

Paweł DĄBROWSKI, Piotr PITUCHA, Czesław SMUTNICKI
Politechnika Wroclawska

SZEREGOWANIE ZADAŃ W SYSTEMIE KOLEKCJONOWANIA CZĘŚCI

Streszczenie. Rozważono problem kolekcjonowania części w systemie montażowym, sprowadzony do problemu szeregowania nowej klasy. Omawiano różne przypadki szczególne. Dla wybranego przypadku podano model matematyczny wraz z wygodnym modelem grafowym. Bazując na właściwościach modelu zaproponowano i przebadano algorytmy rozwiązania wykorzystujące metody poszukiwań lokalnych

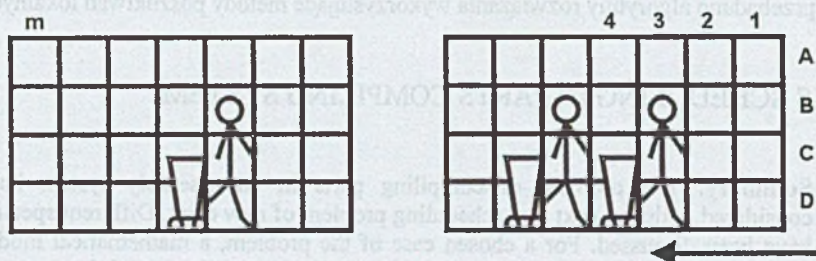
TASKS SCHEDULING IN PARTS COMPILING SYSTEM

Summary. The problem of compiling parts in an assembly system has been considered, reduced next to a scheduling problem of new class. Different special cases have been discussed. For a chosen case of the problem, a mathematical model with convenient graph model has been introduced. Basing on the model features, solution algorithms, which use the local search approach, have been proposed and evaluated.

1. Wprowadzenie

Opisany fragment systemu obejmuje dział przygotowania elementów do montażu instalacji elektrycznych, wykorzystywanych następnie w linii montażowej samochodów. Skład wiązki elektrycznej wraz z elementami towarzyszącymi zależy od wyposażenia samochodu zamówionego przez klienta (typ klimatyzacji, obrotomierz, ABS, ogrzewanie, czujniki etc.). Dla pojedynczego zamówienia punktem początkowym jest zebranie (skolekcjonowanie) odpowiedniego zestawu części składowych, których trzeba wybrać 20-50 spośród 100-300 różnych typów części dostępnych. Wszystkie typy części są dostępne w kontenerach umieszczonych w pionowym regale prostokątnym (tablicy), o rozmiarze m kolumn wysokości w każda, przy czym każdy kontener, umieszczony w pozycji (i, j) , zawiera wiele części określonego jednakowego typu, i jest uzupełniany na żądanie, w sposób nie

zaburzający pracy systemu, rys.1. Logiczna prostokątna organizacja regału może odpowiadać fizycznej budowie dwu- lub wielodzielnej, której rzut na płaszczyznę podstawy ma kształt litery I, U, L lub złożenia tych liter. Kolekcjonowanie części dla pojedynczego zamówienia polega na transporcie pojemnika (bin) wzdłuż regału, przy czym do pojemnika pobierane są tylko niektóre części, zaś proces pobierania jest nadzorowany przez komputer. W systemach o niskim stopniu zautomatyzowania przesuwanie pojemnika i pobór części wykonuje pracownik pod dyktando informacji otrzymywanych na wyświetlaczu z komputera. Pobór części z określonej kolumny regału blokuje dostęp do części w ustalonej liczbie kolumn sąsiednich ze względu na geometrię pojemnika i pobieraka. Czas pobierania części, jak i czas transportu pojemnika jest w przybliżeniu stały, jednak z jednej kolumny regału może być pobieranych kilka części, zaś niektóre kolumny mogą w ogóle nie być odwiedzane. Nie dopuszcza się (lub dopuszcza) wzajemnego wymijania pojemników.



