Jakub Holý
holyjak1@fit.cvut.cz

MI-PAA

úkol č.4

Zadání

- Prozkoumejte citlivost metod řešení problému batohu na parametry instancí generovaných generátorem náhodných instancí. Máte-li podezření na další závislosti, modifikujte zdrojový tvar generátoru.
- Na základě zjištění navrhněte a proveďte experimentální vyhodnocení kvality řešení a výpočetní náročnosti
- Pozorujte zejména závislosti výpočetního času (B&B, DP, heuristika) a rel. chyby (heuristika) na:
  - maximální váze věcí
  - maximální ceně věcí
  - poměru kapacity batohu k sumární váze
  - granularitě

Řešení

K řešení byly použity programy vytvořené v předchozích úlohách, pouze s drobnými změnami. Pro implementaci byl použit jazyk C++ na platformě GNU/Linux. Údaje byly měřeny na desktopové stanici s dostatečným množstvím DDR2 1066MHz paměti, procesorem je Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz. V případě příliš krátkých časů byla měření opakována a vydělena svým počtem.

Pomocí dodaného generátoru instancí byly vygenerovány instance o velikosti 20 prvků, tato hodnota nebyla měněna, neboť proměnnou velikost instance jsem již měřil v předchozích úlohách.

Základní parametry pro vygenerování testovacích instancí jsem zvolil následující:

./gen -I _id_ -n 20 -N 25 -m 0.5 -W 100 -C 100 -k 1 -d 0

Ostatní parametry se různě měnily, v závislosti na měřeném efektu veličiny.

Hodnoty jednotlivých parametrů:

- poměr kapacity batohu k sumární váze: 0,1; 0,3; 0,5; 0,7; 0,9 //(kap 1-5)
- max. cena věci: 10, 100, 1000, 10000, 100000 //(cena 1-5)
- max. váha věci: 10, 100, 1000, 10000, 100000 //(vaha 1-5)
- pro každou granularitu
  - -1 [granumin] – především malé věci
  - +1 [granlus] – především velké věci

    jsem použil následující exponenty:
    - 0,2; 0,4; 0,6; 0,8; 1,0; 1,2; 1,4; 1,6; 1,8; 2,0 //(1-10)

Naměřené výsledky

Čas – dynamické programování [s]

cena1	0,000026409311294556
cena2	0,000026082859039307
cena3	0,000026202983856201
cena4	0,000026283149719238
cena5	0,000026132154464722
granlus1	0,000028065500259399
granlus2	0,000030375156402588
granlus3	0,000032620553970337
granlus4	0,000034236431121826
granlus5	0,000036858348846436
granlus6	0,000038198041915894
granlus7	0,000039427652359009
granlus8	0,000039574518203735
granlus9	0,000040763216018677
granlus10	0,000040663194656372
granmin1	0,000023245840072632
granmin2	0,000022001371383667
granmin3	0,000020401926040649
granmin4	0,000017760362625122
granmin5	0,000015252189636230
granmin6	0,000014656963348389
granmin7	0,000013293809890747
granmin8	0,000012390766143799
granmin9	0,000011859884262085
granmin10	0,000010653085708618
kap1	0,000006959845542908
kap2	0,000017233810424805
kap3	0,000026080236434937
kap4	0,000034633579254150
kap5	0,000042500762939453
vaha1	0,000026190080642700
vaha2	0,000213997173309326
vaha3	0,002823120117187500
vaha4	0,036841316223144500
vaha5	0,364396371841431000

Čas – Branch and Bounds [s]

cena1	0,007974636077880860
cena2	0,007641694068908690
cena3	0,007969226837158200
cena4	0,007857519149780270
cena5	0,007774863243103030
granlus1	0,007822330474853520
granlus2	0,008069346427917480
granlus3	0,008302164077758790
granlus4	0,007835750579833990
granlus5	0,007948696136474610
granlus6	0,007798406600952150
granlus7	0,008016176223754880
granlus8	0,008795668601989740
granlus9	0,008017352104187010
granlus10	0,008001084327697750
granmin1	0,007871592521667480
granmin2	0,007511663436889650
granmin3	0,007322852134704590
granmin4	0,007460236549377440
granmin5	0,007372064590454100
granmin6	0,007997797012329100
granmin7	0,007238473892211910
granmin8	0,007374328613281250
granmin9	0,007671813964843750
granmin10	0,007514507293701170
kap1	0,000019688606262207
kap2	0,001431341171264650
kap3	0,007669626235961910
kap4	0,009458068847656250
kap5	0,008161419868469240
vaha1	0,007619112968444820
vaha2	0,007656396865844730
vaha3	0,007813100814819340
vaha4	0,008153854370117190
vaha5	0,007634117126464850

