

HARMONOGRAMOWANIE ZADAŃ W PRZEPEŁYWOWYM SYSTEMIE PRODUKCYJNYM Z OGRANICZONĄ LICZBĄ OPERATORÓW MASZYN

Jarosław PEMPERA

Streszczenie: Praca poświęcona jest harmonogramowaniu zadań w przepływowym systemie produkcyjnym z ograniczoną liczbą operatorów maszyn. W pracy, sformułowano problem optymalizacyjny, zaproponowano model matematyczny i grafowy, przedstawiono efektywny sposób wyznaczania harmonogramu dla zadanej kolejności wykonywania zadań na maszynach oraz operacji przez operatorów. Posługując się algorytmem opartym na metodzie symulowanego wyżarzania przeprowadzono szereg testów mających na celu określenie wpływu ograniczonej liczby operatorów na wydłużenie czasu realizacji wszystkich zadań.

Słowa kluczowe: problem przepływowy, szeregowanie, ograniczona liczba operatorów, symulowane wyżarzanie, optymalizacja.

1. Wstęp

Coraz większe trudności z pozyskiwaniem wysokiej klasy specjalistów wymuszają na przedsiębiorstwach działania mające na celu jak największe wykorzystanie czasu pracy zatrudnionych pracowników. Szczególnie istotne jest to w przedsiębiorstwach średniej wielkości, w których produkcja jest bardzo zróżnicowana i podporządkowana realizacji unikalnych zamówień klientów. Park maszynowy w takich przedsiębiorstwach składa się z wielu maszyn, przy czym tylko część z nich wykorzystywana jest bardzo intensywnie, pozostałe wykorzystywane są sporadycznie do realizacji specjalizowanych czynności. Z innej strony, ze względu na wielkość przedsiębiorstwa, dysponują one stosunkowo niewielką liczbą doświadczonych oraz dobrze motywowanych pracowników.

Zdecydowana większość systemów produkcyjnych występujących w tego typu firmach ma charakter przepływowy. Produkty w takich systemach przetwarzane są kolejno na maszynach reprezentujących kolejne etapy procesu wytwarzania, dodatkowo niektóre produkty mogą być przetwarzane na maszynach specjalizowanych.

Przepływowe systemy produkcyjne są intensywnie badane już od 50 lat [1]. Niemniej dopiero w ostatnim dziesięcioleciu obserwuje się duże zainteresowanie problemami produkcyjnymi z różnego rodzaju dodatkowymi ograniczeniami: bez czekania [2, 3], bez magazynowania [4,5], z ograniczonym składowaniem [6] oraz ograniczoną liczbą palet [7]. Praktycznie w każdym z rzeczywistych systemów występują niektóre z tego typu ograniczeń i mają istotny wpływ na przebieg procesu wytwarzania. W pracy przedstawiony jest sposób modelowania rzadko rozpatrywanego do tej pory ograniczenia jakim jest ograniczona liczba operatorów maszyn, przedstawiono wyniki testów komputerowych mających na celu określenie wpływu ograniczonej liczby operatorów na wydłużenie czasu realizacji wszystkich zadań.

2. Sformułowanie problemu

W systemie produkcyjnym składającym z m maszyn technologicznych ze zbioru $M=\{1, \dots, m\}$ należy wykonać n zadań ze zbioru $J=\{1, \dots, n\}$. Zadanie $j \in J$ wykonywane jest na wszystkich maszynach w kolejności zgodnej z indeksami (numerami) maszyn. Wykonywanie zadania $j \in J$ możemy podzielić na m czynności technologicznych (operacji) numerowanych kolejnymi liczbami naturalnymi $r_{j+1}, r_{j+2}, \dots, r_{j+m}$, gdzie $r_j = (j-1)m$. Operacja $o = r_j + k \in O$ zadania $j \in J$ wykonywana jest bez przerw na maszynie $k \in M$ przez czas $p_o > 0$. Oznaczmy przez $e(o)$ numer zadania do którego należy operacja o oraz przez $\mu(o)$ numer maszyny przyporządkowanej do wykonania tej operacji. Liczby te możemy wyznaczyć z następujących zależności: $e(o) = \lfloor o/m \rfloor + 1$, $\mu(o) = (o-1) \bmod m + 1$.

