

Marcin Molga*, Czesław Smutnicki*

Modelowanie, analiza i optymalizacja pracy systemów kompletowania**

1. Wprowadzenie

Systemy kompletowania elementów wchodzą w skład większości współczesnych systemów montażowych, występujących obecnie dość często w systemach wytwarzania. Celem systemu kompletowania jest napełnienie zbioru pojemników zadaną z góry mieszanką elementów różnych typów (różną dla różnych pojemników). Proces napełniania jest oceniany przez dedykowaną funkcję kryterialną. Nowość problematyki kompletowania, złożoność modeli matematycznych oraz kombinatoryczny charakter odpowiednich problemów optymalizacyjnych, powodują kłopoty z uzyskaniem rozwiązania satysfakcjonującej jakości. Podejmowane do tej pory próby modelowania i optymalizacji systemów kompletowania obejmowały takie metodologie, jak sieci kolejkowe [3], teoria sterowania [4–5], podejście wieloagentowe [6–8] czy sieci Petriego [9]. W tej pracy podjęto próbę stworzenia, na bazie klasycznej teorii szeregowania, modelu dla „półautomatycznego” systemu kompletowania z geometrią liniową, obsługiwanego przez jednego realizatora, ocenianego przez kryterium czasowe. Celem badań jest analiza przydatności zaproponowanej metodologii oraz analiza możliwości optymalizacji wydajności rzeczywistych systemów kompletowania. Kolejne części pracy zawierają formalną definicję problemu, prezentację łatwego w implementacji i analizie grafowego modelu systemu, szybkie i efektywne algorytmy optymalizujące sekwencję przetwarzania (obsługi) pojemników ze względu na kryterium czasu, wyniki badań numerycznych oraz wypływające z nich wnioski.

2. Opis problemu

Rozpocniemy od opisu werbalnego. Celem jest napełnienie zbioru pojemników (transportowanych taśmociągami) elementami różnych typów, przechowywanych w magazynach (stoiskach, regałach) umieszczonych wzdłuż prostoliniowego odcinka taśmociągu. Odległości pomiędzy magazynami mogą (ale nie muszą) być identyczne. Pojemnik jest

* Instytut Informatyki Automatyki i Robotyki Politechniki Wrocławskiej

** Praca była finansowana ze środków KBN w latach 2003–2006, jako projekt badawczy nr T11A01624

automatycznie zatrzymywany przy każdym magazynie oraz automatycznie identyfikowany za pomocą skanerów laserowych na podstawie kodów kreskowych na nim umieszczonych. Lista elementów, które mają być umieszczone w danym pojemniku, prezentowana jest na monitorze ciekłokrystalicznym. Umieszczenie elementu w pojemniku jest realizowane przez pracownika (realizatora) i wymaga obecności realizatora przez cały czas obsługi zadania przy danym stoisku (napełniania pojemnika przy danym magazynie). Nieruchomy pojemnik oczekujący na napełnienie blokuje pojemniki poruszające się za nim. Realizator krąży w systemie, dokonując obsługi kolejnych pojemników; wybór strategii jego przemieszczania warunkuje odpowiednio szybkie wykonanie wszystkich zleceń (napełnienie pojemników). Formalny opis problemu jest następujący.

W systemie kompletowania o geometrii liniowej należy umieścić w n pojemnikach ze zbioru $\{1, 2, \dots, n\}$ produkty znajdujące się w m stoiskach (magazynach) ze zbioru $1, 2, \dots, m$. Pojemniki przewożone są przy użyciu systemu taśmociągów, który przemieszcza wszystkie pojemniki w identycznej kolejności, tj. najpierw do pierwszego stoiska, później do drugiego itd. W systemie znajduje się również stanowisko wyładownicze (stanowisko 0), w którym zawartość pojemników jest wyładowywana. Również w tym stanowisku następuje wprowadzanie pojemników do systemu. W danej chwili przy danym stanowisku (w jego przestrzeni roboczej) może znajdować tylko jeden pojemnik oraz dany pojemnik nie może być wypełniany produktami pochodzącymi z dwóch różnych stoisk. Włożenie pojemnika na taśmociąg, transport oraz umieszczenie w przestrzeni roboczej następnego stanowiska odbywa się w sposób całkowicie zautomatyzowany, przy czym czynności te są wykonywane niezwłocznie jedna po drugiej. Oznacza to brak możliwości umieszczania pojemników poza przestrzeniami roboczymi stoisk oraz na możliwość umieszczania ich na taśmociągu tylko podczas transportu. Nie dopuszcza się wymijania pojemników zarówno podczas transportu na taśmociągu, jak i w obrębie przestrzeni roboczej stoisk. Czas transportu pomiędzy stanowiskami k oraz $k+1$ nie zależy od pojemnika (jego ciężaru i/lub rozmiarów) i wynosi t_k . Bez zmniejszania ogólności rozważań przyjmiemy, że czas $t_0 = 0$ oraz $t_m = 0$.

