# 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 navrhněte 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 vypozorovat.

## 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 granulalita, 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

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

### 2.2 Metoda větví a hranic (B&B)

U této metody očekávám velkou čitlivost 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 paramentry 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 několika násobným průchodem pro menší instance. Pro každý parametr bylo vygenerováno 100 instancí.

### 4.1 Závislost na poměru součtu vah předmětů k nosnosti batohu

### 4.2 Závislost na maximální ceně předmětů

### 4.3 Závislost na maximální váze předmětů

### 4.4 Závislost na granularitě instance

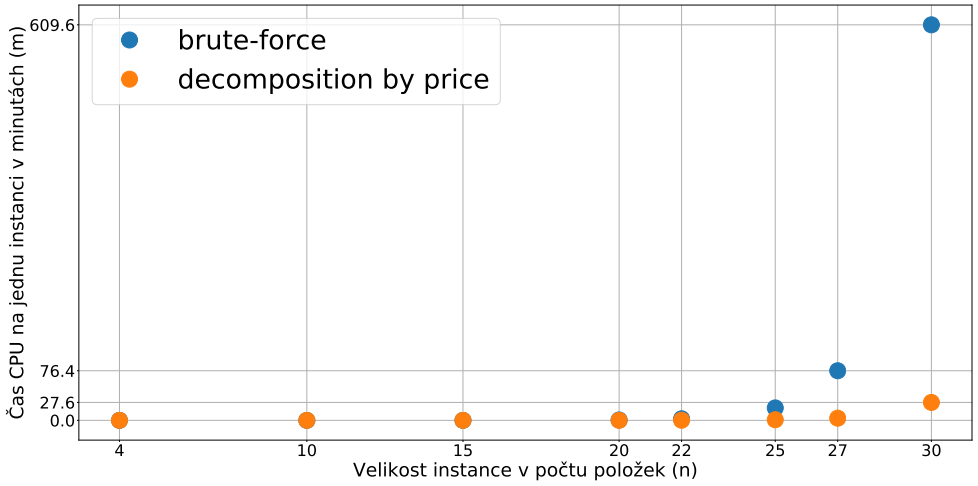


Figure 1: Brute-force ve srovnání s dynamickým programováním s dekompozicí podle ceny. Časová náročnost. Na grafu jsou průměrné hodnoty.

# 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.