Rys. 1. Schemat systemu kompletowania części

Fig. 1. Scheme of parts compiling system

Istnieje co najmniej kilka problemów wynikających z funkcjonowania opisanego systemu. Posiadając określony (dany) zestaw zamówień należy wybrać kolejność ich realizacji, posługując się przyjętym kryterium, np. minimalizacją czasu realizacji wszystkich zamówień. W przypadku gdy pojemnikowi towarzyszy pracownik, kryterium może być minimalizacja długości trasy wykonanej przez niego. Dalej, interesować nas może sposób rozmieszczenia kontenerów w regale mający wpływ na wartość różnych kryteriów. Wreszcie, zakładając pewną informację aprioryczną o napływających zleceniach dziennych, należy wybrać adaptacyjną strategię rozmieszczania kontenerów w regale oraz strategię obsługi zamówień, które minimalizują pewne kryterium szeregowania. Dodatkowym ograniczeniem może być liczba pojemników, jakie znajdują się w systemie. Innym rozszerzeniem jest wprowadzenie kilku kontenerów zawierających ten sam typ części, zlokalizowanych w różnych punktach regału. W tym przypadku marszruta pojemnika nie jest jednoznacznie określona przez profil zlecenia, bowiem miejsce pobrania części może podlegać wyborowi.

W niektórych systemach przesuw pojemnika jest automatyczny, bez wymijania, lecz pobieranie części jest wykonywane ręcznie przez pracownika, który musi w tym celu podejść do każdego pojemnika.

2. Model matematyczny

W niniejszej pracy omówimy system o architekturze liniowej, z jednym regałem, z ograniczonymi możliwościami tworzenia kolejki. Nie dopuszcza się wzajemnego wymijania się pojemników w trakcie trasy wewnątrz systemu. Liczba zleceń oraz ich charakterystyka są znane a priori. Podobnie znany i ustalony jest rozkład elementów w regałach. W dalszym ciągu, przez segment będziemy rozumieli zestaw w kontenerów ulokowanych w jednej kolumnie regału.

Podany problem praktyczny jest modelowany jako problem przepływowy szeregowania z ograniczeniami NS, LS (no store, limited store) oraz z nową klasą ograniczeń nazywanych dalej „ograniczeniami geometrycznymi”. Ograniczenie tego typu utrudnia lub blokuje możliwość przetwarzania zadań na stanowiskach sąsiednich lub przyległych. Niech i będzie wybranym stanowiskiem (maszyną), zaś j zadaniem na nim przetwarzanym w czasie p_j . W praktycznym przykładzie zadanie polega na skompletowaniu w pojemniku j -tego zestawu części według receptury; operacja polega na pobraniu przez pracownika (lub automatyczny pobierak) wybranych części z i -tego segmentu regału, zaś czas wykonywania operacji zależy od liczby różnych części pobieranych aktualnie z tego segmentu. Stąd, w kategoriach teorii szeregowania, jedno zlecenie kompletowania odpowiada zadaniu, zaś segment odpowiada maszynie. Do chwili zakończenia procesu pobierania elementu z segmentu i -tego b kolejnych segmentów występujących bezpośrednio za i jest niedostępnych, ze względu na geometrię układu pracownik-pojemnik lub geometrię pobieraka. Oznaczmy przez S_j, C_j odpowiednio terminy rozpoczęcia oraz zakończenia operacji zadania j na stanowisku i . Przyjmując, że kolejność realizacji zadań dana jest pewną permutacją $\pi = (\pi(1), \dots, \pi(n))$ na zbiorze $\{1, \dots, n\}$ (pracownicy lub kontenery nie mogą „wymijać” się na swojej trasie), otrzymujemy oczywiste ograniczenia technologiczne

$$C_{i,j} \leq S_{i+1,j}, \quad i = 1, \dots, m-1, \quad j = 1, \dots, n, \quad (1)$$

