AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/2

Wojciech Chmiel*, Piotr Kadłuczka*, Konrad Wala**, Stanisław Jędrusik***

Algorytmy heurystyczne w trójwymiarowym zagadnieniu pakowania

1. Wprowadzenie

W obecnym czasie potrzeby wynikające z nasilającej się globalizacji oraz wzrostu konkurencji są powodem dynamicznego rozwojem systemów wspierania produkcji, organizacji pracy i logistyki. Ważnym elementem takich systemów są szybkie algorytmy pozwalające na ustalenie sposobu pakowania kontenerów. Opracowanie efektywnego algorytmu pakowania wymaga rozwiązania złożonych zagadnień optymalizacyjnych. Poziom złożoności wzrasta, gdy rozważana funkcja celu jest funkcją wielokryterialną. Rozważane trójwymiarowe zagadnienie pakowania cechuje:

- wykładniczo rosnący czas obliczeń algorytmów dokładnych – problem *NP-trudny*,
- duża liczba ograniczeń równościowych i nierównościowych,
- duża liczność przestrzeni rozwiązań – związana z liczbą zmiennych decyzyjnych, która w praktyce wynosi od kilkuset do kilkunastu tysięcy,
- wielomodalność funkcji celu,
- chropowatość krajobrazu przestrzeni rozwiązań, związana ze słabą korelacją rozwiązań w sąsiedztwie,
- zwodniczość krajobrazu przestrzeni rozwiązań.

Praktyczne problemy charakteryzujące się dużym rozmiarem oraz wykładniczo rosnącym czasem obliczeń dla algorytmów dokładnych, powodują, że mimo stale rosnących mocy obliczeniowych komputerów nie jesteśmy w stanie w znaczący sposób wpłynąć na zwiększenie rozmiaru, rozwiązywalnych w akceptowalnym czasie problemów metodami ścisłymi. W badaniach zaprezentowanych w artykule koncentrujemy się na algorytmach

* Katedra Automatyki, Wydział EAIiE, Akademia Górniczo-Hutnicza w Krakowie
** Wyższa Szkoła Biznesu, Dąbrowa Górnicza
*** Katedra Informatyki Stosowanej, Wydział Zarządzania, Akademia Górniczo-Hutnicza w Krakowie

heurystycznych, dobranych pod kątem wykorzystania specyfiki opracowanych modeli pakowania oraz reprezentacji dopuszczalnych punktów wstawienia.

Algorytmy heurystyczne są często jedyną drogą dla uzyskania rozwiązań, zadowalających z punktu widzenia praktyki, zarówno co do rozmiaru rozwiązywanych w rozsądnym czasie przykładów, jak i jakości (odległości od rozwiązania optymalnego) otrzymanych wyników [1, 8, 2, 5, 13].

W punkcie 2 dokonano przeglądu metod rozwiązywania zagadnień pakowania, natomiast w punkcie 3 sformalizowano rozważany problem. Algorytmy konstrukcyjne i algorytm popraw, przedstawiono odpowiednio w punktach 4 i 5. Punkty 6 i 7 zawierają uzyskane wyniki eksperymentów wraz z podsumowaniem.

2. Przegląd metod stosowanych dla zagadnienia pakowania

Praktyczny aspekt wykorzystania algorytmów w zautomatyzowanych systemach kompletacji i pakowania, wspomagających zarządzanie procesami magazynowymi i logistycznymi przedsiębiorstw, wymusza konieczność rozwiązywania zadań o znacznym rozmiarze w zadawalającym czasie. Wynika to z dużej liczby przetwarzanych zamówień, obsługi klientów detalicznych, dużego asortymentu produktów (towarów) oraz ograniczeń czasowych dla systemu sterowania. W literaturze światowej prezentowane są efektywne metody dedykowane dla wąskich, ściśle określonych zagadnień pakowania. Liczba artykułów na ten temat zestawiona w przeglądzie Dyckhoffa i Finke (1992) oraz Wäschera (2006) dochodzi do tysiąca. Znaleźć wśród nich można zarówno algorytmy ścisłe, jak i przybliżone (z przewagą tych drugich).

W pierwszej grupie należy wymienić:

- Algorytmy bazujące na metodzie podziału i ograniczeń. Przykładem tego typu algorytmu, zastosowanego do rozwiązywania zagadnienia MKP (*multiple knapsack problem*), jest opisany w [14]. Dolne i górne ograniczenie wyznacza się na podstawie relaksacji Lagrange'a, oraz dodatkowo stosuje się zachłanną heurystykę i redukcję rozmiaru problemu. W efekcie, uzyskano algorytm dokładny pozwalający wydajnie rozwiązywać instancje opisujące problem pakowania 32 000 pakunków do 50 kontenerów.
- Metody enumeracyjne – np. przeszukiwanie drzewa rozwiązań (w głąb, wszerz), zestawiane często z innymi metodami. Są proste w implementacji, ale efektywne tylko dla zadań o niewielkim rozmiarze, stąd często używane są do rozwiązywania podproblemów po dekompozycji instancji wyjściowej.
- Metody analityczne – stosowane dla zagadnień pakowania jednorodnych przedmiotów, np. brył o często skomplikowanym kształcie.

