

3. Experimentální hodnocení kvality algoritmů

Ladislav Martínek

1 Zadání úlohy

- 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 navrhnete a proveďte experimentální vyhodnocení kvality řešení a výpočetní náročnosti
- Zkoumejte zejména následující metody
 1. hrubá síla (pokud z implementace není evidentní úplná necitlivost na vlastnosti instancí)
 2. metoda větví a hranic, případně ve více variantách
 3. dynamické programování (dekompozice podle ceny a/nebo hmotnosti). FPTAS algoritmus není nutné testovat, pouze pokud by bylo podezření na jiné chování, než DP
 4. heuristika - poměr cena/váha
- Pozorujte zejména závislosti výpočetního času (případně počtu testovaných stavů) a rel. chyby (v případě heuristiky) na:
 1. maximální váze věcí
 2. maximální ceně věcí
 3. poměru kapacity batohu k sumární váze
 4. granularitě (pozor - zde si uvědomte smysl exponentu granularity)
- Doporučuje se zafixovat všechny parametry na konstantní hodnotu a vždy plynule měnit jeden parametr. Je nutné naměřit výsledky pro aspoň čtyři (opravdu minimálně) vhodně zvolené hodnoty parametru, jinak některé závislosti nebude možné vypořádat.

2 Rozbor řešení

Pro určení a sledování citlivosti na různé instance problému jsem využil generátor náhodných instancí, u kterého lze nastavovat jednotlivé parametry. U instancí problému jsou nastavovány parametry jako granularita, maximální cena, maximální váha a poměr sumární váhy ke kapacitě batohu. Pokusím se odhadnout chování algoritmů při změnách parametrů jednotlivých instancí. Tedy sledovat citlivost algoritmů na dané parametry.

2.1 Metoda hrubé síly

Metoda hrubé síly nebude v tomto experimentu zkoumána, protože je zřejmé, že pokaždé projde všechny instance a tedy vůbec není citlivá na jiné parametry, kromě parametru n , který ale již by prozkoumán v 1. a 2. úloze.