Elementy umieszczane są przez realizatora zgodnie z określoną specyfikacją dotyczącą danego pojemnika i danego stoiska. Para (j, k) określa czynność załadunkową, która polega na umieszczeniu odpowiedniego zestawu elementów ze stanowiska k w pojemniku j . Czas obsługi (napełnienia elementami) pojemnika j przy stanowisku k wynosi $p_{kj} \geq 0$. Jeżeli specyfikacja danego pojemnika nie zawiera elementów z danego stoiska, wówczas pojemnik taki nie jest obsługiwany przez realizatora, co wcale nie oznacza, że pojemnik nie zostanie obsłużony w tym stanowisku. Zostaną wykonane opisane wyżej czynności realizowane przez system transportowy, tj. umieszczenie i późniejsze pobranie pojemnika. Ze względu jednak na pomijalnie mały czas wykonywania tych czynności przyjmujemy, że wynosi on zero. Realizator po umieszczeniu wszystkich elementów w danym pojemniku z danego stanowiska przesuwa się i/lub pozostaje przy min w celu zrealizowania innej czynności załadunkowej. Czas przesunięcia realizatora od stanowiska i do stanowiska k wynosi $r_{ik} \geq 0$.

Dla pojemnika j oraz stoiska k , przez S'_{kj} oznaczmy moment umieszczenia pojemnika w przestrzeni roboczej stoiska, przez S_{kj} i C_{kj} oznaczmy odpowiednio, moment rozpoczęcia i moment zakończenia obsługi tego pojemnika, natomiast przez C'_{kj} oznaczmy moment zwolnienia przez pojemnik przestrzeni roboczej stoiska. Niech $\pi = (\pi(1), \dots, \pi(n))$

będzie permutacją elementów ze zbioru $\{1, \dots, n\}$ określającą sekwencję wprowadzania pojemników do systemu, natomiast niech $\sigma = (\sigma(1), \dots, \sigma(l))$, będzie permutacją par ze zbioru $\Omega = \{(k, j): p_{kj} > 0, k = 1, \dots, m, j = 1, \dots, n\}$ określającą kolejność wykonywania czynności załadunkowych przez realizatora. Wówczas dla każdego pojemnika i dla każdego stoiska ww. zestaw wielkości musi spełniać następujące ograniczenia:

$$S'_{kj} \leq S_{kj}, \quad C_{kj} = S_{kj} + p_{kj}, \quad C_{kj} \leq C'_{kj} \quad \text{dla } k = 1, \dots, m, j = 1, \dots, n \quad (1)$$

$$C'_{kj} + t_k = S'_{k+1, j}, \quad \text{dla } k = 1, \dots, m-1, j = 1, \dots, n \quad (2)$$

$$C'_{k, \pi(j)} \leq S'_{k, \pi(j+1)}, \quad \text{dla } k = 1, \dots, m, j = 1, \dots, n-1 \quad (3)$$

$$C_{\sigma(j)} + R_{\sigma(j), \sigma(j+1)} \leq S_{\sigma(j+1)}, \quad \text{dla } j = 1, \dots, l-1 \quad (4)$$

gdzie $R_{(i, j), (k, l)} = r_{i, k}$.

Znaczenie ograniczenia (1) jest oczywiste. Ograniczenie (2), modeluje zachowanie automatycznego systemu transportowego i oznacza, że jeżeli pojemnik j opuści stanowisko k w momencie C'_{kj} , to pojawi się w przestrzeni roboczej następnego stanowiska dokładnie po t_k jednostkach czasu. Z ograniczenia (3) wynika, że w przestrzeni roboczej stanowiska k może znaleźć się następny po $\pi(j)$ w kolejności określonej przez π pojemnik dopiero, gdy ten ją opuści. Ograniczenie (4) wynika z ograniczonych czasowo i/lub przestrzennie możliwości realizatora i oznacza, że rozpoczęcie załadunku pojemnika b przy stanowisku a , gdzie $\sigma(j+1) = (a, b)$, może rozpocząć się dopiero po zakończeniu realizacji czynności załadunkowej opisanej przez parę $\sigma(j) = (c, d)$ i przemieszczeniu realizatora ze stanowiska a do c .

Nawet pobieżna analiza nierówności (1)–(4) prowadzi do następujących pytań: „Czy dowolna permutacja określająca sekwencję wprowadzania zadań do systemu jest dopuszczalna?” i „Czy dowolna permutacja czynności załadunkowych jest też dopuszczalna?” Odpowiedź na pierwsze pytanie jest twierdząca, natomiast na drugie – przecząca. W dalszej części pracy pokażemy sposób, w jaki można generować permutacje dopuszczalne, tj. takie, że spełnione są dla nich ograniczenia (1)–(4).

