# MI-PAA 2015 1. a 2.ukol

Tomas Nesrovnal
nesrotom@fit.cvut.cz

November 10, 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 Hruba sila

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 Hruba sila

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).

### 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 Lubuntu 14.04 64bit

### 5.3 Grafy

Vsechny grafy byly vygenerovany skriptem (error.sh). Cas byl meren pomoci knihovny OpenMPI. Sript byl spusten pres prikaz **sudo time nice -n -20 ./error.sh** a trval 180 sekund.

Presna cisla lze nalezt v ***.plot** souborech.

Chyba heuristiky: Z kazdeho batohu spocitana relativni chyba. V grafu jsou pak secteny relativni chyby pro celou sadu. Celkem 3 heuristiky.

Figure 1: Doba behu reseni heuristikou (cena/vaha). 5000 opakovani.

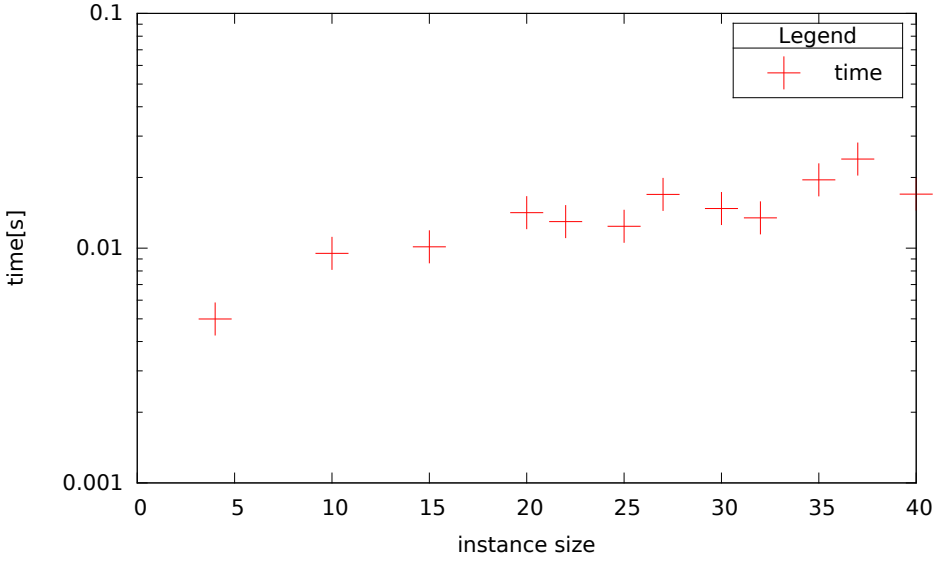



Figure 2: Doba behu reseni hrubou silou. 5 opakovani.

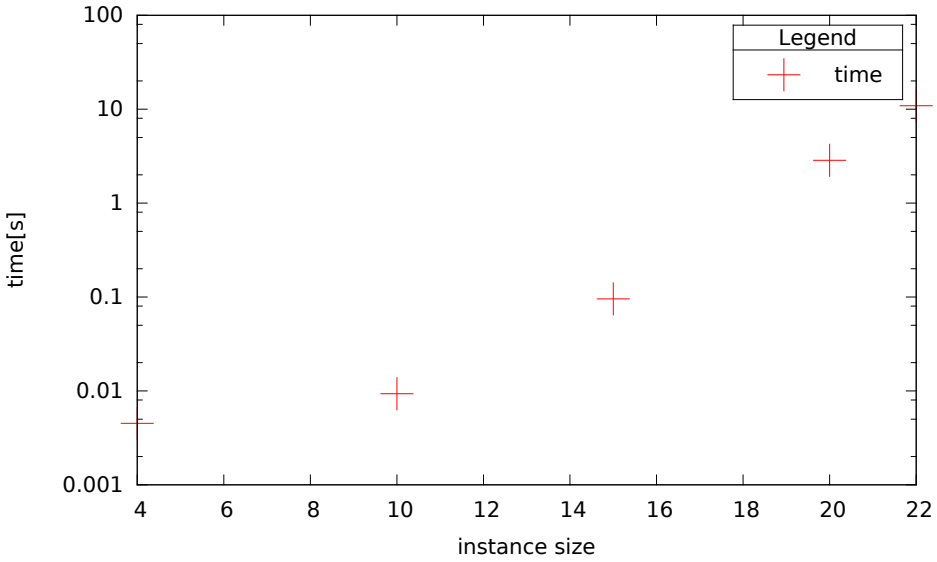


Figure 3: Doba behu reseni hrubou silou (s orezavanim). 5 opakovani.

