

# Specifikace úlohy

Cílem třetí úlohy bylo zjistit závislost výpočetního času různých algoritmů na ostatních parametrech problému batohu. Ostatními parametry je myšleno:

- W - maximální váha předmětu
- V - maximální cena předmětu
- poměr kapacity batohu k sumární váze předmětů
- k a d - distribuce lehčích/těžších předmětů

Dále byly zadány následující výroky k vyšetření jejich pravdivosti

- Metoda větví a hranic bývá rychlejší, než DP
- Výpočetní náročnost DP může být citlivá na max. cenu
- Výkon metod, které vycházejí ze stavu prázdný batoh, se může lišit od metod, které vycházejí ze stavu plný batoh podle poměru sumární váha ku kapacitě
- Není jasné, jakou roli hraje granularita instance (distribuce váhy předmětů)

## Rozbor možných variant řešení

K vyřešení úlohy je především nutné vygenerovat správná data. K dispozici je parametrický generátor instancí, který se dá použít jako unixový nástroj. Jiné řešení jsem neuvažoval.

## Rámcový popis postupu řešení

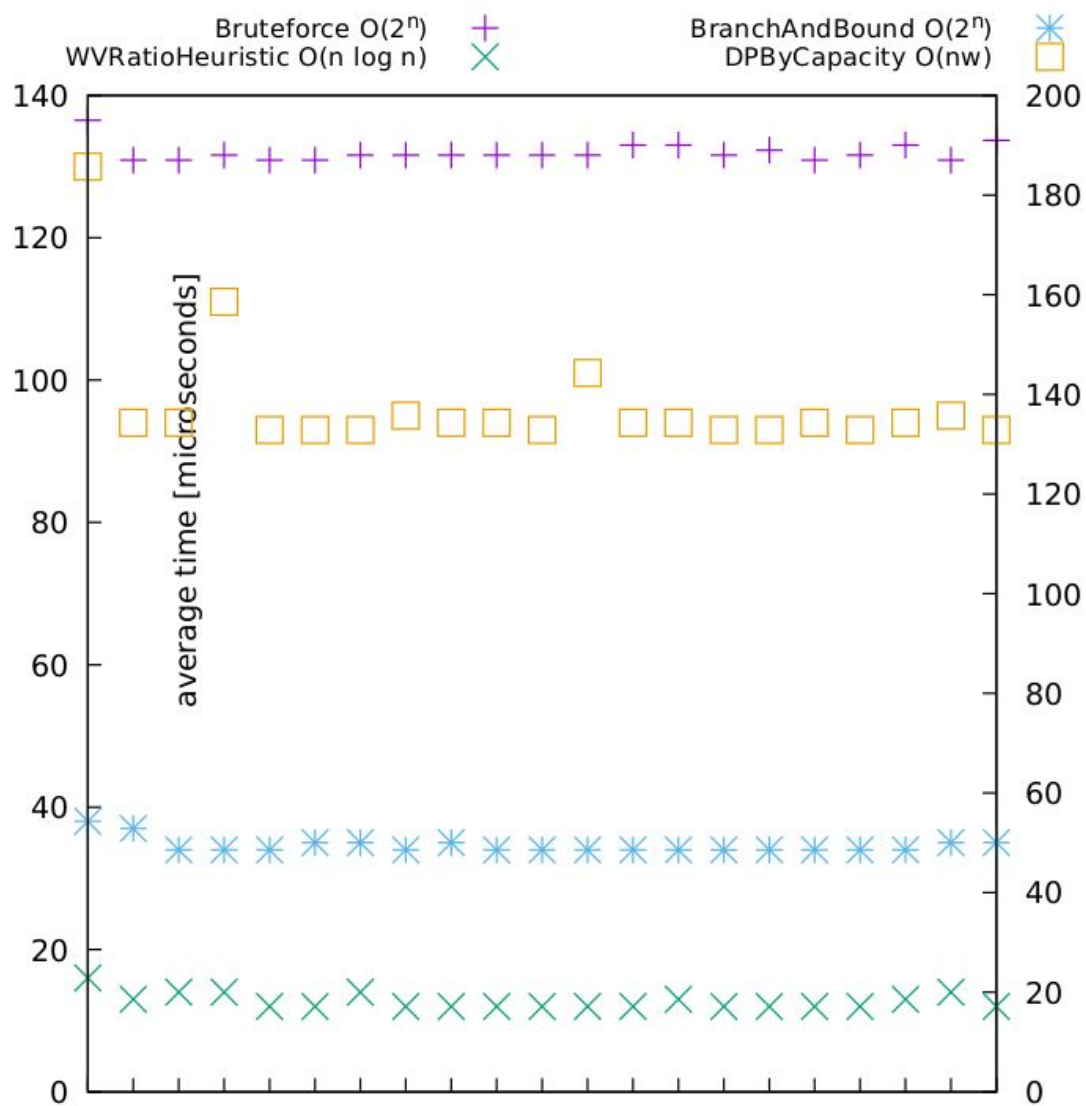
1. Implementace metod pro obsluhu generátoru řešení
2. Hledání vhodného rozsahu hodnot pro jednotlivé parametry generátoru
3. Implementace metod pro automatické vyhodnocení řady generovaných sad instancí
4. Generování grafů
5. Vyhodnocení