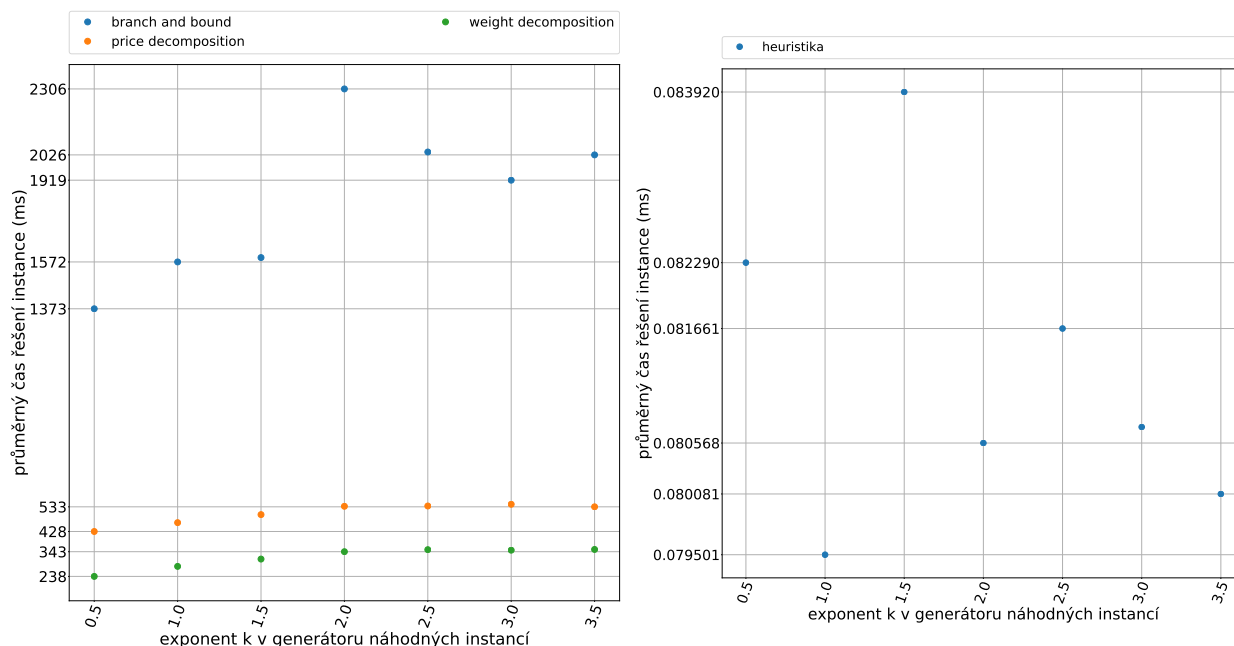


Figure 1: Na grafech je zobrazena časová náročnost jednotlivých metod v závislosti na granularitě instance. Zde jsou uvedeny grafy pro převahu velkých instancí, kde k je exponent z generátoru náhodných instancí, který určuje pravděpodobnost přijetí daného předmětu.

2.2 Metoda větví a hranic (B&B)

U této metody očekávám velkou citlivost na poměr celkové váhy a kapacity batohu, dále by metody mohla ovlivnit granularita. Tato metoda nemá horní mez a proto její čas může narůst až na hodnotu metody hrubé síly.

2.3 Metody dynamického programování (obě dekompozice)

U dekompozic očekávám citlivost vždy na daný parametr, podle kterého je dekompozice prováděna. U dekompozice podle ceny tedy citlivost na maximální cenu a u dekompozice podle váhy na maximální váhu.

2.4 Řešení heuristikou poměr cena/váha

Očekávám, že heuristická metoda bude datově citlivá a to především na parametry jako poměr celkové váhy a kapacity batohu nebo granularita. Vliv maximální ceny a váhy neočekávám.

3 Popis kostry algoritmu

Všechny algoritmy a průběh experimentu, zůstali stejné jako v úloze 2. Byli pouze změněny soubory s instancemi, které byli vygenerovány před experimentem.

4 Experimenty

Experimenty jsem prováděl v režimu jednoho vlákna na starším datovém serveru v podobě starého notebooku, který v době výpočtu nebyl používán. Výsledky tedy nejsou ovlivněny jinými běžícími programy. Procesor na testovacím stroji: *Intel Pentium T3400 (2 cores)*. *Taktován na 2.16 GHz s 1 MB cache*. Měření času CPU probíhalo v knihovně *timeit* s tisíci násobným průchodem pro jednotlivá měření pro přesnost měření.

