

Lech JAMROŻ, Jerzy RASZKA  
Politechnika Krakowska

## ALGORYTM GENETYCZNY DO RÓWNOWAŻENIA OBCIĄŻEŃ MASZYN W ELASTYCZNYM SYSTEMIE PRODUKCYJNYM

**Streszczenie.** W artykule przedstawiono algorytm do równoważenia obciążeń maszyn w elastycznym systemie cyklicznych procesów produkcyjnych. Rozważane jest zagadnienie utrzymania założonej produktywności systemu przy minimalnej liczbie wyrobów przebywających jednocześnie w systemie. Modelowanie i sterowanie odbywa się na poziomie operacyjnym. Do modelowania systemu wykorzystano czasowe sieci Petriego. Czas cyklu pracy systemu przyjęto jako miarę wydajności systemu. Do rozwiązania rozważanego problemu użyto algorytmu genetycznego.

## A GENETIC ALGORITHM FOR MACHINE LOADING IN FLEXIBLE PRODUCTION SYSTEM

**Summary.** The paper presents an algorithm for balance the machine workloads in flexible system of cyclic production processes. The problem work-in-process reduction while reaching a given productivity is considered. The modelling and control on operational level of production control hierarchy are developed. Timed Petri nets for modelling is utilized. As a performance evaluation measure of system the cycle time is selected. The maximum throughput of system is achieved for minimal cycle time. Due to known of complexity of this problem a genetic algorithm is provided.

### 1. Wprowadzenie

Współczesne Elastyczne Systemy Produkcyjne (ESP) dostarczają nowych do rozwiązania problemów, które związane są z ich optymalizacją i sterowaniem [2,10]. Wymaganie wieloasortymentowej i krótkoseryjnej produkcji wymusza stosowanie nowych narzędzi analizy ESP.

Z punktu widzenia sterowania na poziomie operacyjnym ESP widziany jest jako system potokowych procesów cyklicznych realizowanych współbieżnie. Pojedynczy proces związany jest z określoną sekwencją operacji wykonywanych zgodnie z zadany programem produkcyjnym. Poszczególne operacje wykonywane są na podstawie zasobów systemowych, a dostęp do nich wymaga koordynacji procesów.

Realizacja pojedynczego procesu potokowego związana jest z wykonaniem określonego typu wyrobu. Zakładamy, że partie wyrobów mogą być wykonywane w trybie cyklicznym. Potokowy charakter procesu odnosi się do wytwarzania kolejnego wyrobu tego samego typu, którego produkcja rozpoczyna się w miarę dostępności maszyn jeszcze przed ukończeniem rozpoczętych.

Procedury do symulacji dynamiki systemu zbudowane zostały na bazie czasowych sieci Petriego [ 4]. Mechanizm synchronizacji zapewniający bezblokadowy przebieg procesów oparty jest na protokole „wzajemnego wykluczania”. Wyznaczenie czasu cyklu pracy systemu pozwala określić wydajność pracy systemu. Dla minimalnego czasu cyklu utrzymywana jest największa szybkość przebiegu procesów. Pozwala to z kolei uzyskać największą przepustowość systemu [ 8].

## 2. Specyfikacja systemu

Rozważany jest system produkcyjny typu job-shop. Niech  $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$  będzie zbiorem typów wyrobów. Wykonanie wyrobu typu  $J_k \in J$  związane jest z wykonaniem wg zadanego programu produkcyjnego ciągu operacji  $J_k = \langle o^k_1, o^k_2, \dots, o^k_i, \dots, o^k_{n_k} \rangle$ ,  $o^k_{i-1} < o^k_i < o^k_{i+1}$ ,  $k=1, 2, \dots, n$ ,  $i=2,3, \dots, n_k-1$ . Z każdym wyrobem  $J_k$  związana jest marszruta technologiczna  $m^k = \langle m^k_1, m^k_2, \dots, m^k_{n_k} \rangle \in M^k$ , gdzie  $M^k \subset M = \{m_1, m_2, \dots, m_r\}$  jest zbiorem maszyn. Operacja  $o^k_i \in J_k$  jest wykonywana na maszynie  $m_i$  w czasie  $\tau^k_i$ .