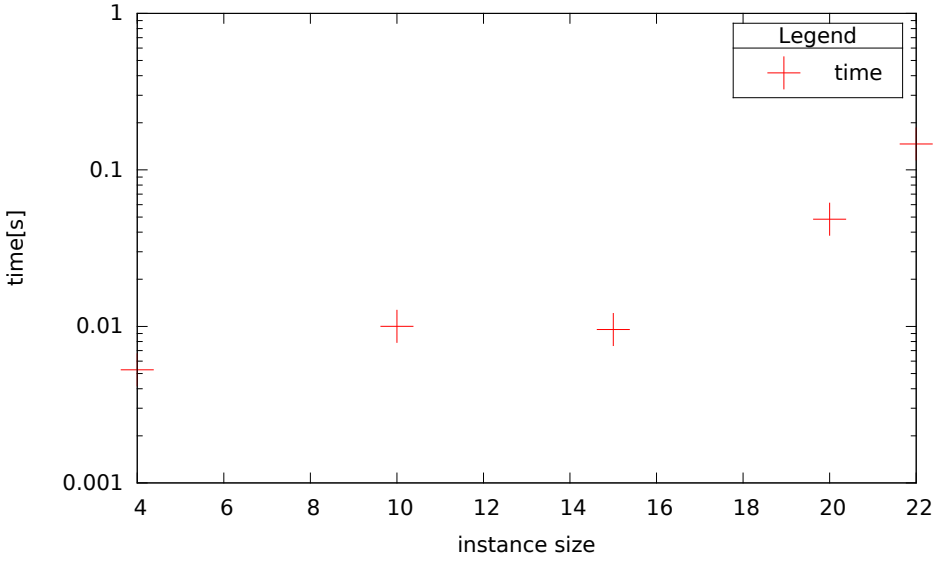



Figure 4: heuristika (podle ceho se radilo): pomer cena/vaha

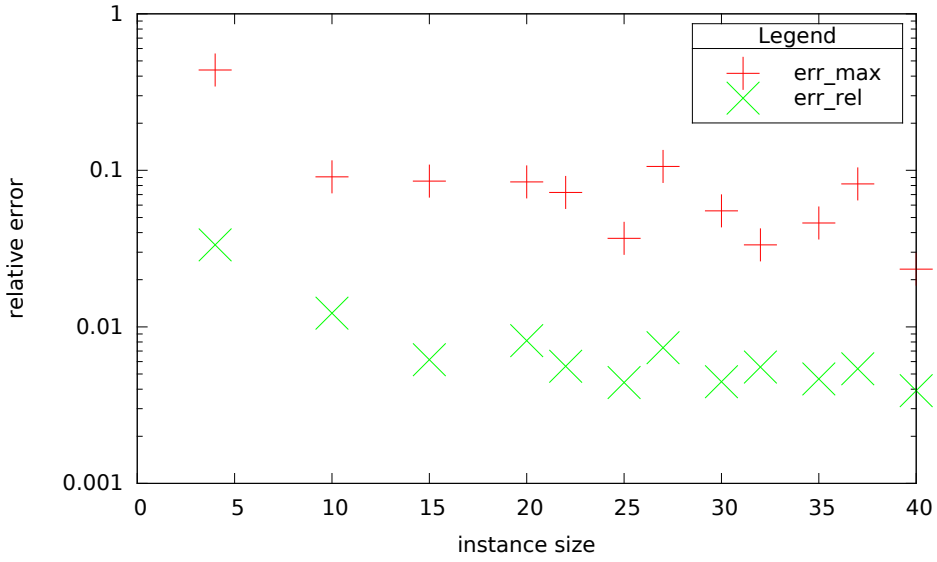


Figure 5: heuristika (podle ceho se radilo): cena

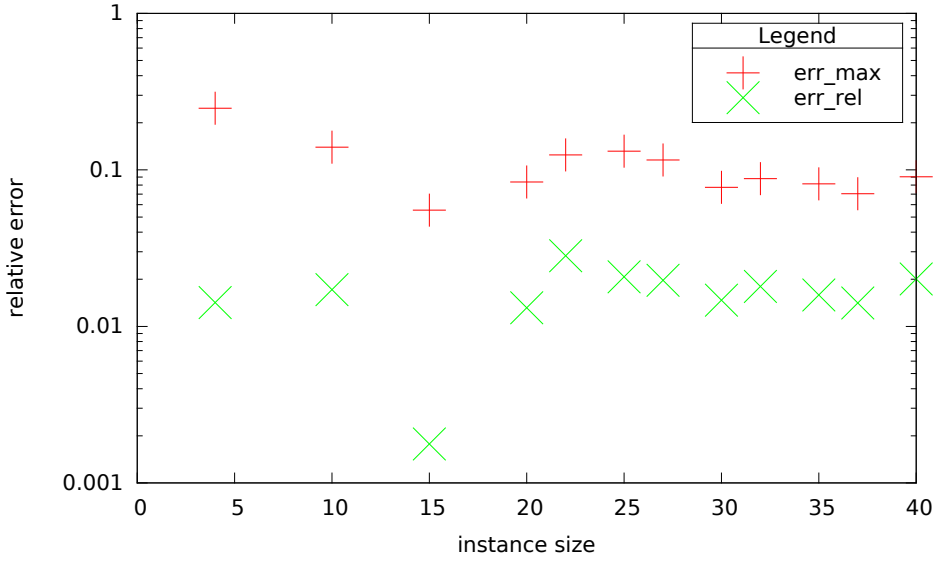
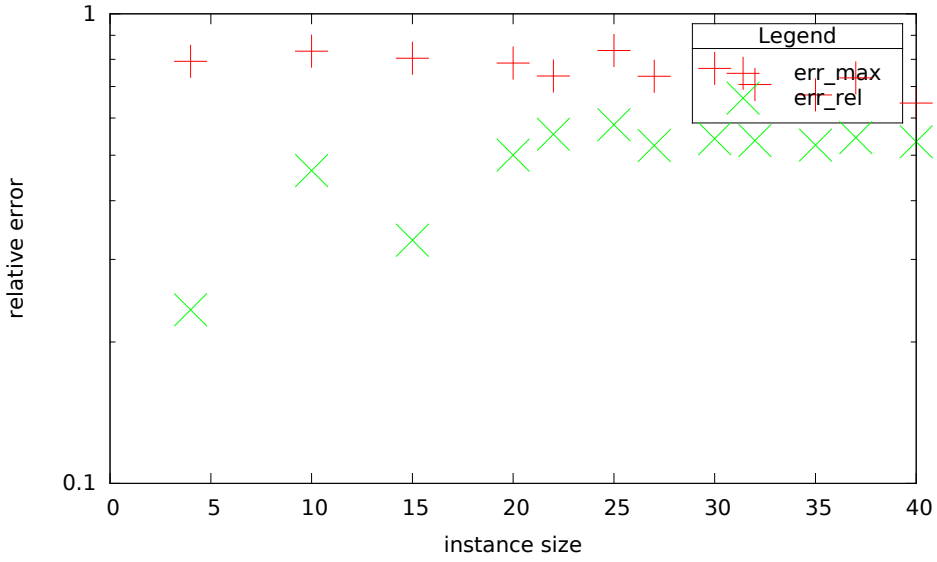



Figure 6: heuristika (podle ceho se radilo): vaha




# 6 Zaver 1. uloha

Vysledky se shoduji s rozborem reseni. Hruba sila je opravdu pomala a byt jednoducha heuristika nevraci zas tak spatne reseni.