Zauważmy, że realizacja wszystkich czynności wykonywanych przez realizator zakończy się w momencie zakończenia realizacji ostatniej czynności w sekwencji σ . Z oczywistych względów odpowiada on załadowaniu ostatniego w kolejności π pojemnika na stoisku m . Oznaczmy przez $C_{\max}(\pi, \sigma)$ ten moment.

Chcemy znaleźć permutację $\pi^* \in \Pi$ oraz $\sigma^* \in \Omega^D$, taką że

$$C_{\max}(\pi^*, \sigma^*) = \min_{\pi^* \in \Pi, \sigma^* \in \Omega^D} C_{\max}(\pi, \sigma) \quad (5)$$

gdzie:

Π – zbiór wszystkich permutacji elementów zbioru $\{1, \dots, n\}$,

Ω^D – zbiór wszystkich dopuszczalnych kolejności wykonywania czynności przez realizator, zbiór Ω^D , zależy od π .

3. Model grafowy

Rozważany w pracy problem (dla ustalonych π, σ) możemy zamodelować również za pomocą grafu skierowanego $G(\pi, \sigma)$. W tym celu dokonamy przekształceń ograniczeń (1)–(4). Korzystając z oczywistego faktu, że $C'_{mj} = C_{mj}$, $j = 1, \dots, n$ oraz z równości $C_{kj} = S_{kj} + p_{kj}$ (1) i $C'_{kj} + t_k = S'_{k+1,j}$ (2) możemy wyeliminować zmienne C_{kj} i C'_{kj} oraz równość (2). W konsekwencji otrzymujemy równoważny zestaw nierówności:

$$S'_{kj} \leq S_{kj}, \text{ dla } k = 1, \dots, m, j = 1, \dots, n \quad (6)$$

$$S_{kj} + p_{kj} + t_k \leq S'_{k+1,j}, \text{ dla } k = 1, \dots, m-1, j = 1, \dots, n \quad (7)$$

$$S'_{k+1,\pi(j)} - t_k \leq S'_{k,\pi(j+1)}, \text{ dla } k = 1, \dots, m-1, j = 1, \dots, n-1 \quad (8)$$

$$S_{m,\pi(j)} + p_{m,\pi(j)} \leq S'_{m,\pi(j+1)}, \text{ dla } j = 1, \dots, n-1 \quad (9)$$

$$S_{\sigma(j)} + p_{\sigma(j)} + R_{\sigma(j),\sigma(j+1)} \leq S_{\sigma(j+1)}, \text{ dla } j = 1, \dots, l-1 \quad (10)$$

Budujemy graf siatkowy postaci $G(\pi, \sigma) = (W \cup W^*, R \cup T \cup F(\pi) \cup F_m(\pi) \cup E(\sigma))$ ze zbiorem węzłów $W \cup W^*$ i zbiorem łuków $R \cup T \cup F(\pi) \cup F_m(\pi) \cup E(\sigma)$. Odpowiednie zbiory definiujemy następująco

$$W = \{(k, j) : k = 1, \dots, m, j = 1, \dots, n\} \quad (11)$$

Węzeł $(k, j) \in W$ kojarzymy ze zdarzeniem $S_{k,\pi(j)}$. Waga węzła $(k, j) \in W$ wynosi $p_{k,\pi(j)}$.

$$W^* = \{(k, j)^* : k = 1, \dots, m, j = 1, \dots, n\} \quad (12)$$

Węzeł $(k, j)^* \in W^*$ kojarzymy ze zdarzeniem $S'_{k,\pi(j)}$. Waga węzła $(k, j) \in W^*$ wynosi zero. Dla uproszczenia oznaczeń przyjmujemy, że węzły odpowiadające zdarzeniom $S_{k,\pi(j)}$ oraz $S'_{k,\pi(j)}$ posiadają taką samą numerację (k, j) , lecz są rozróżniane symbolem $()^*$

$$R = \bigcup_{k=1}^m \bigcup_{j=1}^n \{((k, j)^*, (k, j))\} \quad (13)$$

Łuki ze zbioru R modelują ograniczenia (6) i obciążone są wagą zero

$$T = \bigcup_{k=1}^{m-1} \bigcup_{j=1}^n \{((k, j), (k+1, j)^*)\} \quad (14)$$

Łuki ze zbioru T modelują ograniczenia (7) i obciążone są wagą równą t_k

$$F(\pi) = \bigcup_{k=1}^{m-1} \bigcup_{j=1}^{n-1} \{((k+1, \pi(j))^*, (k, \pi(j+1))^*)\} \quad (15)$$