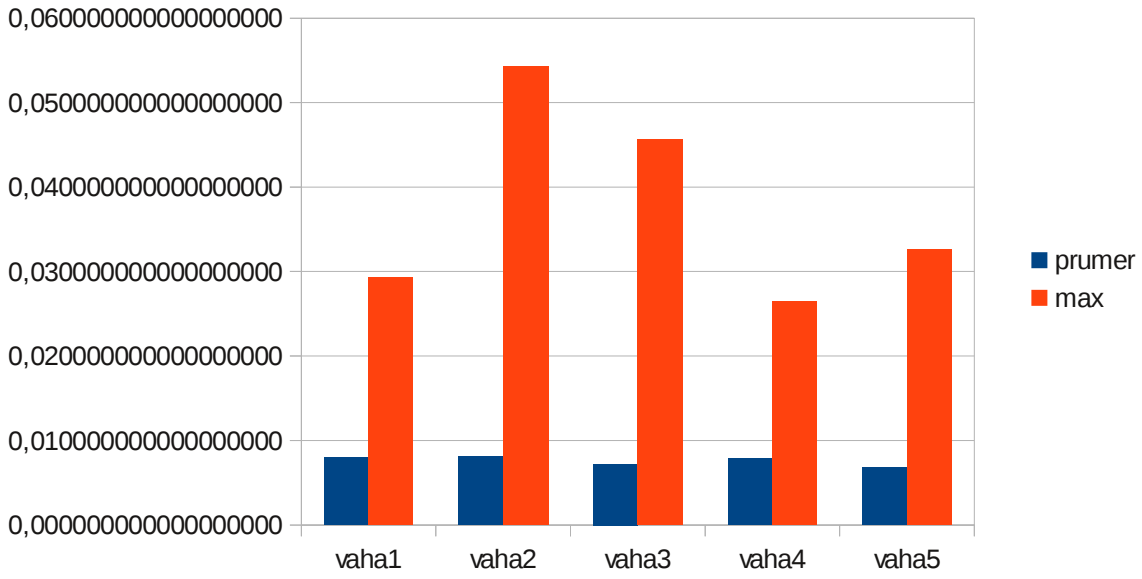
Čas – Poměrová heuristika [s]

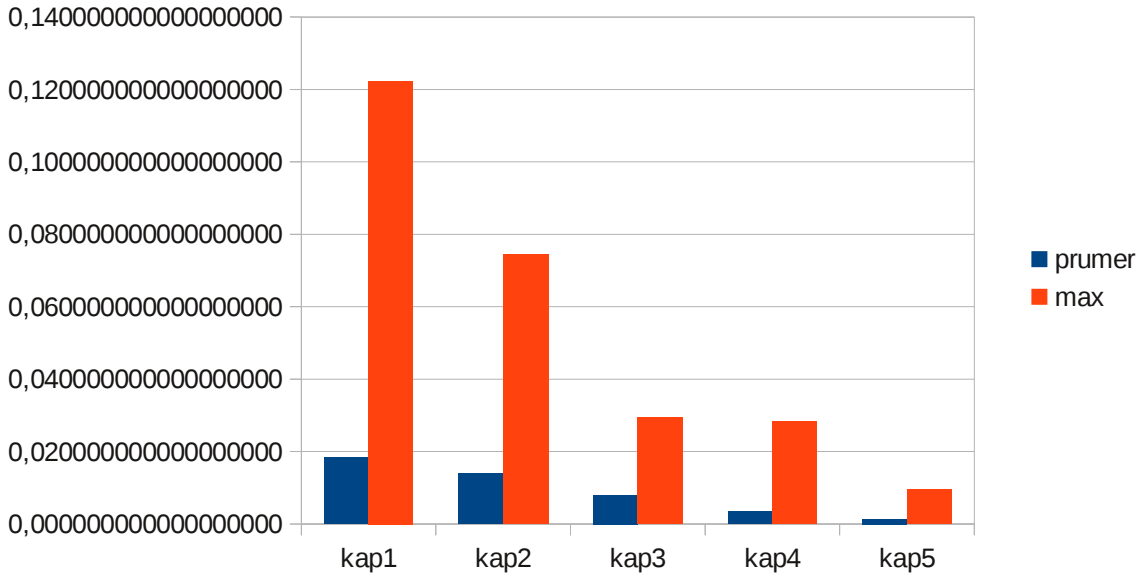
cena1	0,000000799456596375
cena2	0,000000779542922974
cena3	0,000000783847808838
cena4	0,000000794847488403
cena5	0,000000784064292908
granlus1	0,000000786794662476
granlus10	0,000000780176162720
granlus2	0,000000787844657898
granlus3	0,000000789518356323
granlus4	0,000000799944877624
granlus5	0,000000783286094666
granlus6	0,000000790229797363
granlus7	0,000000797041893005
granlus8	0,000000789599418640
granlus9	0,000000800131797790
granmin1	0,000000790993690491
granmin2	0,000000784831047058
granmin3	0,000000783871650696
granmin4	0,000000781847000122
granmin5	0,000000786771774292
granmin6	0,000000770986557007
granmin7	0,000000781661987305
granmin8	0,000000793608665466
granmin9	0,000000775195121765
granmin10	0,000000771871566772
kap1	0,000000771288871765
kap2	0,000000782349586487
kap3	0,000000787690162659
kap4	0,000000781633377075
kap5	0,000000763410568237
vaha1	0,000000782857894897
vaha2	0,000000783548355103
vaha3	0,000000787112236023
vaha4	0,000000779580116272
vaha5	0,000000780212402344

