

Adam Tyński*, Mariusz Makuchowski*

Modelowanie i algorytm rozwiązywania problemu przepływowego z transportem

1. Wstęp

Czas wykonania zadania produkcyjnego przez maszyny stanowi zaledwie kilka procent całkowitego czasu pobytu zadania w systemie produkcyjnym. Pozostały czas zadanie spędza m.in. w buforach maszyn, bądź jest transportowane. Uwzględnienie czasów transportu już na etapie konstrukcji harmonogramu pracy takiego systemu w dość oczywisty sposób znajduje przełożenie na wzrost jego wydajności. Wpływ na wydajność systemu ma również odpowiedni dobór środków transportu. Wśród urządzeń transportujących dużą popularnością cieszą się wózki AGV. Wykorzystanie wózków AGV wiąże się ze wzrostem elastyczności, lepszym wykorzystaniem przestrzeni roboczej i niższymi kosztami pracy systemu w stosunku do systemów wykorzystujących inne środki transportu. Przegląd osiągnięć w zakresie zastosowania wózków AGV do transportu w systemach produkcyjnych znajduje się m.in. w pracy [1].

Wykorzystanie wózków AGV wymaga odpowiedniej organizacji całego systemu produkcyjnego. Jedno z rozwiązań polega na zaopatrzeniu systemu w stację załadowniczą i wyładowniczą, jeden lub więcej wózków AGV, zaś określoną liczbę maszyn organizuje się w układ typu pętla (*single loop*). Aby wyeliminować ryzyko wystąpienia konfliktów transportowych, na system narzuca się jednokierunkowy przepływ zadań, zaś na wózkach wymusza się cykliczny, jednokierunkowy tryb pracy. W poszczególnych cyklach pracy każdego wózka do systemu wprowadzane jest jedno zadanie przez stację załadowniczą, jedno zadanie opuszcza system przez stację wyładowniczą, zaś wszystkie pozostałe zadania są transportowane w kierunku kolejnych maszyn w pętli. Układy tego typu, ze względu na swoją prostotę i jednocześnie duży potencjał, znajdują szerokie zastosowanie w przemyśle i tym samym przykuwają uwagę badaczy [2–4].

Ze względu na narzucony kierunek przepływu zadań oraz tryb pracy wózków, optymalizacja harmonogramu pracy systemu wiąże się z koniecznością rozwiązania permutacyjne-

* Instytut Informatyki Automatyki i Robotyki, Politechnika Wrocławska

go problemu przepływowego. W literaturze opisano wiele metod rozwiązywania wspomnianego problemu. Wśród najlepszych algorytmów „klasycznych” wymieniany jest algorytm TSAB, wykorzystujący technikę poszukiwań z zabronieniami [5]. Algorytmy te w trakcie swojej pracy nie uwzględniają jednak czasów transportu. Wykorzystanie harmonogramów „klasycznych” spowoduje stosunkowo nieefektywną pracę systemów produkcyjnych, w których czasy transportu odgrywają istotną rolę w całości procesu produkcyjnego. Jedną z szans na poprawę jakości harmonogramów pracy takich systemów upatruje się w próbie zaadaptowania algorytmów klasycznych do rozwiązywania permutacyjnego problemu przepływowego z uwzględnieniem transportu, co czyni się w niniejszej pracy.

W pracy rozważany jest elastyczny system produkcyjny typu pętla o strukturze przepływowej permutacyjnej, w którym do transportu międzystanowiskowego wykorzystano identyczne wózki AGV. Problem polega na wyznaczeniu takiej kolejności wykonania zadań przez maszyny, by moment zakończenia wykonywania procesu technologicznego przyjął wartość minimalną. Do rozwiązania przedstawionego w pracy problemu zastosowano odpowiednio zmodyfikowany algorytm rozwiązywania przybliżonego, wykorzystujący technikę poszukiwań z zabronieniami. Algorytm został poddany badaniom numerycznym w celu określenia jakości generowanych przez niego rozwiązań między innymi w zależności od liczby uwzględnionych wózków AGV. Rozważania przedstawione w tej pracy stanowią kontynuację rozważań z prac [6, 7], w których opisywano systemy typu pętla wyposażone w jeden wózek AGV.