Łuki ze zbioru $F(\pi)$ modelują ograniczenia (8) i obciążone są wagą równą $-t_k$

$$F_m(\pi) = \bigcup_{j=1}^{n-1} \{((m, \pi(j)), (m, \pi(j+1)))^*\} \quad (16)$$

Łuki ze zbioru $F_m(\pi)$ modelują ograniczenia (9) i obciążone są wagą równą zero

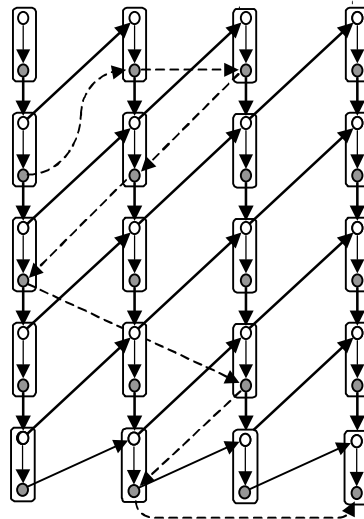
$$E(\sigma) = \bigcup_{j=1}^{l-1} \{(\sigma(j), \sigma(j+1))\} \quad (17)$$

Łuki ze zbioru $E(\sigma)$ modelują ograniczenia (10) i obciążone są wagą równą $R_{\sigma(j), \sigma(j+1)}$.

Przykładowy graf dla danych z tabeli 1, $\pi = (1, 2, 3, 4)$ oraz pewnej σ pokazano na rysunku 1.

Tabela 1
Macierz definiująca przykładową instancję problemu

Stoiska	Zadania			
	1	2	3	4
1	0	1	3	0
2	2	5	0	0
3	3	0	0	0
4	0	0	2	0
5	0	2	0	2



Rys. 1. Graf G dla danych z tabeli 1 i naturalnej permutacji wejściowej pojemników do systemu

Każdemu zadaniu odpowiada jedna kolumna grafu, każda czynność reprezentowana jest przez dwa węzły i łuk (cienki), każde dwa wiersze grafu odpowiadają czynnościom wykonywanym w tym samym stanowisku, regularne łuki ukośne wynikają z ograniczeń (8) i (9), nieregularne łuki ukośne wynikają z σ (ograniczenie (10)).

Para permutacji (π, σ) wyznacza rozwiązanie dopuszczalne problemu, jeśli $G(\pi, \sigma)$ jest acykliczny. Generowanie dopuszczalnych (π, σ) można zrealizować hierarchicznie. Niech π będzie dowolne ustalone (wszystkie π są dopuszczalne do tego działania). Łuki $E(\sigma)$ zależą od permutacji σ , która określa kolejność wykonywania czynności przez pracownika. Obsługuje on wyłącznie czynności z $p_{ij} > 0$ w pewnej kolejności $\sigma \in \Omega$, która ma uwzględniać ograniczenia wynikające z następstw zdarzeń określonych grafem $G(\pi) = (W \cup W^*, R \cup T \cup F(\pi) \cup F_m(\pi))$. Zatem σ można otrzymać na bazie pewnego porządku topologicznego wierzchołków w $G(\pi)$. Dla jednoznaczności opisu i analizy zakładamy, że porządek ten spełnia dodatkowo warunek: jeśli nie istnieje droga w $G(\pi)$ z $(i + 1, j)$ do (k, l) to: albo $(i + 1, j)$ występuje przed (k, l) albo $(k + 1, l)$ występuje przed (i, j) . Przeciwnie, jeśli istnieje w $G(\pi)$ droga z wierzchołka $(i + 1, j)$ do wierzchołka (k, l) , oraz $p_{ij} > 0, p_{kl} > 0$, to czynność (i, j) musi być obsłużona przez pracownika przed (k, l) bowiem węzły $(i, j), (i + 1, j)$ wystąpią w porządku topologicznym przed węzłami $(k, l), (k + 1, l)$. Odpowiednio, zbiór wszystkich różnych σ możemy generować na bazie wszystkich tak określonych porządków w $G(\pi)$. Zauważmy, że różne porządki topologiczne $G(\pi)$ mogą definiować tę samą trasę przejścia realizatora, bowiem są redundantne w stosunku do σ (zawierają dodatkowo węzły dla których $p_{ij} = 0$).