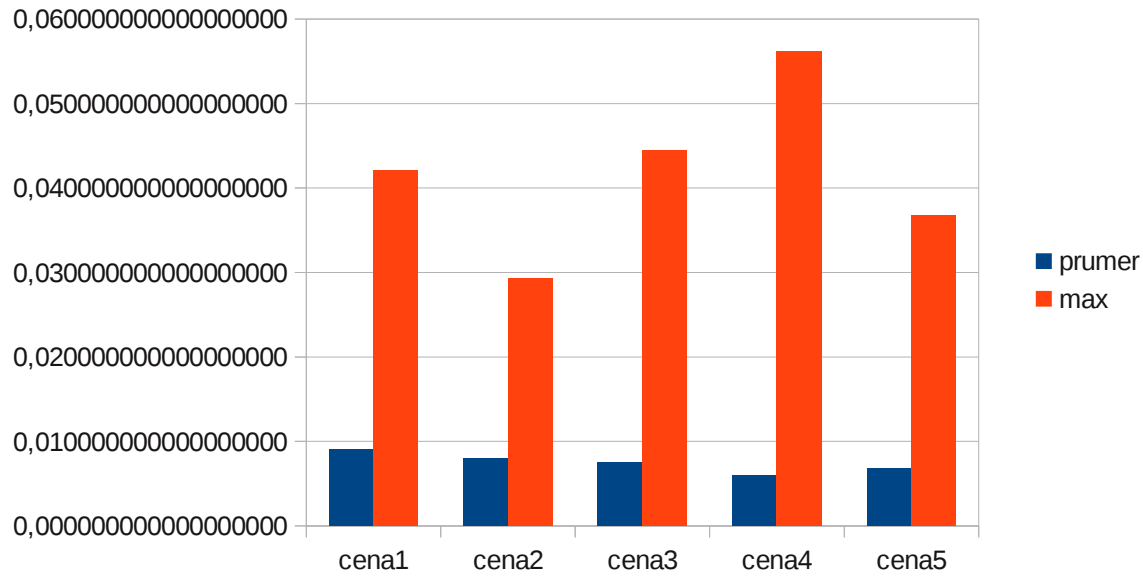
Průměrná a maximální chyba u heuristiky

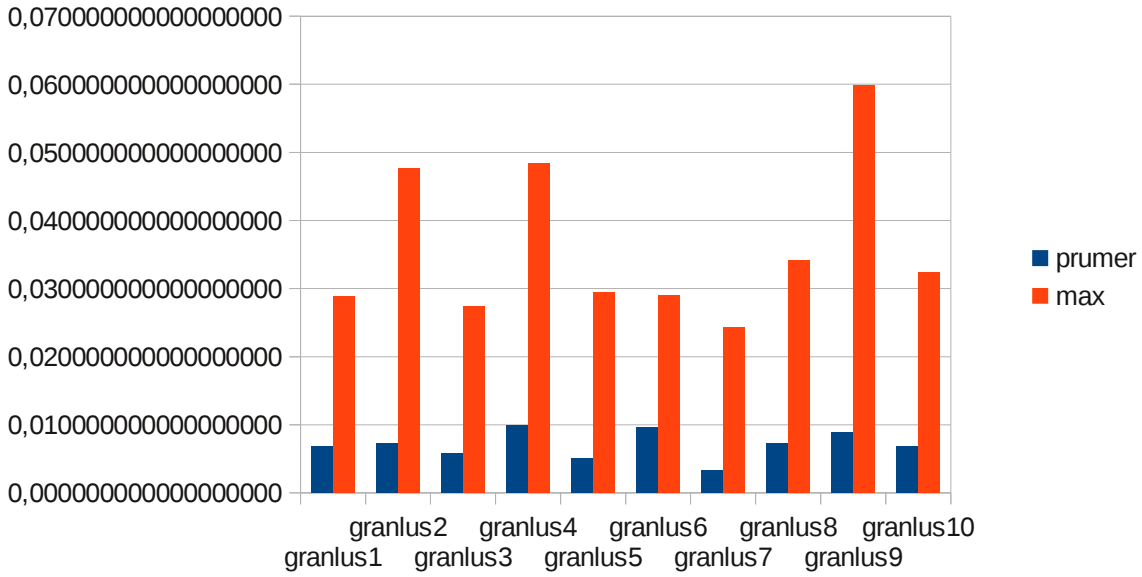
	prumer	max
cena1	0,009012967024179000	0,042105263157895000
cena2	0,007959749115347100	0,029304029304029000
cena3	0,007509378179068500	0,044390637610977000
cena4	0,005941598219910900	0,056121753615507000
cena5	0,006778608496724400	0,036797008510124000
granlus1	0,006797690546326700	0,028860028860029000
granlus2	0,007297124020520400	0,047729918509895000
granlus3	0,005830169001489200	0,027465667915106000
granlus4	0,009954082998024400	0,048411497730711000
granlus5	0,005015910414645300	0,029374201787995000
granlus6	0,009572045918354100	0,028985507246377000
granlus7	0,003309650930508500	0,024355300859599000
granlus8	0,007292422867746200	0,034090909090909000
granlus9	0,008843680170659600	0,059914407988588000
granlus10	0,006879121527022400	0,032305433186490000
granmin1	0,005605588580756500	0,031358885017422000
granmin2	0,005497510096477700	0,032558139534884000
granmin3	0,005368941452904400	0,031161473087819000
granmin4	0,003666341687634100	0,021301775147929000
granmin5	0,011519757386656000	0,048030739673391000
granmin6	0,007317254999286700	0,035339063992359000
granmin7	0,008286385100605700	0,035502958579882000
granmin8	0,008014817520115100	0,049735449735450000
granmin9	0,016590434443139000	0,086631016042781000
granmin10	0,004428442727933000	0,017201834862385000
kap1	0,018486225430791000	0,122388059701490000
kap2	0,013940056134085000	0,074363992172211000
kap3	0,007959749115347100	0,029304029304029000
kap4	0,003366668428402500	0,028169014084507000
kap5	0,001364545880184900	0,009469696969697000
vaha1	0,007959749115347100	0,029304029304029000
vaha2	0,008116795553134000	0,054303278688525000
vaha3	0,007218114183522900	0,045685279187817000
vaha4	0,007904558341650100	0,026415094339623000
vaha5	0,006816461649348400	0,032620922384702000

