Projekt AAL – Strip packing problem

Dokumentacja

Filip Łęczycki

29.11.2014

# 1. Opis problemu

Dany jest zbiór prostopadłościanów {a1, … , an} oraz obszar prostopadłościenny o zadanych dwóch bokach nie mniejszych od największego z boków prostopadłościanów w zbiorze. Należy wyznaczyć, stosując różne metody heurystyczne, a także przeszukiwanie systematyczne, najmniejszą dlugość trzeciego boku obszaru pozwalającą na rozmieszczenie ortogonalne bez kolizji zbioru w obszarze. Porównać czas obliczeń i wyniki różnych metod.

Wprowadzam nazewnictwo:

Prostopadłościan należący do {a1,...,an} – paczka

Obaszar prostopadłościwenny o zadanych dwóch bokach - kosz

# 2. Proponowane rozwiązanie

Założenia ogólne:

* Paczka reprezentowana będzie przez obiekt zawierający pozycję lewego, dolnego, tylniego rogu paczki (x,y,z), jej wymiary (width,length,depth) oraz aktualny obrót (enum::rotation)
* Istnieje 6 możliwych obrotów paczki, ich kolejność zdefiniowana będzie za pomocą enumeracji
* Wszędzie gdzie mowa będzie o położeniu paczki należy rozumieć to jako położenie lewego, dolnego, tylniego rogu paczki
* Kosz reprezentowany będzie przez obiekt zawierający wymiary kosza(width, length, actual height) oraz zbiór zapakowanych paczek, domyślnie pusty. Lewy dolny tylni róg kosza znajduje się w punkcie (0,0,0)

## a. Algorytm naiwny

Rozwiązanie naiwne polegać będzie na zwykłym, sekwencyjnym pakowaniu paczek do kosza aż do momentu wyczerpania się zbioru niezapakowanych paczek. Paczki posortowane są wedlug malejącej objętości. Algorytm umieszcza największą paczkę w punkcie (0,0,0) tak aby wysokość była minimalna. Wysokość zostaje zapisana do zmiennej *actualHeight*. Następnie warstwa jest ‘obcinana’ a algorytm rozpoczyna pętlę od początku umieszczając największą paczkę w punkcie (0,0,*actualHeight*). Podejście to zapewni rozwiązanie niskiej jakości ale uzyskane dość szybko.

Szacowana złożoność:

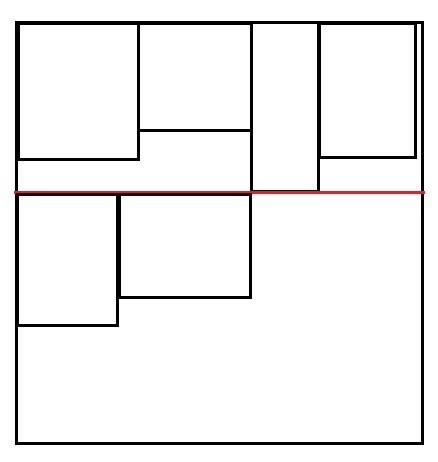
Jako że koszt umieszczenia paczki w koszu jest jednostkowy, złożoność wynosi O(n) = 6n

Opis w pseudokodzie:



## b. Algorytm warstwowy : półkowy

Algorytm dzieli kosz na warstwy każdą rozwiązując jako problem dwuwymiarowy. Dwuwymiarową reprezentacją paczki w tym algorytmie jest jej górna powierzchnia przy obrocie paczki w taki sposób, że jej wysokość jest najmniejsza. Wykorzystany został algorytm półkowy tzn. Paczki umieszczane są na warstwie rzędami o szerokości równej szerokości najszerszej paczki(ilustracja). Orientacja rzędów zależy od tego czy kosz jest szerszy czy dłuższy. Gdy żadna paczka nie zmieści się już na daną warstwę, jest ona obcinana, wysokość zostaje zwiększona i rozpoczyna się pakowanie kolejnej warstwy. Procedura jest powtarzana do momentu umieszczenia wszystkich paczek w koszu.



*Podział warstwy na półki*

Szacowana złożoność:

Mamy N paczek, zakładając pesymistycznie że na warstwę mieści się tylko jedna paczka ale sprawdzamy każdą z nich przy każdej warstwie. Dodatkowo na początku dla każdej paczki sprawdzamy każdy z 6 obrotów szukając optymalnego. Koszt umieszczenia paczki w koszu ciągle jest jednostkowy dlatego szacowana złożoność wynosi O(n) = 6n + n^2 ~ n^2

Opis w pseudokodzie:

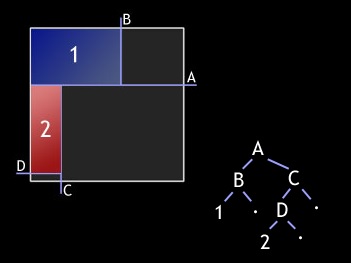


## c. Algorytm warstwowy z wykorzystaniem drzewa binarnego

Algorytm dzieli kosz na warstwy każdą rozwiązując jako problem dwuwymiarowy. Dwuwymiarową reprezentacją paczki w tym algorytmie jest jej największa powierzchnia boczna. Warstwa reprezentowana jest jako drzewo wolnej przestrzeni. Początkowo jest tylko jeden węzeł, reprezentujący całą powierzchnię warstwy. Algorytm bierze pierwszą paczkę ze zbioru, jeżeli mieści się do węzła w ten sposób, że nie zajmuje całej jego powierzchni, do węzła dodawane są dwa liście,dzielące dostępną powierzchnię na dwie części pionowo lub poziomo w zależności od rozmiarów paczki. Następnie podejmowana jest próba umieszczenia paczki w każdym z liści. Procedura jest powtarzana do momentu gdy paczka idealnie dopasuje się do miejsca w liściu lub wcale nie zmieści się na warstwę. Gdy warstwa jest zapełniona, następuje jej obcięcie, wyczyszczenie drzewa i następuje kolejny przebieg pętli. Dzieje się tak do momentu umieszczenia wszystkich paczek w koszu.

Szacowana złożoność:

Podobnie jak przy poprzednim algorytmie, szacowana złożonośc wynosi O(n^2)



*Podział przestrzeni*

Opis w pseudokodzie:

## Podsumowanie

Najlepsze wyniki przy rozsądnym czasie obliczeń zapewnia niewątpliwie algorytm ‘drzewiasty’. Jednak przy odpowiedniej strukturze danych wejściowych ( duża powierzchnia kosza w stosunku do wymiarów paczek) i przy dużej liczbie paczek (~10 000) algorytm trzeci uzykuje wyniki jedynie o 30% lepsze od algorytmu drugiego, przy dwukrotnie dłuższym czasie wykonania. Algorytm pierwszy daje zadowalające wyniki jedynie w przypadku układania paczek zbliżonych rozmiarami do kosza.