W systemie produkcyjnym należy wykonać nm operacji ze zbioru $O = \{1, \dots, nm\}$. W danej chwili można wykonywać tylko jedno zadanie na danej maszynie oraz w danej chwili można wykonywać tylko jedną operację danego zadania. Dodatkowo wymagana jest identyczna kolejność wykonywania zadań na wszystkich maszynach.

W systemie znajduje się z operatorów maszyn ze zbioru $Z = \{1, \dots, z\}$. Podczas wykonywania operacji $o \in O$ niezbędna jest interwencja operatora przez czas p_o . Operator inicjuje wykonywanie operacji (wkłada detal na maszynę), nadzoruje przebieg wykonywania operacji oraz kończy wykonywanie operacji (usuwa detal z maszyny). Operator w każdej chwili może wykonywać (nadzorować) tylko jedną operację, każdy operator może obsługiwać dowolną maszynę, czas przejścia operatora pomiędzy stanowiskami k oraz l , $k \neq l$, $k, l \in M$, wynosi $t_{kl} \geq 0$. Oczywiście wartość $t_{kl} = 0$ dla $k = l$.

Harmonogram wykonywania zadań w systemie możemy jednoznacznie opisać przy pomocy pary wektorów (S, A) , gdzie S_o określa termin rozpoczęcia wykonywania operacji $o \in O$, natomiast a_o określa numer operatora przyporządkowanego do wykonania tej operacji. Termin zakończenia wykonania operacji $o \in O$ następuje w momencie $C_o = S_o + p_o$.

Celem optymalizacji jest wyznaczenie dopuszczalnego (spełniającego w/w ograniczenia) harmonogramu wykonywania operacji przez operatorów na maszynach minimalizującego czas wykonania wszystkich zadań.

Niech $\sigma = (\sigma(1), \dots, \sigma(n))$ będzie permutacją określoną na zbiorze $\{1, \dots, n\}$. Permutacja σ określa kolejność wykonywania zadań na każdej z maszyn w systemie produkcyjnym. Dla ułatwienia dalszych rozważań, oznaczmy przez $o' = o + 1$ ($o' = 0$, gdy operacja o jest ostatnią operacją danego zadania) następnika technologicznego operacji $o \in O$, natomiast przez o^σ oznaczmy następnika maszynowego operacji $o \in O$ ($o^\sigma = 0$, gdy operacja o jest ostatnią operacją wykonywaną na maszynie $\mu(o)$ w kolejności σ). Numer następnika maszynowego można wyznaczyć ze wzoru: $o^\sigma = \sigma(ps^\sigma(e(o)) + 1)m + \mu(o)$, gdzie $ps^\sigma(x)$ oznacza pozycję elementu x w permutacji σ .

Kolejność wykonywania operacji przez operatora $l \in Z$ możemy opisać przy pomocy permutacji $\pi_l = (\pi_l(1), \pi_l(2), \dots, \pi_l(n_l))$, gdzie n_l oznacza liczbę operacji wykonywanych przez operatora $l \in Z$. Kolejność wykonywania operacji przez wszystkich operatorów możemy opisać przy pomocy zestawu permutacji $\pi = (\pi_1, \pi_2, \dots, \pi_z)$. Wszystkie możliwe kolejności generowane są przez zbiór:

$$\Pi = \{ \pi : \pi_s \in \Pi(N_s) \wedge N_s \text{ jest podziałem rozłącznym zbioru } O \}, \quad (1)$$

gdzie: $\Pi(X)$ jest zbiorem wszystkich permutacji elementów zbioru X .

