

MI-PAA 2015 2.ukol

Tomas Nesrovnal
nesrotom@fit.cvut.cz

December 7, 2015

1 Specifikace ulohy

Problem 0-1 batohu.

2 Rozbor moznych variant reseni

Ulohu muzu resit hrubou silou. Ziskam tam presny vysledek, ale vypocet bude pomaly. Dalsim resenim je pouzit heuristiku, jejiz vyledek nebude nejlepsi mozny ale vypocet probehne rychle.

3 Ramcovy popis postupu reseni

3.1 Hrubá síla

Zkusim vsechny moznosti a vyberu tu nejlepsi.

3.2 Heuristika

Vkladam do bahothu nejlepsi predmety s pomerem cena/vaha, dokud mi jeste staci kapacita.

4 Popis kostry algoritmu

4.1 Hrubá síla

Vytvorim pole, ktere udava ktery predmet je v batohu. Rekurzivne zkousim vsechny moznosti (zavolam rekurzi bez prvku, pak prvek pridam a zavolam rekurzi znovu). Ulozim si nejlepsi reseni.

Druha varianta obsahuje vylepseni: Pokud ve stromu reseni narazim na to, ze se do batohu uz vic nevejde, vetev zariznu.

4.2 Heuristika

Seradim si pole s predmety podle pomeru cena/vaha (nebo jine varianty, viz grafy). Cele pole sestupne prochazim a pokud se tam predmet vejde, tak ho tam vlozim.

5 Namerene vysledky

5.1 Spravnost vysledku

Pomoci skriptu byla overena spravnost vysledku (porovnanim s referencnim resenim). U method s nepresnym vysledem byla zaznamenana maximalni a relativni chyba.

5.2 Na cem bylo mereno

Intel(R) Core(TM) i3-2328M Processor (3M Cache, 2.20 GHz), gcc 4.9.2 (-Ofast), OS GNU/Linux Ubuntu 14.04 64bit

6 Branch and Bound

Orezaval jsem ze shora i z dola. Pred tim jsem pole seradil jako v heuristice, aby bylo orezavani efektivnejsi.

7 Dynamicke programovani

Implementoval jsem dva zpusoby dymanickeho programovani. Jeden zpusob plnil tabulku od spoda nahoru (dva cykly v sobe). Druhy resil odshora dolu (rekurze). Obe reseni vracely spravne vysledky, ale prvni zpusob byl mnohem rychlejsi.

Obe tyto reseni meli dekompozici podle vahy.

8 FPTAS

FPTAS je podobny heuristice. Nejdriv ale spocitam maximalni chybu a podle toho a podle nastavene mozne chyby epsilon preskaluji vstupni data.

Pro chybu $\epsilon=0$ byly tedy výsledky správné, stejně jako u dynamického programování.

Složitost FPTAS se odvíjí od přednastavené chyby: $O(n^3/\epsilon)$

9 Grafy

Grafy představují lineární funkce vytvořené z diskrétních hodnot.