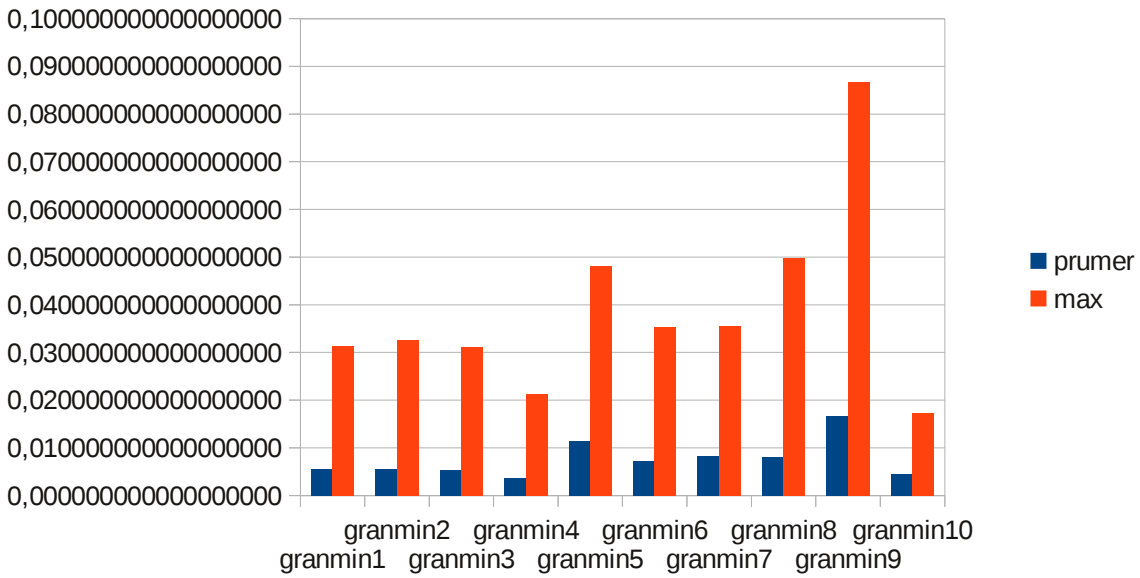
Průměrná a maximální chyba heuristiky v grafech

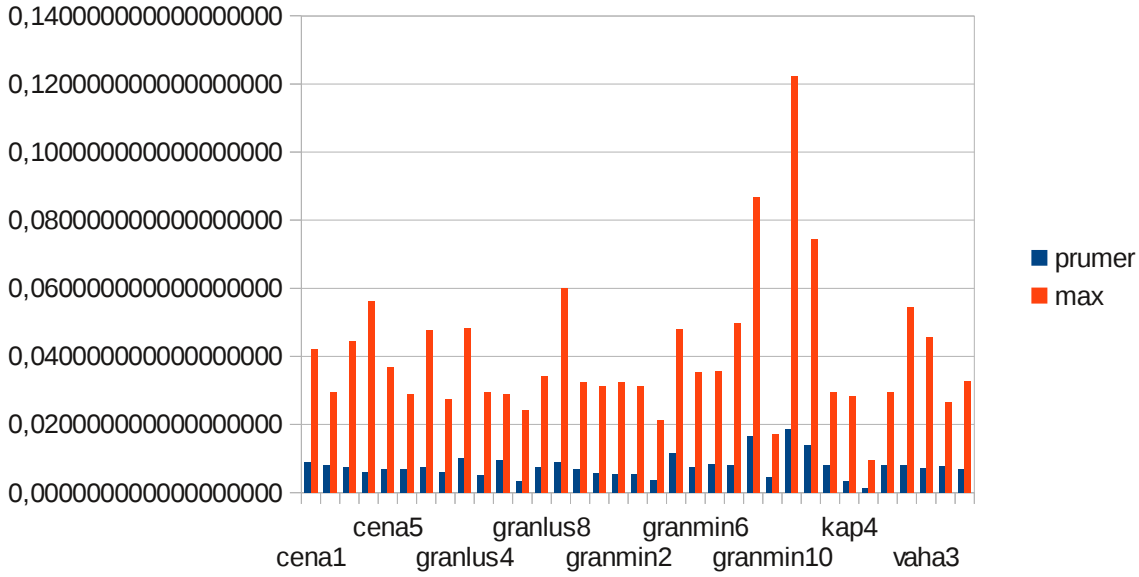





0,060000000000000000
0,050000000000000000
0,040000000000000000
0,030000000000000000
0,020000000000000000
0,010000000000000000
0,000000000000000000
cena1
cena2
cena3
cena4
cena5
prumer
max


0,070000000000000000
0,060000000000000000
0,050000000000000000
0,040000000000000000
0,030000000000000000
0,020000000000000000
0,010000000000000000
0,000000000000000000
granlus1
granlus2
granlus3
granlus4
granlus5
granlus6
granlus7
granlus8
granlus9
granlus10
prumer
max