ograniczenia kolejnościowe

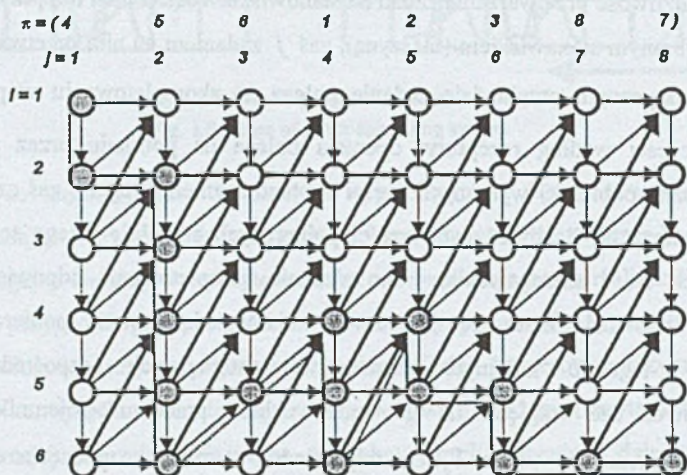
$$C_{i\pi(j)} \leq S_{i,\pi(j+1)}, \quad i = 1, \dots, m, \quad j = 1, \dots, n-1, \quad (2)$$

oraz dodatkowe ograniczenia wynikające z geometrii układu

$$C_{i\pi(j)} \leq S_{i-b,\pi(j+1)}, \quad i = b+1, \dots, m, \quad j = 1, \dots, n-1. \quad (3)$$

Odewołując się do modelu z grafem siatkowym dla przepływowego problemu szeregowania, [3], możemy stworzyć także odpowiedni model dla tego nietypowego przypadku. W grafie z rys. 2 wiersze odpowiadają maszynom, kolumny odpowiadają zadaniom umieszczonym w kolejności wynikającej z permutacji π . Obciążone węzły grafu odpowiadają operacjom, nieobciążone łuki pionowe reprezentują ograniczenia (1), nieobciążone łuki poziome – ograniczenia (2), nieobciążone łuki ukośne $((i, j), (i-b, j+1))$ – ograniczenia (3).

Tak skonstruowane warunki (1)-(3) domyślnie zezwalają na tworzenie nieograniczonej kolejki pomiędzy stanowiskami $i-b$ oraz $i-b-1$, jednakże intuicyjnie nie jest to dopuszczalne w omawianym przykładzie praktycznym. Istotnie, wstrzymane zadania, ze względu na brak miejsc składowania, tworzą kolejkę w obszarze stanowisk powodując ich sukcesywną blokadę. Warunki zapewniające w tym przypadku dopuszczalność są podobne do ograniczeń buforowania [4] i mogą być wprowadzone jak następuje:



Rys. 2. Graf reprezentujący ograniczenia problemu

Fig. 2. Graph representing problem constraints

Zadanie $\pi(j)$ może zostać rozpoczęte na stanowisku i , pod warunkiem że zadanie blokujące je $\pi(j-1)$, wykonywane wcześniej, zostało zakończone oraz opuściło stanowisko

$i + b$. Ponieważ opuszczenie stanowiska $i + b$ przez zadanie $\pi(j - 1)$ następuje w chwili jego rozpoczęciem w stanowisku $i + b + 1$, stąd otrzymujemy oczywiste warunki

$$S_{i\pi(j)} \geq S_{i+b+1,\pi(j-1)}, \quad i = 1, \dots, n - b - 1, \quad j = 2, \dots, n, \quad (4)$$

które następnie przekształcone do wygodniejszej postaci

$$C_{i\pi(j)} - p_{i\pi(j)} \leq S_{i-b-1,\pi(j+1)}, \quad i = b + 2, \dots, n, \quad j = 1, \dots, n - 1, \quad (5)$$