## 3. Model sieciowy systemu. Pojęcia i oznaczenia

Model rozważanego systemu formułowano na podstawie czasowych sieci Petriego [ 3]. Czasowa znakowana sieć Petriego (*TMSP*) jest uporządkowaną piątką  $TMSP = (P, T, F, M_0, \tau)$ , gdzie:  $P = \{p_1, p_2, \dots, p_n\}$  - zbiór miejsc,  $T = \{t_1, t_2, \dots, t_m\}$  - zbiór tranzycji,  ${}^{\circ}p = \{t : (t, p) \in F\}$  - zbiór wejściowych tranzycji miejsca  $p$ ,  $p^{\circ} = \{t : (p, t) \in F\}$  - zbiór wyjściowych tranzycji miejsca  $p$ .  $M_0: P \rightarrow \mathbb{N}$  - funkcja znakowania (markowania) początkowego, ( $\mathbb{N} = \{0, 1, \dots\}$ ),  $\tau: T \rightarrow \mathbb{R}^+$  - funkcja czasu wzbudzenia tranzycji ( $\mathbb{R}^+$  jest zbiorem dodatnich liczb rzeczywistych),  $F \subset (P \times T) \cup (T \times P)$  - zbiór łuków.

Sieć Petriego *TMSP*, dla której spełnione jest  $|{}^{\circ}p| = |p^{\circ}| = 1$  dla każdego  $p \in P$ , jest czasowym znakowanym grafem (*TMG*). Czasy operacji technologicznych odwzorowywane są za pomocą czasów wzbudzeń tranzycji  $\tau_i = \tau(t)$ ,  $t \in T$ . Tranzycja  $t^k_i \in T$  w modelu sieciowym reprezentuje operację  $o^k_i \in J_k$ .

Rozważmy cykliczny proces wzbudzeń tranzycji w grafie  $TMG$ . Czas, po którym zostaje odtworzone znakowanie grafu  $TMG$ , nazywamy czasem cyklu [ 3]. W celu obliczenia czasu cyklu wyznaczmy elementarne obwody w grafie  $TMG$ . Elementarnym obwodem  $\gamma$  nazywamy ciąg wierzchołków  $\{v_0, v_1, \dots, v_i, v_j, \dots, v_m\}$  takich, że  $v_i \in P \cup T$ ,  $v_0 = v_m$ ,  $v_i \neq v_j$ , dla każdego  $i, j = 1, 2, \dots, m-1$ . Niech  $\Gamma$  będzie zbiorem elementarnych obwodów w grafie  $TMG$ .

Dla każdego  $\gamma \in \Gamma$  czas cyklu obwodu jest definiowany w postaci [ 8]:

$$\beta(\gamma) = (\tau(\gamma)/M(\gamma)), \tag{1}$$

gdzie:  $\tau(\gamma) = \sum_{t \in \gamma} \tau_t$ ,  $M(\gamma) = \sum_{p \in P} M(p)$ ,  $M(p) \in \{0, 1\}$

$$\beta^* = \beta(\gamma^*) = \max_{\gamma \in \Gamma} \beta(\gamma). \tag{2}$$

W grafie  $TMG$  dla danego czasu cyklu  $\beta^*$  najwcześniejszy moment wzbudzenia tranzycji  $t_p \in T$  wynosi:

$$x_{p^o}(k) + \tau(t_p) \leq x_{p^o}(k) + \beta^* M_0(p), \quad \forall p \in P, \quad k = 1, 2, \dots \tag{3}$$

gdzie:  $x_{p^o}$  - jest chwilą k-tego wzbudzenia tranzycji  $t_p \in T$

#### 4. Sformułowanie problemu

Niech  $\Pi$  będzie zbiorem czasowych sekwencji  $\pi$  wzbudzeń tranzycji w grafie  $TMG$ , tj.  $\pi = \{ \langle t_i; x_i(1), x_i(2), \dots, x_i(k), \dots \rangle; t_i \in T \}$ . Zakładamy, że liczba  $C(\pi)$  jest czasem cyklu dla grafu  $TMG$ , jeżeli istnieje czasowa sekwencja wzbudzeń  $\pi \in \Pi$  taka, że znakowanie grafu w chwilach  $kC(\pi)$ ,  $k \in \{1, 2, \dots\}$  jest równe znakowaniu początkowemu.