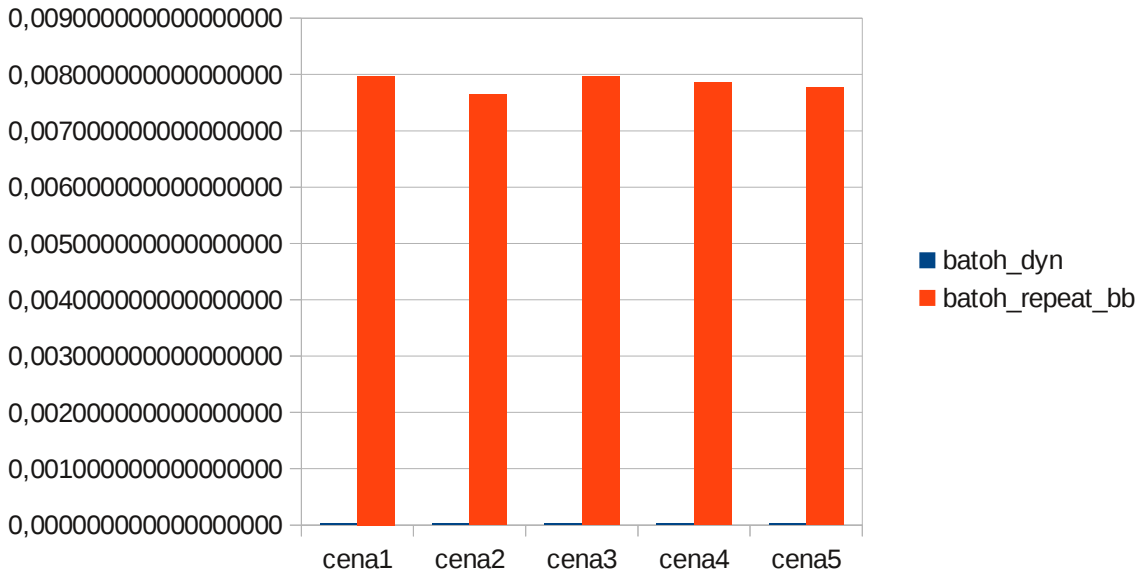


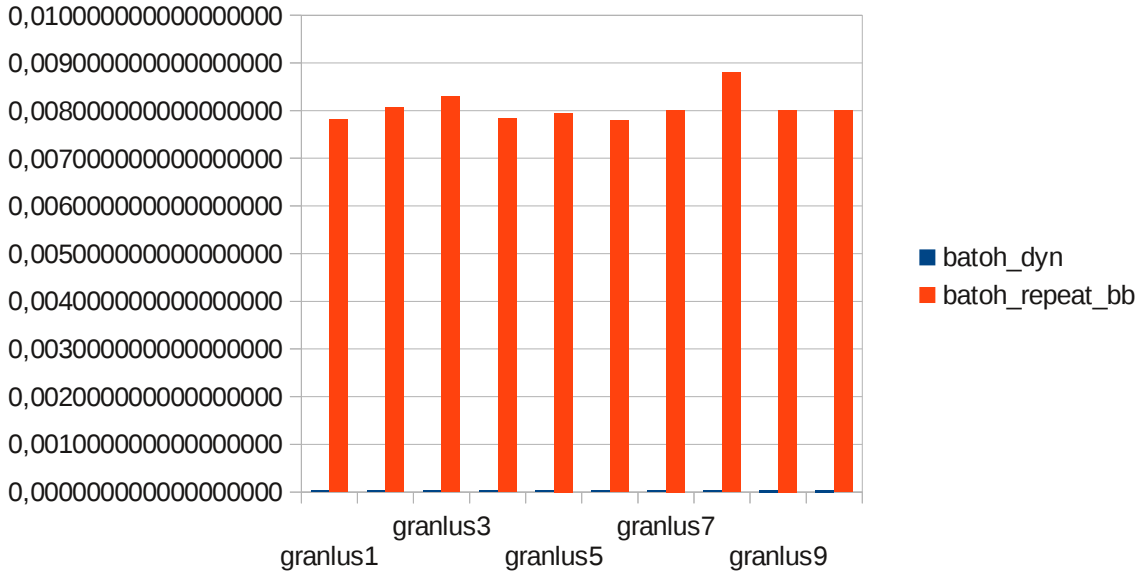


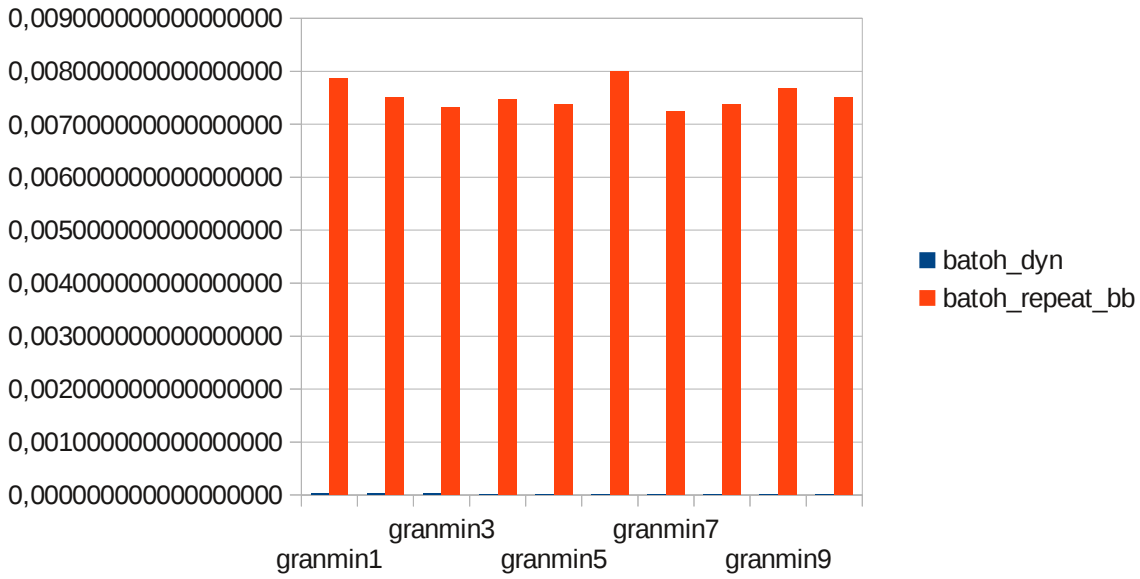
V těchto datech/grafech nedošlo k žádným neočekávaným událostem, i ta maximální špička v kap1 je pravděpodobná, neboť při velmi malém batohu je možné že