mogą być reprezentowane jako ukośne łuki $((i, j), (i - b - 1, j + 1))$ z obciążeniem minus $p_{i\pi(j)}$ w grafie z rys. 2. Zauważmy dalej, że niektóre łuki ukośne pochodzące z (3) w połączeniu z tymi z (5) są nadmiarowe i mogą być usunięte (tylko w przypadku zerowych czasów transportu pomiędzy stanowiskami). Istotnie, rozważmy węzły (i, j) oraz $(i - b - 1, j + 1)$, pomiędzy którymi istnieją dwie drogi. Pierwsza o długości zerowej zawiera jeden łuk $((i, j), (i - b, j + 1))$ pochodzący z (3). Druga zawiera łuk $((i, j), (i + 1, j))$, obciążony węzeł $(i + 1, j)$ (obciążenie $p_{i+1,\pi(j)}$) oraz łuk $((i + 1, j), (i - b, j + 1))$ (obciążenie $-p_{i+1,\pi(j)}$). Długość tej ostatniej drogi jest większa lub równa zero, jeśli tylko łuk $((i, j), (i + 1, j))$ posiada obciążenie nieujemne (czas transportu). Zatem łuk $((i, j), (i - b, j + 1))$ może zostać pominięty dla $i = b + 1, m - 1, j = 1, \dots, n - 1$.

Dla kompletności modelu podamy sposób uwzględnienia dekompozycji segmentów na regały (rozłączne ciągi segmentów). Niech i będzie ostatnim segmentem w pewnym regale, zaś $i + 1$ pierwszym segmentem w regale następnym. Przypadek, gdy pomiędzy i oraz $i + 1$ można utworzyć nieograniczoną kolejkę, jest oczywisty, nie potrzeba wprowadzać żadnych ukośnych łuków w grafie pomiędzy węzłami w kolumnach i oraz $i + 1$. Z kolei ograniczenia geometryczne blokujące $b' < b$ stanowisk można modelować opisaną wcześniej techniką. Dopelnieniem jest czas transportu niezależny od zadań i stanowisk oraz pewne ustalone kryterium optymalności.

Zauważmy, że dla $b = m$ ograniczenia geometryczne odpowiadają trywialnemu systemowi z pojemnością $c = 1$, dla którego wartość kryterium $C_{\max}(\pi)$ nie zależy od permutacji (jest stała), kryterium $\sum C_j$ jest minimalizowane przez zmodyfikowaną regułę SPT (uszereguj według niemalejących wartości $\sum_{j=1}^m p_{ij}$), zaś kryterium $\sum w_j C_j$ jest minimalizowane przez zmodyfikowaną regułę WSPT (uszereguj według niemalejących

wartości $\sum_{i=1}^m \frac{P_{ij}}{w_j}$). Problemy z ograniczeniami geometrycznymi posiadają własności takie same lub analogiczne jak problemy z buforowaniem. Przykładowo, w rys. 2 pokazano przebieg ścieżki krytycznej odpowiadającej kryterium długości uszeregowania.

3. Badania eksperymentalne algorytmów

Zaproponowany model grafowy został wykorzystany do wyznaczania wartości funkcji celu w dwóch algorytmach poszukiwań lokalnych bazujących na zaburzeniach typu „insert” dla permutacji [3]: poszukiwania zstępującego (DS) i symulowanego wyżarzania (SA). W DS zastosowano pełny przegląd otoczenia, z dokładnym obliczeniem wartości kryterium dla wszystkich rozwiązań, z metodą najszybszego spadku (do najlepszej poprawy), losowym rozwiązaniem początkowym oraz zatrzymaniem w pierwszym ekstremum lokalnym. W SA wykorzystano wariant z automatycznym doбором parametrów i logarytmicznym schematem studzenia [2]. Pojedyncza iteracja dla wybranego wariantu algorytmu SA składa się z $n(n-1)$ losowych ruchów przy stałym poziomie akceptacji.