Problem minimalizacji czasu cyklu pracy systemu sformułowany jest następująco:

$$C(\pi^*) = \min_{\pi \in \Pi} C(\pi) \tag{4}$$

przy ograniczeniach:

$$\tau(\gamma)/M_0(\gamma) \leq \beta^*, \quad \forall \gamma \in \Gamma, \tag{5}$$

$$\tau(t_p) \leq x_{p^o}(k) - x_{p^o}(k) + \beta^* M_0(p), \quad \forall p \in P. \tag{6}$$

Problem (4) - (6) jest rozwiązywany z wykorzystywaniem algorytmu genetycznego.

## 5. Metoda rozwiązania

Z uwagi na NP-zupełność rozwiązania rozważanego problemu [1] do jego rozwiązania zaproponowano metodę wykorzystującą algorytm genetyczny (AG) [5, 9]. Zachowując charakterystyczne dla poszukiwań iteracyjne otrzymywanie rozwiązań, spodziewane polepszanie uzyskiwane jest ewolucyjną zasadą przeżycia najlepiej przystosowanych osobników. Wykorzystując podstawowe cechy i operacje tego algorytmu oparto się na mechanizmach doboru naturalnego i dziedziczności. Zastosowano kodowanie binarne wektora znakowania początkowego  $M_0$  oraz modyfikacje położenia znaczników wg podstawowych operacji AG. Nowe rozwiązania uzyskiwane są w wyniku wymiany położenia znaczników w zbiorze najlepiej przystosowanych wektorów znakowania  $M_0$ .

Wykorzystano ideę podstawowego algorytmu genetycznego, w którym zastosowano trzy operacje:

- 1) selekcja - operacja, w której uzyskuje się zbiór wektorów znakowania początkowego poprzez ich powielenie w stopniu zależnym od aktualnie otrzymanej wartości funkcji celu,
- 2) krzyżowanie - operacja, dla której z pewnym średnim prawdopodobieństwem zachodzi:
  - a) kojarzenie w pary wektorów znakowania ze zbioru uzyskanego w pkt 1),
  - b) losowy wybór wspólnego dla każdej pary wektorów punktu krzyżowania przez dokonanie podziału wektora (dwa podciągi kodowe reprezentujące położenie znaczników),
  - c) wymiana wyznaczonych przez punkt krzyżowania podciągów pomiędzy wektorami,
- 3) mutacja - występująca sporadycznie operacja wprowadzenia lub usunięcia pojedynczego znacznika.

W literaturze [1, 5, 11] istnieje wiele innych propozycji operacji zakwalifikowanych do zaawansowanych algorytmów genetycznych, jak np.: dominacja, inwersja, duplikacja, translokacja i segregacja. W przyjętym zakresie badań dokonano pewnych adaptacji pod kątem rozważanego problemu. W szczególności dotyczy to uporządkowania i segmentacji kodu znakowania początkowego wg struktury modelu sieciowego oraz wprowadzenie nowych operacji rekonfigurujących.

## 6. Założenia do programu obliczeniowego

Opracowany program ma strukturę modułową i oparty jest na dwóch modułach: PTSIM oraz PTGA. Moduł PTSIM dokonuje symulacji zachowania się modelu systemu

i pozwala na wyznaczenie współczynnika wykorzystania maszyn i czasu cyklu. Procedury inicjujące tego modułu specyfikują parametry modelu sieciowego. Poprzez określenie obwodów możliwe jest oszacowanie górnej granicy niezbędnego czasu symulacji  $t_{\max} \leq \tau(\gamma)$  [6]. Moduł PTGA zawiera procedury związane z algorytmem genetycznym i generuje wektory znakowania początkowego. Dla tych wektorów reprezentujących uzyskiwane rozwiązania  $M_0$  zastosowano kodowanie binarne. Zapewnia ono maksymalną liczbę wykrywanych podobieństw strukturalnych (schematów podobieństwa) oraz w przypadku rozważanego problemu w sposób bezpośredni reprezentuje pozycje znaczników. Procedury inicjujące modułu PTGA określają początkowy zbiór wektorów znakowania  $M_0$  (pokolenie startowe).

