
Problém batohu

Pavol Lupták <P.Luptak@sh.cvut.cz>

Obsah

1. Zadanie č.4	1
1.1. Kvalita riešenia	1
1.2. Výpočetná náročnosť	1
1.3. Data pre experiment a jeho usporiadanie	1
1.4. Zadanie domácej úlohy	2
2. Riešenie	2
3. Výsledky	2
4. Implementácia a zdrojové súbory	10
5. Záver	10

1. Zadanie č.4

Na riešenie problému batohu existuje mnoho algoritmov ako exaktných tak približných. O ich vlastnostiach sa toho veľa nevie, iba pre kombinované heuristiky bola dokázaná maximálna chyba. Ak chceme vedieť viac, musíme vyhodnocovať experimentálne. Budeme sledovať dva podstatné parametre kvalitu riešenia a výpočetnú náročnosť. Ak uvedieme obe tieto veličiny v prehľade, dozvieme sa pre ktoré kombinácie nárokov na čas a kvalitu je ten ktorý algoritmus najvyhodnejší.

1.1. Kvalita riešenia

Pre inštancie, kde vieme exaktné riešenie sa dá kvalita merať absolútne. Tam, kde porovnávame heuristiky medzi sebou, môžeme porovnávať iba relatívne. Tieto dva spôsoby hodnotenia je potrebné rozlišovať a konzistentne medzi nimi voliť.

1.2. Výpočetná náročnosť

Výpočetná náročnosť sa meria ešte horšie. Celkový čas výpočtu zahŕňa všetky vplyvy, zlyháva však pri porovnaní výsledkov z rôznych strojov. Zahŕňa tiež vplyvy, ktoré nie sú dôležité - napríklad spôsob implementácie dátových štruktúr. Preto ako merítko výpočtovej zložitosti volíme počet testovaných stavov. Pretože celkové počty stavov inštancií je možné jednoducho odvodiť, máme merítko účinnosti výpočtovej metódy.

1.3. Data pre experiment a jeho usporiadanie

Každý experiment má za úlohu zodpovedať nejakú otázku. Musí byť navrhnutý tak, aby nedával falošné odpovede, aby napríklad neodpovedal na otázku obmedzenú. Data teda musia byť reprezentatívne, čo

je niekedy vážny problém. Dokonca ake sa pokúšame data generovať náhodne, môžeme nevedomky generovať inštancie s určitými pevnými charakteristikami. Preto je generátor inštancií rozsiahle parametrizovaný. Najskôr je potrebné zistiť citlivosť jednotlivých metód na parametre riešených inštancií. Isté podozrenia sú na:

- výpočtová zložitosť dynamického programovania, môže byť citlivá na maximálnu cenu.
- výkon metód, ktoré vychádzajú zo stavu "prázdny batoh" sa môže líšiť od metód vyhádzajúcich zo stavu "plný batoh", podľa pomeru celková váha/kapacita batohu
- nie je jasné, akú rolu hrá granularita inštancie (prevaha malých alebo veľkých vecí)

1.4. Zadanie domácej úlohy

Preskúmajte citlivosť metód riešenia problému batohu na parametre inštancií generovaných generátorom náhodných inštancií. Ak máte podozrenie na ďalšie závislosti, modifikujte zdrojový tvar generátoru. Na základe zistení navrhnete a vykonajte experimentálne vyhodnotenie kvality riešenia a výpočtovej náročnosti. Pokiaľ možno prezentujte algoritmy ako body na ploche, ktorej súradnice sú vyššie uvedené kritéria.

2. Riešenie

Pri našom experimentálnom hodnotení metód nás budú zaujímať nasledujúce vlastnosti metód:

- *Výpočtová náročnosť metód* určená pomerom krokov algoritmu 1 a celkovej veľkosti stavového priestoru danou vzťahom $2n$, vyjadrená v percentách
- *Presnosť metód*, ktorá má význam len u nepresných (heuristických) algoritmov. Je určená pomerom relatívnej chyby a maximálnej ceny získanej exaktnými metódami (v mojom prípade som použil cenu získanu metódou dynamického programovania).

Každý riadok v tabuľke bol získaný priemerom 50 problémov s danými parametrami, vygenerovaný upravenou verziou knapgen.