Badania przeprowadzono dla podstawowego wariantu systemu z jednym regałem z n kontenerami i liczbą m różnych kolumn w regale. Optymalizacja została przeprowadzona dla dwóch typowych funkcji kryterialnych: długości uszeregowania $C_{\max} = \max_{1 \leq j \leq n} C_{m,\pi(j)}$ oraz średniego czasu przepływu przy zerowych terminach dostępności zleceń $\bar{C} = \frac{1}{n} \sum_{j=1}^n C_{m,\pi(j)}$.

Ponieważ problem jest nowy, brak jest standardowych przykładów testowych oraz wyników referencyjnych, do których można by odnieść uzyskiwane wyniki. Dlatego też dla potrzeb eksperymentu instancje testowe problemów zostały wygenerowane losowo. Wartości czasu obsługi dla określonej kolumny regału były losowane ze zbioru $\{0, 10, 20, 30\}$ z następującymi prawdopodobieństwami $\{0.4, 0.3, 0.2, 0.1\}$. Czas transportu pomiędzy kolejnymi kolumnami, o ile nie było to zaznaczone, wynosił 1. Testy zostały wykonane dla wielkości pojemnika b równych 2, 3 i 4. Dla każdego zestawu danych obliczenia zostały przeprowadzone dla serii 10 instancji. Dla każdej serii zostały określone średnie względne procentowe poprawy wartości funkcji celu według wzoru

$$R(C_x) = \frac{(C^{INI} - C_x)}{C^{INI}} \cdot 100\%$$

gdzie: C^{INI} - jest wartością funkcji kryterialnej dla rozwiązanie startowego,
 $C_x \in \{C_{max}^{DS}, C_{max}^{SA}, \bar{C}^{DS}, \bar{C}^{SA}\}$ - jest rozwiązaniem dla algorytm DS lub SA dla kryterium C_{max} lub \bar{C} . Odpowiednio, wielkość IT odnosi się do średniej liczby iteracji algorytmu (czas iteracji jest porównywalny dla SA i DS).

Tablica 1

Jakość algorytmów DS i SA dla kryterium C_{max}

n	m	b=2				b=3				b=4			
		DS		SA		DS		SA		DS		SA	
		R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT
20	5	23,9	11,1	25,8	22,0	19,8	10,8	20,8	22,3	12,2	8,9	12,4	13,4
20	10	23,1	12,8	26,6	28,0	21,3	12,4	23,7	22,5	19,8	13,6	22,6	26,7
20	20	19,9	12,6	23,4	30,8	17,6	13,0	21,9	33,1	17,1	12,4	21,0	28,8
20	30	17,3	13,0	21,3	34,0	16,7	13,5	21,5	36,0	16,1	13,6	19,6	33,7
20	40	19,5	15,6	23,7	35,8	18,1	15,9	21,3	35,5	16,5	14,7	20,7	33,8
30	5	25,5	16,1	27,4	31,9	16,7	13,4	19,0	28,0	10,4	10,1	10,7	19,2
30	10	21,0	15,7	25,1	33,3	20,1	16,5	23,5	36,1	19,2	18,8	21,5	35,8
30	20	19,7	17,3	23,3	41,9	17,6	17,8	20,4	40,8	18,0	20,6	20,7	46,6
30	30	17,5	17,1	21,8	45,5	18,7	20,6	21,6	44,4	16,3	18,3	19,7	45,4
30	40	17,0	16,8	20,8	47,5	17,6	21,3	21,2	47,7	15,6	19,1	19,1	46,1
40	5	25,2	20,6	28,1	40,2	17,9	18,0	19,5	37,3	10,9	13,6	11,6	24,1
40	10	23,3	21,4	26,5	41,7	21,1	24,7	23,7	42,1	19,4	24,0	21,4	46,6
40	20	20,5	22,3	23,6	48,0	18,8	23,5	21,9	52,2	17,7	25,1	20,1	50,7
40	30	17,5	19,6	21,4	55,1	17,7	21,5	20,7	53,8	16,1	22,5	19,8	55,7
40	40	17,6	24,2	21,1	56,1	16,2	21,8	20,6	61,3	15,4	20,2	19,6	60,3
100	50	14,5	35,6	18,5	94,6	13,2	41,0	17,4	100,3	12,5	40,2	16,4	105,2