W grupie metod przybliżonych należy wymienić:

– Algorytmy zachłanne – najliczniej reprezentowana grupa algorytmów. Ze względu na prostotę nadają się do przetwarzania dużych instancji (w sytuacji ograniczeń czasowych), gdzie kosztem jakości rozwiązań mamy krótki czas obliczeń. Przykładem algorytmu zachłannego dla problemu SSBCLP (*single sized bin container loading problem*) jest stworzony przez Pisingera [10], w którym dla każdej wielkości paczki i kontenera wyznaczany jest indeks, będący ilorazem kosztu jednostkowego objętości paczki do współczynnika wypełnienia kontenera dla danej paczki. Wybierana jest zawsze najlepsza paczka i usuwana z listy.
– Algorytmy konstrukcyjne i poprawy rozwiązania – zadaniem pierwszych jest utworzenie rozwiązania początkowego, które następnie próbuje się poprawić. Pierwsze charakteryzują się prostotą – typowa złożoność obliczeniowa – liniowa (w jednej iteracji wyznaczamy jedną składową rozwiązania). Algorytmy popraw bazują na definicji sąsiedztwa rozwiązania (o różnej liczności), z którego wybierane jest najlepsze.
– Algorytmy symulowanego wyżarzania – przedstawione w punkcie 5 artykułu.
– Algorytmy poszukiwania z zabronieniami (*tabu search*) – implementację różnych technik TS dla szerokiej klasy problemów pakowania dwu i trójwymiarowych zaprezentowano w [6].
– Algorytmy ewolucyjne, rojowe oraz inne inspirowane naturą. Przykładowy opis [3] zastosowania algorytmu ewolucyjnego BPGA do rozwiązywania rzeczywistego problemu optymalizacji wskazuje na wysoką efektywność metody. Każda klasa problemów pakowania wymaga zróżnicowanego podejścia, jeśli chodzi o kodowanie rozwiązania i konstrukcję operatorów genetycznych.
– Metaheurystyki – stanowią połączenie wszystkich wymienionych wcześniej metod (dokładne i przybliżone). Przykładem tego typu algorytmu jest autorstwa Miyazawa i Wakabayashi [9], który dzieli wyjściową instancję na kilka podinstancji, stosując do nich różne heurystyki np. NFDH (*next fit deceasing height*) w dwóch wersjach, LL – Lee, Chenga oraz inne, uzyskując rozwiązania częściowe. Są one następnie łączone.

3. Pakowanie kontenera

Standardowo przyjmuje się następujące założenia dotyczące procesu pakowania kontenera:

1) podczas przystępowania do wykonania zlecenia pakowania znana jest ilość asortymentów, jakie zostaną pobrane do jego realizacji;
2) kompletacja może być prowadzona bezpośrednio do kontenerów, lub do pojemników kompletacyjnych, z których następnie nastąpi przepakowanie do docelowych jednostek ładunkowych – kontenerów;

3) wymiary ładowanych opakowań towarów w postaci prostopadłościanów są z góry znane;
4) w momencie rozpoczęcia pakowania znane są wymiary objętości ładunkowej kontenera;
5) nie podaje się rodzaju asortymentu, jakiego dotyczy problem pakowania, specyfikuje się natomiast następujące parametry (ograniczenia), jakie powinny zostać uwzględnione w procesie pakowania jednostek wysyłkowych:
   – dopuszczalna waga pojedynczej jednostki wysyłkowej,
   – waga oraz rozmiary poszczególnych asortymentów, które mają się znaleźć w kontenerze,
   – zdefiniowane typy opakowań wysyłkowych,
   – informacje dodatkowe dotyczące asortymentów (możliwość pakowania z innymi asortymentami, warunki transportu, możliwość piętrzenia, klasa odporności na zniszczenie);
6) proces pakowania może być wieloetapowy (w pierwszej kolejności pakowanie detalicznych artykułów do kartonów, następnie grupowanie kartonów w palety, może również wystąpić grupowanie palet w większe jednostki transportowe).

W pracy przyjęto reprezentację rozwiązania zaczerpniętą z artykułu [11], dla trójwymiarowego zagadnienia pakowania opierającej się na potrójnej sekwencji numerów paczek, permutacji (A, B, C), zawierającej relatywne położenie paczek w stosunku do pozostałych. Powyższa reprezentacja nie pozwala na opis wszystkich możliwych sposobów upakowania, jednak pozwala na reprezentację ich dużej części. W szczególności pozwala na reprezentację rozwiązań, realizowalnych przez robota (*robot packing*), tzn. jako pakowanie od lewego, dolnego, tylnego rogu kontenera, gdzie kolejne paczki wstawiane są naprzeciw, powyżej lub po prawej, w stosunku do wszystkich, poprzednio wstawionych. Ta metoda pakowania modeluje rzeczywiste zagadnienie pakowania kontenera występujące w większości produkcyjnych systemów załadowczych. Aby zapobiec kolizji pomiędzy ramieniem robota oraz paczkami już wstawionymi, zakłada się, że żadna z paczek nie może blokować ruchu ramienia. Można wykazać, że to ograniczenie nie ma dużego wpływu na jakość pakowania [10].

