Úkol č. 2 z předmětu MI-PAA

Řešení problému batohu dynamickým programováním,
metodou větví a hranic a aproximativním algoritmem

Jan Bělohoubek
belohja4@fit.cvut.cz

27.10.2012
aktualizace 31.10.2015

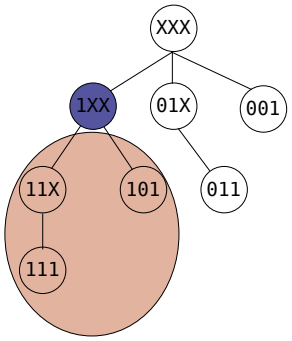
1 Popis použitých metod

1.1 Metoda hrubé síly (BF)

Metoda hrubé síly je založena na průchodu celým stavovým prostorem. Tato metoda zaručuje nalezení nejlepšího řešení, ale pracuje v exponenciálním čase.

1.2 Metoda větví a hranic (B&B)

Metodu hrubé síly je možno optimalizovat tak, že se neprochází celý stavový prostor, ale vynechávají se části, o kterých je známo, že nemohou obsahovat řešení. To je vidět na obrázku 1.



Obrázek 1: Pokud je označený stav nevyhovující z pohledu hmotnosti, je jasné, že přidávání dalších věcí do batohu nemá smysl, protože povede k ještě většímu přetížení batohu. Stejně tak pokud ani po přidání všech (v dané větvi povolených) věcí do batohu nebude celková cena věcí umístěných v batohu větší než cena nejlepšího nalezeného řešení, nemá také smysl zkoušet přidávat do batohu další věci. Tudíž není třeba dále prohledávat následovníky tohoto stavu - vyznačený podstrom.

1.3 Metoda dynamického programování s dekompozicí podle ceny

Metoda dynamického programování řeší problémy tak, že nejprve řeší podproblémy daného problému a z nich je následně složeno řešení složitějších problémů, až je nakonec vyřešen nejsložitější problém. Pro dekompozici podle

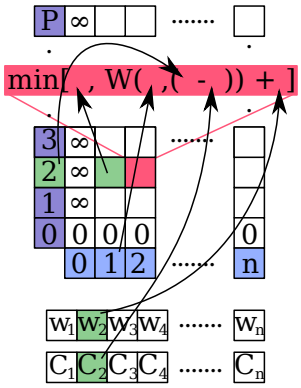
ceny platí, že se nejprve řeší postupně problémy se stanovenou cenou

$$0, 1, 2, ...(P-1), P$$

kde P je suma cen všech věcí, které je možné umístit do batohu. Postup lze realizovat v několika krocích. Mezivýsledky se zapisují do matice W velikosti

$$(n+1) \texttt{ x } (P+1),$$

kde n je počet věcí v batohu.



Obrázek 2: Vyplňovaní matice W.

Prvky matice lze popsat následujícím způsobem:

$$W(i,c),\ \ kde\ i \in < 0, n >\ \ a\ c \in < 0, P >$$

vyplňujeme ji podle následujících pravidel (viz obrázek 2):

- $W(0,0) = 0$
- $W(0,c) = \infty,\ \forall c > 0$
- $W(i+1,c) = min(W(i,c), W(i, c - C_{i+1}) + w_{i+1}),\ \forall i > 0,$ kde C_{i+1} je cena $(i+1)$. věci a w_{i+1} její váha. Pokud $c - C_{i+1} < 0$, pak $W(i, c - C_{i+1}) = \infty$.

Po vyplnění matice W z ní můžeme zkonstruovat vektor řešení takto:

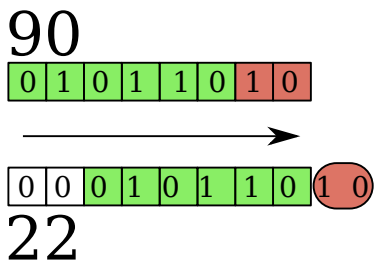
- procházíme sloupec n matice W od indexu P směrem k 0, dokud nenajdeme prvek $W(n,i)$ takový, že $W(n,i) < kapacita\ batohu$
- pokud $W(n,i) == W(n-1,i)$, pak $n - tý$ prvek není součástí řešení.

- pokud jsme prvek přidali do batohu, snížíme *kapacit batohu* o hmotnost přidaného prvku
- pokračujeme stejně ve sloupci $n - 1$, až získáme celé řešení.

`Zdroj: https://edux.fit.cvut.cz/courses/MI-PAA/homeworks/knapsack/dynprog`

1.4 Aproximativní algoritmus (FPTAS) – pouze myšlenka (pozor na chybu!)

Aproximativní algoritmus vychází z metody dynamického programování s dekompozicí podle ceny. Urychlení oproti dynamickému programování se dosahuje redukcí cen věcí. Konkrétní implementace je založena na zanedbání několika nejnižších bitů z ceny (viz obrázek 3). Tím dosáhneme urychlení, ale algoritmus již nenajde zaručeně nejlepší řešení.



Obrázek 3: Posun o dva bity doleva je ekvivalentní zanedbání 2 nejnižších bitů z ceny. Na obrázku je znázorněna redukce ceny „90“ na cenu „22“.