Figure 1: Doba vypočtu

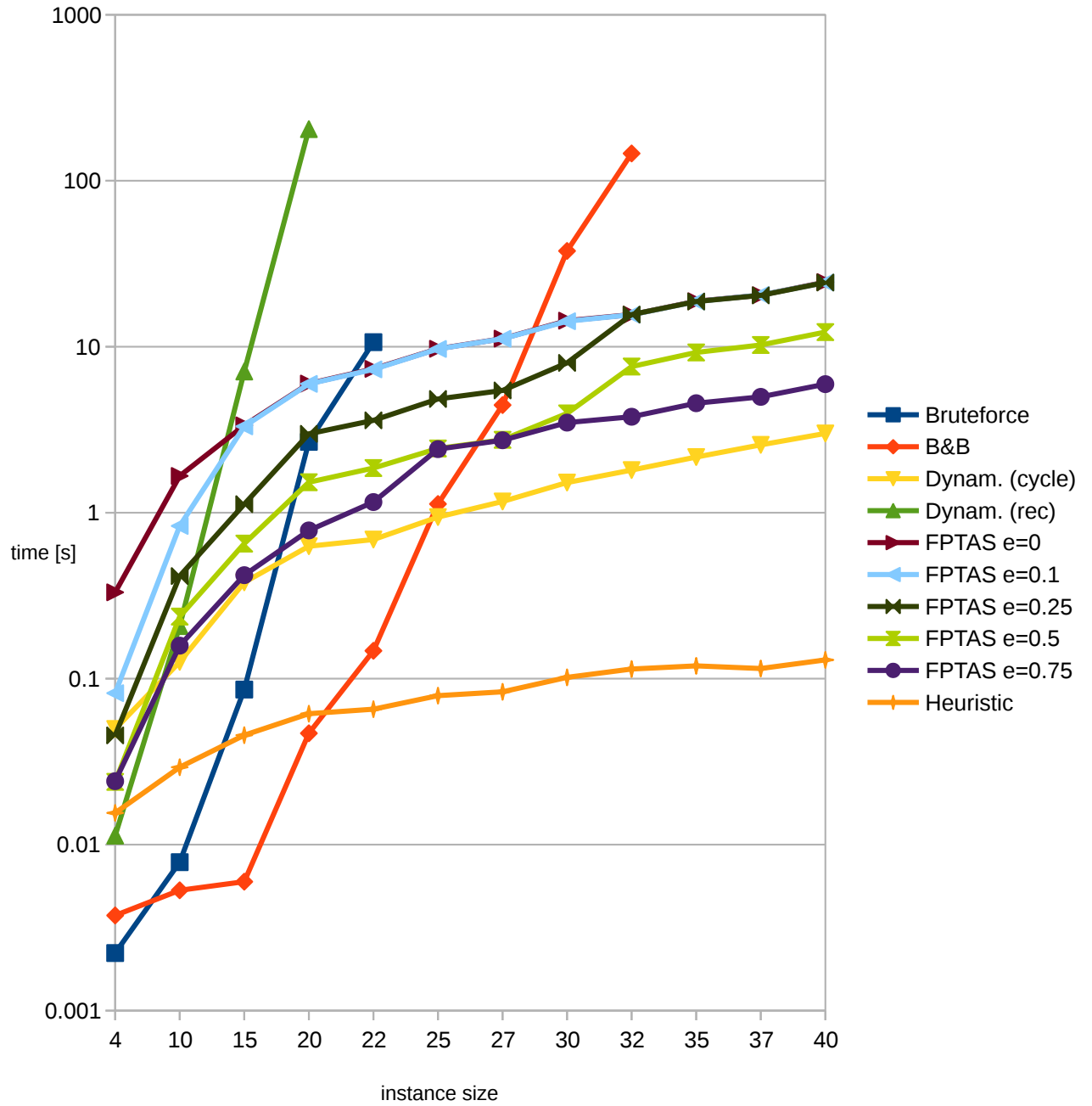
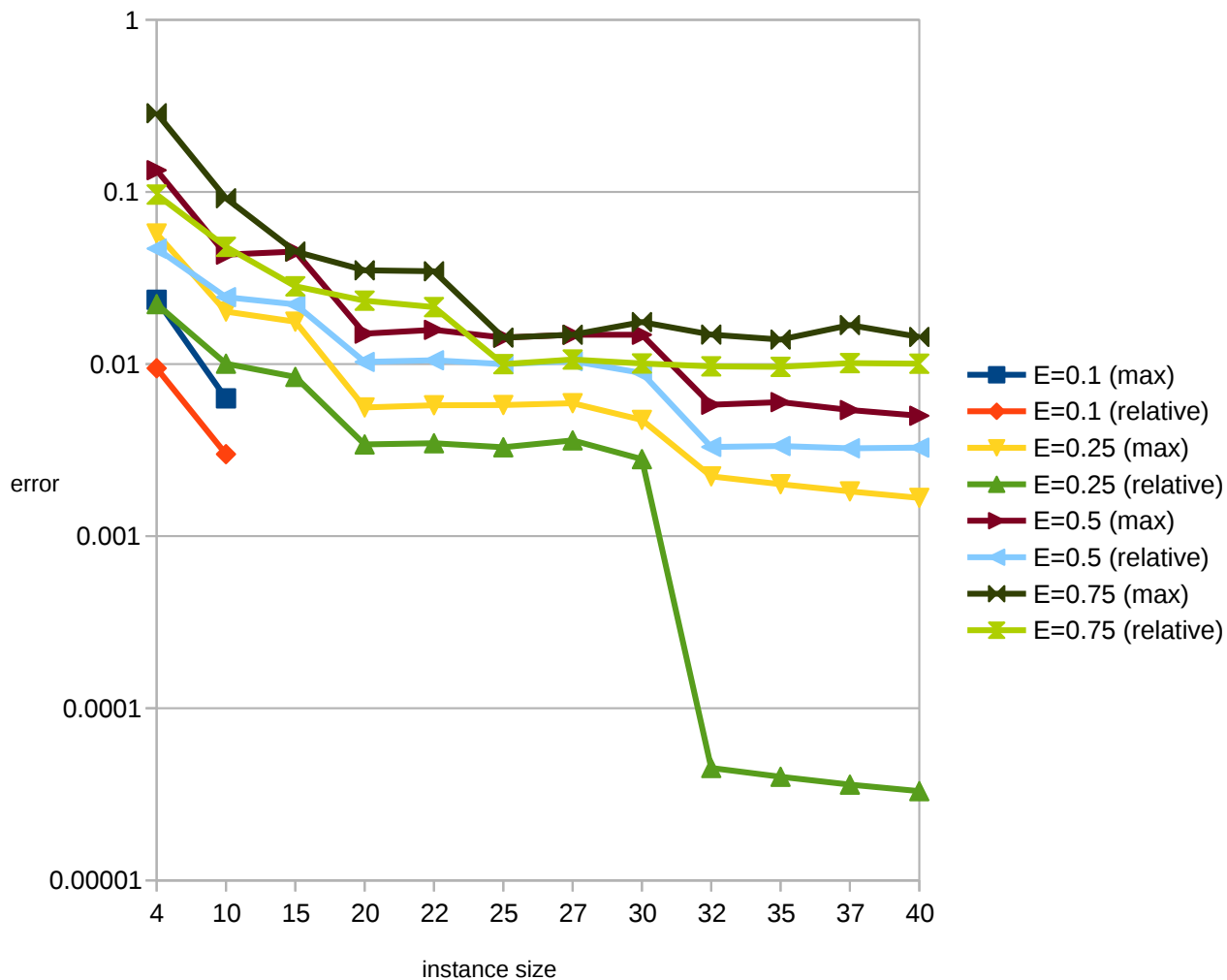


Figure 2: Maximalni a relativni chyba FPTAS



10 Zaver

BB (optimalizovana v C) vyrazne pomohla, ale je hodne citliva na vstupni data. Ovsem asymptoticka slozitosť zustava $O(2^n)$, protoze B&B nezlepsí nejhorsí prípad.

Dynamicke programovani je lepsi (dle namerenych udaju) provadet odspoda a je lepsi se vyhnout rekurzi.

FPTAS pocita s chybou, ale pri dobre zvolenem eps mame malou chybu a rychle reseni.

10.1 Zhodnoceni

Ocekavene vysledky potvrdilo mereni. Metoda B&B se ukazala byt velice dobra pro maly pocet dat. FPTAS se ukazal byt rychly a relativne presny, ale klasicke dynamicke programovani bylo stale lepsi. Heuristika je stale nejrychlejsi, ale její vysledky jsou radove min presne nez u ostatnich metod.