3. Výsledky

Tabuľka 1. Závislosť na veľkosti inštancie

¹Uvažujeme množstvo prehľadaných stavov, v prípade heuristických metód sa počítajú aj kroky triediaceho algoritmu, v dynamickom programovaní počet nenulových položiek iterujúcej tabuľky

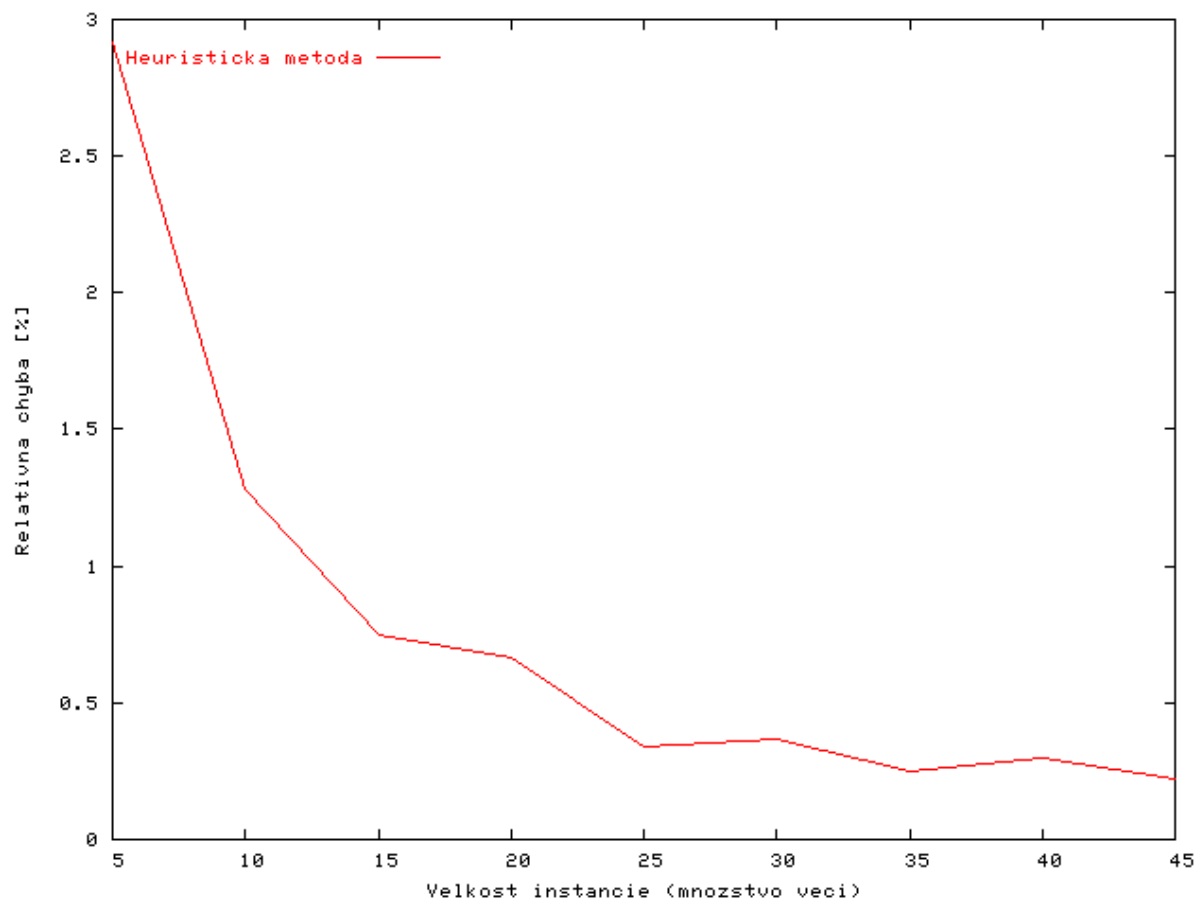
Maximálna hmotnosť vecí = 100, maximálna cena vecí = 250, počet problémov = 50, vyrovnané rozdelenie vecí, pomer kapacity k celkovej váhe = 0.6							
Veľkosť inštancie	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcennejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
5	82.625000	71.875000	2.918172	90.625000	2.918172	10806.562500	60.375000
10	85.259766	7.128906	1.279783	8.203125	1.279783	1289.595703	16.390625
15	86.143677	0.451660	0.746874	0.500488	0.746874	86.707581	4.206726
20	87.175144	0.023651	0.666777	0.025654	0.666777	4.925323	0.972893
25	88.717563	0.001112	0.339485	0.001189	0.339485	0.237305	0.222174
30	0.000000	0.000049	0.368425	0.000052	0.368425	0.010296	0.047434
35	0.000000	0.000002	0.247098	0.000002	0.247098	0.000436	0.015172
40	0.000000	0.000000	0.295934	0.000000	0.295934	0.000018	0.002690
45	0.000000	0.000000	0.220560	0.000000	0.220560	0.000001	0.000249