Zauważmy, że podanie kolejności π jednocześnie określa również przydział operatorów

do operacji. Dla zadanej kolejności wykonywania zadań na maszynach σ oraz zadanej kolejności wykonywania operacji przez operatorów π harmonogram wykonywania operacji określony przez terminy rozpoczęcia S_o oraz zakończenia C_o wykonywania operacji $o \in O$ musi spełniać następujące warunki:

$$S_o \geq 0 \quad o \in O, \quad (2)$$

$$S_{o_i} \geq C_o, \quad o = r_j + k, j = 1, \dots, n, k = 1, \dots, m-1, \quad (3)$$

$$S_{o_{\sigma}} \geq C_o, \quad o = r_{\sigma(j)} + k, j = 1, \dots, n-1, k = 1, \dots, m, \quad (4)$$

$$S_{\pi_k(j+1)} \geq C_{\pi_k(j)} + t_{\mu(\pi_k(j)), \mu(\pi_k(j+1))} \quad j = 1, \dots, n_k-1, k = 1, \dots, z, \quad (5)$$

$$C_o \geq S_o + p_o \quad o \in O. \quad (6)$$

Ograniczenie (2) jest oczywiste, natomiast ograniczenie technologiczne (3) oznacza, że moment rozpoczęcia wykonywania następnej operacji tego samego zadania w ciągu technologiczny może nastąpić dopiero po zakończeniu realizacji danej operacji. Ograniczenie maszynowe (4) oznacza, że rozpoczęcie wykonywania następnej operacji na tej samej maszynie może nastąpić dopiero po zakończeniu danej. Nierówności (5) modelują ograniczenia wynikające z kolejności realizacji operacji przez operatorów, ograniczenie uwzględnia czas przejścia operatora pomiędzy maszynami. Równanie (6) wiąże moment zakończenia realizacji operacji z momentem jej rozpoczęcia. Para (σ, π) określa dopuszczalną kolejność wykonywania operacji na maszynach oraz przez operatorów, jeżeli istnieje rozwiązanie układu nierówności (2–6).

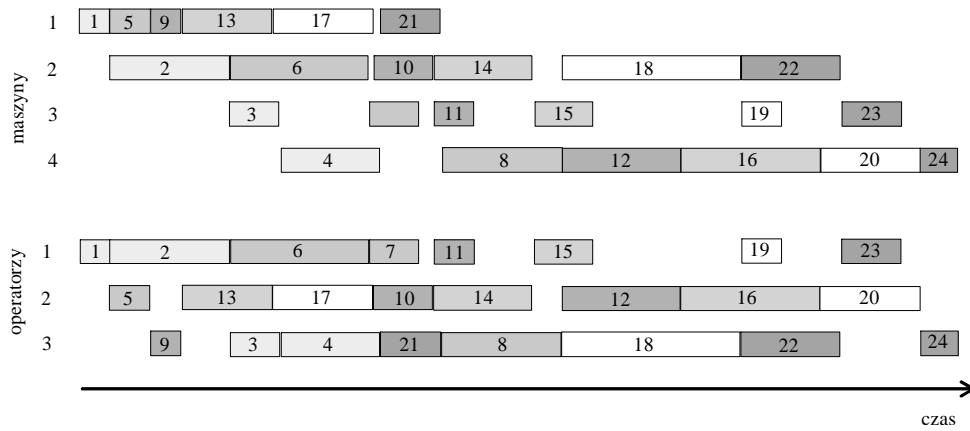
Dla ustalonej kolejności wykonywania operacji (σ, π) definiujemy maksymalny moment zakończenia realizacji wszystkich zadań w następujący sposób:

$$C_{\max}(\sigma, \pi) = \max_{o \in O} C_o \quad (6)$$

Zadanie optymalizacji polega na znalezieniu takiej dopuszczalnej pary (σ^*, π^*) , aby maksymalny moment zakończenia wykonywania wszystkich operacji był minimalny.

