Projektowanie efektywnych algorytmów

Autor:
Tymon Tobolski (181037)

Prowadzący:
Mgr inż. Karolina Mokrzysz

Wydział Elektroniki
III rok
Pn TP 11.15 - 1.00

14 listopada 2011

1 Opis problemu

Dla podanej listy zadań znaleźć najbardziej optymalne uszeregowanie, minimalizujące opóźnienia.

Każde zadanie posiada określony czas wykonania (p), czas w którym powinno być wykonane (d) oraz wagę (w). W przypadku gdy zadanie jest zostało wykonane do czasu d naliczane jest opóźnienie. Celem algorytmu jest takie uszeregowanie zadań, aby suma kosztów opóźnień wszystkich zadań była jak najmniejsza. Koszt opóźnienia zadania wyraża się wzorem:
$l(t) = w * max(0, t - d)$

2 Opis algorytmu

Dla opisanego problemu zostały sprawdzone trzy wersje algorytmu szeregującego.

2.1 Przegląd zupełny

Przegląd zupełny polega na sprawdzeniu każdej możliwej permutacji list zadań, policzeniu koszty każdej z nich, a następnie wybranie tej o najmniejszym koszcie.

2.2 Pierwsza procedura eliminacyjna

Jednym ze sposobów ulepszenia przeglądu zupełnego jest obliczanie na bieżąco kosztu rozwiązania częściowego i porównanie go z dotychczasowym najmniejszym kosztem. W przypadku gdy jego koszt jest większy niż najmniejszy obecny, rozwiązanie takie zostaje odrzucone wraz ze wszystkimi rozwiązaniami, które powstały by w przypadku kontunuacji permutacji. W przypadku gdy koszt jest mniejszy następuje dalsze sprawdzanie rozwiązań, a koszt tego rozwiązania częściowego zostaje zapisany jako najmniejszy obecny.

2.3 Druga procedura eliminacyjna

Inna metodą poprawy działania algorytmu jest "usinanie" drzewa rozwiązań w przypadku gdy po zamianie miejscami 2 zadań kosz nowopowstałego uszeregowania jest mniejszy niż przed zamianą. Dla danego częściowego rozwiązania algorytm zakłada zamianę ostatniego zadania z każdym poprzed-

nim. Np. dla [A,B,C,D] algorytm sprawdza koszt uszeregowań [A,B,D,C], [A,D,B,C] oraz [D,A,B,C].

3 Środowisko testowe

Program testowy został napisany w języku Haskell przy pomocy kompilatora GHC w wersji 7.0.3.

Testy zostały przeprowadzone na komputerze o poniższych parametrach:

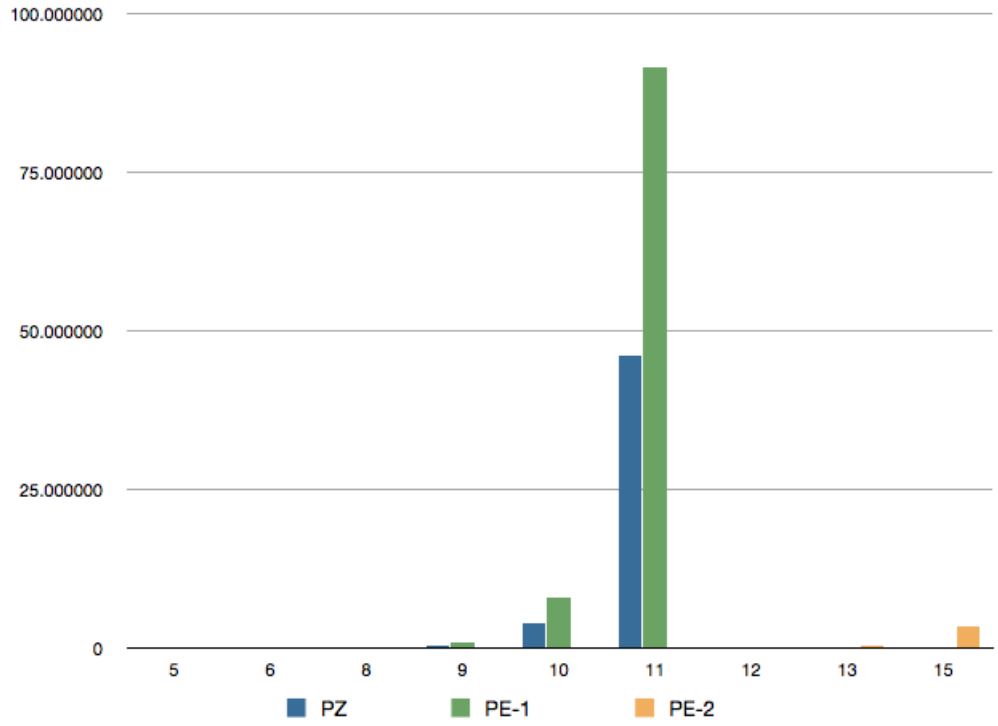
- System operacyjny: Max OS X 10.7.1
- Procesor: 2.4 GHz Intel Core 2 Duo
- Pamięc RAM: 8 GB 1067 MHz

4 Wyniki

Uśredniony czas (w sekundach) wykonania jednego szeregowania w zależności od ilości zadań N:

N	PZ	PE-1	PE-2
5	0.006454	0.006206	0.007205
6	0.006537	0.008704	0.005996
8	0.050714	0.080831	0.007391
9	0.389817	0.728662	0.011458
10	3.836483	7.951507	0.020667
11	45.928979	91.460791	0.055901
12	-	-	0.171354
13	-	-	0.438758
15	-	-	3.276497

Puste miejsca oznaczają zbyt długi czas wykonania.



Rysunek 1: Czas potrzebny do wyznaczenia optymalnego uszeregowania

5 Wnioski

Po przeprowadzeniu analizy wyników można jednoznaczenie stwierdzić, iż druga procedura eliminacyjna jest nieporównywalnie szybsza od pierwszej jak i od przeglądu zupełnego. Wykres obrazuje jak duża jest różnica w czasie działania. Niewątpliwie wpływ na to ma wczesne odrzucanie gałęzi rozwiązań zastosowane w procedurze drugiej, które pozwala znacznie zmniejszyć ilość potrzebnych obliczeń.

Po wykonaniu profilowania kodu okazało się, że najwięcej czasu (ok 65-70%) przeznaczone jest na obliczanie kosztu uszeregowania. Ma to szczególnie wpływ na pierwsza procedure eliminacyjna, która sprawdza koszt o wiele więcej razy niż przegląd zupełny, dlatego też w tym przypadku jest ona wolniejsza.