Heuristika cena/vaha byla nejlepsi z heuristik.

Orezavani bruteforce melo za nasledek zrychleni o 12 sekund u instance o velikosti 25.

Bruteforce ma slozitost $O(2^n)$, zatimco heuristika jen $O(nlogn)$. Z tohoto duvodu bylo mereni na heuristice opakovano 50000x, zatimco na bruteforce pouze 5x.

Jeste nutno podotknout, ze na takhle malych datech (navic asi nahodne vygenerovanych) muzou byt velke odchylky a anomalie.

# 7 2. uloha

# 8 Branch and Bound

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

# 9 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.

# 10 FPTAS

FPTAS je podobny heuristice. Nejdriv ale spocitam maximalni chybu a podle toho a podle nastavene mozne chyby epsilon preskaluji vstupni data. Pro chybu eps=0 byly tedy vysledky spravne, stejne jako u dynamickeho programovani.

# 11 Zaver 2. uloha

BB vyrazne pomohla, ale je hodne citliva na vstupni data.

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

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

# 12 Grafy

Nasleduje kompletni vycet grafu, kde time ukazuje rychlost vypoctu (pro pomale alrogritmy nebyly pouzite velke instance). "err" ukazuje chybu. f znamena algoritmus FPTAS a cislo za nim je nastavene eps v procentech. d je dynamicke programovani, 1 je shora dolu, 2, z dola nahoru. b je bruteforce. b0 je neoptimalizovane, b1 s BB. h je heuristika jeste z prvni ulohy.