Rozważany problem optymalizacyjny jest problemem NP-trudnym, ponieważ permutacyjny problem przepływowy jest szczególnym przypadkiem rozważanego problemu. Łatwo można, że dla liczby operatorów równej liczbie maszyn tj. $z=m$, dla dowolnej kolejności wykonywania zadań na maszynach istnieje optymalny przydział operatorów do operacji taki, że operator k wykonuje wszystkie operacje przydzielone do maszyny k w kolejności zgodnej z kolejnością wykonywania zadań na tej maszynie.

Na Rys. 1 przedstawiono w postaci wykresu Gantta harmonogram wykonywania 7 zadań produkcyjnych na 4 maszyn z wykorzystaniem 3 operatorów maszyn. Dla uproszczenia, przyjęto $t_{kl}=0$, $k, l \in M$ oraz naturalną kolejność wykonywania zadań na maszynach tj. $\sigma=(1,2,3,4,5,6,7)$, natomiast $\pi=(\pi_1, \pi_2, \pi_3)$, gdzie $\pi_1=(1,2,6,7,11,15,19,23)$, $\pi_2=(5,13,17,10,14,12,16,20)$, $\pi_3=(9,3,4,21,8,18,22,24)$.



Rys. 1. Wykres Gantta dla kolejności (σ, π)

3. Model grafowy

Problem wyznaczenia dopuszczalnego harmonogramu wykonywania operacji dla ustalonej (σ, π) określającej kolejność wykonywania operacji na maszynach oraz przez operatorów, można sprowadzić do problemu wyznaczenia długości najdłuższych dróg w odpowiednio skonstruowanym grafie skierowanym $G(\sigma, \pi) = (N, T \cup F(\sigma) \cup E(\pi))$. Graf $G(\sigma, \pi)$ określony jest przez zbiór obciążonych węzłów N oraz zbiór łuków $T \cup F(\sigma) \cup E(\pi)$, gdzie :

$$N = O, \quad (5)$$

węzeł $j \in N$ reprezentuje $\mu(j)$ -tą operację zadania $e(j)$ i obciążony jest wagą p_j ,

$$T = \bigcup_{j=1}^n \bigcup_{k=1}^{m-1} \{(r_j + k, r_j + k + 1)\}, \quad (6)$$

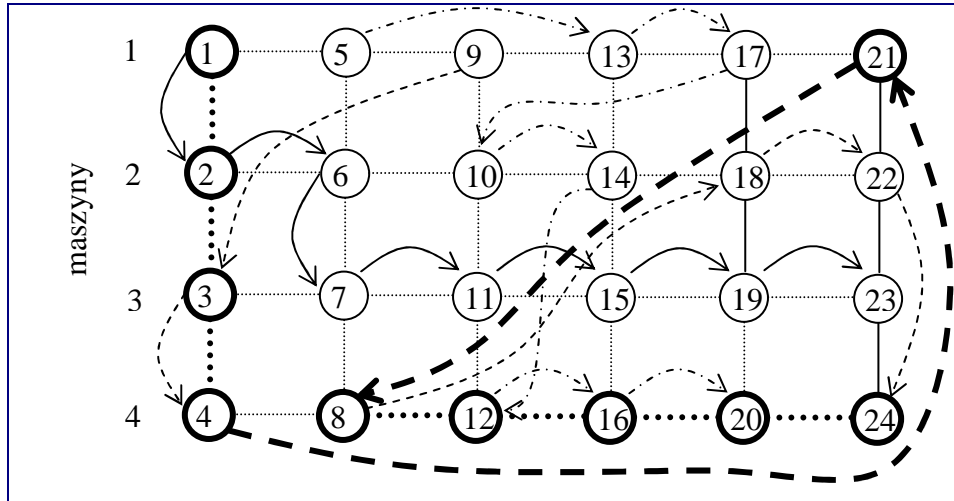
zbiór T zawiera łuki modelujące ograniczenia technologiczne (3),

$$F(\pi) = \bigcup_{k=1}^m \bigcup_{j=1}^{n-1} \{(r_{\sigma(j)} + k, r_{\sigma(j+1)} + k)\}, \quad (7)$$

każdy łuk $(r_{\sigma(j)} + k, r_{\sigma(j+1)} + k) \in F(\sigma)$ modeluje ograniczenia maszynowe (4),

$$E(\pi) = \bigcup_{k=1}^z \bigcup_{j=1}^{n_k-1} \{(\pi_k(j), \pi_k(j+1))\}, \quad (8)$$

każdy łuk $(\pi_k(j), \pi_k(j+1)) \in E(\pi)$ modeluje ograniczenia wynikające z kolejności wykonywania operacji przez operatorów (5).