3.1. Definicja 3-sekwencji

W publikacjach [11] i [12] zaproponowano abstrakcyjną reprezentację sekwencji pakowania w oparciu o trzy permutacje A, B i C. Dla każdej z tych permutacji, możemy zdefiniować pewną relację ω_{ij}, która będzie oznaczała, że w permutacji liczba i występuje przed j. Załóżmy również, dla uproszczenia zapisu, że $\sim\omega_{ij} \Leftrightarrow \omega_{ji}$. Niech (x_i, y_i, z_i) i (w_i, d_i, h_i), będą odpowiednio współrzędnymi dolnego, lewego, tylnego wierzchołka paczki oraz jej szerokością, głębokością i wysokością. W związku z tym możemy zdefiniować trzy relacje porządku częściowego α_{ij}, β_{ij}, χ_{ij}:

$$\alpha_{ij} \Leftrightarrow (x_i + w_i \le x_j \vee y_i \ge y_j + h_j \vee z_i \ge z_j + d_j) \tag{1}$$

paczka i jest na lewo, ponad lub z przodu w stosunku do paczki j,

$$\beta_{ij} \Leftrightarrow (x_i \le x_j + w_j \vee y_i \le y_j + h_j \vee z_i \le z_j) \tag{2}$$

paczka i jest na lewo, pod lub z tyłu w stosunku do paczki j,

$$\chi_{ij} \Leftrightarrow (x_i \ge x_j + w_j \vee y_i + h_i \le y_j \vee z_i \ge z_j + d_j) \tag{3}$$

paczka i jest na prawo, pod lub z przodu w stosunku do paczki j.

Zgodnie z zasadą pakowania przez robota, każda nowa paczka jest pakowana, w stosunku do znajdujących się w kontenerze, najbardziej na lewo, nad lub z tyłu. Można też udowodnić, że każdej z sekwencji A, B i C można przypisać odpowiadające im ułożenie paczek. Aby określić wzajemne położenie paczek i oraz j, opierając się na (1), (2) oraz (3) [11], można zastosować następujące relacje:

$$\alpha_{ij} \wedge \beta_{ij} \wedge \sim\chi_{ij} \Rightarrow \ i \text{ jest na lewo od } j \tag{4}$$

$$\sim\alpha_{ij} \wedge \beta_{ij} \wedge \chi_{ij} \Rightarrow \ i \text{ jest poniżej } j \tag{5}$$

$$\sim\alpha_{ij} \wedge \beta_{ij} \wedge \sim\chi_{ij} \Rightarrow i \text{ jest z tyłu } j \tag{6}$$

$$\alpha_{ij} \wedge \beta_{ij} \wedge \ \chi_{ij} \Rightarrow i \text{ jest z tyłu } j \tag{7}$$

Wyrażenia (6) i (7), powodują preferowanie jednego kierunku pakowania, mając negatywny wpływ na uzyskaną jakość pakowania. Jednak ich istnienie jest wymagane, ponieważ zapewnia, że każdy zestaw permutacji A, B, C definiuje dopuszczalną sekwencję pakowania. Za pomocą powyższych zależności można zdefiniować procedurę określającą pozycję nowej paczki na podstawie paczek już wstawionych.

3.2. Procedura wstawiania paczek do kontenera

Paczki wstawiamy do kontenera w kolejności określonej przez sekwencję, gdzie pierwsza paczka ma współrzędne (0, 0, 0). Niech zbiór P zawiera paczki załadowane już do kontenera. Pozycję, punkt (x_j, y_j, z_j), kolejnej paczki $j \notin P$, wstawianej do kontenera, wyznaczamy na podstawie załadowanych paczek $i \in P$. Niech dalej:

– $P_x \subseteq P$ jest podzbiorem paczek, dla których spełniona jest zależność

$$\alpha_{ij} \wedge \beta_{ij} \wedge \sim\chi_{ij},$$

– $P_y \subseteq P$ jest podzbiorem paczek, dla których spełniona jest zależność

$$\sim\alpha_{ij} \wedge \beta_{ij} \wedge \chi_{ij},$$

– $P_z \subseteq P$ jest podzbiorem paczek dla których spełniona jest zależność

$$(\sim\alpha_{ij} \wedge \beta_{ij} \wedge \sim\chi_{ij}) \vee (\alpha_{ij} \wedge \beta_{ij} \wedge \chi_{ij}).$$

Przyjęto, że współrzędne wstawienia paczki j są wyznaczone za pomocą następujących reguł:

$$x_j = \max \{0; \max (x_i + w_i), i \in P_x\} \tag{8}$$

$$y_j = \max \{0; \max (y_i + d_i), i \in P_y\} \tag{9}$$

$$z_j = \max \{0; \max (z_i + h_i), i \in P_z\} \tag{10}$$

gdzie (w_i , d_i , z_i) są wymiarami krawędzi paczki j po uwzględnieniu jej rotacji. Jeżeli po wstawieniu paczki j okaże się, że wystaje ona poza wymiary kontenera, to paczka nie jest ładowana do kontenera, inaczej jest ładowana i wstawiana do zbioru P.