Figure 7: time_f0.pdf

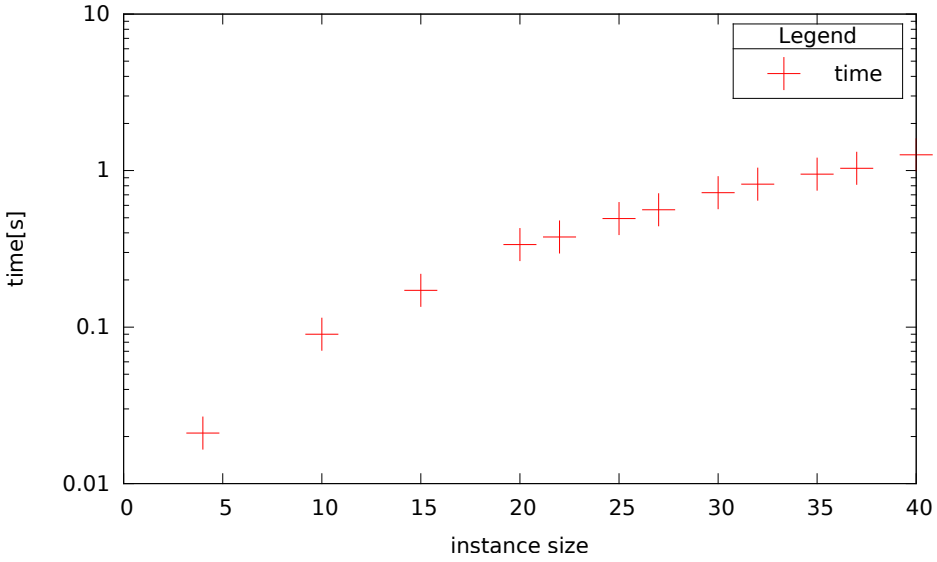


Figure 8: time_f25.pdf

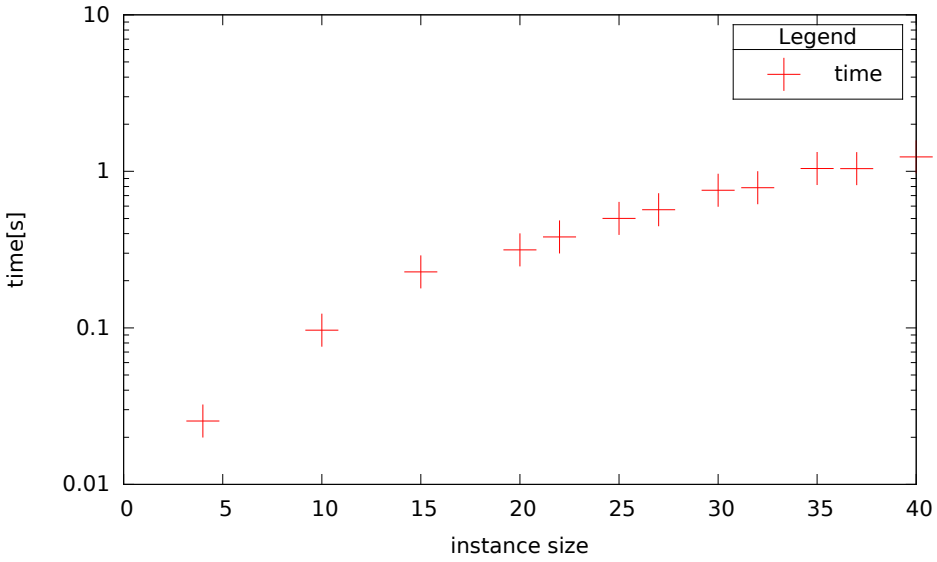



Figure 9: time_f50.pdf

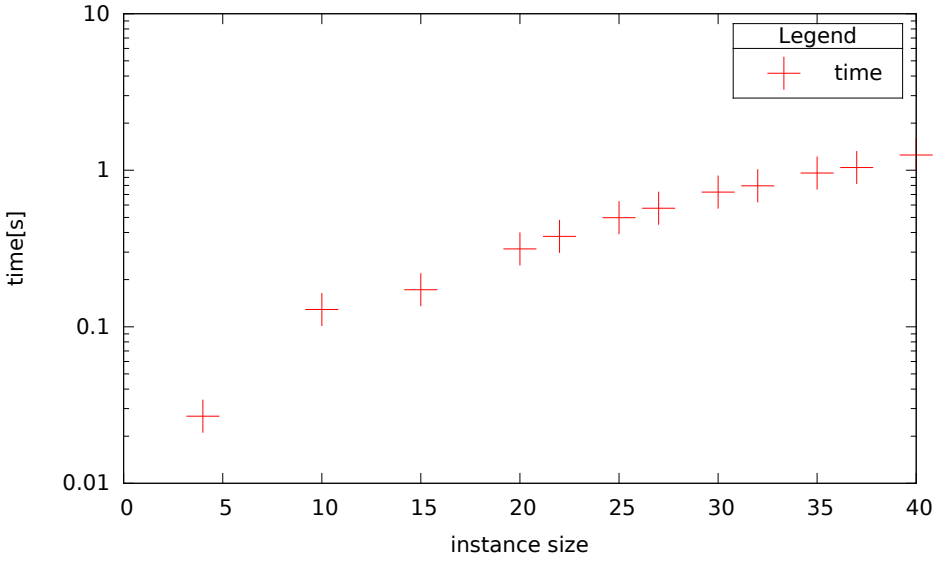


Figure 10: time_f75.pdf