V prípade metódy hrubej sily som v dôsledku neúmernej časovej náročnosti výpočtu, analyzoval len problémy s maximálnou veľkosťou inštancí 25, preto pre vyššie inštancie je uvedená v tabuľke 0.

Problémy s inštanciami 40 a 45 majú tak veľký stavový priestor, že pomer krokov v heuristickej metóde a veľkosti celkového stavového priestoru limituje k 0.

Z tabuľky je zrejmé, že so vzrastajúcim množstvom inštancií rastie výpočtová zložitosť **metódy hrubej sily** veľmi mierne, zložitosť **heuristických metód** vzhľadom na celkový stavový priestor prudko klesajú, podobne sa so vzrastajúcim množstvom inštancií oplácajú **metódy dynamického programovania a vetvenia**.

So vzrastajúcim množstvom inštancií, klesá relatívna chyba **heuristických metód**. Tento jav zachytuje nasledujúci graf.

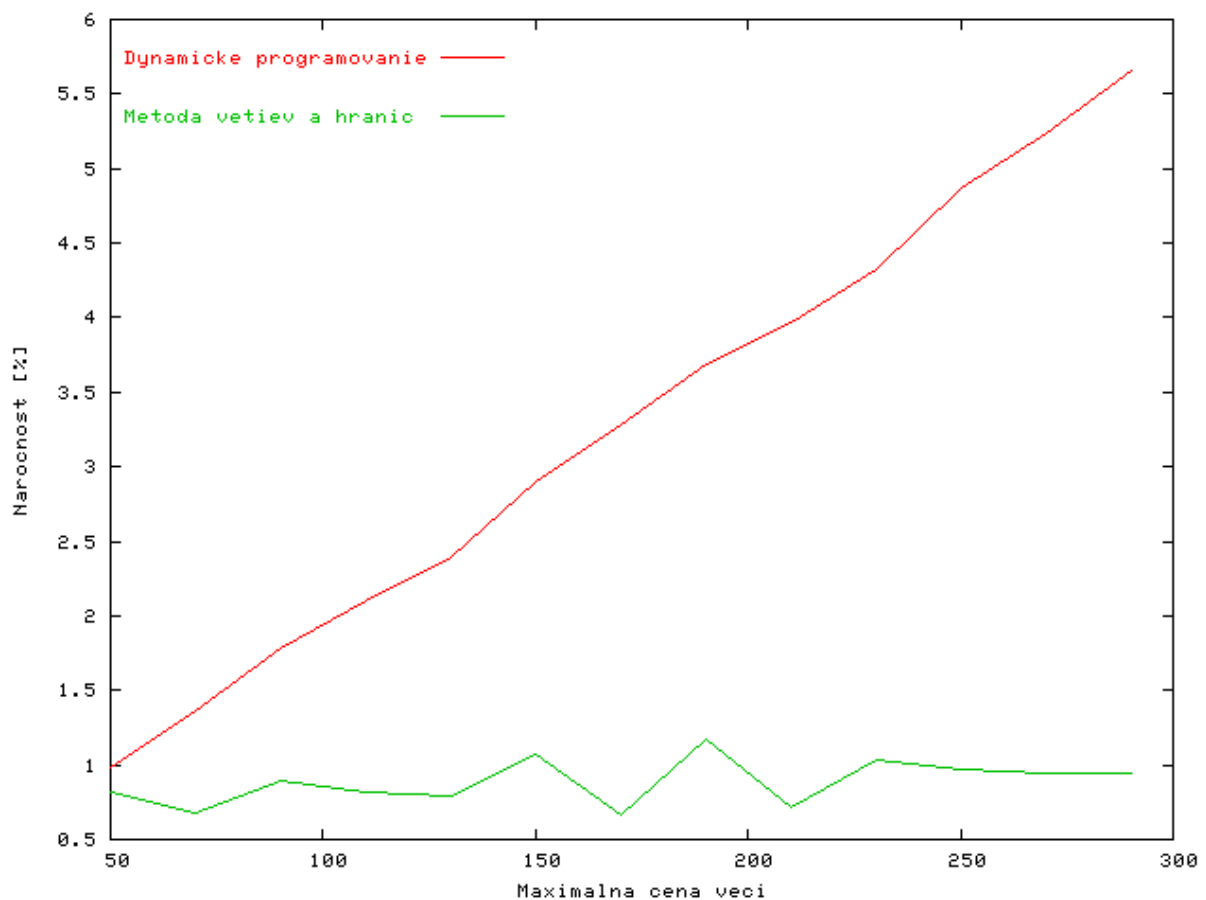


Tabuľka 2. Závislosť na maximálnej cene vecí

Maximálna hmotnosť vecí = 100, množstvo vecí v batohu = 20, počet problémov = 50, vyrovnané rozdelenie vecí, pomer kapacity k celkovej váhe = 0.6							
Maximálna cena vecí	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcenejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
50	87.598274	0.023651	0.358119	0.025654	0.358119	0.988739	0.814495
70	87.794128	0.023651	0.412185	0.025654	0.412185	1.364925	0.681734
90	87.757942	0.023651	0.422320	0.025654	0.422320	1.789030	0.889265
