

### 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 provedte 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 které 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.

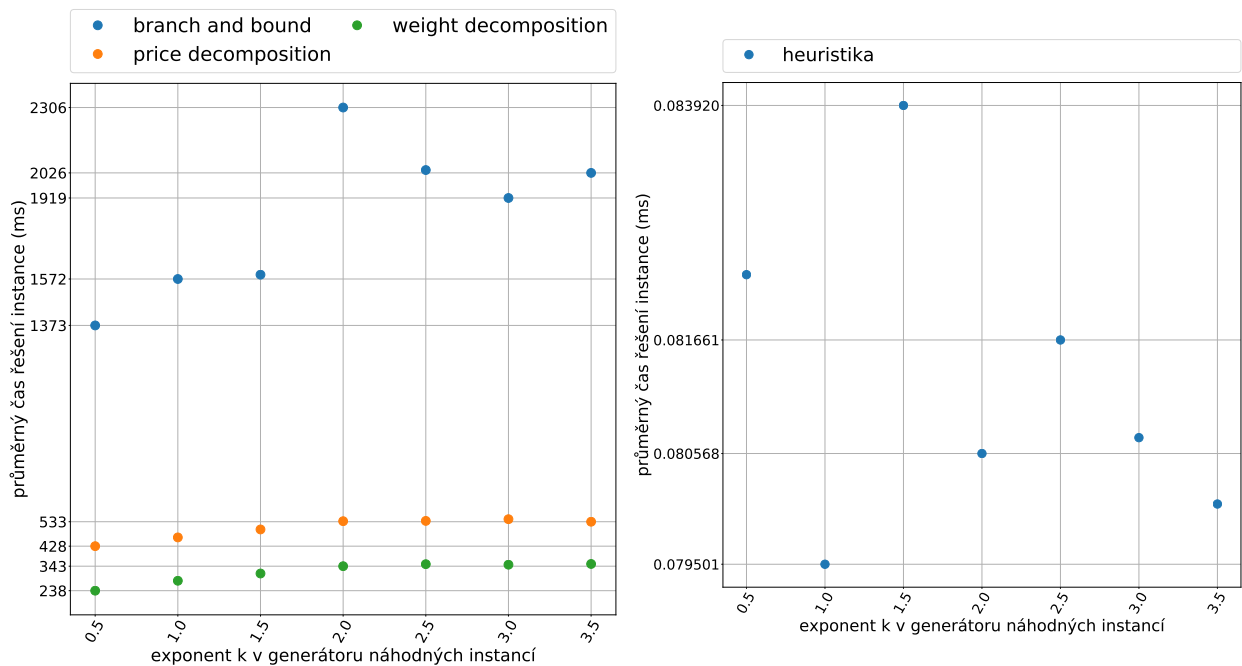


Figure 1: Na grafech je zobrazena časová náročnost jednotlivých metod v závislosti na granularitě instance. Zde jsou uvedeny grafy pro převahu velkých instancí, kde  $k$  je exponent z generátoru náhodných instancí, který určuje pravděpodobnost přijetí daného předmětu.

## 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 metody hrubé síly.

## 2.3 Metody dynamického programování (obě dekompozice)

U dekompozic očekávám citlivost vždy na daný parametr. 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 byly 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 tidící násobným průchodem pro jednotlivá měření pro přesnost měření.

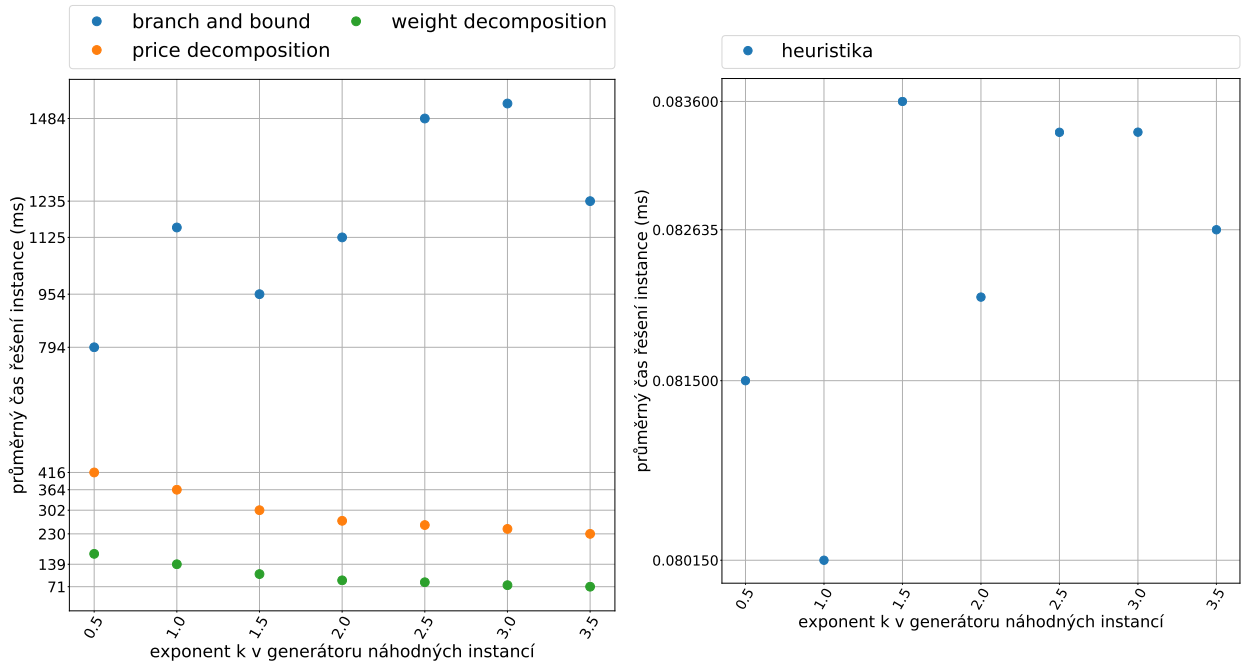


Figure 2: Na grafech je zobrazena časová náročnost jednotlivých metod v závislosti na granularitě instance. Zde jsou uvedeny grafy pro převahu malých instancí, kde  $k$  je exponent z generátoru náhodných instancí, který určuje pravděpodobnost přijetí daného předmětu.

