

Sprawozdanie

Temat: Pakowanie palet

Opis

Głównym założeniem algorytmu jest sprawdzenie jak największej ilości kombinacji, dlatego też sam w sobie jest bardzo prosty.

W pierwszej kolejności sortuje paczki zgodnie z malejącą objętością, następnie przydziela każdą z nich do pierwszego nośnika, na którym można ją zmieścić. Jeżeli taki nośnik nie istnieje, tworzy nowy.

Następnie wykonuje 9 iteracji, każda trwająca 100ms. W obrębie każdej iteracji losuje 12 – numer iteracji paczek, zdejmując je z nośników i przydziela na nowo w losowej kolejności, co powtarzane jest dopóki nie skończy się czas przeznaczony na iterację.

Algorytm nie został w pełni zaimplementowany (po znalezieniu nowego optimum nie jest zachowywany stan przetwarzania), jednak póki co działa jednowątkowo, więc finalna wydajność powinna być i tak dużo większa (obecnie udaje się każdorazowo przeprowadzić około $1.5 \cdot 10^7$ iteracji).

Pseudokod

```
in
paczki : List of Paczka
palety : List of Paleta

var
nośniki : List of Nośnik
wybranePaczki : Set of Paczka

out
najlepszyWynik : Wynik

paczki.sort();
foreach p in paczki
    nośniki.add(p)

for it = 1..9
    dopóki czas < 100 ms
        for p = 1..(12-it)
            wybranePaczki.add(losuj(paczki\wybranePaczki))
            nośniki.remove(wybranePaczki)
            foreach p in random(wybranePaczki)
                nośniki.add(p)
            wybranePaczki.clear()
            jeżeli nośniki.wynik > najlepszyWynik
                najlepszyWynik = wynik
```

Wyniki

Instancja	Minimum		Mediana		Maksimum	
pp101.in	576	1200	576	1200	576	1200
pp102.in	720	240	720	240	720	240
pp103.in	384	480	384	480	384	480
pp104.in	240	240	240	240	240	240
pp105.in	192	480	192	480	192	480
pp106.in	192	720	192	720	192	720
pp107.in	144	3960	144	3960	144	3960
pp108.in	288	240	288	240	288	240
pp109.in	384	720	432	720	432	720
pp110.in	336	480	336	480	336	480

Specyfikacja środowiska:

Procesor: Intel Xeon 1230v3

Pamięć RAM: 16 GB

Java: 1.7.0_40-b43

Wykorzystywane rdzenie: 1 rdzeń