Tablica 1 przedstawia wyniki dla funkcji kryterialnej C_{max} . Można zaobserwować zbliżone wartości względnych popraw dla obu algorytmów. W niektórych grupach instancji widoczna jest nieznaczna przewaga algorytmu SA nad DS, jednak przy co najmniej dwukrotnie większej liczbie iteracji. Dodatkowo, geometria regału (rozumiana jako wartości m oraz w) wpływa na wartość względnych popraw. I tak, dla przypadków z mniejszą liczbą segmentów w regałach uzyskane poprawy są większe przy jednakowej wysokości segmentu. Wpływ wielkości geometrycznej pojemnika b jest widoczny dla większości przypadków, gdzie wraz ze wzrostem wielkości pojemnika względne poprawy maleją. Najbardziej jest to widoczne dla przypadków $m=5$, gdzie wielkość geometryczna pojemnika jest znacząca, w stosunku do liczby kolumn w regale, i istotnie redukuje możliwość optymalizacji. Wyniki dla

przypadku funkcji kryterialnej \bar{C} zawiera tablica 2. Podobnie jak w przypadku kryterium C_{\max} , różnice pomiędzy wartościami popraw uzyskanymi przez oba algorytmy są niewielkie, przy co najmniej dwukrotnie większej liczbie iteracji wykonanej przez algorytm SA. Podobne są także zależności pomiędzy geometrią regału i wielkością geometryczną pojemnika a uzyskiwanymi poprawami. Jednak w przypadku rozpatrywanej funkcji kryterialnej uzyskiwane wartości R(DS) i R(SA) są większe. Wynika to z innej charakterystyki funkcji kryterialnej \bar{C} , posiadającej większy zakres zmienności.

Tablica 2

Jakość algorytmów DS i SA dla funkcji kryterialnej \bar{C}

		b=2				b=3				b=4			
		DS		SA		DS		SA		DS		SA	
n	m	R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT	R[%]	IT
20	5	32,5	15,2	35,2	30,8	35,4	16,7	36,2	34,4	26,7	16,5	26,4	31,9
20	10	28,4	19,2	31,5	39,5	26,7	18,0	28,8	39,0	26,6	19,3	29,3	38,3
20	20	22,7	19,7	24,2	48,6	20,4	20,3	22,9	44,7	22,2	20,4	24,7	45,5
20	30	18,2	21,7	18,8	48,9	19,0	23,9	21,6	52,3	18,0	19,9	19,6	48,9
20	40	21,0	28,3	22,0	53,3	18,9	25,1	21,3	56,1	17,9	25,7	19,8	52,8
30	5	37,8	23,3	39,4	50,2	32,7	27,1	34,3	51,2	28,7	27,8	29,2	52,1
30	10	27,7	24,4	30,1	52,3	27,9	24,7	29,8	51,0	27,8	26,4	30,2	55,5
30	20	20,9	24,4	24,5	60,3	20,4	25,2	23,7	59,0	21,7	26,3	24,7	60,2
30	30	19,8	29,0	22,8	65,6	21,9	31,6	23,9	64,9	18,7	30,3	21,6	63,9
30	40	17,5	30,8	20,9	69,8	18,8	31,3	21,2	70,7	17,6	32,9	20,2	72,2
40	5	37,7	30,3	41,3	60,4	31,6	32,6	34,1	62,9	30,1	37,4	31,0	65,6
40	10	31,1	34,4	34,4	64,8	30,1	33,2	32,6	64,0	29,0	35,8	29,9	64,3
40	20	24,1	36,2	26,8	74,2	23,1	33,6	26,4	73,2	22,0	34,5	25,0	74,2
40	30	19,8	39,0	23,5	76,3	21,1	36,0	24,1	81,9	19,7	38,6	22,1	80,8
40	40	19,3	39,7	22,6	88,2	16,9	34,9	20,5	84,0	19,3	44,2	21,6	86,0
100	50	17,4	67,5	21,5	143,9	17,3	76,4	21,4	143,1	20,4	89,9	21,4	152,6