Jeżeli przechowujemy tablicę, w której pozycja każdej załadowanej paczki i jest zanotowana w trzech sekwencjach A, B, C, to można w stałym czasie sprawdzić relacje α_{ij}, β_{ij}, χ_{ij} dla dwóch paczek i, j, stąd wstawienie paczki za pomocą porównań z pierwotnie wstawionymi za pomocą wymienionych powyżej reguł może być wykonane w czasie $O(|P|)$. Wstawienie wszystkich paczek jest więc realizowalne w czasie $O(n^2)$.

Działanie procedury wstawiania można trochę przyśpieszyć, usuwając ze zbioru P paczki całkowicie zakryte przez nowo wstawioną paczkę j. Paczka i jest zakryta przez paczkę j, jeżeli $x_i + w_i \le x_j + w_j$, $y_i + h_i \le y_j + h_j$, $z_i + d_i \le z_j + d_j$, i wtedy nie wpływa ona na proces wstawiania następnych paczek. Usuwając taką paczkę z P, unikamy niepotrzebnych sprawdzeń podczas wstawiania kolejnych paczek do kontenera.

4. Algorytm konstrukcyjny

Jako algorytm konstrukcyjny zastosowano algorytm szeregowania listowego zawierający dwa następujące podstawowe kroki:

1. Utwórz listę paczek sortując je według wybranego parametru.
2. Zgodnie z kolejnością określoną przez listę wstawiaj paczki do kontenera według przedstawionej powyżej procedury wstawiania.

W kroku pierwszym stosowano następujące sposoby sortowania:

– od największej objętości paczki do najmniejszej (alg. *ATD*), tj.:

$$v_1 \geq v_2 \geq ... \geq v_n \text{ , gdzie: } v_j = d_j s_j w_j \text{ , } j = 1, 2, ..., n \tag{11}$$

– od największego boku maksymalnego paczki do najmniejszego (alg. *BTD*) :

$$\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_n \text{ , gdzie: } \alpha_j = \max\{ d_j, s_j, w_j \}, \; j = 1, 2, ..., n \tag{12}$$

– od największego boku minimalnego paczki do najmniejszego (alg. *CTD*) :

$$\beta_1 \geq \beta_2 \geq ... \geq \beta_n \text{ , gdzie: } \beta_j = \min\{ d_j, s_j, w_j \}, \; j = 1, 2, ..., n \tag{13}$$

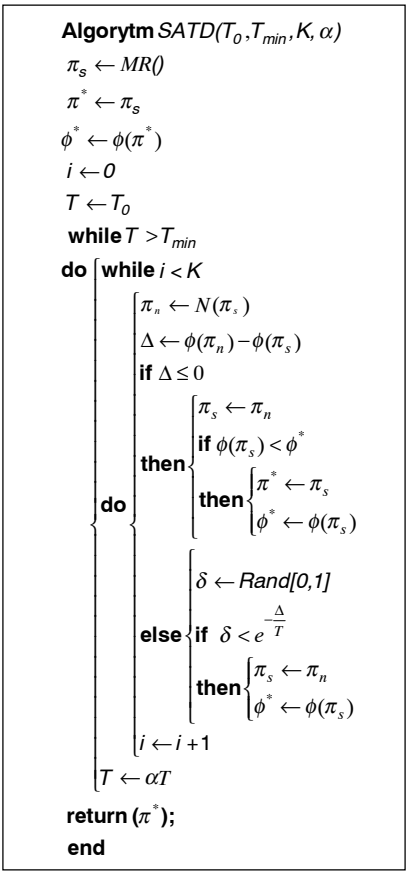
Wszystkie powyższe reguły sortowania wyznaczają różne warianty uszeregowania zachłannego paczek.

5. Algorytm symulowanego wyżarzania *SATD*

Algorytm symulowanego wyżarzania przetwarza rozwiązania określające zarówno sekwencje pakowania, jak również rotację paczki. Parametrami algorytmu są: T_0 – temperatura początkowa, $T_{\min}$ – temperatura końcowa, α – parametr sterujący zmianą temperatury w kolejnych iteracjach $\alpha \in (0, 1)$, K – liczba wygenerowanych rozwiązań w stałej temperaturze (dojście do stanu równowagi termodynamicznej). Każde rozwiązanie π ma postać $\pi = \{A, B, C, R\}$, gdzie: A, B, C są permutacjami n-elementowego zbioru określającymi porządek pakowania, gdzie n – oznacza liczbę paczek, natomiast wektor R, określa rotację paczki w stosunku do położenia początkowego $R = \{r_1, r_2, ..., r_n\}$, gdzie $r \in \{0, ..., 5\}$. Pseudokod algorytmu $N(\pi)$, tworzącego rozwiązane w otoczeniu rozwiązania aktualnego, przedstawiono na rysunku 1, a na rysunku 2 algorytm *SATD*.

```
Algorytm N(π(A,B,C,R),n)
  i ← Rand[0,3]
  if(i == 0) then SWAP(A)
  if (i == 1) then SWAP(B)
  if (i == 2) then SWAP(C)
  if (i == 3) then RCHAN(R)
  return(π_new)
end
```