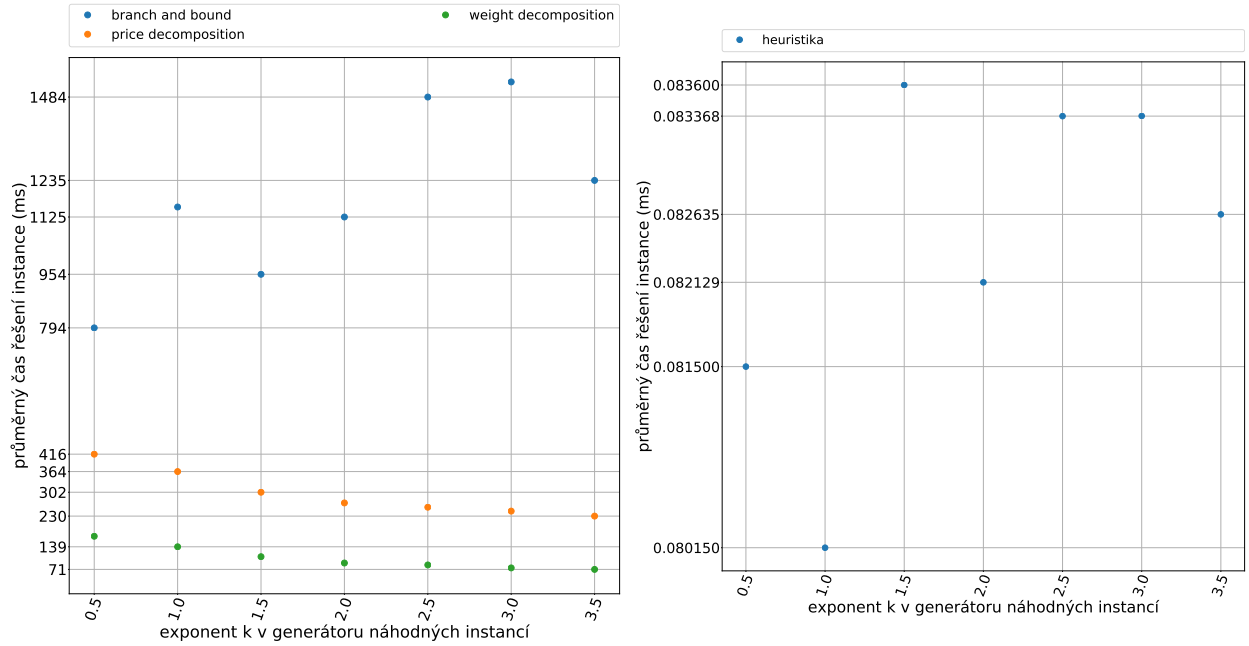


Figure 2: Na grafech je zobrazena časová náročnost jednotlivých metod v závislosti na granularitě instance. Zde jsou uvedeny grafy pro převahu malých instancí, kde k je exponent z generátoru náhodných instancí, který určuje pravděpodobnost přijetí daného předmětu.

Pro každý parametr bylo vygenerováno 100 instancí o počtu předmětů 27. Ostatní základní parametry jsem uvedl v tabulce 1. Při jednotlivých experimentech byl vždycky měněn jeden parametr. Na grafech jsem vždy rozdělil exaktní metody a heuristiku z důvodu nízkých časů u heuristiky, u které by nebyly rozdíly s exaktními metodami vůbec ukázané.

Parametr	Hodnota
Poměr kapacity batohu k sumární váze	0.5
Maximální váha věci	50
Maximální cena věci	50
Exponent k	1
Velké nebo malé věci v batohu	0

Table 1: Tabulka základního nastavení parametrů generátoru.

4.1 Závislost na granularitě instance

První jsem se podíval na granularitu instancí. Ta je řízena následovně. V generátoru je možno ovlivnit, zda instance bude obsahovat spíše malé nebo spíše velké věci. Pro převahu malých věcí je pravděpodobnost p , že věc s váhou w bude v instanci zahrnuta

$$p = \frac{1}{w^k}$$

. Kde k je volitelný exponent. Pro převahu velkých věcí platí symetrický vztah

$$p = \frac{1}{(w_{max} - w)^k}$$

Na grafu 1 můžeme vidět porovnání pro převahu velkých věcí. Lze vidět, že pro metody větví a hranic je zde jasně patrný trend. S větším počtem velkých předmětů tedy roste její výpočetní náročnost. Tento trend, avšak ne tak patrný, může být pozorován i na jednotlivých dekompozicích.