2. Model matematyczny problemu

Problem można przedstawić następująco. W rozważanym systemie produkcyjnym znajduje się zbiór maszyn $M = M^P \cup M^t = \{1, 2, \dots, m^P\} \cup \{m^P + 1\}$, gdzie $m^P > 2$ jest liczbą wszystkich maszyn produkcyjnych, zaś $m^t \geq 1$ jest liczbą wózków AGV (maszyn transportowych) znajdujących się w systemie. Każda maszyna o numerze $2, \dots, m^P - 1$ posiada bufor wejściowy i wyjściowy o jednostkowej pojemności, zaś maszyny $1, m^P$ utożsamiają odpowiednio stację załadowniczą i wyładowniczą, wyposażone w bufory o Nielimitowanych pojemnościach. Wszystkie maszyny zorganizowane są w układ typu pętla, tj. rozmieszczone są wzdłuż zamkniętej drogi w ten sposób, że maszynę l , $1 < l \leq m^P$, poprzedza maszyna o numerze $l - 1$. Dodatkowo stacja załadownicza (maszyna 1) znajduje się w bezpośredniej bliskości stacji wyładownczej (maszyny m^P).

Czas przejazdu wózka $v \in M^t$ pomiędzy maszyną l i maszyną $l + 1$, $1 \leq l < m^P$, będzie oznaczany przez $t(l, l + 1) > 0$. Podobnie, przez $t(m^P, 1) > 0$ będzie oznaczany czas przejazdu wózka pomiędzy stacją wyładowniczą i załadowniczą. Zakłada się, że wózki mogą poruszać się tylko w jednym kierunku, tj. odwiedzając kolejno maszyny $1, 2, \dots, m^P - 1, m^P$. W połączeniu ze sposobem rozmieszczenia maszyn łatwo jednoznacznie określić czas przejazdu wózków

$$t(l, k) = \begin{cases} \sum_{i=l}^{k-1} t(i, i+1), & l < k \\ \sum_{i=1}^{m^p-1} t(i, i+1) + t(m^p, 1) - \sum_{i=k}^{l-1} t(i, i+1) & l > k \\ 0, & l = k \end{cases} \quad (1)$$

między dowolną parą maszyn produkcyjnych $l, k \in M^p$.

W systemie należy wykonać $r > 1$ zadań ze zbioru $J = \{1, 2, \dots, r\}$. Każde zadanie $j \in J$ podzielone jest na $m = 2m^p - 1$ operacji, w tym m^p operacji produkcyjnych i $m^p - 1$ operacji transportowych. Operację produkcyjną l zadania $j \in J$, notowaną jako para (l, j) , należy wykonać na maszynie $l \in M^p$ w czasie $p(l, j) > 0$. Zbiór wszystkich n^p operacji produkcyjnych będzie oznaczany symbolem $O^p = \{(l, j) : l \in M^p, j \in J\}$, gdzie $|O^p| = m^p \cdot r = n^p$. Pomiedzy każdą parą operacji produkcyjnych (l, j) , $(l+1, j)$, $1 \leq l < m^p, j \in J$, wykonywana jest operacja transportowa, notowana jako para $(m^p + l, j)$. Operacja ta polega na przetransportowaniu przez wózek $v \in M^t$ palety z elementem wykonywanym w ramach zadania j pomiędzy maszynami $l, l+1 \in M^p$ i wykonywana jest w czasie $p(m^p + l, j) = p(m^p + l) = t(l, l+1)$. Zbiór wszystkich n^t operacji transportowych będzie oznaczany symbolem $O^t = \{(m^p + l, j) : 1 \leq l < m^p, j \in J\}$, gdzie $|O^t| = (m^p - 1) \cdot r = n^t$. W dalszych rozważaniach przydatny będzie również zbiór wszystkich n operacji $O = O^p \cup O^t = N \times J$, gdzie $N = \{1, 2, \dots, m\}$, oraz $|O| = n^p + n^t = m \cdot r = n$.