4. Algorytmy rozwiązywania

Model z rozdziału 3 wykorzystano do budowy klasy priorytetowych algorytmów konstrukcyjnych, wyznaczających σ dla danego π . (Dotyczy to przypadku gdy system kompletowania nie pozwala na wybór sekwencji wejściowej π , ale pozostawia swobodny wybór σ). Generacja odpowiednich porządków topologicznych polega na dodawaniu do częściowego uporządkowania węzłów elementów z kolejki, do której iteracyjnie dodawane są węzły posiadające już zanalizowane poprzedniki. Ponieważ pomiędzy węzłami oczekującymi w kolejce nie istnieje droga, zatem dozwolone jest ich pobieranie według zdefiniowanych przez nas reguł:

- 1) **FIFO** – analizuj węzły w kolejności wprowadzania ich do kolejki;
- 2) **FGi FIFO** – preferuj węzły (i, j) , dla których $p_{ij} = 0$, w kolejności niemalejących wartości i , tj. indeksu stoiska; pozostałe według reguły FIFO;
- 3) **FGj FIFO** – preferuj węzły (i, j) , dla których $p_{ij} = 0$, w kolejności niemalejących wartości j ; pozostałe według reguły FIFO;
- 4) **FGi NN** – preferuj węzły (i, j) , dla których $p_{ij} = 0$, w kolejności niemalejących wartości i ; pozostałe według zasady NN (*Nearest Neighbour*, reguła polega na wyborze pierwszego węzła reprezentującego stoisko znajdujące się w najmniejszej odległości od aktualnej pozycji pracownika);
- 5) **FGi mNNv1** – preferuj węzły (i, j) , dla których $p_{ij} = 0$, w kolejności niemalejących wartości i ; pozostałe według zmodyfikowanej reguły NN (modyfikacja polega na preferowaniu zadań wykonywanych nie tylko na najbliższym stoisku, ale także „bliskich” aktualnie wykonywanemu zadaniu w sensie indeksu zadania);

- 6) **FGi mNNv2** – preferuj węzły (i, j) , dla których $p_{ij} = 0$, w kolejności niemalejących wartości i ; pozostałe według zmodyfikowanej reguły NN (inna modyfikacja polegająca na tym, że do obsługi wybierany jest ostatni w kolejce węzeł reprezentujący stoisko znajdujące się w najmniejszej odległości od aktualnej pozycji).

5. Kryterium optymalności

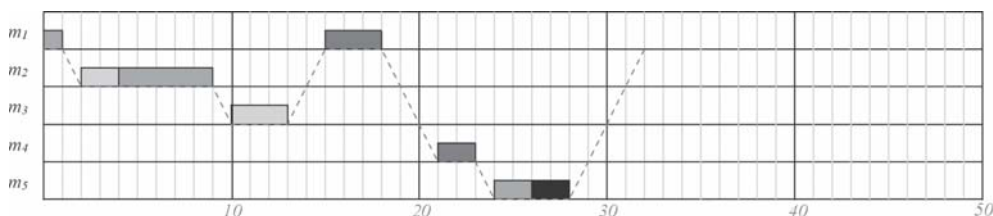
W analizie przyjęto kryterium 5), mimo iż ogólność modelu pozwala na operowaniu na dowolnym innym regularnym kryterium czasowym. Dodatkowo zakłada się, że realizator rozpoczyna i kończy pracę przy pierwszym stoisku ($i = 1$). Oznacza to, że po wykonaniu wszystkich zadań wymagających obsługi musi powrócić do stoiska o indeksie 1. Założenie to można łatwo zrealizować wprowadzając fikcyjne zadania wymagające obsługi wyłącznie przy stoisku 1.

W przypadku wskaźników jakości operujących na drodze pokonanej przez realizatora (patrz [1–2]) prędkości realizatora i taśmociągu są nieistotne, to jednak, gdy stosuje się kryteria czasu, konieczne będzie rozróżnienie następujących przypadków:

- prędkość realizatora większa od prędkości taśmociągu,
- prędkość realizatora równa prędkości taśmociągu,
- prędkość realizatora mniejsza od prędkości taśmociągu.

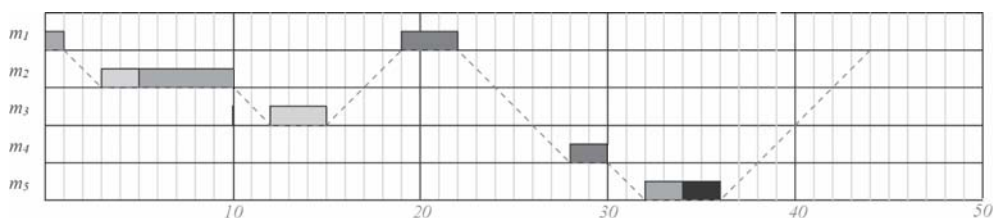
Prędkość realizatora definiujemy jako liczbę jednostkowych odległości pokonanych przez realizator w jednostce czasu, zaś prędkość taśmociągu jako ilość jednostkowych odległości pokonanych przez pojemnik na taśmociągu w jednostce czasu.

Harmonogram obsługi zadań z tabeli 1 wprowadzanych do systemu w naturalnej kolejności, dla różnego stosunku prędkości realizatora do prędkości taśmociągu przedstawiono na rysunkach 2–4. Czynności wchodzące w skład tego samego zadania zaznaczono tym samym kolorem. Linia przerywana oznacza marszrutę realizatora.