110	87.249672	0.023651	0.738741	0.025654	0.738741	2.098509	0.818603
130	88.294651	0.023651	0.292918	0.025654	0.292918	2.395050	0.793253
150	87.510887	0.023651	0.493276	0.025654	0.493276	2.900743	1.074720
170	87.849840	0.023651	0.419786	0.025654	0.419786	3.276344	0.667393
190	87.736202	0.023651	0.679202	0.025654	0.679202	3.689938	1.179146

Maximálna hmotnosť vecí = 100, množstvo vecí v batohu = 20, počet problémov = 50, vyrovnané rozdelenie vecí, pomer kapacity k celkovej váhe = 0.6							
Maximálna cena vecí	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcenejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
210	87.570629	0.023651	0.456986	0.025654	0.456986	3.970232	0.717690
230	88.079308	0.023651	0.631170	0.025654	0.631170	4.324125	1.030182
250	87.260054	0.023651	0.499919	0.025654	0.499919	4.864786	0.964029
270	88.294266	0.023651	0.402282	0.025654	0.402282	5.233889	0.947519
290	87.111149	0.023651	0.655983	0.025654	0.655983	5.660580	0.947210

Z tabuľky je zrejmé, že so vzrastajúcou maximálnou cenou vecí (a tým aj vzrastajúcou celkovou cenou vecí v batohu) rastie takmer lineárne výpočtová náročnosť **dynamického programovania**, v prípade **metódy vetiev a hraníc** mierne kolíše. U ostatných metód nebolo pozorovaná výrazná zmena správania v závislosti od maximálnej ceny vecí.