Przyjęta struktura systemu oraz kierunek przepływu zadań wymuszają jednakową kolejność przejścia zadań przez każde ze stanowisk (system ma strukturę przepływową permutacyjną). Dalej, niech permutacja zbioru zadań $\pi = (\pi(1), \pi(2), \dots, \pi(r))$ oznacza kolejność wykonywania zadań ze zbioru J przez każdą z maszyn produkcyjnych, zaś niech Π będzie zbiorem wszystkich permutacji. Dla permutacji $\pi \in \Pi$ przyjmuje się arbitralną kolejność wykonywania operacji transportowych przez każdy wózek, generowaną na bazie π w sposób opisany poniżej. Nie trudno zauważyć, że wszystkie wózki wykonują w sumie $l_C = m^p + r - 2$ cykli, gdzie cykl i pracy każdego wózka, $1 \leq i \leq l_C$, przebiega następująco. Jeśli $i \leq r$, to wózek czeka przy maszynie 1 na zakończenie się operacji załadowniczej $(1, \pi(i))$, po czym wykonuje operację transportową $(m^p + 1, \pi(i))$ w czasie $p(m^p + 1, \pi(i)) = t(1, 2)$. W przeciwnym przypadku, jeżeli ostatnie z zadań do wykonania zostało już wprowadzone do systemu, tzn. jeśli $i > r$, wózek wykonuje przejazd pusty (przejazd bez załadunku) w kierunku maszyny 2 (w czasie $t(1, 2)$). Następnie, jeśli $1 \leq i - 1 \leq r$, to wózek oczekuje na zakończenie się operacji $(2, \pi(i - 1))$, po czym wykonuje operację transportową $(m^p + 2, \pi(i - 1))$. W przeciwnym razie wózek wykonuje przejazd pusty w kierunku maszyny 3. Proces powtarza się aż do momentu przyjazdu wózka do maszyny m^p . Wykonanie przejazdu pustego pomiędzy stacją wyładowniczą i załadowniczą w czasie $t(m^p, 1)$ kończy cykl.

Arbitralna kolejność wykonania operacji transportowych wiąże się z arbitralnym działaniem każdego wózka $v \in M^t$ do wykonania operacji

$$(m^p + 1 + f_{uw}, \pi(e_u - f_{uw})), \quad u = 0, \dots, u_{\max}, \quad w = 0, \dots, w_{\max}(u) \quad (2)$$

gdzie

$$e_u = u \cdot m^t + v \quad (3)$$

$$f_{uw} = f_{u0} + w \quad (4)$$

$$f_{u0} = \max(0, e_u - r) \quad (5)$$

$$u_{\max} = \{u : e_u \leq lc, e_{u+1} > lc\} \quad (6)$$

$$w_{\max}(u) = \{w : f_{uw} < \min(m^P - 1, e_u), f_{uw+1} \geq \min(m^P - 1, e_u)\} \quad (7)$$

Jak przedstawiono w pracy [6], przyjęcie arbitralnej kolejności realizacji operacji transportowych jest uzasadnione sposobem programowania komputera pokładowego wózków AGV oraz metodą koordynacji pracy systemu produkcyjnego. Co więcej, przyjęcie innego trybu pracy wózków zasadniczo nie prowadzi do poprawy wydajności systemu. Przyjęta kolejność gwarantuje również, że wózki nie będą „wyprzedzać się” w trakcie realizacji harmonogramu, co jest częstym wymogiem w wielu rzeczywistych systemach produkcyjnych.