W prezentowanej wersji AG wprowadzone zostały pewne adaptacje w zakresie struktury danych i operacji. Biorąc pod uwagę często przypadkowy sposób kodowania, który może mieć wpływ na przebieg rozwiązania, uporządkowane zostały pozycje w ciągu kodowym. Poprzez odpowiednie zaetykietowanie miejsc  $p_i \in P$  w modelu sieciowym wstępnie przyjęto podział na obwody synchronizacji  $\gamma_i^S$ ,  $i \in (1, 2, \dots, r)$  i obwody procesów  $\gamma_i^P$ ,  $i \in (1, 2, \dots, n)$ . W zakresie każdego z tych obwodów etykietowanie uwzględnia naturalną kolejność technologiczną. Ponadto w obwodach synchronizacji  $\gamma_i^S$  założono występowanie dokładnie jednego znacznika.

Z uwagi na silne ograniczenia i wynikający stąd niewielki stosunkowo obszar dopuszczalnych rozwiązań w celu określenia pokolenia początkowego zastosowano adaptację algorytmu polegającą na zwielokrotnieniu prób losowych. W efekcie tego w tym pokoleniu uzyskuje się zwiększenie średniego wskaźnika przystosowania i zmniejszenie uzależnienia procesu poszukiwań od początkowego ustawienia generatora losowego

W module PTGA generowane kolejne populacje uzyskiwane są poprzez selekcję, krzyżowanie i mutację, przy czym dwie ostatnie operacje realizowane są jedynie dla stanów znaczników reprezentujących obwody procesów  $\gamma_i^P$ . Jest to istotna modyfikacja algorytmu genetycznego, polegająca na zamianie tych operacji na operację losowego przesunięcia znacznika w obwodach synchronizacji  $\gamma_i^S$ .

## 7. Wyniki eksperymentów numerycznych. Przykład obliczeniowy

W celu przetestowania algorytmu został napisany program w języku C++. Badania wykonano dla przykładowego systemu procesów, w którym poszczególne obwody procesów są postaci:

$$\gamma_1^P: < t_1 p_1 t_2 p_2 t_3 p_3 t_1 >$$

$$\gamma_2^P: < t_4 p_4 t_5 p_5^c t_4 >$$

$$\gamma_3^P: < t_6 p_6 t_7 p_7 t_8 p_8^c t_6 >$$

natomiast obwody synchronizacji:

$$\gamma_1^S: < t_1 p_9 t_4 p_{10} t_6 p_{11} t_1 >$$

$$\gamma_2^S: < t_2 p_{12} t_8 p_{13} t_2 >$$

$$\gamma_3^S: < t_3 p_{14} t_5 p_{15} t_7 p_{16} t_6 >$$

Tablica 1

Dane specyfikujące procesy

ma- szyny.	obwody		
	$\gamma_1^P$	$\gamma_2^P$	$\gamma_3^P$
m <sub>1</sub>	$\tau_1=1$	$\tau_4=3$	$\tau_6=2$
m <sub>2</sub>	$\tau_2=5$	-	$\tau_8=4$
m <sub>3</sub>	$\tau_3=1$	$\tau_5=2$	$\tau_7=3$
	$\tau_i = \tau(t_i)$		

Tablica 1 zawiera czasy wzbudzeń tranzycji w obwodach  $\gamma_i^P$ .

Miarą stopnia wykorzystania maszyn w cyklu pracy systemu  $C(\pi^*)$  jest współczynnik  $\xi = \sum_i \xi_i / \tau$ , gdzie:  $\xi_i = \Delta T_i / C(\pi^*)$  jest stopniem wykorzystania maszyny  $m_i$  pracującej przez czas  $\Delta T_i$  w cyklu  $C(\pi^*)$ .

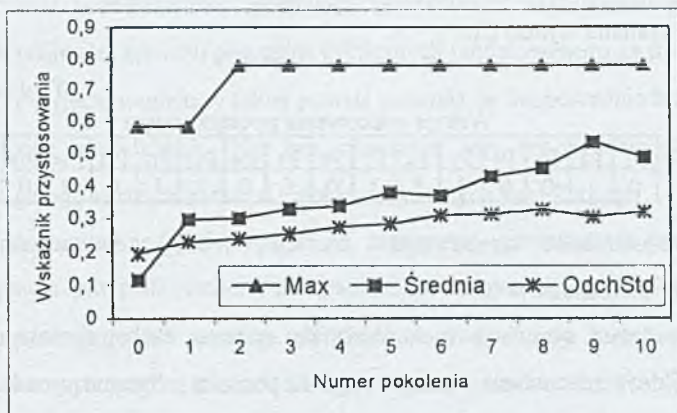
Przykładowe wyniki z przebiegu procesu obliczeniowego dla wybranych rozwiązań przedstawiono w tablicy 2. Przedstawia ona uzyskane wskaźniki w postaci ilości znaczników w poszczególnych obwodach procesów  $\gamma_i^P \in \Gamma$  i synchronizacji  $\gamma_i^S \in \Gamma$ , stopień wykorzystania maszyn  $\xi$  oraz czas cyklu  $C=C(\pi^*)$  w zależności od generowanego wektora znakowania początkowego  $M_0$ .