Tabuľka 3. Závislosť na maximálnej hmotnosti vecí

Maximálna cena vecí = 250, množstvo vecí v batohu = 20, počet problémov = 50, vyrovnané rozdelenie vecí, pomer kapacity k celkovej váhe = 0.6							
Maximálna hmotnosť vecí	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcennejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
50	88.040878	0.023651	0.464989	0.025654	0.464989	4.837955	0.891594
70	87.021069	0.023651	0.503697	0.025654	0.503697	4.889122	0.845903
90	87.433453	0.023651	0.606262	0.025654	0.606262	4.950184	0.822773
110	86.819271	0.023651	0.431190	0.025654	0.431190	4.945763	1.008396
130	87.676691	0.023651	0.387799	0.025654	0.387799	4.758343	0.905699
150	87.750463	0.023651	0.556595	0.025654	0.556595	4.939297	0.865868
170	86.760309	0.023651	0.587964	0.025654	0.587964	4.645336	0.773983
190	87.287167	0.023651	0.538731	0.025654	0.538731	4.780819	0.913645
210	86.889250	0.023651	0.509333	0.025654	0.509333	4.991465	0.811443
230	88.050451	0.023651	0.676527	0.025654	0.676527	4.912186	0.924118
250	86.825300	0.023651	0.610915	0.025654	0.610915	4.891788	0.916685
270	87.290495	0.023651	0.407910	0.025654	0.407910	4.860201	0.837236
290	88.245352	0.023651	0.397466	0.025654	0.397466	4.846344	0.832487

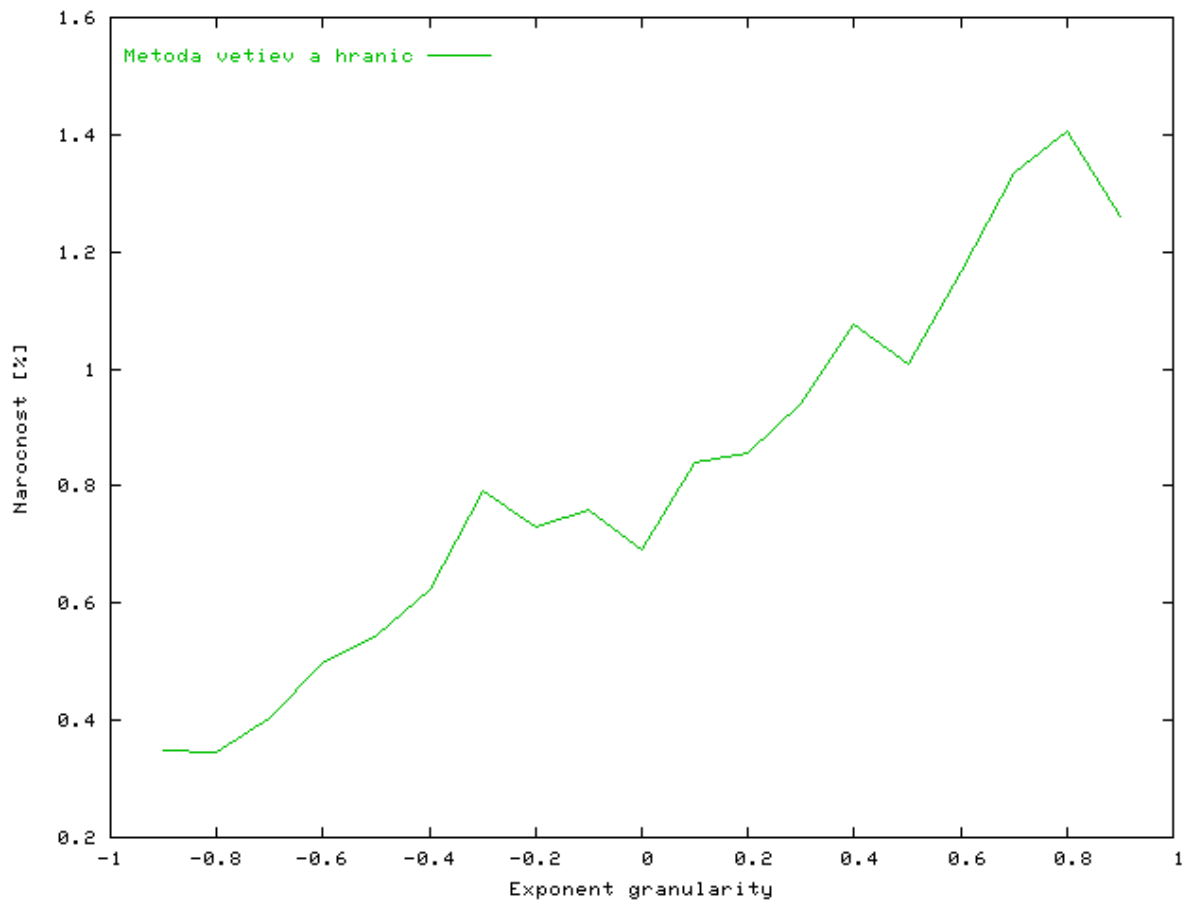
Pri testoch so vzrastajúcou maximálnou hmotnosťou vecí mierne kolísala výpočtová náročnosť u **metódy vetiev a hraníc** ako aj **dynamického programovania**. V tomto prípade najväčšia výpočtová náročnosť bola zaznamenaná u maximálnej hmotnosti 100-110. Pri ostatných metódach nebola pozorovaná výrazná závislosť od maximálnej hmotnosti vecí.