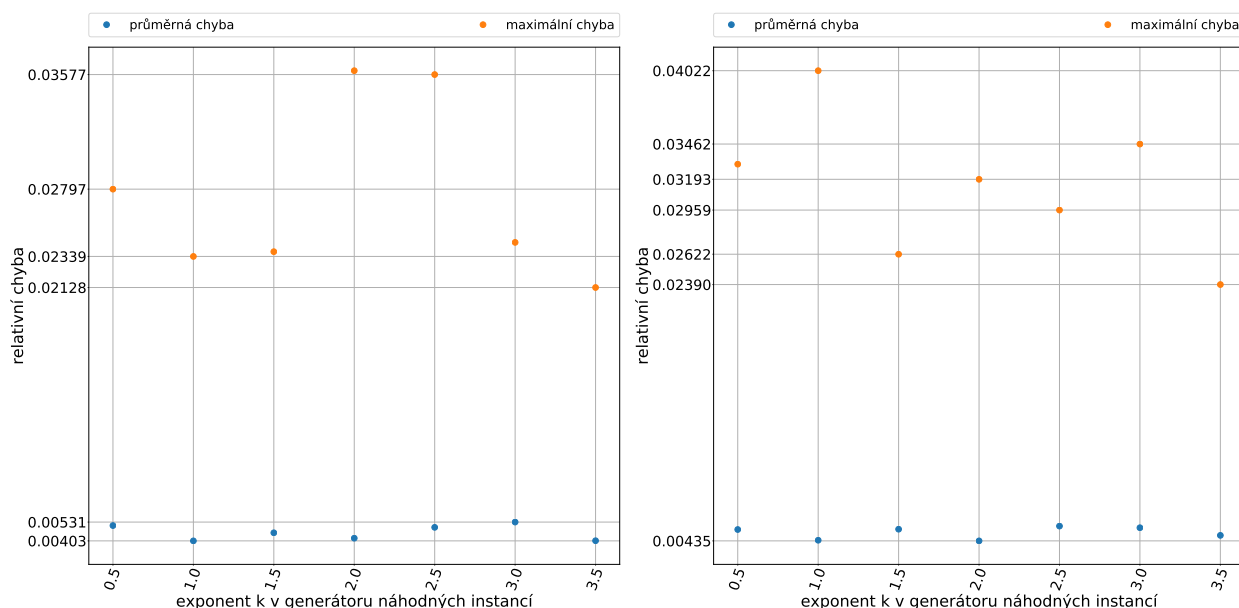


Figure 3: Na grafech je ukázána vždy průměrná a maximální chyba heuristiky v závislosti na granularitě instance. Na levém grafu pro převahu malých instancí a na pravém grafu pro převahu velkých instancí.

Na grafu 2 vidíme porovnání pro převahu malých věcí. I zde jsou patrné citlivosti jednotlivých exaktních metod. Ale mírně odlišně než u převahy velkých věcí. U metody větví a hranic je stále s převahou menších instancí roste její časová náročnost. Naopak u dekompozicích tento čas klesá.

Citlivost u dekompozic bych si také mohl vysvětlit jako citlivost na celkovou sumu váhy nebo ceny podle druhu dekompozice. Protože převaha velkých nebo malých předmětů tuto sumu ovlivní. Citlivost na tyto parametry prozkoumáme v další podkapitole.

U heuristické metody na grafech můžeme vidět veliký rozptyl hodnot. Její závislost na granularitě se tady nepotvrdila. O této metodě bych tedy neřekl, že její výkonost ovlivní granularita jednotlivých instancí.

Na grafech 3 je ukázána průměrná a maximální chyba heuristické metody při změnách granularity instancí. Zde není patrná téměř žádná závislost chyby na granularitě instance. Při podrobnější zkoumání hodnot maxim, by možná bylo možné říct, že maximální chyba klesá s převahou malých i velkých předmětů, ale tento trend by bylo nutné podrobněji prozkoumat a podrobit mnohem větší analýze.

4.2 Závislost na poměru součtu vah předmětů k nosnosti batohu

Pro tento parametr jsem očekával největší citlivost u metody větví a hranic. Tato citlivost se potvrdila a je patrná na grafu 4. Je tedy zjevné, že metoda selhává v momentě, kdy se poměr pohybuje mezi hodnotami 0.2 - 0.5. Také je na grafu vidět minimální čas pro velké hodnoty a minimální čas pro hodnotu 1. Tento jev je očekávaný z implementace. V rekurzivním volání se snažím předmět přidávat dokud nedojde k prořezu a v tomto případě končí po přidání všech předmětů. Generování instance se tedy skládaly z velkého množství předmětů a najít řešení bylo přímočařejší.

Pro dynamické programování se výpočetní náročnost také zvyšuje. Toto může být způsobeno tím, že přestává být tak účinné prořezání a je nutné projít více větví rekurze.

Pro heuristiku je také patrný trend, že výpočetní čas roste s poměrem, tedy roste náročnost této metody s poměrem. Zajímavý je tento výsledek ve srovnání s chybou.