Rys. 2. Graf $G(\sigma, \pi)$

Łuki ze zbiorów T oraz $F(\sigma)$ są nieobciążone natomiast łuk $(\pi_k(j), \pi_k(j+1)) \in E(\pi)$ obciążony jest wagą $t_{\mu(\pi_k(j)), \mu(\pi_k(j+1))}$.

Z modelu grafowego wynika szereg własności, które znacznie ułatwiają określenie czy kolejność (σ, π) jest dopuszczalna oraz wyznaczenie najwcześniejszego harmonogramu realizacji operacji dla dopuszczalnej kolejności (σ, π) .

Własność 1. Kolejność wykonywania operacji na maszynach oraz przez operatorów określona przez parę (σ, π) jest dopuszczalna wtedy i tylko wtedy gdy graf $G(\sigma, \pi)$ jest acykliczny.

Własność 2. Wyznaczenie najwcześniejszego terminu zakończenia wykonywania operacji $o \in O$ równoważne jest wyznaczeniu długości najdłuższej drogi dochodzącej do węzła reprezentującego operację o w grafie $G(\sigma, \pi)$.

Własność 3. Wyznaczenie najdłuższych dróg do wszystkich węzłów w grafie $G(\sigma, \pi)$ oraz określenie czy graf $G(\sigma, \pi)$ jest acykliczny można zrealizować w czasie $O(nm)$.

Długość najdłuższej drogi (ścieżki krytycznej) w $G(\sigma, \pi)$ jest równa długości najdłuższej drogi do węzła $r_{\sigma(n)} + m$, tj. do węzła reprezentującego ostatnią operację ostatniego w kolejności σ zadania. Wielkość ta określa najwcześniejszy moment zakończenia wykonywania wszystkich zadań. Na Rys. 2 przedstawiono graf $G(\sigma, \pi)$ odpowiadający harmonogramowi wykonywania operacji przedstawionego na Rys. 1. Liniami kropkowanymi zaznaczono łuki technologiczne i maszynowe, natomiast pozostałymi typami linii łuki reprezentujące ograniczenia operatorowe. Pogrubioną linią zaznaczono ścieżkę krytyczną.

4. Procedura wyznaczania dopuszczalnej kolejności wykonywania operacji

Liczba wszystkich możliwych rozwiązań problemu, nawet dla problemów o stosunkowo niewielkiej liczbie zadań jest astronomicznie duża, niestety tylko nieliczna frakcja tych rozwiązań jest dopuszczalna. Na przykład dla jednej maszyny, jednego operatora oraz 10 zadań ($m=1, z=1, n=10$) mamy $(10!)^2$ możliwych rozwiązań z czego 10!

jest dopuszczalnych, natomiast dla ustalonej kolejności wykonywania na maszynie mamy $10!$ wszystkich możliwych kolejności wykonywania operacji przez operatora, z których tylko jedna jest dopuszczalna (identyczna z kolejnością wykonywania na maszynie).

Do wyznaczenia dopuszczalnej kolejności oraz harmonogramu wykonywania operacji można zastosować procedurę opartą na metodzie zachłannej przedstawioną na Rys 3. Parametrami procedury oprócz danych problemu jest kolejność wykonywania zadań na maszynach σ , natomiast kolejność wykonywania operacji przez operatorów wyznaczana jest podczas działania tej procedury. W każdej iteracji procedury, przyporządkowywany jest do jednej odpowiednio wybranej operacji operator oraz ustalany jest moment rozpoczęcia oraz zakończenia jej wykonywania. Aby uwzględnić ograniczenia technologiczne oraz maszynowe w danej iteracji można szeregować tylko te operacje, których poprzedniki technologiczne oraz maszynowe zostały już wcześniej uszeregowane. Spośród wszystkich operacji, które można uszeregować w danej iteracji (gotowych do wykonania) wybierana jest operacja, której wykonanie można zakończyć najwcześniej biorąc pod uwagę obsługę przez każdego operatora.