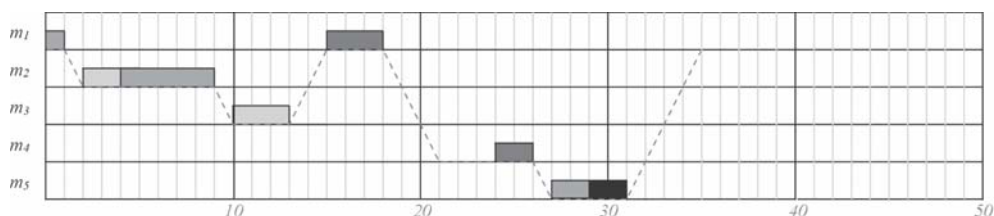
Rys. 2. Wykres Gantta dla sekwencji z tabeli 1 $v_r = 1, v_t = 1$

Objaśnienia w tekście



Rys. 3. Wykres Gantta dla sekwencji z tabeli 1 $v_r = 0,5, v_t = 1$

Objaśnienia w tekście



Rys. 4. Wykres Gantta dla sekwencji z tabeli 1 $v_r = 1$, $v_t = 0,5$
Objaśnienia w tekście

6. Badania testowe algorytmów

Z powodu braku w literaturze przykładów testowych dla analizowanego problemu jakość przedstawionych algorytmów oceniono na drodze eksperymentu komputerowego. Wygenerowano populację losowych przykładów dla liczby magazynów równej 5, 10 i 20 oraz liczby pojemników równej 20, 50 i 100. Dla każdej kombinacji liczby zadań i liczby stoisk wygenerowano trzy grupy po 10 instancji każda. Przykłady należące do danej grupy posiadały ustalone (jednakowe dla każdego stanowiska) prawdopodobieństwo FF wystąpienia obsługi zadania w danym stanowisku – im wyższy był współczynnik, tym bardziej pracochłonne było zadanie (wymagało obsługi przy większej liczbie stoisk). Współczynnik ten dla poszczególnych grup wynosił: 0,05; 0,10 i 0,25. Czas napełnienia pojemnika przy poszczególnych stoiskach wygenerowano z rozkładu jednostajnego na przedziale 1–10. Odległości pomiędzy dwoma stoiskami x , y przyjęto równe $d_{xy} = |x - y|$. Czas przesunięcia realizatora pomiędzy stoiskami x i y wyznaczony został ze wzoru $r_{xy} = d_{xy}/v_r$, gdzie v_r oznacza prędkość realizatora, natomiast czas transportu pojemnika pomiędzy stoiskami k oraz $k + 1$ ze wzoru $t_k = d_{k,k+1}/v_t$, gdzie v_t oznacza prędkość taśmociągu.

Wszystkie instancje zostały zanalizowane dla trzech przypadków:

- 1) identycznej prędkości realizatora i taśmociągu,
- 2) większej prędkości realizatora,
- 3) większej prędkości taśmociągu.

Tabela 2

Szczegółowe wyniki algorytmów. $FF = 0,05$, $v_r = 0,5$, $v_t = 1$

M	N	FIFO	FGi	FGj	FGi	FGi	FGi
			FIFO	FIFO	NN	mNN v1	mNN v2
5	20	0,41	0,41	2,97	1,33	1,74	1,33
	50	4,62	4,34	3,86	0,40	1,63	0,40
	100	3,62	3,57	2,99	0,20	1,18	0,52
10	20	8,99	11,70	15,41	3,62	3,18	2,61
	50	12,35	18,45	20,24	1,54	1,81	0,35
	100	12,11	17,53	18,95	1,08	2,14	1,18
20	20	21,44	29,83	38,10	2,57	1,65	1,90
	50	37,57	46,54	51,06	0,87	2,48	1,05
	100	43,01	53,97	56,25	0,97	1,57	0,80

W tabelach 2–10 zamieszczono średnie procentowe błędy względne rezultatów wygenerowanych przez poszczególne procedury. Aby obliczyć błąd względny, posłużono się rozwiązaniem referencyjnym F_{REF} , za które przyjęto najlepsze z rozwiązań wygenerowanych dla danej instancji przez wszystkie używane algorytmy.