Pro chybu na grafu 5 je jasně patrný klesající trend, tedy že chyba je závislá na poměru sumární váhy ke kapacitě batohu. Čím vyšší je poměr, tím menší je chyba, protože klesá i pravděpodobnost

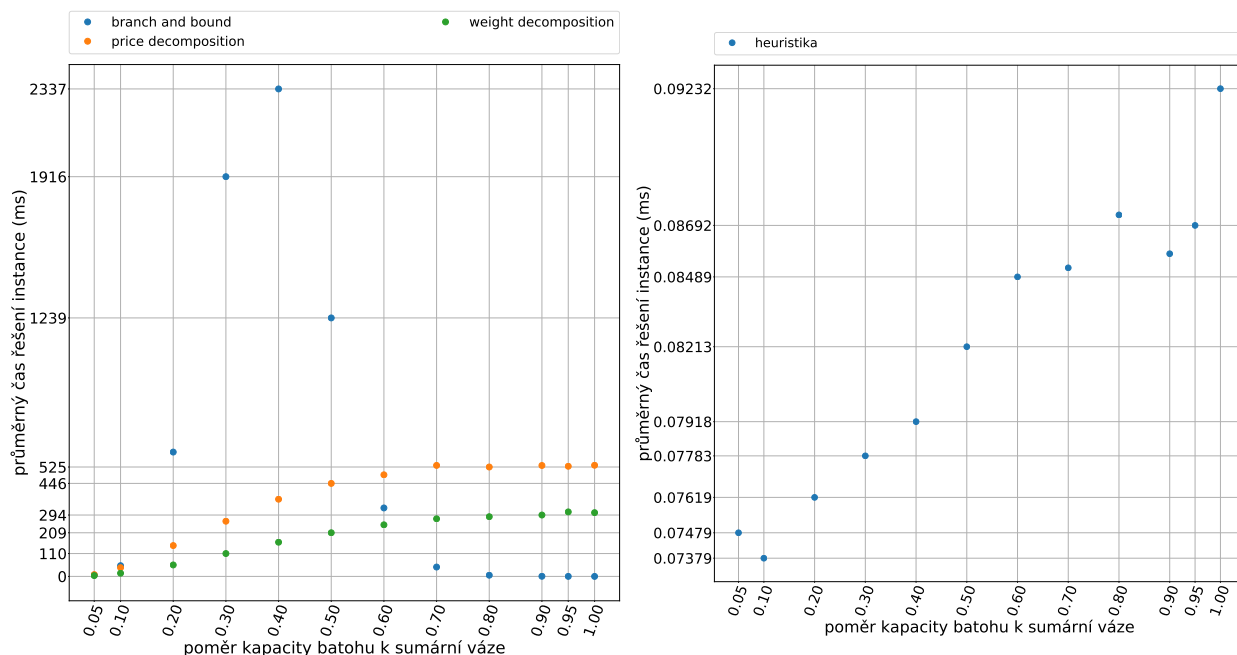


Figure 4: Na grafech je vidět závislost výpočetní náročnosti na poměru součtu vah a nosnosti batohu. Vlevo pro exaktní metody a vpravo pro heuristiku.

toho, že předmět nebude v řešení. Nulová chyba pro poměr 1 je očekávána, neboť jsou přidány všechny předměty do batohu.

4.3 Závislost na maximální ceně předmětů

Z grafu 6 je vidět závislost výpočetní náročnosti na maximální ceně věcí. Zde je opět vidět očekávaný nárůst složitosti u dekompozice podle ceny. Výpočetní náročnost je závislá na tomto parametru, z důvodu akolování prostředků pro tabulku dynamického programování. Tabulka má rozměr v závislosti na součtu vah všech předmětů a počtu předmětů.

Na grafu heuristiky je patrný trend, ale ve velmi malém rozsahu časů, ale i tak s roztoucí váhou to vypadá, že se snižuje výpočetní náročnost. Tuto závislost by bylo nutné prozkoumat podrobněji na větších instancích, zdali se tento trend stále projevuje.

Na grafu 8 vpravo je zobrazena relativní chyba v závislosti na parametru maximální ceny. Z velkého rozptylu v grafu je patrné, že chyba není závislá na maximální váze.

4.4 Závislost na maximální váze předmětů

Závislost na maximální ceně je zobrazena na grafu 7. Je zde jasně patrné, že na tomto parametru je závislá dekompozice podle váhy, kde její čas velice rychle nárůstá. Ostatní exaktní metody nejsou závislé na maximální váze předmětu.

Heuristika z grafu vypadá, že na maximální váhu citlivá není. Ale test nebyl proveden v tak velkém rozsahu, aby bylo možné stanovit jasný závěr.