Wyniki statystyczne z przebiegu poszukiwań w postaci wartości maksymalnej (Max), średniej (Śred) i standardowego odchylenia (OdchStd) dla wskaźnika przystosowania  $\xi$  w kolejnych pokoleniach przedstawia rys. 1.

Tablica 2

Wybrane przykładowe wyniki z przebiegu poszukiwań

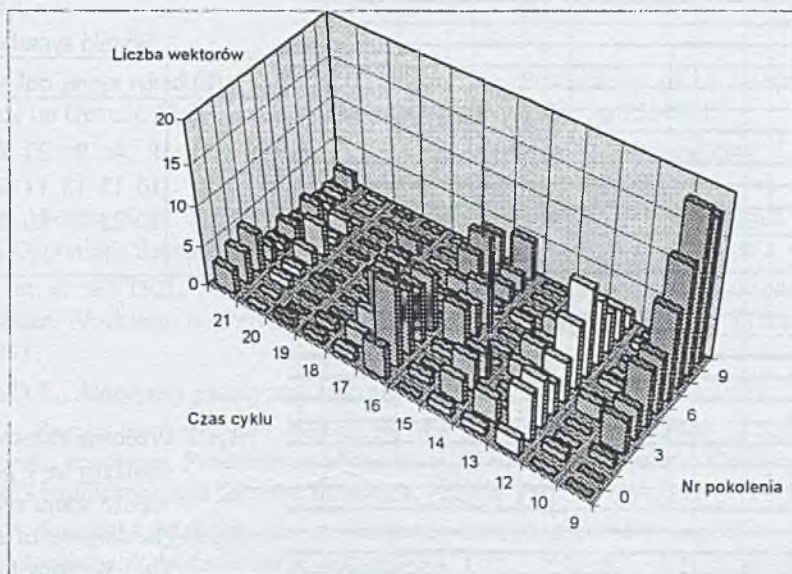
Liczby znaczników procesy synchr. {Σ=1 2 3   1 2 3}	Wskaźn.		Pozycje znaczników w $M_0$						
	$\xi$	C	$\gamma_1^P$	$\gamma_2^P$	$\gamma_3^P$	$\gamma_1^S$	$\gamma_2^S$	$\gamma_3^S$	
<b>Pokolenie nr 0</b>									
{8=2 1 2   1 1 1}	0,44	16	011	10	110	010	10	100	
{8=2 1 2   1 1 1}	0,33	21	110	10	011	001	01	010	
{7=2 1 1   1 1 1}	0,41	17	110	10	010	001	01	010	
{7=2 1 1   1 1 1}	0,50	14	110	10	100	100	10	001	
{7=2 1 1   1 1 1}	0,37	19	110	01	100	001	10	001	
{7=2 1 1   1 1 1}	0,39	18	110	01	010	010	01	001	
{8=2 1 2   1 1 1}	0,44	16	011	01	110	001	01	001	
{8=2 1 2   1 1 1}	0,58	12	011	01	110	001	10	010	
{7=1 1 2   1 1 1}	0,44	16	010	10	110	010	10	100	
{8=2 1 2   1 1 1}	0,33	21	101	10	110	100	01	100	
{8=2 1 2   1 1 1}	0,33	21	101	10	101	001	01	001	
<b>Pokolenie nr 9</b>									
{8=2 1 2   1 1 1}	0,78	9	011	01	110	001	10	001	
{7=1 1 2   1 1 1}	0,78	9	010	01	110	001	10	001	
{8=2 1 2   1 1 1}	0,78	9	011	01	110	001	10	001	
{8=2 1 2   1 1 1}	0,78	9	011	01	110	001	10	001	
{7=1 1 2   1 1 1}	0,44	16	010	01	110	001	01	001	
{7=1 1 2   1 1 1}	0,44	16	010	01	110	001	01	001	
{8=2 1 2   1 1 1}	0,58	12	011	01	110	001	10	010	
...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	
{8=2 1 2   1 1 1}	0,47	15	011	01	110	001	10	100	
{8=2 1 2   1 1 1}	0,78	9	011	01	110	001	10	001	
{7=1 1 2   1 1 1}	0,78	9	010	01	110	001	10	001	
{7=1 1 2   1 1 1}	0,78	9	010	01	110	001	10	001	