Pro každý parametr bylo vygenerováno 100 instancí o počtu předmětů 27. 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.

#### 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.

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.

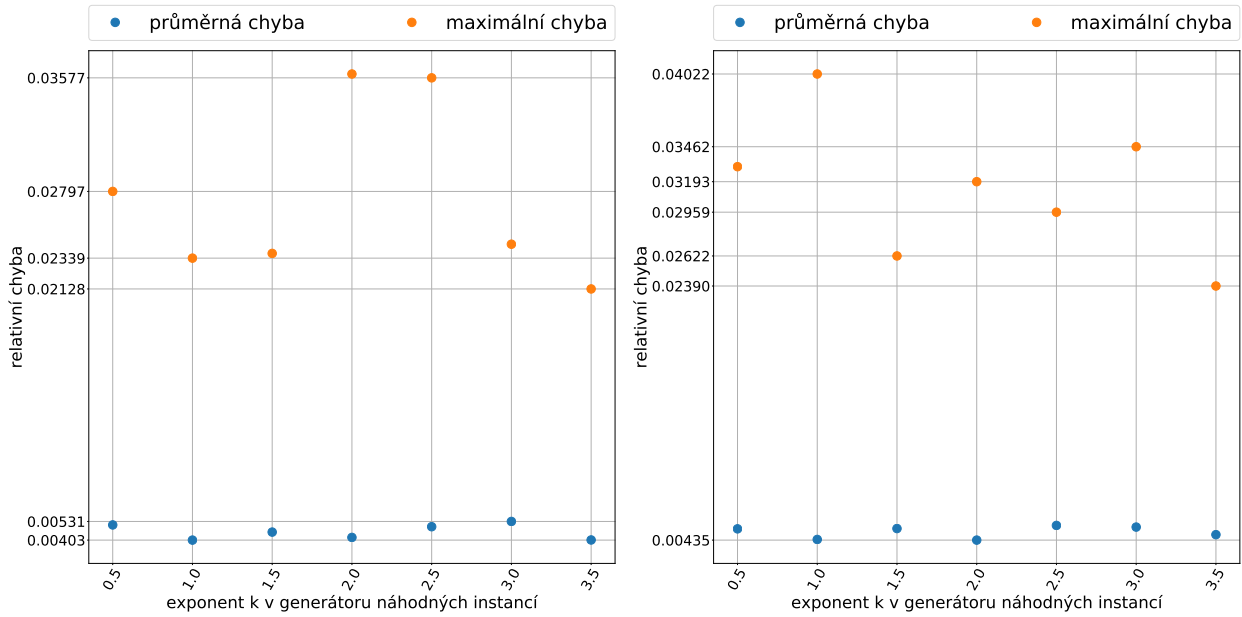


Figure 3: Na grafech je ukázána vždy průměrná a maximální chyba heuristiky v závislosti na granularitě instance. Na levém grafu pro převahu malých instancí a na pravém grafu pro převahu velkých instancí.

Na 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

#### 4.3 Závislost na maximální ceně předmětů

#### 4.4 Závislost na maximální váze předmětů

### 5 Závěr

Během experimentu jsem otestoval velké množství instancí s různými parametry a sledoval citlivost algoritmů na tyto parametry. Byli pozorovány časy exaktních algoritmů a u heuristiky byla také měřena relativní chyba.

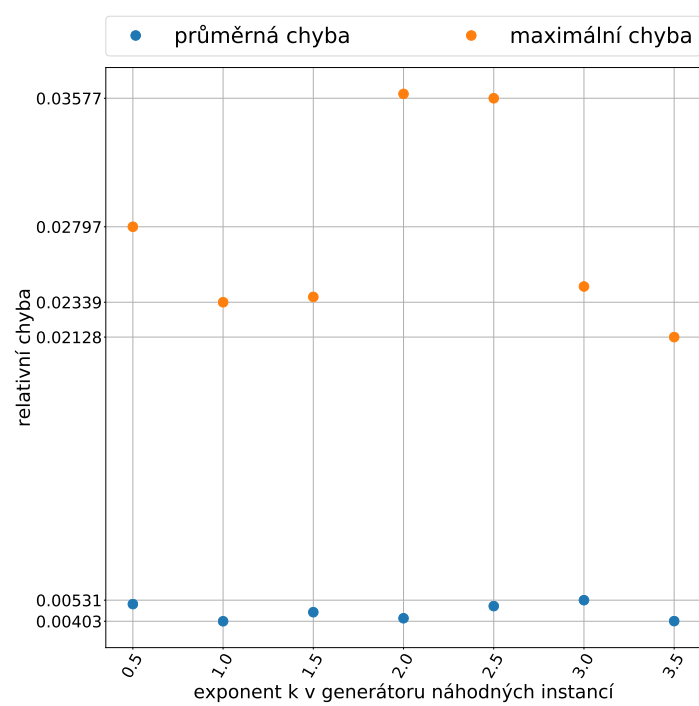


Figure 4: Brute-force ve srovnání s dynamickým programováním s dekompozicí podle ceny. Časová náročnost. Na grafu jsou průměrné hodnoty.