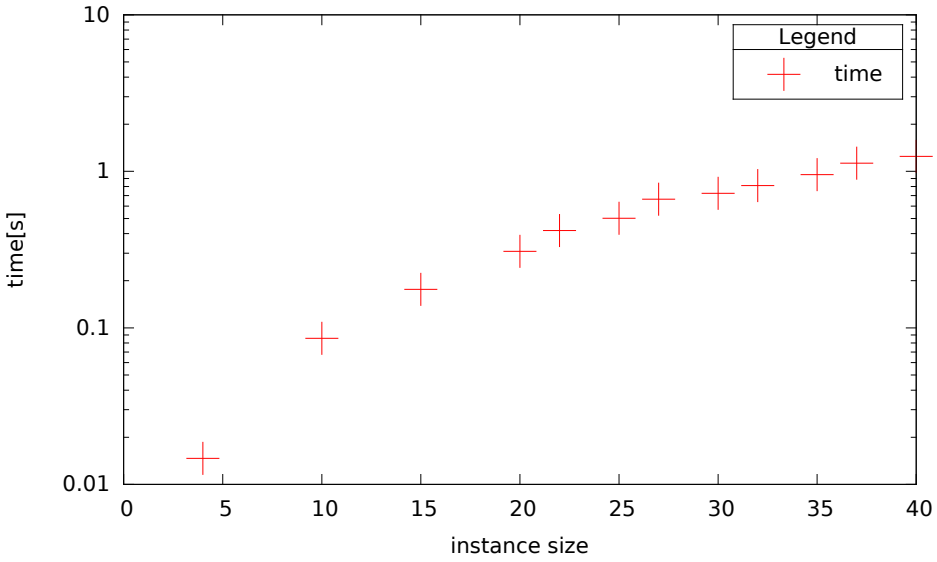



Figure 11: time_f100.pdf

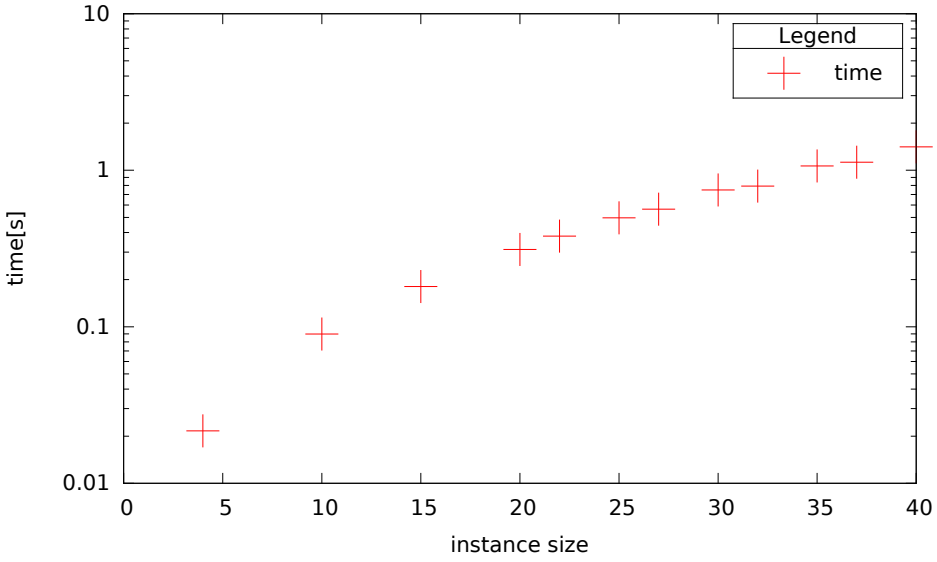


Figure 12: time_d1.pdf

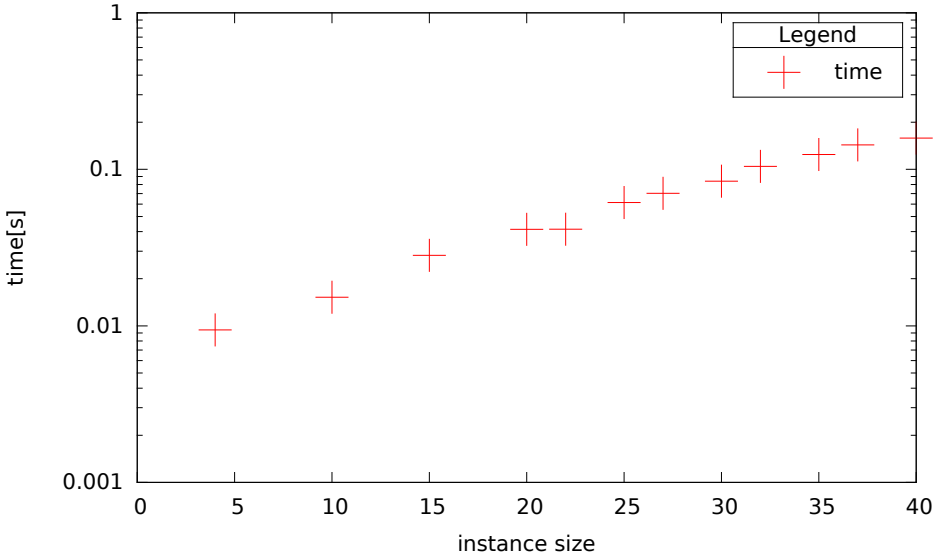



Figure 13: time_d2.pdf

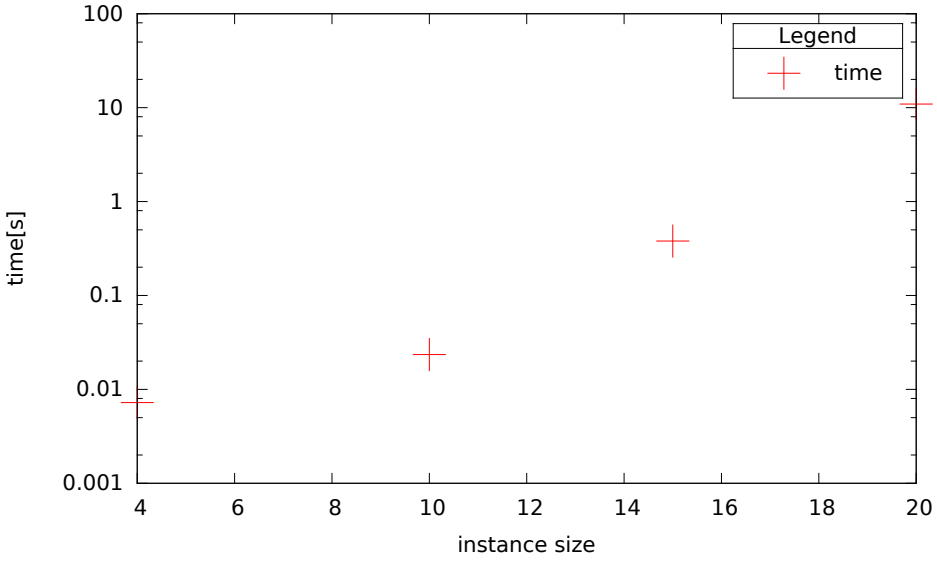


Figure 14: time_b0.pdf