Tabela 3Szczegółowe wyniki algorytmów. $FF = 0,10$, $v_r = 0,5$, $v_t = 1$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	6,28	8,99	10,80	2,81	0,00	1,07
	50	6,58	7,90	9,11	0,18	2,27	0,15
	100	7,18	7,15	8,77	0,37	2,79	0,10
10	20	17,15	26,86	28,15	1,11	3,65	1,91
	50	25,61	32,69	34,01	0,82	2,42	2,31
	100	21,12	28,13	29,02	0,59	2,64	0,27
20	20	40,48	52,78	59,86	2,91	3,56	0,89
	50	59,17	64,87	63,34	0,64	2,65	0,64
	100	63,84	65,99	63,65	0,19	4,70	0,69

Tabela 4Szczegółowe wyniki algorytmów. $FF = 0,25$, $v_r = 0,5$, $v_t = 1$

M	N	FIFO	Fgi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	10,98	10,25	11,35	0,61	2,05	0,75
	50	14,55	14,77	17,22	0,07	3,23	0,45
	100	13,05	14,18	16,73	0,04	2,16	0,66
10	20	36,84	38,32	39,70	0,75	3,99	0,24
	50	39,17	38,33	38,20	0,61	3,71	0,80
	100	40,98	38,84	38,86	0,29	4,25	0,29
20	20	69,89	60,16	59,12	0,91	6,08	0,47
	50	89,19	63,96	57,14	0,67	6,34	0,88
	100	91,52	67,42	57,84	1,04	5,33	0,35

Tabela 5Szczegółowe wyniki algorytmów. $FF = 0,05$, $v_r = 1$, $v_t = 1$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	0,25	0,25	1,80	0,80	1,05	0,80
	50	2,85	2,67	2,37	0,25	0,98	0,25
	100	2,24	2,20	1,85	0,13	0,74	0,32
10	20	5,97	7,81	10,27	2,40	2,18	1,71
	50	8,16	12,14	13,35	1,01	1,16	0,24
	100	7,92	11,46	12,39	0,70	1,40	0,77
20	20	15,20	21,17	27,12	1,81	1,20	1,35
	50	26,22	32,41	35,58	0,60	1,73	0,73
	100	29,92	37,55	39,13	0,68	1,09	0,56

Tabela 6Szczegółowe wyniki algorytmów. $FF = 0,10$, $v_r = 1$, $v_t = 1$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	3,90	5,56	6,63	1,75	0,00	0,71
	50	3,96	4,73	5,46	0,10	1,36	0,09
	100	4,27	4,26	5,23	0,22	1,66	0,06
10	20	11,03	17,24	18,12	0,71	2,32	1,22
	50	16,28	20,79	21,63	0,52	1,53	1,48
	100	13,50	17,94	18,52	0,40	1,69	0,17
20	20	27,05	35,25	39,99	1,97	2,38	0,60
	50	39,58	43,40	42,39	0,42	1,78	0,42
	100	42,61	44,05	42,48	0,13	3,09	0,46

Tabela 7Szczegółowe wyniki algorytmów. $FF = 0,25$, $v_r = 1$, $v_t = 1$

M	N	FIFO	Fgi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	6,53	6,12	6,76	0,36	1,21	0,45
	50	8,51	8,64	10,08	0,04	1,89	0,27
	100	7,64	8,29	9,79	0,02	1,24	0,39
10	20	22,92	23,89	24,77	0,47	2,50	0,15
	50	23,93	23,42	23,34	0,37	2,27	0,49
	100	25,06	23,75	23,77	0,17	2,60	0,18
20	20	44,18	38,08	37,42	0,58	3,85	0,30
	50	56,47	40,50	36,17	0,42	4,01	0,56
	100	57,67	42,48	36,44	0,65	3,36	0,23

Tabela 8Szczegółowe wyniki algorytmów. $FF = 0,05$, $v_r = 1$, $v_t = 0,5$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	0,38	0,38	2,97	1,35	1,72	1,35
	50	2,47	2,31	3,10	0,25	1,01	0,25
	100	3,05	2,70	2,41	0,13	1,10	0,32
10	20	6,47	6,43	15,16	2,95	3,08	2,13
	50	8,51	10,31	15,48	1,20	2,28	0,00
	100	8,31	10,18	14,35	0,69	2,40	0,61
20	20	12,97	18,05	26,33	2,56	3,48	1,70
	50	23,62	28,38	34,20	0,52	3,22	0,33
	100	27,79	34,41	39,06	0,59	2,05	0,38

Tabela 9Szczegółowe wyniki algorytmów. $FF = 0,10$, $v_r = 1$, $v_t = 0,5$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	4,19	5,48	9,55	2,79	0,12	0,66
	50	5,11	4,92	6,88	0,20	2,87	0,04
	100	5,24	4,65	7,21	0,42	3,02	0,12
10	20	10,32	14,25	19,67	1,08	3,58	1,42
	50	15,62	18,53	22,88	0,57	3,04	1,31
	100	13,54	16,15	20,95	1,03	4,17	0,05
20	20	23,54	30,36	38,20	2,62	4,04	0,33
	50	35,26	38,51	39,08	0,73	4,42	0,14
	100	38,95	39,83	39,54	0,17	5,43	0,43