heuristika poměru snadno vybere jinou variantu než je optimální. Nicméně i tak vyšla heuristika v průměru velmi dobře.

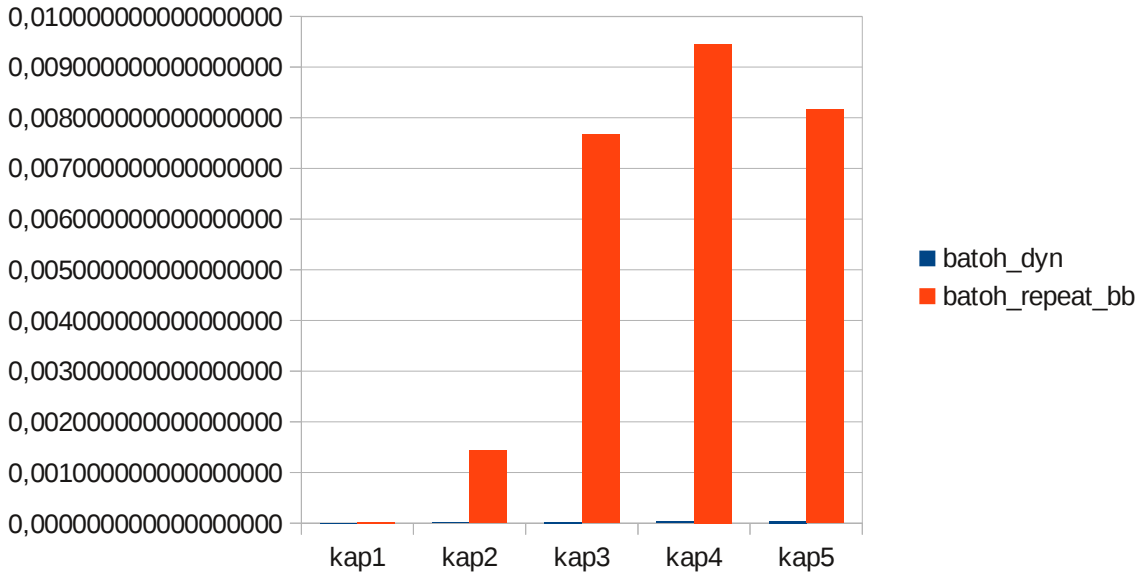
Čas výpočtu této heuristiky nemá cenu srovnávat, jelikož pro jednu velikost instance už ze své podstaty konstantní.