Rys. 1. Wartości maksymalne, średnie i odchylenia standardowego dla wskaźnika przystosowania w kolejnych pokoleniach

Fig. 1. Maximum, average and standard deviation values for fitness in following generations

Na rys. 2 przedstawiono liczbę otrzymanych identycznych wektorów znakowania w funkcji obliczonego czasu cyklu w kolejnych pokoleniach.



Rys.2. Liczba wektorów markowania początkowego w funkcji czasu cyklu używana w kolejnych pokoleniach

Fig.2. Number of first mark vectors in function of cycle time obtained for following generations

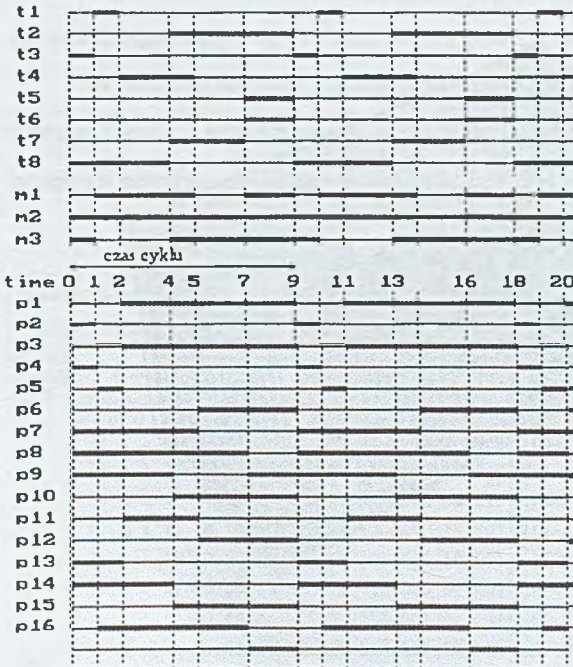
Otrzymany stan początkowy znakowania przedstawiony w tablicy 3 uzyskano na podstawie wyników z 9 populacji uwzględniając warunki minimalnego znakowania. Minimalny czas cyklu pracy systemu wynosi  $C(\pi^*) = 9$ .

Tablica 3

Wektor znakowania początkowego															
$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$	$p_8$	$p_9$	$p_{10}$	$p_{11}$	$p_{12}$	$p_{13}$	$p_{14}$	$p_{15}$	$p_{16}$
0	1	0	0	1	1	1	0	0	0	1	1	0	0	0	1

Prawdopodobieństwa krzyżowania i mutacji wynosiły odpowiednio 0,6 i 0,03, natomiast rozmiar pokolenia był 40.

Przeprowadzono symulację zachowania się systemu dla otrzymanego algorytmem genetycznym wektora znakowania początkowego za pomocą programu symulacyjnego [ 6 ] i aparatu MaxPlus algebry [7]. W tablicy 4 przedstawiono wyniki w postaci wektora  $x(k)$  dla  $k = 0, 1, 2, \dots$  zawierającego dyskretne chwile wzbudzeń tranzycji w kolejnych cyklach. Z tablicy tej można określić czas cyklu systemu  $C^* = x(k+1) - x(k) = 9$ . Wyniki w postaci wykresów czasowych przedstawia rys. 3.



Tablica 4

Wyniki symulacji

$x(k)$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
$x(0)$	-	-	0	-	-	-	-	0
$x(1)$	1	4	9	2	7	7	2	9
$x(2)$	10	13	18	11	16	16	11	18
$x(3)$	19	22	27	20	25	25	20	27
...	...	...	...	...	...	...	...	...