Rys. 1. Algorytm $N(\pi)$



Rys. 2. Algorytm symulowanego wyżarzania *SATD* dla zagadnienia trójwymiarowego pakowania

gdzie:

MR() – generuje losowo, bazując na rozkładzie równomiernym rozwiązania początkowego,

$N(\pi)$ – generuje nowe rozwiązanie w otoczeniu rozwiązania π,

$\phi(\pi)$ – określa wartości funkcji przystosowania (funkcji celu dla rozwiązania) jako współczynnik wypełnienia kontenera zdefiniowany następująco:
$\phi(\pi) = \sum_{i \in S} v_i / V_c$ [%], gdzie: S to zbiór indeksów paczek wstawionych do kontenera, natomiast $V_c = W \times D \times H$, to całkowita pojemność kontenera,

Rand[*a, b*] – generator liczb pseudolosowych z zakresu [*a*, *b*], w oparciu o generator *Mersenne twister* [7],

SWAP(*x*) – dokonuje losowej zmiany położenia dwóch paczek w losowo wybranej permutacji z rozkładem równomiernym,

RCHAN(*x*) – dokonuje losowej modyfikacji, zgodnie z rozkładem równomiernym, jednej ze współrzędnych w wektorze *R*,

π^* – najlepsze rozwiązanie znalezione przez algorytm *SATD*,

SWAP(x) – dokonuje losowej zmiany położenia dwóch paczek w losowo wybranej permutacji z rozkładem równomiernym.

6. Wyniki eksperymentów obliczeniowych

W eksperymentach obliczeniowych do rozwiązywania zagadnienia pakowania zaimplementowano w języku C# algorytm symulowanego wyżarzania *SATD* oraz trzy algorytmy konstrukcyjne: *ATD* (realizujący sortowanie paczek wg wzoru (11)), *BTD* (realizujący sortowanie paczek wg wzoru (12)), *CTD* (realizujący sortowanie paczek wg wzoru (13)) oraz *LCD* – algorytm określający kolejność wstawiania w sposób losowy.

Stworzono również oprogramowanie do wizualizacji uzyskanych rozwiązań. Jako zadania testowe zastosowano sześć instancji, wśród których można wydzielić dwa typy zadań:

(a) paczki otrzymano w wyniku cięcia gilotynowego kontenera, długości boków paczki określano z rozkładem równomiernym, ograniczonym od góry przez długość boku kontenera,

(b) paczki wygenerowano w sposób losowy, zgodnie z rozkładem normalnym, gdzie $\mu = 3$ oraz $\sigma^2 = 2$.

W tabelach 1–3 przyjęto następujące oznaczenia n – liczba paczek, $\phi(\pi^*)$ [%] – funkcja celu najlepszego uzyskanego przez algorytm rozwiązania, ($W \times D \times H$) – odpowiednio szerokość, głębokość, wysokość paczki, T_0 – temperatura początkowa zastosowana w algorytmie *TDSA* oraz K – liczba przeszukanych rozwiązań przy ustalonej temperaturze.

Tabela 1

Wyniki testów algorytmu *SATD* dla trzech instancji testowych uzyskanych poprzez cięcie gilotynowe kontenerów

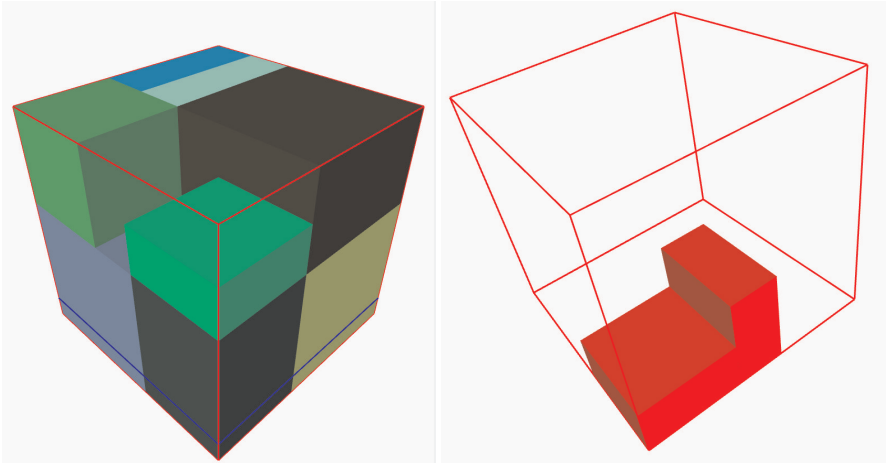
	K	T_0	$\phi(\pi^*)$		K	T_0	$\phi(\pi^*)$		K	T_0	$\phi(\pi^*)$		K	T_0	$\phi(\pi^*)$
n=20 W=5 D=5 H=5	3	100	95,20	n=60 W=8 D=8 H=8	3	100	48,02	n=100 W=10 D=10 H=10	3	100	29,70	n=200 W=15 D=15 H=15	3	100	19,52
		200	93,60			200	43,63			200	34,35			200	16,43
		400	96,83			400	42,01			400	30,20			400	18,54
		800	93,60			800	44,95			800	28,24			800	19,35
		1000	92,05			1000	43,01			1000	29,84			1000	24,66
	6	100	95,20		6	100	47,00		6	100	33,80		6	100	20,67
		200	96,86			200	42,84			200	31,33			200	20,58
		400	93,66			400	48,6d			400	37,61			400	19,49
		800	88,87			800	49,25			800	33,60			800	21,09
		1000	93,62			1000	43,60			1000	36,60			1000	24,60
	10	100	99,20		10	100	50,20		10	100	35.40		10	100	21,70
		200	94,40			200	45,91			200	40,90			200	21,54
		400	91,20			400	52,04			400	35,60			400	23,36
		800	89,61			800	47,03			800	38,60			800	24,10
		1000	98,40			1000	45,10			1000	35,70			1000	19,50