Tabuľka 4. Závislosť od exponentu granularity

Maximálna hmotnosť vecí = 100, maximálna cena vecí = 250, počet problémov = 50, počet vecí v batohu = 20, pomer kapacity k celkovej váhe = 0.6							
Exponent granularity	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcenejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
-0.9	84.666655	0.023651	0.590485	0.025654	0.590485	4.828989	0.349880
-0.8	86.049198	0.023651	0.631383	0.025654	0.631383	4.814699	0.344686
-0.7	84.763887	0.023651	0.564406	0.025654	0.564406	4.907339	0.404060
-0.6	85.579203	0.023651	0.580685	0.025654	0.580685	4.862869	0.496885
-0.5	85.673723	0.023651	0.517169	0.025654	0.517169	4.704090	0.545002
-0.4	85.159388	0.023651	0.693227	0.025654	0.693227	4.757297	0.623573
-0.3	86.603065	0.023651	0.583879	0.025654	0.583879	4.808519	0.793186
-0.2	86.792942	0.023651	0.487751	0.025654	0.487751	4.770998	0.730387
-0.1	86.827856	0.023651	0.433686	0.025654	0.433686	4.717062	0.758097
0	87.339354	0.023651	0.353490	0.025654	0.353490	4.767120	0.690256
0.1	87.369532	0.023651	0.427752	0.025654	0.427752	4.779162	0.841581
0.2	87.540209	0.023651	0.372901	0.025654	0.372901	4.785126	0.857387
0.3	88.378489	0.023651	0.300180	0.025654	0.300180	4.862206	0.940382
0.4	88.079794	0.023651	0.425434	0.025654	0.425434	4.851618	1.075394
0.5	88.230534	0.023651	0.478919	0.025654	0.478919	4.921389	1.008306
0.6	88.103785	0.023651	0.424902	0.025654	0.424902	4.829382	1.166470
0.7	86.940748	0.023651	0.512478	0.025654	0.512478	4.986691	1.333408
0.8	88.551344	0.023651	0.405647	0.025654	0.405647	4.877504	1.407234
0.9	88.503706	0.023651	0.547098	0.025654	0.547098	4.848450	1.260880

Exponent granularity bol volený od 0.9 do 0 pri charaktere granularity -1 (viac malých vecí) a od 0 po 0.9 pri charaktere granularity 1, čo zabezpečilo plynulý prechod od malých vecí k veľkým v generovaných problémoch.

Zmena exponentu granularity, t.j. prevaha ťažkých vecí výrazne ovplyvňuje výpočtovú náročnosť **metódy dynamického programovania**, čo môžeme vidieť na grafe.



Tabuľka 5. Závislosť na pomere kapacity batohu k celkovej váhe vecí

Maximálna hmotnosť vecí = 100, maximálna cena vecí = 250, počet problémov = 50, vyrovnané rozdelenie vecí, počet vecí v batohu = 20							
Kapacita batohu / celková váha	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcenejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
0.05	0.033800	0.023651	3.343149	0.025654	3.196521	5.148569	0.027422
0.1	0.210493	0.023651	2.257686	0.025654	2.257686	5.120224	0.127686
0.15	0.840630	0.023651	1.343914	0.025654	1.343914	5.099316	0.368048
0.2	2.541372	0.023651	0.908660	0.025654	0.908660	5.082130	0.797327
0.25	6.196190	0.023651	0.962629	0.025654	0.962629	5.066160	1.320295
0.3	12.047749	0.023651	0.713859	0.025654	0.713859	4.813774	1.689945
0.35	22.033264	0.023651	0.842981	0.025654	0.842981	4.859760	2.579853
0.4	35.427807	0.023651	1.053979	0.025654	1.053979	4.828110	2.728226