Na grafu 8 vlevo je zobrazena relativní chyba v závislosti na parametru maximální váhy. Z velkého rozptylu v grafu je patrné, že chyba není závislá na maximální váze.

5 Závěr

Během experimentu jsem otestoval velké množství instancí s různými parametry a sledoval citlivost algoritmů na tyto parametry. Byly pozorovány časy exaktních algoritmů a u heuristiky byla také měřena relativní chyba v závislosti na změnách konkrétních parametrů instancí.

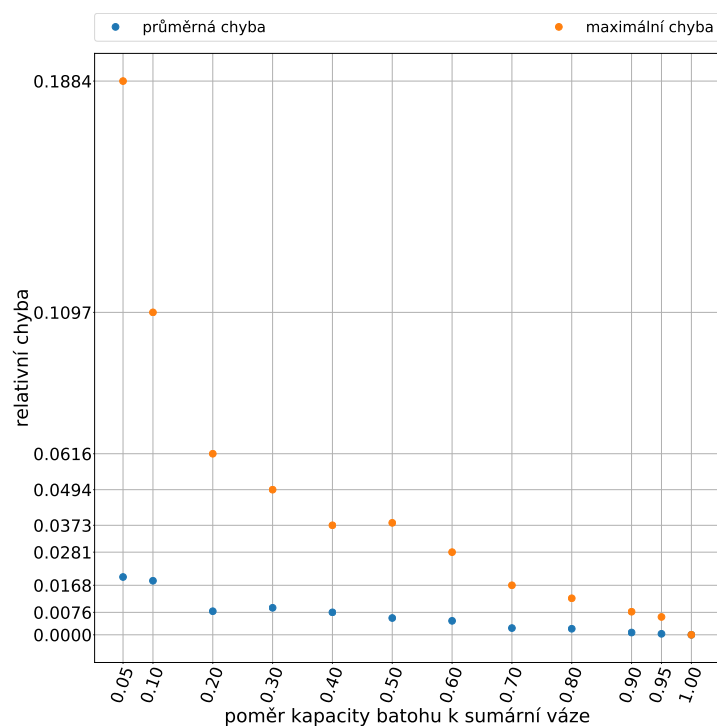


Figure 5: Relativní chyba heuristiky v závislosti na poměru součtu vah a nosnosti batohu.