Tabela 2
Wyniki testów algorytmu *SATD* dla trzech instancji testowych uzyskanych poprzez losową generację paczek z rozkładem normalnym, w którym $\mu = 3$ oraz $\sigma^2 = 2$

	K	T_0	$\phi(\pi^*)$		K	T_0	$\phi(\pi^*)$		K	T_0	$\phi(\pi^*)$
		100	*82,40*			*100*	*19,45*			*100*	*48,44*
	3	200	*88,80*		3	*200*	*16,41*		3	*200*	*53,13*
		400	*85,60*			*400*	*18,52*			*400*	*52,15*
		800	*83,20*			*800*	*18,29*			*800*	*50,59*
		1000	*80,00*			*1000*	*24,59*			*1000*	*48,44*
		100	*90,40*			*100*	*20,65*			*100*	*53,52*
n=125		2 00	*88,00*	*n*=200		*200*	*20,50*	*n*=400		*200*	*54,69*
W=5	6	400	*92,80*	*W*=15	6	*400*	*19,44*	*W*=8	6	*400*	*54,49*
D=5		800	*89,60*	*D*=15		*800*	*20,98*	*D*=8		*800*	*51,76*
H=5		1000	*84,00*	*H*=15		*1000*	*20,06*	*H*=8		*1000*	*57,42*
		100	*59,00*			*100*	*21,66*			*100*	*58,98*
		200	*52,30*			*200*	*21,51*			*200*	*64,45*
	10	400	*57,23*		10	*400*	*23,32*		10	*400*	*59,77*
		800	*54,69*			*800*	*24,06*			*800*	*54,69*
		1000	*65,04*			*1000*	*19,47*			*1000*	*65,04*

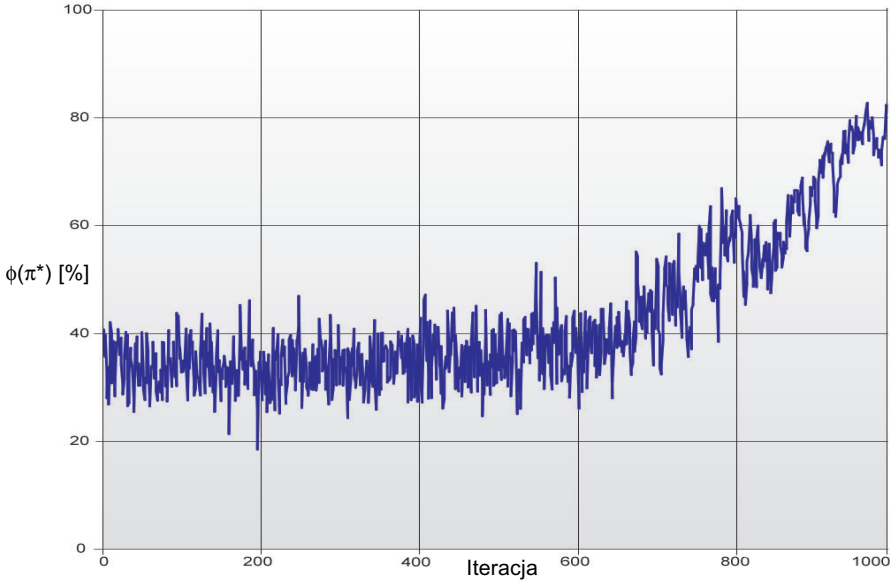
Tabela 3
Wyniki testów algorytmu *SATD* dla trzech instancji testowych uzyskanych poprzez losową generację paczek z rozkładem normalnym, w którym $\mu = 3$ oraz $\sigma^2 = 2$

Nr	*n*	*W*	*D*	*H*	*Algorytm*	$\phi(\pi^*)$
1	20	5	5	5	*LTD*	28,80
					ATD	31,20
					BTD	30,40
					CTD	18,40
2	60	8	8	8	*LTD*	16,60
					ATD	9,57
					BTD	15,04
					CTD	16,02
3	200	15	15	15	*LTD*	7,35
					ATD	5,78
					BTD	6,43
					CTD	5,36
4	125	5	5	5	*LTD*	20,80
					ATD	40,00
					BTD	24,80
					CTD	33,60
5	400	8	8	8	*LTD*	18,55
					ATD	17,97
					BTD	21,68
					CTD	12,70