Niech U oznacza zbiór operacji jeszcze nieuszeregowanych,
 $R(U)$ zbiór operacji gotowych do uszeregowania,
 Z_k -moment zakończenia obsługi przez operatora $k \in M$.

- 1 **podstaw** $U=O, Z_k=0, k \in M, n_i=0, i \in Z$.
- 2 **dopóki** $U \neq \emptyset$ **wykonaj**
- 3 **wyznacz** $o = \arg \min_{s \in R(U)} \min_{i \in Z} \max(C'_s, Z_i + t_{\mu(\pi_i(n_i)), \mu(s)} + p_s)$,
gdzie $C'_s = \max(C_{s'}, C_{s\sigma}) + p_s$.
- 4 **wyznacz** $k = \arg \min_{i \in Z} \max(C'_o, Z_i + t_{\mu(\pi_i(n_i)), \mu(o)} + p_o)$,
- 5 **oblicz** $C_o = \max(C'_o, Z_k + t_{\mu(\pi_k(n_k)), \mu(o)} + p_o)$, $S_o = C_o - p_o$,
- 6 **podstaw** $n_k = n_k + 1, \pi(n_k) = o$,
- 7 **podstaw** $Z_k = C_o$,
- 8 **podstaw** $U = U / \{o\}$,

Rys. 3. Procedura wyznaczenia dopuszczalnej kolejności wykonywania operacji

4. Eksperyment komputerowy

Celem badań eksperymentalnych była ocena wpływu ograniczonej liczby operatorów na zwiększenie długości optymalnych harmonogramów wykonywania zadań przez operatorów na maszynach. Z powodu NP-trudnego charakteru problemu rozpatrywanego w pracy, zrezygnowano z konstruowania algorytmu dokładnego. Harmonogramy generowane były przez algorytm oparty na metodzie symulowanego wyżarzania [8]. Liczne badania eksperymentalne wskazują, że algorytmy oparte na tej metodzie należą do najskuteczniejszych dla wielu problemów szeregowania zadań. Parametry algorytmu SA zostały wyznaczone automatycznie metodą opisaną między innymi w pracy [9]. Zastosowano otoczenie typu wstaw, algorytm wykonywał 1000 iteracji przy czym w każdej temperaturze dokonywane było $n^2/4$ próbkowań.

Eksperyment komputerowy został przeprowadzony na dwóch grupach danych testowych. Pierwszą grupę instancji stanowił zestaw składający się z pierwszych 3-ch grupach instancji problemu przepływowego $n \times m \in \{20 \times 5, 20 \times 10, 20 \times 20\}$, które zostały zaproponowane przez Taillarda [10]. Każda z takich grup składa się 10 instancji o identycznej liczbie maszyn i zadań. Instancje Taillarda zostały wyselekcjonowane spośród wielu instancji wygenerowanych losowo jako szczególnie trudne. Cechują się one bardzo dużym zróżnicowaniem czasów wykonania operacji.

Druga grupa instancji powstała na podstawie czasów wykonania operacji poprzez skrócenie czasów wykonania na maszynach nieparzystych o połowę. W konsekwencji czego otrzymano maszyny ewidentnie niedociążone. Przyjęto zerowe czasy prześię operatorów pomiędzy maszynami.

Dla każdej instancji i dla każdej liczby operatorów $z = m, m-1, m-2, \dots, m-h$, gdzie $h = \min(m-2, 5)$, algorytm SA został uruchomiony 3-krotnie. W przypadku $z = m$, każdego operatora przyporządkowano do innej maszyny. Algorytm SA został zaimplementowany w środowisku Borland Builder C++ oraz uruchomiony komputerze z procesorem Pentium 1.5 GHz.