Dla podklasy określonego problemu ($n=20$, $m=20$) z funkcją kryterialną C_{\max} został przetestowany wpływ wielkości czasu transportu (przejazdu) pomiędzy poszczególnymi kolumnami regału na uzyskiwane poprawy, patrz tablica 3, przy założeniu że czas przejazdu jest jednakowy pomiędzy dwoma sąsiednimi kolumnami regału. Wzrost wielkości czasu transportu powoduje zmniejszenie wartości R(SA). Efekt jest najbardziej widoczny dla wielkości czasów transportu porównywalnych z czasami załadunku do pojemników. Intuicyjnie, fakt ten jest spodziewany - możliwość optymalizacji zmniejsza się, ponieważ sumaryczne czasy obsługi dla poszczególnych pojemników są zbliżone.

Tablica 3
Wpływ czasów transportu na wartości $R(SA)$ [%]
dla kryterium C_{\max} ($n=20, m=20$)

czas transportu	b=2	b=3	b=4
0	24,1	21,6	22,1
1	22,9	21,2	20,2
2	21,6	19,7	19,9
3	20,5	18,8	18,5
5	19,1	17,1	17,1
10	15,9	13,8	13,6

Kolejne testy polegały na określeniu wpływu występowania ograniczonej liczby pojemników mogących przebywać jednocześnie w systemie (sposób modelowania opisano szczegółowo w pracy [5]) na wartości $R(SA)$. Wyniki dla ustalonej wielkości problemu ($n=20, m=5$) przedstawia tablica 4. Można zauważyć występowanie pewnej wartości granicznej liczby pojemników, po osiągnięciu której nie jest widoczna znacząca poprawa wartości funkcji celu. Wielkość ta odpowiada ograniczeniu pojemności samego systemu, zaś nasycenie jest osiągane w przypadku, gdy liczba pojemników pomnożona przez jego geometryczną wielkość b jest większa od liczby kolumn w regale (np. dla $b=2$ są to 2 pojemniki). Dla przypadku, gdy dostępny jest tylko jeden pojemnik, optymalizacja nie ma sensu, bowiem wartość wymienionych funkcji kryterialnych jest stała i nie zależy od kolejności.

Tablica 4
Wpływ liczby pojemników na $R(SA)$ [%] dla
kryterium C_{\max} ($n=20, m=5$)

Liczba pojemników	b=2	B=3	b=4
1	0	0	0
2	20,9	19,8	10,3
3	24,8	18,8	9,2
4	25,4	18,3	10,3
5	25,8	20,3	11,5
6	24,3	18,1	10,5
7	25,0	18,6	11,1
8	24,7	19,9	10,3