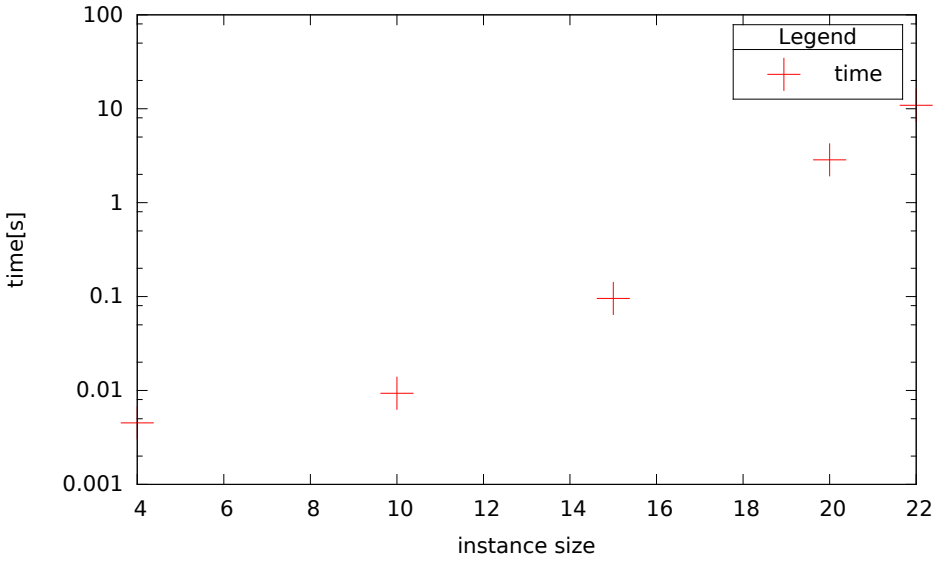



Figure 15: time_b1.pdf

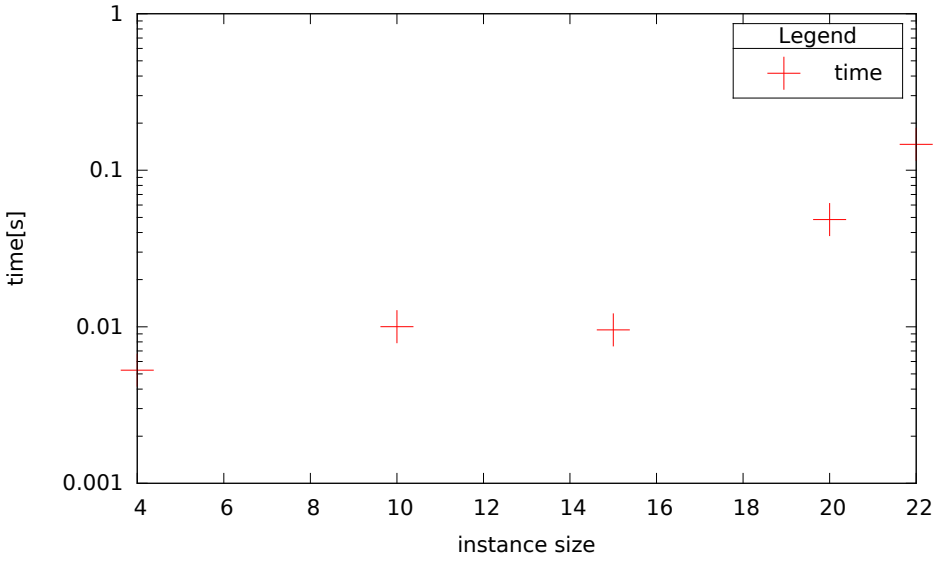


Figure 16: time_h1.pdf

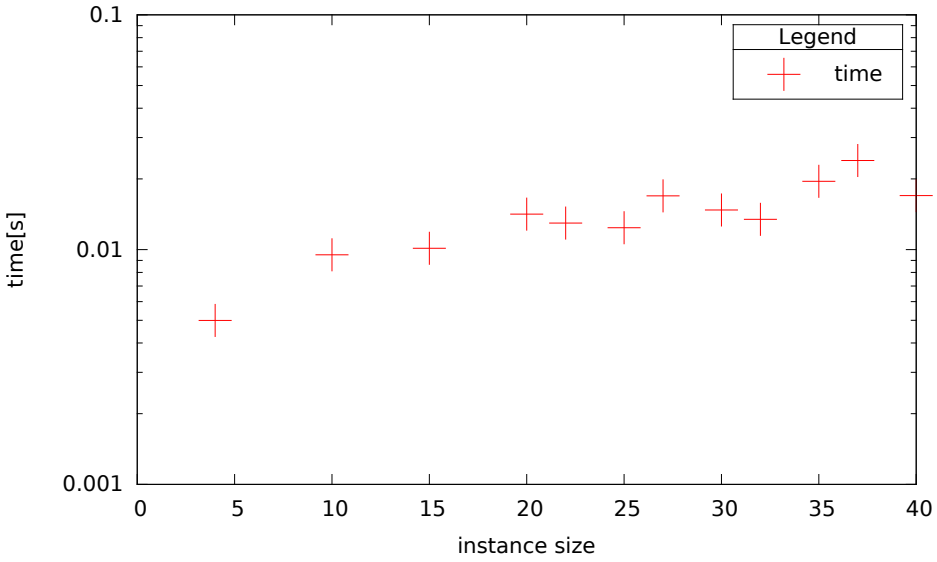



Figure 17: time_h2.pdf

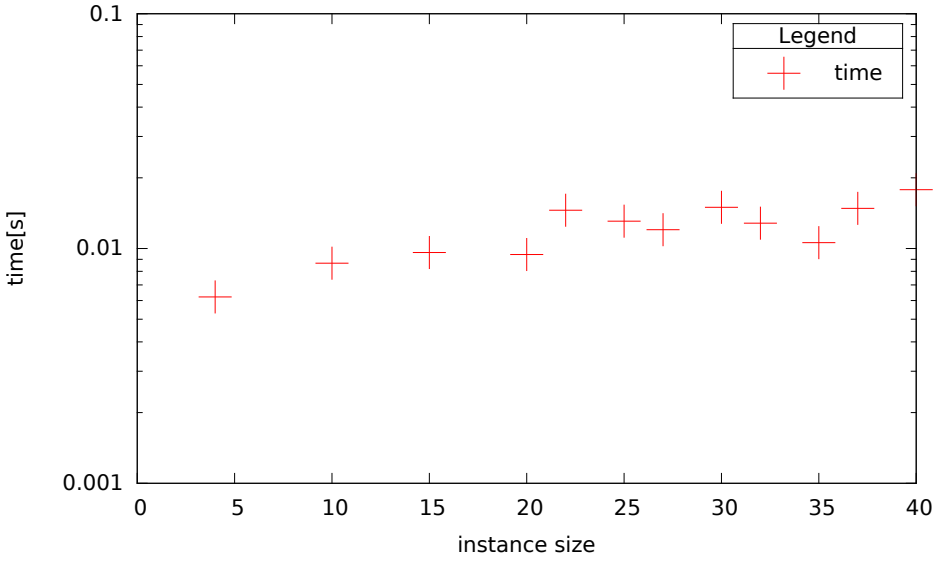