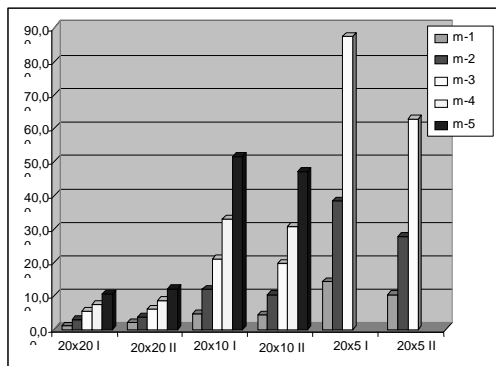
Na podstawie zestawu harmonogramów wyznaczonych dla $z = m$, dla każdej instancji wyznaczono rozwiązanie referencyjne $(\sigma^{ref}, \pi^{ref})$ tj. rozwiązanie o najkrótszym harmonogramie spośród wygenerowanych podczas trzech przebiegów algorytmu SA. Dla każdego rozwiązania (σ, π) wygenerowanego dla każdej instancji i każdej liczby operatorów wyznaczono względny przyrost długości harmonogramu zdefiniowany następująco:

$$L(\sigma, \pi) = 100\% \frac{C_{\max}(\sigma, \pi) - C_{\max}(\sigma^{ref}, \pi^{ref})}{C_{\max}(\sigma^{ref}, \pi^{ref})}, \quad (9)$$

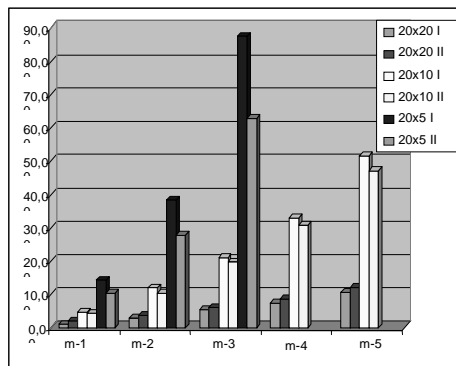
Następnie dla każdej trójki rozwiązań wygenerowanych dla każdej instancji i każdej liczby operatorów wyznaczono wartość minimalną i średnią względnego wydłużenia. W Tabelicy 1 przedstawiono wartości w/w parametrów uśrednione względem grup instancji. Wyniki eksperymentu komputerowego zostały zilustrowane również na rysunkach. Na Rysunku 4 przedstawiono względne wydłużenie harmonogramu w zależności od liczby maszyn i operatorów, natomiast Rysunek 5 umożliwia łatwe porównanie względnego wydłużenia w grupach instancji.

Tab. 1. Rezultaty eksperymentu komputerowego

$N \times m$	Grupa	I					II				
		z	$m-1$	$m-2$	$m-3$	$m-4$	$m-5$	$m-1$	$m-2$	$m-3$	$m-4$
20×5	min	14,1	38,3	87,4			10,2	27,6	62,6		
	śred.	14,3	38,8	87,7			10,5	28,8	63,2		
20×10	min	4,7	11,9	21,1	32,8	51,5	4,2	10,2	19,7	30,7	47,1
	śred.	5,2	12,1	21,5	33,3	51,7	4,6	10,6	20,2	31,4	47,6
20×20	min	1,1	2,7	5,3	7,5	10,6	1,9	3,7	6,1	8,8	12,0
	śred.	1,2	2,9	5,4	7,6	10,6	2,2	4,1	6,6	9,0	12,3



Rys. 4. Wydłużenie harmonogramu w zależności od liczby maszyn i operatorów



Rys. 5. Porównanie wyników dla grup instancji

5. Wnioski

Z analizy rezultatów eksperymentu komputerowego zestawionych w Tab 1. oraz zilustrowanych na Rys. 4 i Rys. 5 wynika, że wraz ze zmniejszaniem się liczby operatorów harmonogram wykonywania zadań wydłuża się, przy czym względny jego przyrost znacznie się zmniejsza wraz ze wzrostem liczby maszyn. W przypadku instancji z liczbą maszyn $m=5$ zmniejszenie liczby operatorów do 4 powoduje, aż 10% wydłużenie harmonogramu. W przypadku systemu z 20 maszynami zmniejszenie liczby operatorów do 19 powoduje tylko 1% wydłużenie harmonogramu, 10% wydłużenie harmonogramu następuje dopiero po zredukowaniu liczby operatorów do 15.