Maximálna hmotnosť vecí = 100, maximálna cena vecí = 250, počet problémov = 50, vyrovnané rozdelenie vecí, počet vecí v batohu = 20							
Kapacita batohu / celková váha	Metóda hrubej sily	Heuristická metóda (c/m)		Heuristická metóda (c/m) s testom najcennejšej veci		Dynamické programovanie	Metóda vetiev a hraníc
	Náročnosť [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Relatívna chyba [%]	Náročnosť [%]	Náročnosť [%]
0.45	50.360456	0.023651	1.005209	0.025654	1.005209	4.872131	2.469399
0.5	65.406799	0.023651	0.575942	0.025654	0.575942	4.904678	1.673573
0.55	77.570896	0.023651	0.644808	0.025654	0.644808	4.737654	1.438406
0.6	88.163517	0.023651	0.482038	0.025654	0.482038	4.911522	1.105522
0.65	93.898119	0.023651	0.554994	0.025654	0.554994	4.787060	0.544182
0.7	97.397188	0.023651	0.435061	0.025654	0.435061	4.837124	0.312136
0.75	99.049292	0.023651	0.270944	0.025654	0.270944	4.682837	0.125696
0.8	99.739485	0.023651	0.361731	0.025654	0.361731	4.940193	0.091042
0.85	99.953547	0.023651	0.189260	0.025654	0.189260	4.838686	0.035521
0.9	99.991972	0.023651	0.096633	0.025654	0.096633	4.769617	0.019329
0.95	99.999159	0.023651	0.095048	0.025654	0.095048	4.651443	0.010401

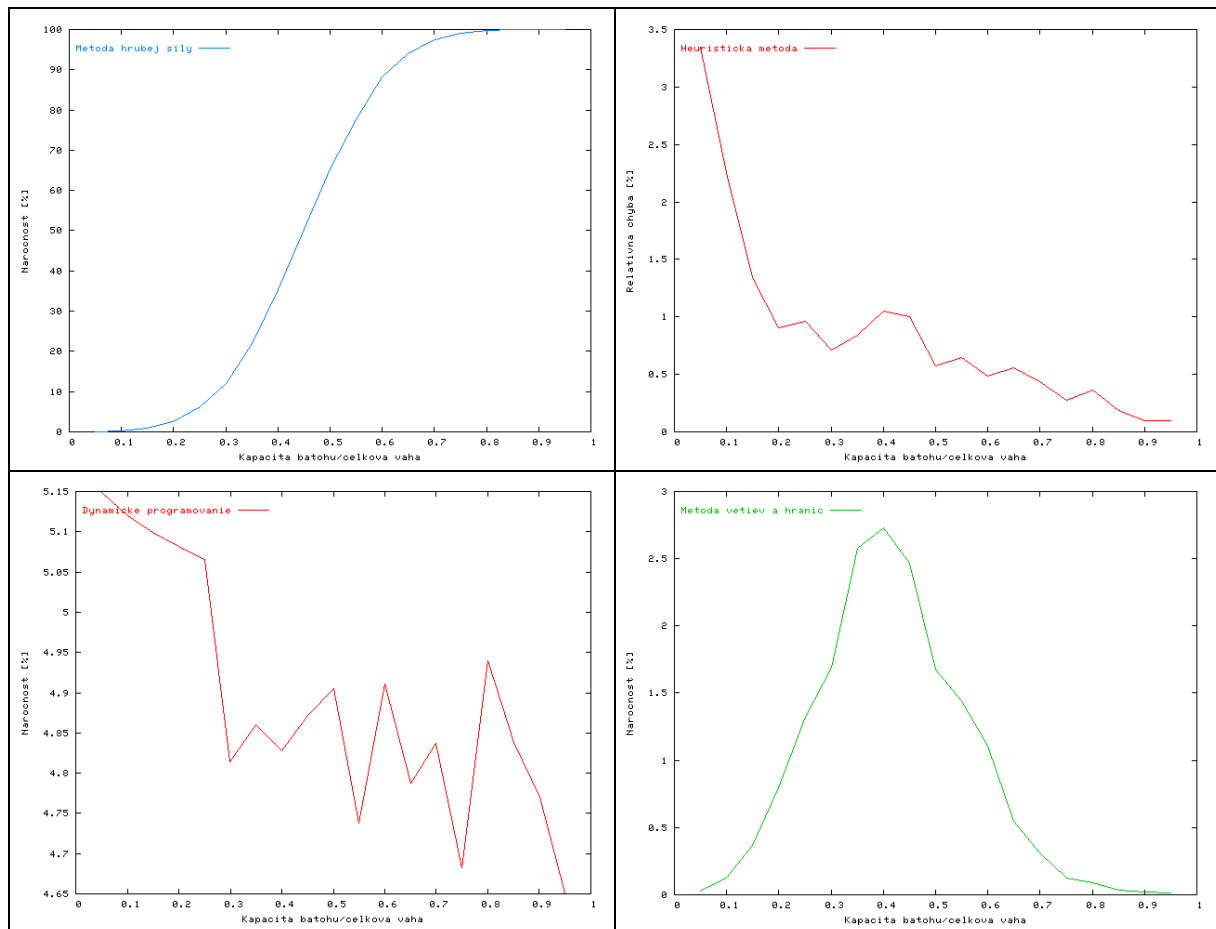
Sledovanie závislosti na pomere kapacity batohu k celkovej váhe vecí prinieslo najzaujímavejšie výsledky, nakoľko zmena tohto parametru ovplyvnila výpočtovú náročnosť a relatívnu chybu každej metódy.