Figure 18: time_h3.pdf

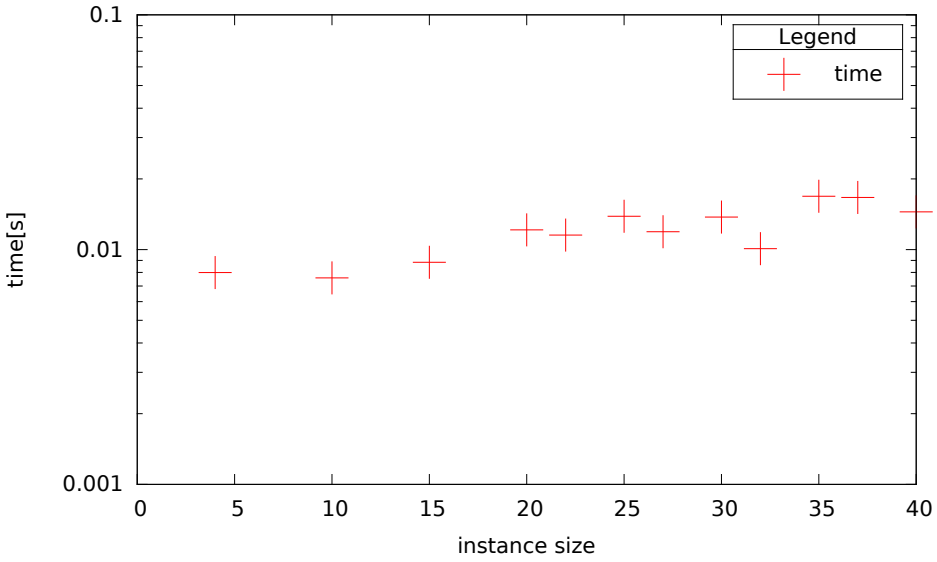



Figure 19: err_f25.pdf

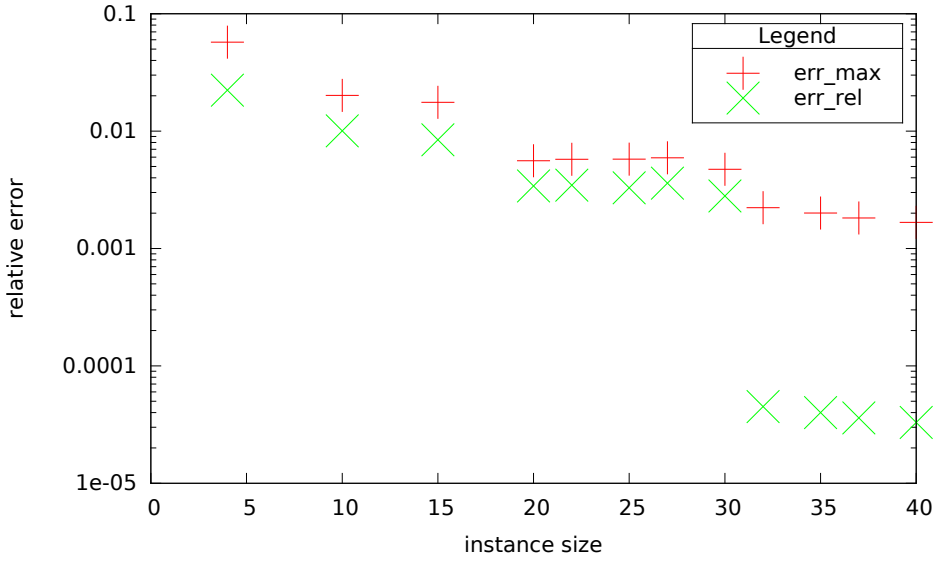


Figure 20: err_f50.pdf

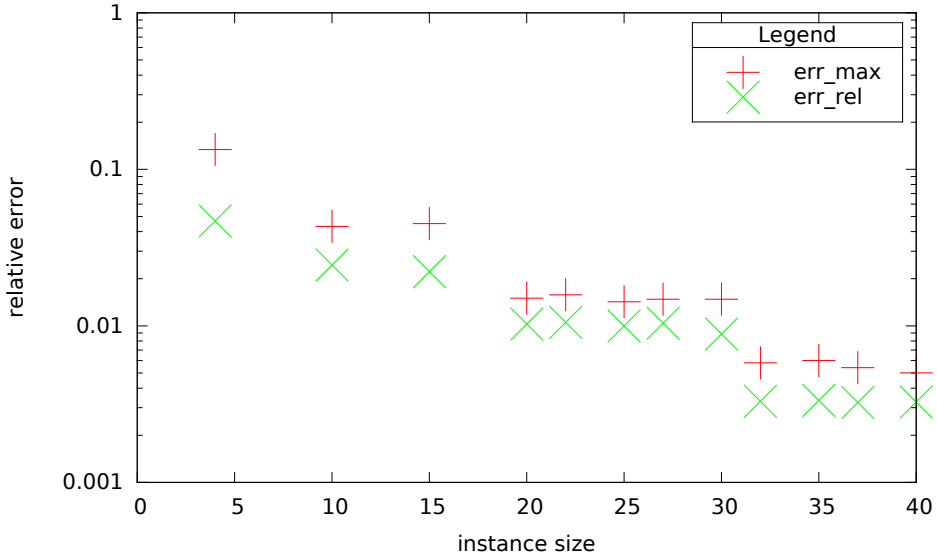



Figure 21: err_f75.pdf

