowych warunek montażu bez powrotów. Możliwa jest także rozbudowa algorytmu w celu podziału zbioru wszystkich wyrobów na rozłączne partie, montowane oddzielnie.

LITERATURA

- [1] Ammons J.C., Lofgren C.B., McGinnis L.F.: A large scale machine loading problem in flexible assembly. *Annals of Operations Research*, vol.3, 1985, pp.319-332.

- [2] Ghosh S., Gagnon R.J.: A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, vol.27, 1989, pp.637-670.
- [3] Glover F.: Tabu Search, Part I. *ORSA Journal of Computing*, vol. 1, no 3, 1989, s.190-206.
- [4] Glover F.: Tabu Search, Part II. *ORSA Journal of Computing*, vol. 2, no 1, 1990, s.4-32.
- [5] Glover F.: Tabu Search: A Tutorial. *Interfaces*, vol. 20, no 4, 1990, s.74-90.
- [6] King J.R., Nakornchai V.: Machine-component group formation in group technology. Review and extensions. *International Journal of Production Research*, vol. 20. No. 2, 1982 s.117-133
- [7] Korcyl A.: Optymalizacja rozdziału zadań i zasobów w elastycznym systemie produkcyjnym. Rozprawa doktorska, AGH 1992.
- [8] Korcyl A., Sawik T.: Algorytm typu tabu dla wyznaczania partii produkcyjnych i obciążenia maszyn w elastycznym systemie produkcyjnym. *Zeszyty Naukowe Politechniki Śląskiej, Automatyka*, z.109, 1992, s.147-157.
- [9] Nowicki E.: Zastosowanie techniki przeszukiwania z zabronionymi ruchami do problemów przepływowych z ustalonymi i zmiennymi czasami wykonywania zadań. *Zeszyty Naukowe AGH, z. 64 Automatyka*, 1993, s.323-335.
- [10] Sawik T.: Modele zadań krótkookresowego planowania produkcji w elastycznym systemie produkcyjnym. *Zeszyty Naukowe Politechniki Śląskiej, Automatyka*, z. 101, 1990, s.249-261.
- [11] Sawik T.: Optymalizacja dyskretna w elastycznych systemach produkcyjnych. WNT, Warszawa 1992.
- [12] Sawik T.: Modele zadań równoważenia obciążeń maszyn w elastycznych systemach montażowych. *Zeszyty Naukowe Politechniki Śląskiej, Automatyka*, 1994.
- [13] Schrage L., Cunningham K., LINGO, Optimization Modeling Language, LINDO Systems Inc., Chicago, 1991.

Recenzent: Prof.dr hab.inż. Jan Węglarz

Wpłynęło do Redakcji do 30.04.1994r.

Abstract

The paper presents an integer programming formulation and a tabu search algorithm for balancing of a flexible assembly system.

A flexible assembly system is made up of a set of assembly stations (usually robots) linked with an automated handling system. Each station has a limited working space, where limited number of part feeders can be placed. As a result each station is capable of performing a limited number of different assembly operations.

The objective of the FAS balancing problem is to assign parts and assembly operations to stations for a set of different assembly types so as to balance workloads of machines and material handling system. A heuristic called ROC_Tabu_Search is presented for solving

the balancing problem. The heuristic is composed of the tabu search and ROC (Rank Order Clustering) algorithms modified appropriately to account for the relations between assembly types and part type.

Simulation studies have been performed to test the performance of the ROC-Tabu-Search algorithm and to compare the solution results with those obtained by using LINGO discrete optimizer. The results have indicated that the ROC-Tabu-Search algorithm yields good solutions in short CPU time.

FORMULATION OF BALANCING PROBLEMS IN A FACTORY
THE SEARCH OF ALGORITHMS AND PRACTICAL SOLUTIONS

Abstract. This paper presents a study on the problem of balancing production lines with respect to work-in-progress (WIP) control. It also discusses the possibility of a related approach where operations are grouped by location in order to reduce the number of material handling operations. The results of a series of simulation studies of material handling and work-in-progress management are presented. The authors conclude that the proposed algorithm is a promising one for the study of line balancing.

PROBLEMY WYRÓWNIENIA PRACY W FABRYCE
SZUKANIE ALGORYTMÓW I PRAKTYCZNYCH ROZWIĄZAŃ

Streszczenie. W tym artykule przedstawiono badanie problemu wyrównania linii produkcyjnych z punktu widzenia kontroli ilości towarów w toku (WIP). Omówiono również możliwość zastosowania podobnego podejścia, w którym operacje są grupowane według lokalizacji, co prowadzi do zmniejszenia liczby operacji obszarowych. Wyniki szeregu badań symulacyjnych dotyczących sterowania przepływem materiałów i zarządzania pracą w toku są również przedstawione. Autorzy konkludują, że zaproponowany algorytm jest obiecującym narzędziem do badania wyrównania linii produkcyjnych.

1. Wprowadzenie

W artykule omówiono problem wyrównania linii produkcyjnych z punktu widzenia kontroli ilości towarów w toku (WIP). Omówiono również możliwość zastosowania podobnego podejścia, w którym operacje są grupowane według lokalizacji, co prowadzi do zmniejszenia liczby operacji obszarowych. Wyniki szeregu badań symulacyjnych dotyczących sterowania przepływem materiałów i zarządzania pracą w toku są również przedstawione. Autorzy konkludują, że zaproponowany algorytm jest obiecującym narzędziem do badania wyrównania linii produkcyjnych.