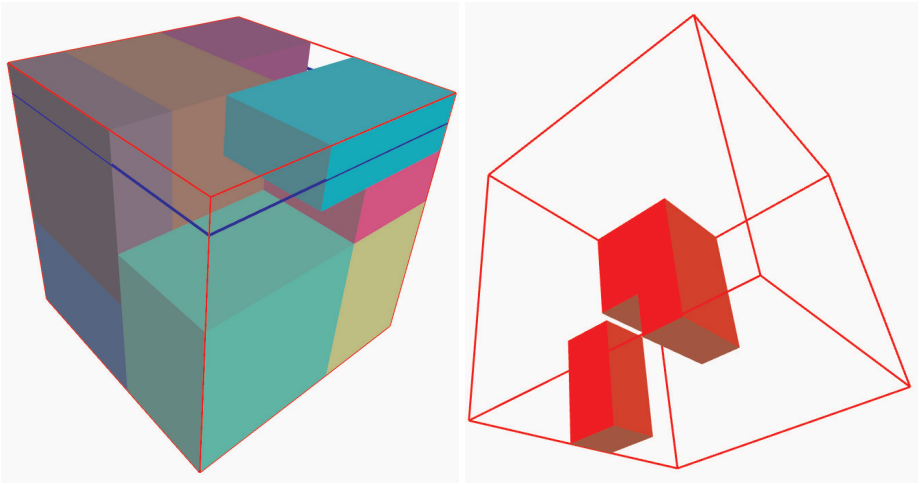


Rys. 3. Przykład pakowania za pomocą algorytmu *SATD* kontenera 5×5×5 dla $n = 125$ paczek uzyskanych poprzez ich losową generację rozkładem normalnym ($\mu = 3$, $\sigma^2 = 2$). Współczynnik wypełnienia $\phi(\pi^*) = 93\%$ (z prawej – paczki, z lewej – niewykorzystane miejsce w kontenerze)

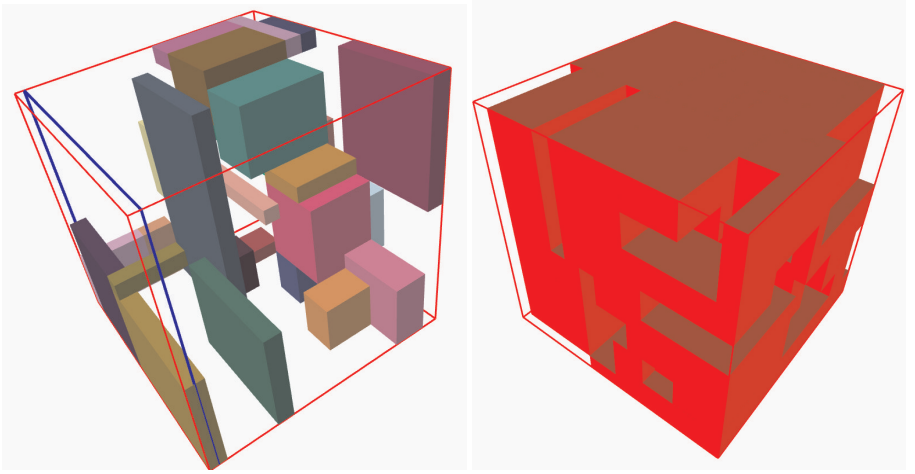
Na rysunkach 3, 5, 6 przedstawiono przykłady pakowania za pomocą algorytmu *SATD* kontenerów, a na rysunku 4 zilustrowano proces szukania najlepszego upakowania za pomocą algorytnu *SATD* kontenera 5×5×5.



Rys. 4. Ilustracja procesu szukania najlepszego upakowania za pomocą algorytmu *SATD* kontenera 5×5×5 dla $n = 125$



Rys. 5. Przykład pakowania za pomocą algorytmu *SATD* kontenera dla $n = 20$ paczek uzyskanych poprzez cięcie gilotynowe. Współczynnik wypełnienia $\phi(\pi^*) = 90\%$ kontenera 5×5×5 (z prawej – paczki, z lewej – niewykorzystane miejsce w kontenerze)



Rys. 6. Przykład pakowania za pomocą algorytmu *SATD* kontenera dla $n = 200$ paczek uzyskanych poprzez cięcie gilotynowe. Współczynnik wypełnienia $\phi(\pi^*) = 20\%$ kontenera 15×15×15 (z prawej – paczki, z lewej – niewykorzystane miejsce w kontenerze)

7. Podsumowanie

Zagadnienie trójwymiarowego pakowania, jest zagadnieniem *NP-trudym*, co jest źródłem problemów z wyznaczeniem wysokiej jakości rozwiązań w akceptowalnym czasie.