Nízka hodnota pomeru kapacity batohu k celkovej váhe vecí znamená, že sa batoh veľmi rýchlo naplnil pomerne ťažkými vecami a **metóda hrubej sily** ďalej neprehľadáva ďalšie zbytočné stavy.

Podobné vlastnosti sa prejavili u **metódy vetiev a hraníc** až na to, že od pomeru 0.45 (t.j. celková kapacita batohu je cca. dvakrát menšia ako celková váha vecí) dochádza k rýchlejšiemu naplneniu batohu a teda k zlepšeniu výpočtovej náročnosti.

U **dynamického programovania** so vzrastúcim pomerom kapacity batohu k celkovej hmotnosti vecí, klesá výpočtová náročnosť veľmi pozvoľne.

Zaujímavé bolo tiež pozorovať so vzrastajúcim pomerom kapacity batohu k celkovej hmotnosti vecí, pokles relatívnej chyby v prípade heuristických metód.



4. Implementácia a zdrojové súbory

Na generovanie potrebných štatistických údajov som upravil `batoh.c` integrujúci všetky testované metódy problému batohu. Testované problémy boli parametrizované pomocou špeciálneho skriptu `[generate.sh]`, ktorý zabezpečoval frontend pre `knapgen.c`, pre každú kombináciu testovaných parametrov generoval 50 problémov, priemeroval ich a na základe priemerov zostavoval tabuľky. AWK skript `[make_xml.sh]` textové tabuľky konvertuje do XML Docbooku.

Táto dokumentácia je k dispozícii tiež v PDF [[vyhodnotenie.pdf](#)]

Testy boli realizované na Athlon XP1900+, OS Linux 2.4.21rc2, pri tvorbe bol použitý gcc 3.3, gnuplot 3.3.7, saxon 6.4.4, fop 0.20.4, docbook 4.2.

5. Záver

Na základe veľkého množstva experimentálnych meraní som zistil:

1. Ak požadujeme, čo najrýchlejšie výsledky problému a presnosť riešenia nie je rozhodujúci faktor,

volíme veľmi rýchle **heuristické metódy**. Pri dostatočne veľkom množstve inštancií majú relatívne malé maximálne chyby.

2. Ak požadujeme exaktné riešenie problému, uprednostníme **metódu hraníc a vetiev** v prípade, že množstvo inštancií problému nepresahuje 25-27, pre problémy s väčším množstvom inštancií volíme **dynamické programovanie**, ktorá pri veľkom množstve inštancií problému vďaka svojmu lineárnemu spomaľovaniu s rastúcou celkovou cenou v batohu, dosahuje najlepšie výsledky.