Tabela 10Szczegółowe wyniki algorytmów. $FF = 0,25$, $v_r = 1$, $v_t = 0,5$

M	N	FIFO	FGi FIFO	FGj FIFO	FGi NN	FGi mNN v1	FGi mNN v2
5	20	6,98	5,74	8,84	0,76	2,67	0,58
	50	9,77	8,80	13,30	0,40	4,10	0,06
	100	8,60	8,04	11,84	0,11	3,25	0,18
10	20	20,83	20,66	22,72	1,11	5,58	0,02
	50	21,98	20,84	21,92	0,51	5,17	0,20
	100	23,24	21,24	22,17	0,62	5,53	0,10
20	20	39,74	33,93	33,91	1,17	6,47	0,20
	50	51,35	35,88	32,02	1,02	8,05	0,16
	100	53,46	38,53	32,75	1,56	6,87	0,10

7. Wnioski i uwagi

Z przedstawionych w poprzednim rozdziale rezultatów wynika jasno (tab. 2–10), że zdecydowanymi liderami wśród zaproponowanych algorytmów są procedury oparte na regule NN. Algorytmy te generują praktycznie równie dobre (różnice błędów nie przekraczają 5%) wyniki niezależnie od rozmiaru problemu, współczynnika wypełnienia czy stosunku prędkości realizatora i taśmociągu.

Łatwo zauważyć, że wraz ze wzrostem rozmiaru problemu rosną również różnice w jakości rozwiązań generowanych przez algorytmy z grupy FIFO i NN. Podobna zależność istnieje także pomiędzy różnicą błędów względnych a współczynnikiem wypełnienia. Należy zwrócić uwagę na niewielkie różnice pomiędzy wynikami generowanymi przez wszystkie algorytmy dla problemów o najmniejszym rozmiarze i najmniejszym współczynniku wypełnienia (tab. 2).

Jeżeli chodzi o stosunek prędkości realizatora i taśmociągu, to widać wyraźnie, że dla zaproponowanych algorytmów trudniejsza jest sytuacja, w której realizator porusza się wolniej od taśmociągu (tab. 2–4). Najgorsze wyniki są ponad 90% gorsze od wiodących dla danego rozmiaru problemu. (tab. 4). Pomiedzy przypadkami identycznych prędkości realizatora i taśmociągu oraz realizatora poruszającego się szybciej nie było większych różnic (tab. 5–10).

Podsumowując, metodologia przedstawiona w niniejszej pracy umożliwia modelowanie szerokiej gamy rzeczywistych systemów produkcyjnych. Model jest łatwy w implementacji i szybki w analizie. Najlepsze z zaproponowanych algorytmów generują jakościowo dobre rozwiązania niezależnie od rozmiaru problemu czy stosunku prędkości realizatora i taśmociągu.

Literatura

- [1] Smutnicki C., Molga M.: *Szeregowanie zadań w systemie kompletowania z ruchomym realizatorem*. Pótrocznik AGH Automatyka, 7(1/2), 2003, 173–178
- [2] Smutnicki C., Molga M.: *Szeregowanie zadań w systemie kompletowania z wieloma ruchomymi realizatorami*. W: Komputerowo zintegrowane zarządzanie. Zbiór prac pod red. Ryszarda Knosali. Warszawa, WNT 2004, 451–458
- [3] De Koster R.: *Performance approximation of pick-to-belt orderpicking systems*. European Journal of Operational Research, 72, 1994, 558–573
- [4] Espelage W., Wanke E.: *Movement minimization for unit distances in conveyor flow shop processing*, Mathematical Methods of Operations Research, 57(2), 2003, 173–206
- [5] Espelage W., Wanke E.: *A linear time approximation algorithm for movement minimization in conveyor flow shop processing*, European Journal of Operations Research, 160(2), 2005, 485–500
- [6] McSherry D., Pendharkar P.C.: *A computational study on design and performance issues of multi-agent intelligent systems for dynamic scheduling environments*. Expert Systems with Applications, 16(2), 1999, 121–133
- [7] Cowling P., Ouelhadj D., Petrovic S.: *A multi-agent architecture for dynamic scheduling of steel hot rolling*. Journal of Intelligent Manufacturing. Special Issue on Agent-Based Manufacturing Process Planning and Scheduling, 14(5), 2003, 457–470

-
- [8] Cowling P., Ouelhadj D., Petrovic S.: *Dynamic scheduling of steel casting and milling using multi-agents*. Production Planning and Control. Special Issue on the Application of Multi Agent Systems to Production Planning and Control, 15, 2004, 1–11
- [9] Gyapay Sz., Pataricza A., Sziray J., Friedler F.: *Petri net-based optimization of production systems*. IEEE International Conference on Intelligent Engineering Systems, 2002, 465–469