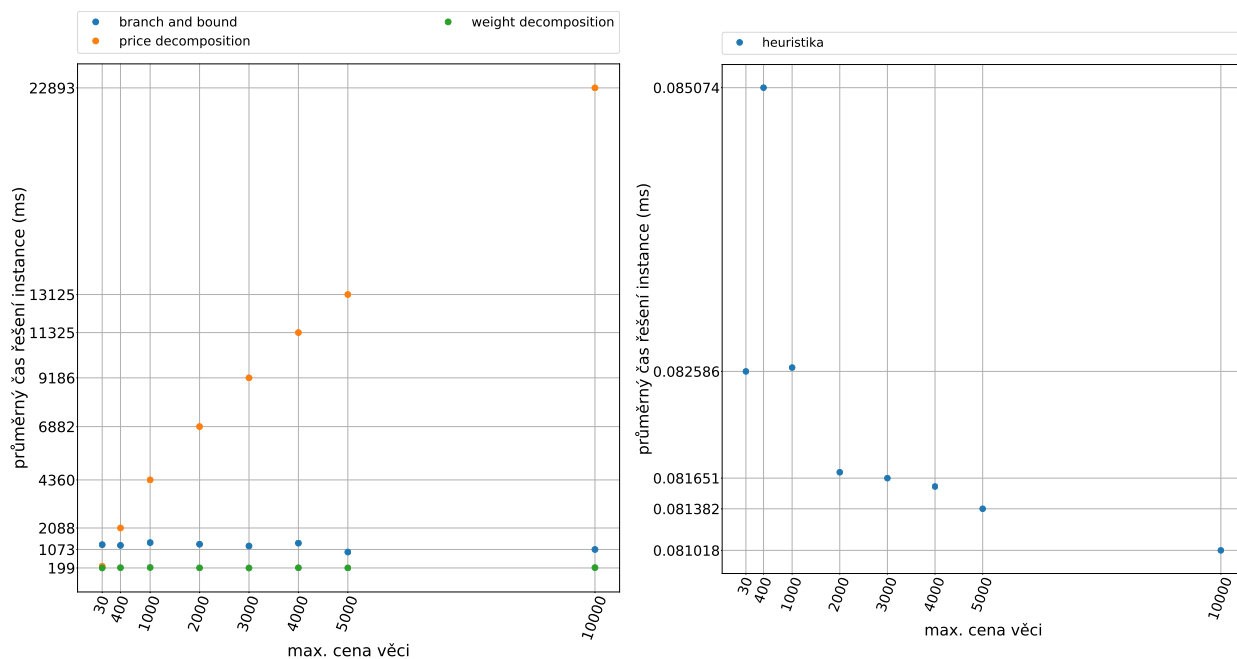


Figure 6: Na grafech je vidět závislost výpočetní náročnosti na maximální ceně předmětu. Vlevo pro exaktní metody a vpravo pro heuristiku.

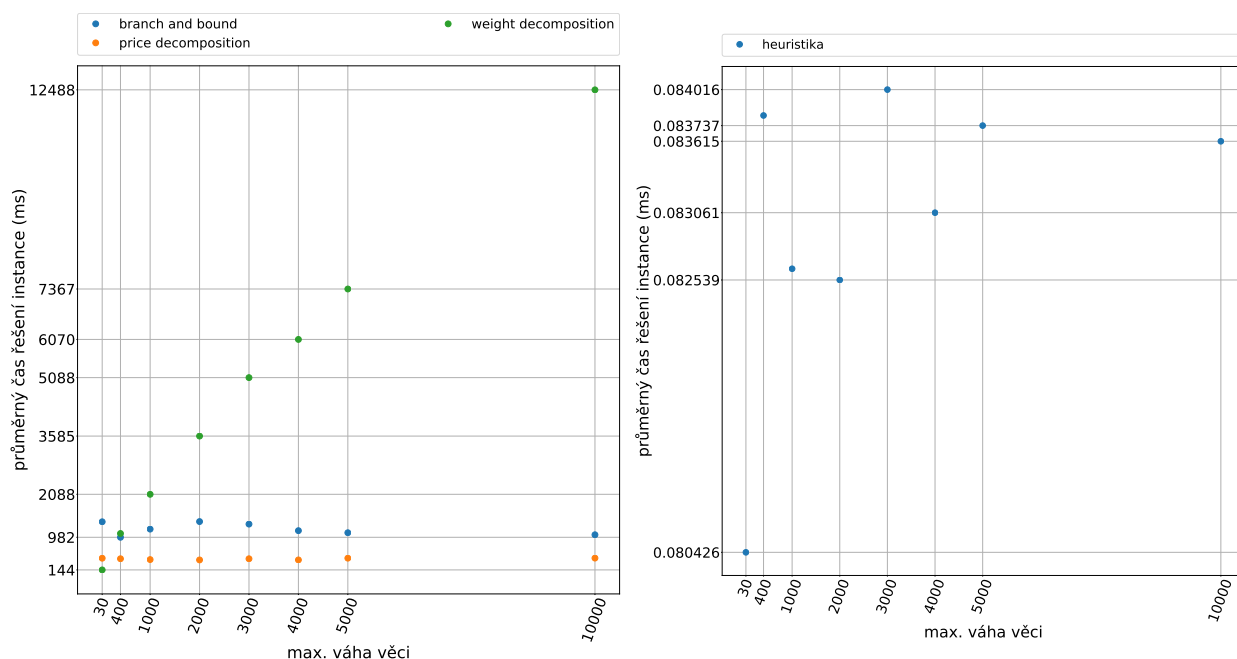


Figure 7: Na grafech je vidět závislost výpočetní náročnosti na maximální váze předmětu. Vlevo pro exaktní metody a vpravo pro heuristiku.