Uszeregowaniem nazywa się zbiór wartości $S(i) \geq 0$, $i \in O$, gdzie $S(i)$ jest momentem rozpoczęcia wykonania operacji i . Uszeregowanie jest dopuszczalne, gdy spełnia wszystkie ograniczenia technologiczne, typowe dla klasycznego problemu przepływowego (patrz. np. [5–7]). Problem polega na odnalezieniu takiego uszeregowania dopuszczalnego, by moment zakończenia się procesu technologicznego, równy $S(m^P, \pi(r)) + p(m^P, \pi(r))$, przyjmował wartość minimalną. Problem jest silnie NP -trudny.

3. Model permutacyjno-grafowy

Przedstawiony poniżej model permutacyjno-grafowy znajduje swoje zastosowanie głównie w algorytmach rozwiązywania; jest szczególnie przydatny na etapie obliczania wartości funkcji celu. Dana jest permutacja $\pi \in \Pi$. Niech $G(\pi) = (O, E)$ będzie grafem skierowanym, reprezentującym dowolną permutację $\pi \in \Pi$, ze zbiorem wierzchołków O i zbiorem łuków $E = E^T \cup E^K \cup E^V \cup E^S$, gdzie

$$E^T = \bigcup_{j \in J} \bigcup_{l=1}^{m^P-1} \{((l, j), (m^P + l, j)), ((m^P + l, j), (l + 1, j))\} \quad (8)$$

$$E^K = \bigcup_{l \in M^P} \bigcup_{j=1}^{r-1} \{((l, j), (l, j + 1))\} \quad (9)$$

$$E^V = \bigcup_{j=2}^r \bigcup_{l=1}^{m^P-2} \{((m^P + l, j), (m^P + l + 1, j - 1))\} \quad (10)$$

Zbiór E^S stanowi sumę zbiorów $E^S = E^{S1} \cup E^{S2}$ określonych następująco:

$$E^{S1} = \begin{cases} \bigcup_{j=2}^{r-m^t+1} \{((a(j), b(j)), (m^P + 1, j + m^t - 1))\}, & r - m^t + 1 \geq 2 \\ \emptyset, & r - m^t + 1 < 2 \end{cases} \quad (11)$$

$$E^{S2} = \begin{cases} \bigcup_{l=\max(2, m^t+2-r)}^{m^P-1} \{((c(l), d(l)), (m^P + l, r))\}, & m^P + 1 \geq \max(2, m^t + 2 - r) \\ \emptyset, & m^P + 1 < \max(2, m^t + 2 - r) \end{cases} \quad (12)$$

gdzie

$$a(j) = \min(m, j + m^P - 1) \quad (13)$$

$$b(j) = \max(1, j - m^P + 1) \quad (14)$$

$$c(l) = \min(m, l + m^P - 1 + r - m^t) \quad (15)$$

$$d(l) = \max(1, l - m^P + 1 + r - m^t) \quad (16)$$

Każdy wierzchołek $(l, j) \in O$ przyjmuje obciążenie $p(l, \pi(j))$. Łuki ze zbiorów E^T, E^K, E^V nie są obciążone. Reprezentują odpowiednio kolejność wykonania operacji w zadaniu, operacji produkcyjnych przez maszyny i operacji transportowych przez wózki. Każdy łuk $(a, b) \in E^S$, gdzie $a = (l, i)$, $b = (k, j)$, przyjmuje obciążenie $t(l - m^P + 1, k - m^P)$. Łuki te reprezentują przejazdy wózka bez załadunku. Definicja zbioru łuków E gwarantuje acykliczność grafu dla każdej permutacji $\pi \in \Pi$. Symbolem $r^\pi(i)$ będzie oznaczana najdłuższa ścieżka dochodząca do wierzchołka $i \in O$ bez jego obciążenia $p(i)$ w grafie $G(\pi)$. Wartość $r^\pi(i)$ jest równa najwcześniejszemu możliwemu momentowi rozpoczęcia operacji i , tj. $r^\pi(i) = S(i)$. Rozważany w pracy problem sprowadza się zatem do odnalezienia permutacji $\pi \in \Pi$, dla której ścieżka krytyczna w grafie $G(\pi)$, równa $C_{\max}(\pi) = r^\pi(m^P, r) + p(m^P, \pi(r))$, przyjmuje wartość minimalną.