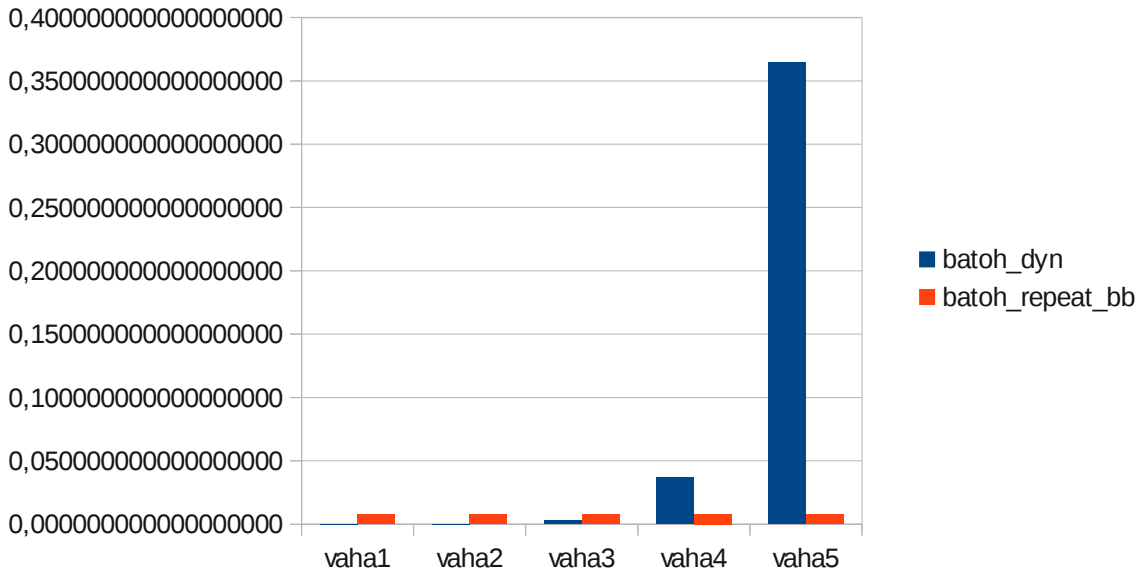
Citlivosti algoritmů na jednotlivé parametry






0,009000000000000000
0,008000000000000000
0,007000000000000000
0,006000000000000000
0,005000000000000000
0,004000000000000000
0,003000000000000000
0,002000000000000000
0,001000000000000000
0,000000000000000000
granmin1
granmin3
granmin5
granmin7
granmin9
batoh_dyn
batoh_repeat_bb


0,010000000000000000
0,009000000000000000
0,008000000000000000
0,007000000000000000
0,006000000000000000
0,005000000000000000
0,004000000000000000
0,003000000000000000
0,002000000000000000
0,001000000000000000
0,000000000000000000
kap1
kap2
kap3
kap4
kap5
batoh_dyn
batoh_repeat_bb



Z grafů vyplývá, že algoritmy jsou necitlivé na většinu měřených veličin.

Dynamické programování vypadá že je citlivé na zvětšenou maximální váhu předmětů – tomu tak ve skutečnosti není. S rostoucí maximální povolenou hodnotou hmotnosti jedné věci roste v generátoru taktéž celková kapacita batohu, aby do něj bylo možné některé prvky vůbec vložit. Na samotný růst maximální ceny prvku tedy algoritmus citlivý není, ale vzhledem ke zvolené dekompozici je citlivý na růst celkové velikosti batohu.

Algoritmus Branch and Bound je citlivý na omezování kapacity batohu – čím více je zmenšena kapacita, tím snáze je možné v daném stromu stavů prořezávat, neboť je jisté, že už se do malého batohu již nic nevejde.

Závěr

Algoritmy nejsou příliš citlivé na data, každý z nich má na něco citlivost, nicméně na obecných datech bych oba viděl jako ekvivalentní, použití bych volil podle paměťových možností – je-li paměti dostatek tak dynamické programování, jinak Branch and Bound.