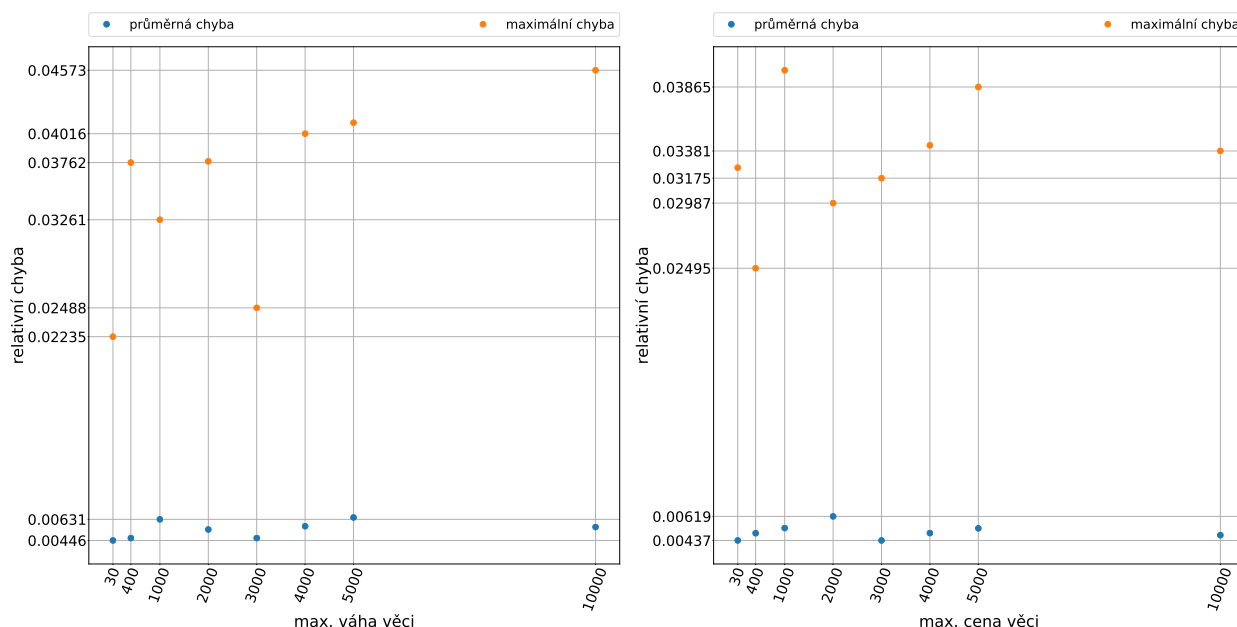


Figure 8: Na grafech je ukázána vždy průměrná a maximální chyba heuristiky v závislosti maximální váze předmětů (vlevo) a maximální ceně předmětů (vpravo). Na levém grafu pro převahu malých instancí a na pravém grafu pro převahu velkých instancí.

Podle předpokladů se ukázala metoda větví a hranic citlivá na poměr sumy vah a kapacity batohu, kde až do hodnoty 0.4 rostla výpočetní náročnost a poté až do 1 klesala. Až pro hodnotu 1 spadl čas na minimum, protože metoda přidá všechny předměty. Na tento poměr se také ukázala být citlivá heuristická metoda, výpočetní náročnost s poměrem narůstala a relativní chyba ta s rostoucím poměrem klesala.

V algoritmech se také projevuje nepatrně koeficient granularity, ale pouze minimálně. Na relativní chyby heuristiky vliv nemá.

Dále se projevila závislost jednotlivých dekompozicích vždy na parametru podle kterého je prováděna. Tato znalost by nám mohla při analýze problému pomoc vybrat správný algoritmus pro minimalizaci výpočetní náročnosti.

Velice zajímavé bylo oproti úkolu 2, že se metoda větví a hranic vykazovala větší náročnost než dynamické programování, oproti poznatku z úkolu 2 na referenčních instancích. Tento poznatek bych připsal zvolenému základnímu poměru součtu vah předmětů k nosnosti batohu. Ten činil 0.5 a tato metoda se na tento poměr ukázala velice citlivá. Pro hodnoty v určitém rozsahu její čas velice narůstal. Pokud by se v referenčních instancích nacházely takové, které mají poměr 0.4, tato metoda by určitě tak dobře z předchozího testu nevyšla.

Náměřené hodnoty však slouží pouze jako podpůrné argumenty i přes větší množství instancí a velké množství opakování. Míra důvěry v naměřené data však souvisí na způsobu provádění experimentu a generovaných datech.