Liczność przestrzeni rozwiązań wykładniczo rośnie wraz ze wzrostem rozmiaru problemu. Uzyskane wyniki wskazują, że zastosowanie algorytmu symulowanego wyżarzania *STAD* pozwala na uzyskanie lepszych rezultatów, niż stosowanie prostych reguł konstrukcyjnych. Można również zauważyć, że niebagatelne znaczenie na uzyskane wyniki ma także rozkład wielkości paczek. Algorytm uzyskuje dobre wyniki, gdy instancja testowa zawiera paczki w przybliżeniu o jednakowej wielkości, co widać w tabeli 1. Dla niewielkiego kontenera (5×5×5), w którym paczki są o podobnym rozmiarze, średnia wartość wypełnienia wynosi 95%. Wyraźnie widać, że ze wzrostem rozmiaru kontenera wzrasta zróżnicowanie wymiarów paczek i algorytm ma coraz poważniejsze problemy z uzyskaniem akceptowalnej jakości rozwiązań. W przypadku kontenera (10×10×10) uzyskany średni współczynnik wypełnienia spada do 36%, a dla kontenera (15×15×15) to już zaledwie średnio 20%.

Tak niekorzystne rozwiązania mają swoje źródło w dwóch przyczynach. Pierwsza to właściwości instancji testowych – wzrost zróżnicowania paczek wraz ze wzrostem rozmiaru kontenera, a druga to sposób wyznaczania pozycji nowej paczki w kontenerze określony równaniami (8), (9), (10), co szczególnie widać na rysunku 6. Sposób ten prowadzi do dużego „marnotrawstwa" przestrzeni, gdy mamy do czynienia z instancją zawierającą „niejednorodne" paczki.

Nieco lepsze wyniki uzyskano dla instancji, w których paczki wygenerowano w sposób losowy, zgodnie z rozkładem normalnym (tab. 2). W tym przypadku, nadmiar paczek daje pewną swobodę wyboru i pozwala na uzyskanie lepszych wartości współczynnika wypełnienia (od średnio $\phi(\pi^*) = 77\%$ dla kontenera (5×5×5), do średnio $\phi(\pi^*) = 55\%$ dla kontenera (8×8×8)).

Należy też zwrócić uwagę na niską wydajność prostych algorytmów konstrukcyjnych, które nieco lepiej sobie radzą w przypadku, gdy mają do wyboru większą liczbę jednorodnych paczek (test 4 w tab. 3).

Pomimo, że formuła określona równaniami (8), (9), (10), pozwala na szybkie wyznaczanie pozycji nowej paczki w czasie $O(n^2)$, to w pewnych przypadkach prowadzi do nikłego wykorzystania przestrzeni kontenera. Należy wiec rozważyć zastosowanie nowej reprezentacji rozwiązania, który w większym stopniu uwzględni ich aktualny rozkład w kontenerze. W pracy [4] zaproponowano nową reprezentację rozwiązania w przestrzeni trójwymiarowej.

Literatura

[1] Aarts E.H.L., Verhoeven M., *Local search, in Annotated bibliographies in Combinatorial Optimization.* (Dell'Amico M., Maffioli F., Martello S., eds.), 1997 163–180.

[2] Ausiello G., Crescenzi P., Gambosi G., Kann V., Marchetti-Spaccamela A., Protasi M., *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. 1999.

[3] Chan F.T.S., Au K.C., Chan L.Y., Lau T.L., *Using genetic algorithms to solve quality-related bin packing problem*. Robotics and Computer-Integrated, 23, 1, 2007, 71–81.

[4] Filipowicz B., Chmiel W., Kadłuczka P., Wala K., *Kontur wypukły w trójwymiarowym zagadnieniu pakowania*. Automatyka (półrocznik AGH), t. 14, z. 3/2, 2010.

[5] Hochbaum D.H., ed., *Approximation Algorithms for NP-Hard problems*. PWS Publishing Company, r. 9, 1997.
[6] Lodi A., Martello S., Vigo D., *TSpack: A Unified Tabu Search Code for Multi-Dimensional Bin Packing Problems*. Annals of Operations Research, 131, 2004, 203–213.
[7] Matsumoto M., Nishimura T., *Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator*. ACM Trans. Model. Comput. Simul., 8, 3, 1998.
[8] Michalewicz Z., Fogel D.B., *How to solve it: Modern Heuristics*. Springer-Verlag, Berlin, Heidelberg, 2004.
[9] Miyazawa F.K., Wakabayashi Y., *Polynomial approximation algorithms for the orthogonal z oriented 3D packing problem*. Technical Report RT-MAC-9512, Instituto de Matematica e Estatica , Universidade de Sao Paulo, 1995.
[10] Pisinger D., *Heuristics for the container loading problem*. European Journal of Operational Research, 141, 2, 2002, 382–392.
[11] Pisinger D., Egeblad J., *Heuristic approaches for the two- and three-dimensional knapsack packing problem.* Computers & Operations Research, vol. 36, 4, 2009, 1026–1049.
[12] Scheithauer G., *A three dimensional bin packing algorithm*. Journal of Information Processing and Cybernetics, 27, 1991, 263–271.
[13] Vazirani V.V., *Approximation Algorithms*. Springer-Verlag, Berlin, 2003.
[14] Yamada T., Takeoka T., *An exact algorithm for the fixed-charge multiple knapsack problem*. European Journal of Operational Research, 192, 2, 2009, 700–705.