Z wykresu przedstawionego na Rysunku 5 wynika, że tylko dla instancji z grupy $m=10$ można zaobserwować mniejsze wydłużenie harmonogramu dla wszystkich liczb operatorów w przypadku instancji z drugiej grupy tj. instancji o zredukowanym czasie wykonania na maszynach nieparzystych. Świadczy to o zdolności algorytmu do identyfikacji niedociążonych maszyn oraz umiejętnym przydzielaniem operacji do operatorów. Czasy działania algorytmu SA dla danych o największym rozmiarze tj. $n=20$, $m=20$ nie przekraczały 20 sekund.

6. Podsumowanie

W pracy przedstawiono nowe modele obliczeniowe dla permutacyjnego problemu przepływowego z ograniczoną liczbą operatorów maszyn. Zaproponowano procedurę generującą dopuszczalną kolejność wykonywania operacji przez operatorów na maszynach. Z rezultatów przeprowadzonego eksperymentu komputerowego wynika, że w przypadku systemów z dużą liczbą maszyn liczbę operatorów można dość znacznie zredukować nieznacznie wydłużając czas realizacji zadań. Krótki czas działania algorytmu umożliwia efektywne jego wykorzystanie w informatycznych systemach wspomagających planowanie wytwarzania w systemach produkcyjnych.

Literatura

1. Gupta J.N.D., Stafford S.: Flowshop scheduling research after five decades. European Journal of Operational Research, 169, 2006, 699–711.

2. Quan-Ke P.; Tasgetiren F.; Yun-Chia L.: A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers and Operations Research*, 2008, 2807-2839.
3. Grabowski J., Pempera J.: Some local search algorithms for no-wait flow-shop problem with makespan criterion. *Computers & Operations Research*. 32, 8, 2005, 2197-2212.
4. Ronconi D.; Henriques L.: Some heuristic algorithms for total tardiness minimization in a flowshop with blocking. *Omega*, 2009, 272-281.
5. Grabowski J., Pempera J.: The permutation flow shop problem with blocking : A tabu search approach. *Omega (Oxford)*, 35, 3, 2007, 302-311.
6. Nowicki E.: The permutation flow shop with buffers: a tabu search approach. *European Journal of Operational Research*. 116/1, 1999, 205-219.
7. Nowicki E.: Gniazdowy system produkcyjny z ograniczoną liczbą palet. Modele i własności. *Komputerowo Zintegrowane Zarządzanie*. WNT, 2002. 194-204.
8. Aarts E.H.L., Laarhoven P.J.M.: Simulated annealing: a pedestrian review of theory and some applications. *Pattern Recognition and Applications*. Eds. Devijver P.A. and Kittler J., Springer, 1987.
9. Smutnicki C.: Algorytmy szeregowania. *Akademicka Oficyna Wydawnicza, EXIT, Warszawa 2002*.
10. Taillard E.: Benchmarks for basic scheduling problems, *European Journal of Operational Research*, 64, 1993, 278–285.
11. Nawaz E., Enscore E.E., Ham I.: A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem. *Omega International Journal of Management Science*, 11, 1993, 91-95.

Dr inż. Jarosław PEMPERA
 Instytut Automatyki, Informatyki i Robotyki
 Politechnika Wrocławska
 50-372 Wrocław, ul. Wybrzeże Wyspiańskiego 27
 tel./fax.: (71) 320-28-34
 e-mail: jaroslaw.pempera@pwr.wroc.pl