Graf $G(\pi)$ skonstruowany jest w ten sposób, że każdy z wierzchołków posiada co najwyżej dwa bezpośrednie poprzedniki i następniki, dzięki czemu stosunkowo łatwo można wyznaczyć wartości $r^\pi(i)$ oraz, w konsekwencji, długość ścieżki krytycznej. Stosunkowo

proste są też modyfikacje powyższego grafu. Graf $G(\delta)$, $\delta \in \Pi$ będzie się różnił od grafu $G(\pi)$ jedynie obciążeniem wierzchołków reprezentujących niektóre operacje produkcyjne, tj. $(l, \delta(j))$, gdzie $l \in M^P$, $\delta(j) \neq \pi(j)$, $j \in J$. Struktura zbioru łuków E oraz czas wykonania operacji transportowych w przyjętym modelu są niezależne od permutacji.

4. Algorytmy rozwiązywania i badania numeryczne

Pomimo zmienionej definicji, możliwy przebieg ścieżek krytycznych w opisywanym grafie nie zmienia się w stosunku do opisu z pracy [7]. W związku z powyższym własności ścieżki krytycznej opisane w [7] w badanym przypadku również są zachowane. Przyjmując zatem za definicję funkcji kryterialnej długość $C_{\max}(\pi)$ ścieżki krytycznej w grafie $G(\pi)$, do rozwiązywania problemu z transportem można zastosować algorytm konstrukcyjny NEH_{AGV} oraz algorytm poszukiwań z zabronieniami $TSAB_{AGV}$, omawiane również w pracy [7].

Celem przeprowadzonych badań numerycznych było (i) określenie relacji czasowych i jakościowych pomiędzy algorytmem NEH_{AGV} i $TSAB_{AGV}$ oraz (ii) określenie wpływu liczby wózków AGV na wartość funkcji celu. Wspomniane algorytmy nie będą porównywane z algorytmami „klasycznymi”, nieuwzględniającymi czasów transportu, gdyż takie porównanie dla systemu z jednym wózkiem AGV zostało wykonane w pracy [7]. W cytowanej pracy zaproponowano również zestaw 70 instancji testowych podzielonych na 7 grup. Każda instancja ma swoją unikalną nazwę w postaci $TFr/m^P/k$, gdzie r jest liczbą zadań, m^P jest liczbą maszyn produkcyjnych, $r \times m^P = 20 \times 5, 20 \times 10, 50 \times 5, 50 \times 10, 100 \times 5, 100 \times 10, 200 \times 10$, zaś zmienna $k \in \{1, 2, \dots, 10\}$ jest numerem instancji w danej grupie $r \times m^P$. Na potrzeby problemu opisywanego w tej pracy do każdej instancji zostanie dodany dodatkowy parametr $m^t = \{1, 2, 3, 6\}$, określający liczbę wózków w systemie. Ostatecznie otrzymujemy zestaw 280 instancji podzielonych na 28 grup, gdzie każda instancja ma nazwę w postaci $TFr/m^P/m^t/k$.