Rys.3. Przebiegi czasowe wzbudzeń tranzycji  $t_i \in T$  pracy maszyny  $m_k \in M$  stanu znakowania  $p_j \in P$   
 Fig.3. The diagram of enable transition  $t_i \in T$  working time of machines  $m_k \in M$  and marking state  $p_i \in P$



## 8. Podsumowanie

W artykule przedstawiono możliwość zastosowania algorytmów genetycznych do oszacowania wydajności systemu procesów cyklicznych modelowanego za pomocą czasowych sieci Petriego. Przyjęcie modelu w takiej postaci pozwala na bezpośrednie binarne kodowanie zakresu dziedziny poszukiwań. Nie bez znaczenie jest też możliwość wykorzystania istniejącego bogatego zestawu narzędzi i metod dotyczących sieci Petriego.

Zaproponowane podejście oparte na algorytmie genetycznym charakteryzuje się wysokim stopniem przypadkowości i nie gwarantuje otrzymania optymalnego rozwiązania. Algorytmy te cechuje jednak wysoka odporność w stosunku do metod czysto losowych. Przejawia się ona w przypadku dużego rozmiaru obszaru poszukiwań globalnych o charakterze poszukiwań z podwyższoną efektywnością.

Z przeprowadzonych badań wg zaproponowanej metody wynika konieczność zwrócenia bacznej uwagi na dobre przygotowanie populacji startowej oraz włączenie mechanizmów ograniczających generację przypadków nieefektywnych z góry skazanych na niepowodzenie.

## LITERATURA

1. Davis L.: Job shop scheduling with genetic algorithms. Proceedings of an International Conference on Genetic Algorithms and Their Applications, 1985, pp.136-140.
2. Di Cesare F., et. al.: Practice of Petri nets in manufacturing. Chapman&Hall, London, 1993.
3. Dubois D., Stecke K.E.: The algebra of timed marked graphs and algorithmic issues. Annals of Operations Research, 1990.
4. Gellot F., et. al.: PETRILAM: A Tool for Petri nets analysis and simulation, Proceedings of 15-th Inter. Workshop on Petri nets and Performance Models, Toulouse, France, Oct. 20-22. 1993.
5. Goldberg D.E.: Algorytmy genetyczne i ich zastosowanie. WNT, Warszawa 1995.
6. Jamroz L., Raszka J.: Simulation method for the performance evaluation of system of discrete cyclic processes. Proceedings of the 16-th IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria. Feb. 17-19th, 1997, 190-193.
7. Jamroz L., Raszka J.: Modelowanie mechanizmu dostępu procesów współbieżnych do wspólnych zasobów. Półrocznik Seria: Automatyka. Tom 3. Zeszyt 1, AGH, Kraków 1999, pp. 167-174.
8. Laftit S., Proth J. M., Xie X. L.: Optimization of invariant criteria for event graphs, IEEE Transactions on Automatic Control, vol.37, no. 5, 1992, pp. 547-555.
9. Michalewicz Z.: Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, Warszawa 1996.

10. Sawik T.: A two-level loading and scheduling of flexible assembly systems. *Automatyka t. 3, z. 1, 1999, pp. 297-305.*
11. Wala K., Chmiel W.: Nowy schemat procesu genetycznego poszukiwania na przykładzie zagadnienia harmonogramowania z maszynami równoległymi. *Zeszyty Naukowe Politechniki Śląskiej, s. Automatyka, z. 114, Gliwice 1994, pp. 67-77.*

Recenzent: Prof.dr hab.inż. J.Grabowski

### Abstract

The paper is concerned with method for the balance machine workloads in system of production cyclic processes, especially Flexible Manufacturing System (FMS). The modelling and control on operational level of the production control hierarchy are developed. Some system parameters as so as throughput of system and loading balancing of machines is choosen. At the same time the processes can have a cooperative and competitive relationship. The access of processes to each common resources is coordinated by mutual exclusion protocol. As a performance evaluation measure of system the cycle time is selected. The usefulness of timed Petri nets for scheduling of machines is investigated. This model is useful to analyze asynchronous and repetitive production processes. The productivity of a FMS can be expresses in terms of cycle time of its timed Petri nets. Reaching a maximum productivity of a FMS is thus equivalent to reach a minimal cycle time. In order to discuss the maximum performance we must consider the minimum distribution of work in process, which leads to the full utilization of the bottleneck machines. In this case the system works at maximal rate and the productivity is highest. The problem consists of maximizing the productivity with work-in-proces reduction. Due to known of complexity of this problem a genetic algorithm is provide.