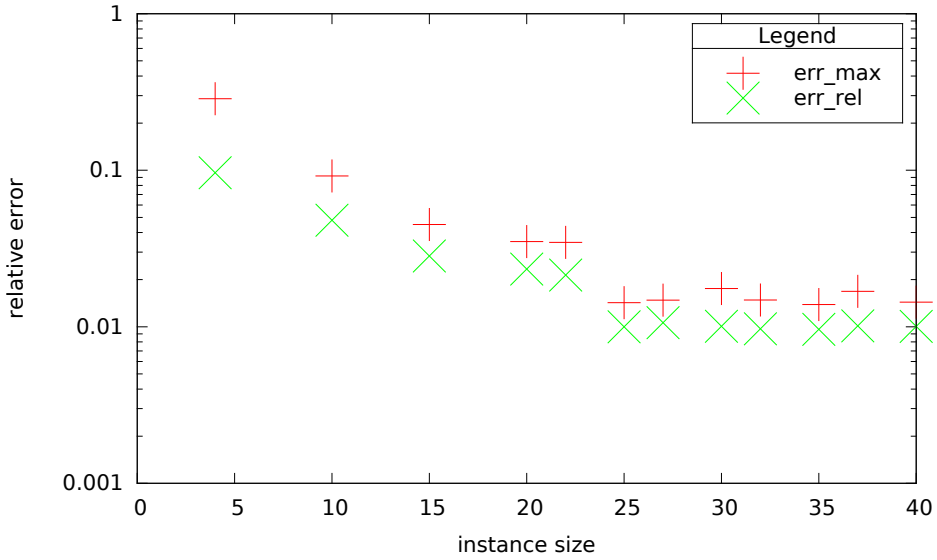


Figure 22: err_f100.pdf

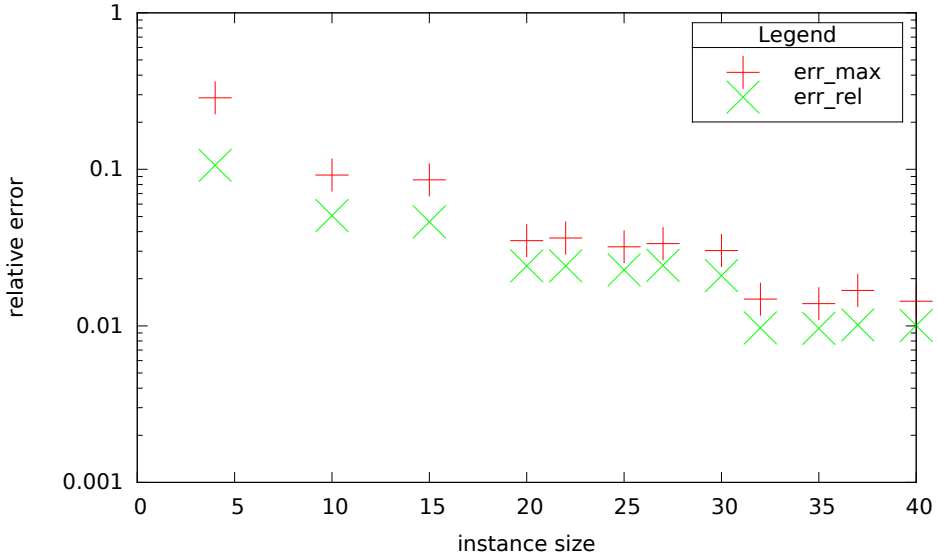



Figure 23: err_h1.pdf

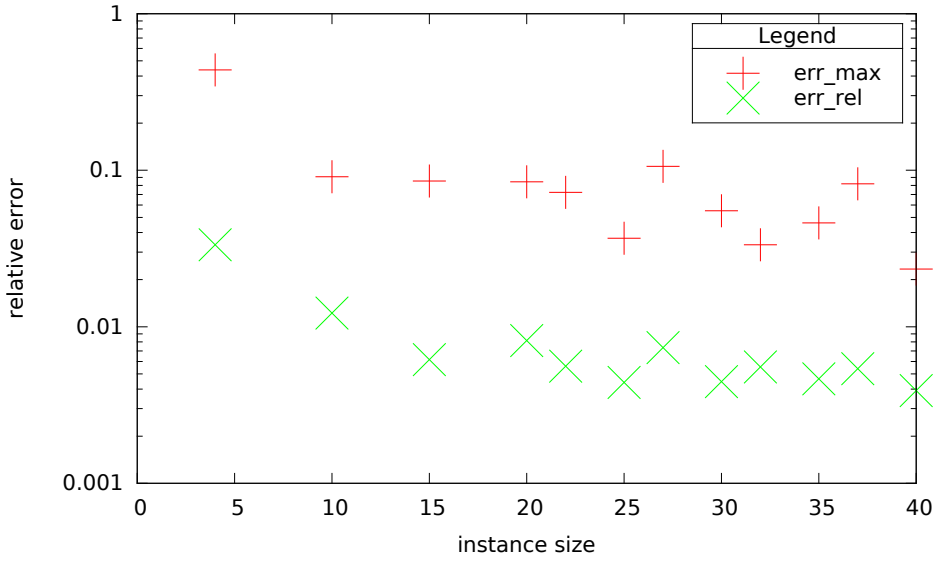


Figure 24: err_h2.pdf

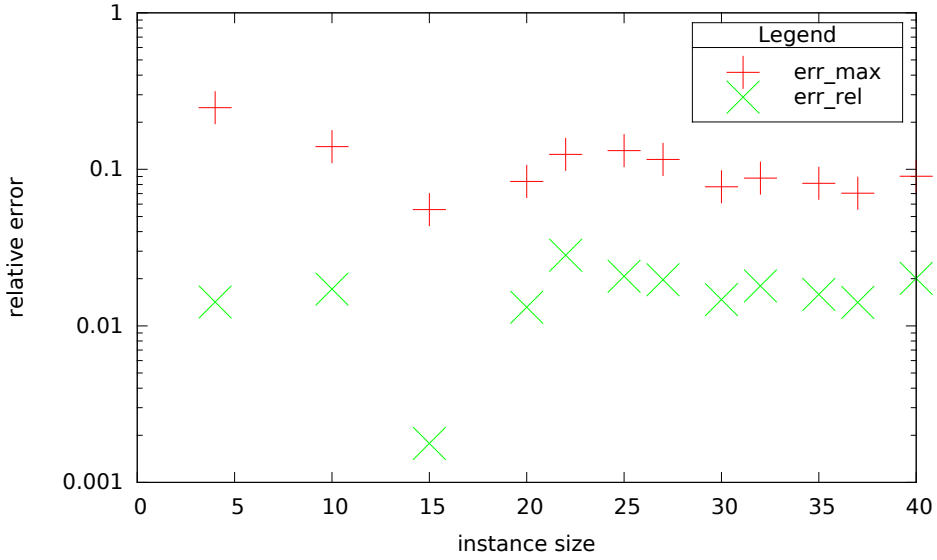



Figure 25: err_h3.pdf