Wszystkie instancje zostały w sposób przybliżony rozwiązane przez następujące algorytmy NEH_{AGV} i $TSAB_{AGV}$, zaimplementowane w Delphi i uruchamiane na komputerze z procesorem Intel Core 2 Duo (2,66 GHz, 22445 MIPS). Permutacja wygenerowana przez algorytm NEH_{AGV} stanowiła permutację początkową algorytmu $TSAB_{AGV}$. Algorytm $TSAB_{AGV}$ dla każdej instancji został uruchomiony z identycznymi wartościami parametrów jak w pracy [7] ($\epsilon = 0$, $maxt = 8$, $maxret = 1000/500$, maksymalny czas pracy algorytmu, równy 10 minut). Wartości funkcji celu permutacji dostarczonych przez algorytm NEH_{AGV} oraz $TSAB_{AGV}$ będą oznaczane (w zależności od liczby wózków AGV uwzględnionych w instancji) odpowiednimi symbolami C^{NHm^t} i C^{TSM^t} .

Dla każdej instancji $TFr/m^P/m^t/k$, została wyznaczona poprawa względna $FI(NHm^t, TSM^t) = 100\% \cdot (C^{NHm^t} - C^{TSM^t}) / C^{NHm^t}$ wartości C^{NHm^t} przez algorytm $TSAB_{AGV}$. Dla każdej grupy instancji została wyznaczona średnia względna poprawa $I(NHm^t, TSM^t)$. Dodatkowo dla instancji (oraz grup instancji), dla których $m^t > 1$ zostały

wyznaczone odpowiednie wartości $FI(TS1, TSm^t)$, $I(TS1, TSm^t)$. Wszystkie wyznaczone średnie poprawy względne zostały przedstawione w tabeli 1. Dla każdej grupy instancji wyznaczone zostały również średnie czasy pracy algorytmów; dla algorytmu TSAB_{AGV} zostały oznaczone przez $T(TSm^t)$ i przedstawione w tabeli 2.

Tabela 1
Średnie względne poprawy dla poszczególnych algorytmów

r / m^p	$I(NH1, TS1)$ [%]	$I(NH2, TS2)$ [%]	$I(NH3, TS3)$ [%]	$I(NH6, TS6)$ [%]	$I(TS1, TS2)$ [%]	$I(TS1, TS3)$ [%]	$I(TS1, TS6)$ [%]
20 / 5	3,77	2,87	2,85	2,85	2,68	2,72	2,72
20 / 10	4,93	6,00	5,86	2,39	13,09	15,11	15,64
50 / 5	3,47	2,49	1,49	1,26	5,06	5,07	5,07
50 / 10	4,99	5,62	6,10	3,82	22,76	19,75	22,52
100 / 5	1,68	1,28	0,80	0,50	6,40	6,47	6,47
100 / 10	3,16	4,62	4,93	3,65	14,80	20,19	24,59
200 / 10	1,09	2,32	3,17	2,74	14,60	20,72	27,06
Średnia	3,30	4,96	3,60	2,46	11,34	12,86	14,87

Tabela 2
Średnie czasy pracy algorytmów dla poszczególnych grup instancji

r / m^p	$T(TS1)$ [s]	$T(TS2)$ [s]	$T(TS3)$ [s]	$T(TS6)$ [s]
20 / 5	5,20	5,12	5,19	4,78
20 / 10	10,68	11,46	15,31	13,83
50 / 5	116,38	97,33	85,72	84,72
50 / 10	287,94	324,30	321,05	300,70
100 / 5	600,00	600,00	589,69	563,02
100 / 10	600,00	600,00	600,00	600,00
200 / 10	600,00	600,00	600,00	600,00
Średnia	317,17	319,99	316,71	309,52

Z tabeli 1 wynika, że uwzględnienie drugiego wózka AGV w trakcie pracy algorytmu TSAB_{AGV} (poprawy $I(TS1, TS2)$) powoduje w niektórych przypadkach nawet ponad dwudziestoprocentowy wzrost jakości generowanych rozwiązań. Dodanie każdego kolejnego wózka powoduje już znacznie mniejszy wzrost jakości rozwiązań co najwyżej o kilka