4. Wnioski

Dla obu przypadków funkcji kryterialnych niewielki wzrost wartości $R(DS)$ może być spowodowany pewną specyfiką przestrzeni rozwiązań określonej przez zastosowane otoczenie. W przypadku prostych algorytmów optymalizacji lokalnej, jakim jest DS, kłopotliwy jest brak możliwości ucieczki z lokalnych minimów przestrzeni rozwiązań, powodujący przedwczesną zbieżność algorytmu. W przypadku bardziej złożonych algorytmów optymalizacji lokalnej, jak np. SA, istnieje (statystyczna) możliwość opuszczenia minimum lokalnego w celu poszukiwania bardziej obiecujących obszarów przestrzeni rozwiązań. Najnowsze badania w optymalizacji dyskretnej pokazują, że zachowanie każdego z algorytmów powinno być analizowane szczegółowo w kontekście krajobrazu przestrzeni rozwiązań. W przypadku analizowanego problemu można zaobserwować względnie „głębokie” minima lokalne, wymuszające wykonanie co najmniej kilkunastu do kilkudziesięciu iteracji dla algorytmu DS. Algorytm SA wykonując podobną liczbę iteracji znacząco zmniejsza wartość parametru temperatury, a tym samym prawdopodobieństwo zaakceptowania rozwiązania pogarszającego funkcję celu. Tak więc, SA po wykonaniu takiej samej liczby iteracji jak algorytm DS, osiągając minimum lokalne ma zwykle wartość temperatury zbyt małą, by uzyskać możliwości opuszczenia go. Niewielkie różnice pomiędzy wynikami obu algorytmów pozwalają także przypuszczać, że przestrzeń rozwiązań zawiera wiele ekstremów lokalnych o zbliżonej wartości funkcji celu. Niewątpliwie więcej informacji można by uzyskać po wykonaniu odpowiednich badań lokalizujących obszar wielkiej doliny.

Prezentowane zagadnienie należy do klasy problemów optymalizacji pracy magazynów. W niniejszej pracy analizowaliśmy pewien szczególny problem szeregowania zleceń w systemie kompletowania części, mający swoje źródło w przemysłowych systemach produkcyjnych. Uzyskane wyniki sugerują, że do rozwiązywania problemu najlepiej nadaje się algorytm poszukiwań zstępujących, ponieważ daje względnie dobre rozwiązania w stosunkowo niewielkim czasie. Słabe zachowanie algorytmu SA skłania do wniosku, że istnieje potrzeba konstrukcji bardziej wyrafinowanych metod wykorzystujących więcej szczególnych własności problemu. Problemy optymalizacji pracy magazynu są w niewielkim stopniu opisywane w dostępnej literaturze. Opisane we wprowadzeniu różne warianty systemów tego typu oraz zyski z wprowadzenia optymalizacji rozmieszczenia elementów w regałach sprawiają, że problematyka tej klasy może się stać tematem wielu kolejnych badań.

LITERATURA

1. de Koster R.: Performance approximation of pick-to-belt orderpicking system, *European Journal of Operational Research*, 1994, s. 558-573.
2. Das H., Cummnings P., Le Van M.: Scheduling of serial multiproduct batch processes via simulated annealing, *Computers and Chemical. Engineering*, 1990, s. 1351-1362.
3. Nowicki E., Smutnicki C., A fast tabu search algorithm for the flow shop. *European Journal of Operational Research*, 1996, s. 160-175.
4. Smutnicki C.: A two-machine permutation flow shop scheduling problems with buffers. *OR Spectrum* 20, 1998, s. 229-235.
5. Smutnicki C.: Algorytmy szeregowania. Jak to rozwiązać. 1-300, 2002 (w druku).

Recenzent: Prof. dr hab. inż. Andrzej Świerniak

Abstract

The problem of compiling parts in an assembly system is formulated as follows. Parts are stored in the rectangle shelf (table) of width m and height w , where mw is of order 200-300. Each of n orders (tasks) has to collect approximately 20-30 parts, accordingly to the individual order. Parts are collected to the bin moved along the shelf by a worker. Because of the bin-and-worker geometry, access to the column of the shelf blocks access of other workers to some neighboring columns. Because of the workers route, formed queues blocks also access to shelf for other workers. Assuming that successive bins are serviced accordingly to FIFO rule, find the sequence of processing of orders, which minimizes the given criteria value. The problem is converted next to the flow shop scheduling problem with classical no-store constraints and constraints of new class called "geometric constraints". Both types of constraints can be advantageously modeled with the help of grid graph, derived from the conventional flow-shop model. By extending this graph representation, one can find special properties useful for solution algorithms. At the current stage, two algorithms DS (descending search) and SA (simulated annealing) have been implemented and tested. Using these algorithms and the rich set of random benchmarks, some basic numerical properties of the problem have been detected, which influence on the further research directions.