punktów procentowych. Dla instancji, dla których liczba maszyn jest niewielka ($m^p = 5$), dodanie trzeciego i kolejnych wózków praktycznie nie wpływa na jakość generowanych rozwiązań. Najprawdopodobniej dla tych instancji algorytm generuje rozwiązania bliskie optimum już dla liczby wózków, wynoszącej $m^t = 2$. Największy wzrost jakości rozwiązań w zależności od liczby wózków obserwowalny jest dla instancji charakteryzujących się dużą liczbą zadań i maszyn, tj. dla $r \geq 100$, $m^p = 10$. Podobne obserwacje można poczynić, porównując rozwiązania generowane przez algorytm $TSAB_{AGV}$ w stosunku do NEH_{AGV} . Pewnym wyjątkiem są instancje z grup $TFr/5/m^t$, dla których obserwowany jest spadek przewagi algorytmu popraw nad algorytmem konstrukcyjnym wraz ze wzrostem liczby wózków.

Czas pracy algorytmów NEH_{AGV} , $TSAB_{AGV}$ nie zależy w istotnym stopniu od liczby wózków AGV. Wraz ze wzrostem liczby wózków obserwowalny jest nieznaczny spadek czasu pracy algorytmu dla grup instancji $TFr/5/m^t$ oraz nieznaczny wzrost czasu pracy dla instancji z grup $TFr/10/m^t$. Czasy pracy algorytmu NEH_{AGV} nie przekroczyły 3 sekund i nie zostały ujęte w tabeli 2.

5. Podsumowanie

W pracy jest rozważany elastyczny system produkcyjny typu „pętla”, w którym transporty realizowane są przez zbiór identycznych wózków AGV. W pracy został przedstawiony model matematyczny, reprezentacja permutacyjno-grafowa opisywanego systemu oraz został sformułowany problem szeregowania zadań. Następnie, dla problemu został zaproponowany algorytm konstrukcyjny NEH_{AGV} i algorytm lokalnych poszukiwań $TSAB_{AGV}$ oraz instancje testowe. Pod koniec pracy zostały zaprezentowane wyniki badań numerycznych, których celem było określenie relacji jakościowych pomiędzy algorytmami jak również wpływ liczby wózków AGV na jakość generowanych przez nie rozwiązań.

Po przeprowadzonych badaniach okazało się, że dla instancji uwzględniających już dwa wózki AGV był obserwowalny znaczny wzrost jakości generowanych przez algorytmy rozwiązań w stosunku do rozwiązań instancji z jednym wózkiem. Dla instancji z większą liczbą wózków wzrost jakości rozwiązań nie był już tak istotny jak w przypadku instancji z dwoma wózkami. Wzrost ten wahał się od zera do kilku punktów procentowych. Nie zaobserwowano również istotnego wpływu liczby wózków na czas pracy badanych algorytmów.

Literatura

- [1] Ganesharajah T., Hall N., Sriskandarajah C., *Design and operational issues in AGV-served manufacturing systems*. Annals of Operations Research, 76, 1998, 109.
- [2] Hall N., Sriskandarajah C., Ganesharajah T., *Operational decisions in AGV-served flowshop loops: scheduling*. Annals of Operations Research, 107, 2001, 161.
- [3] Bozer Y.A., Srinivasan M.M., *Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops*. IIE Transactions, 23, 1991, 72.

- [4] Bozer Y.A., Srinivasan M.M., *Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems*. European Journal of Operational Research, 63, 1992, 173.
- [5] Nowicki E., Smutnicki C., *A fast tabu search algorithm for the permutation flow-shop problem*. European Journal of Operational Research, 9, 1996, 160.
- [6] Smutnicki C., Tyński A., *Modelowanie przepływu zadań w elastycznym systemie produkcyjnym z wózkami AGV*. Automatyka, 9, 2005, 223.
- [7] Tyński A., *Zastosowanie techniki poszukiwań z zabronieniami w rozwiązywaniu problemu przepływowego z transportem*. Komputerowo Zintegrowane Zarządzanie, 2, 2006, 595.