## Popis kostry algoritmu

není relevantní/viz předchozí bod

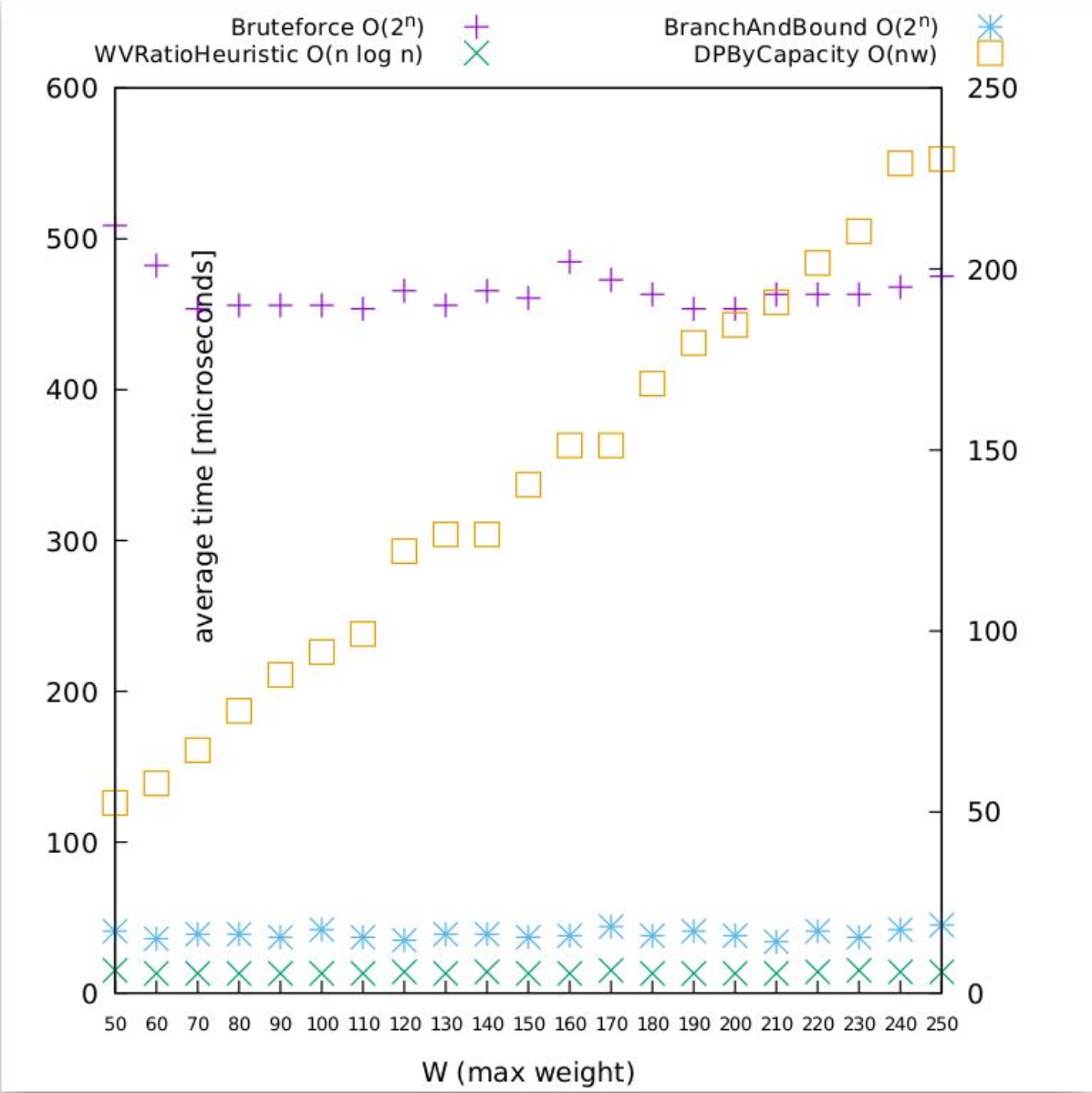
## Naměřené výsledky

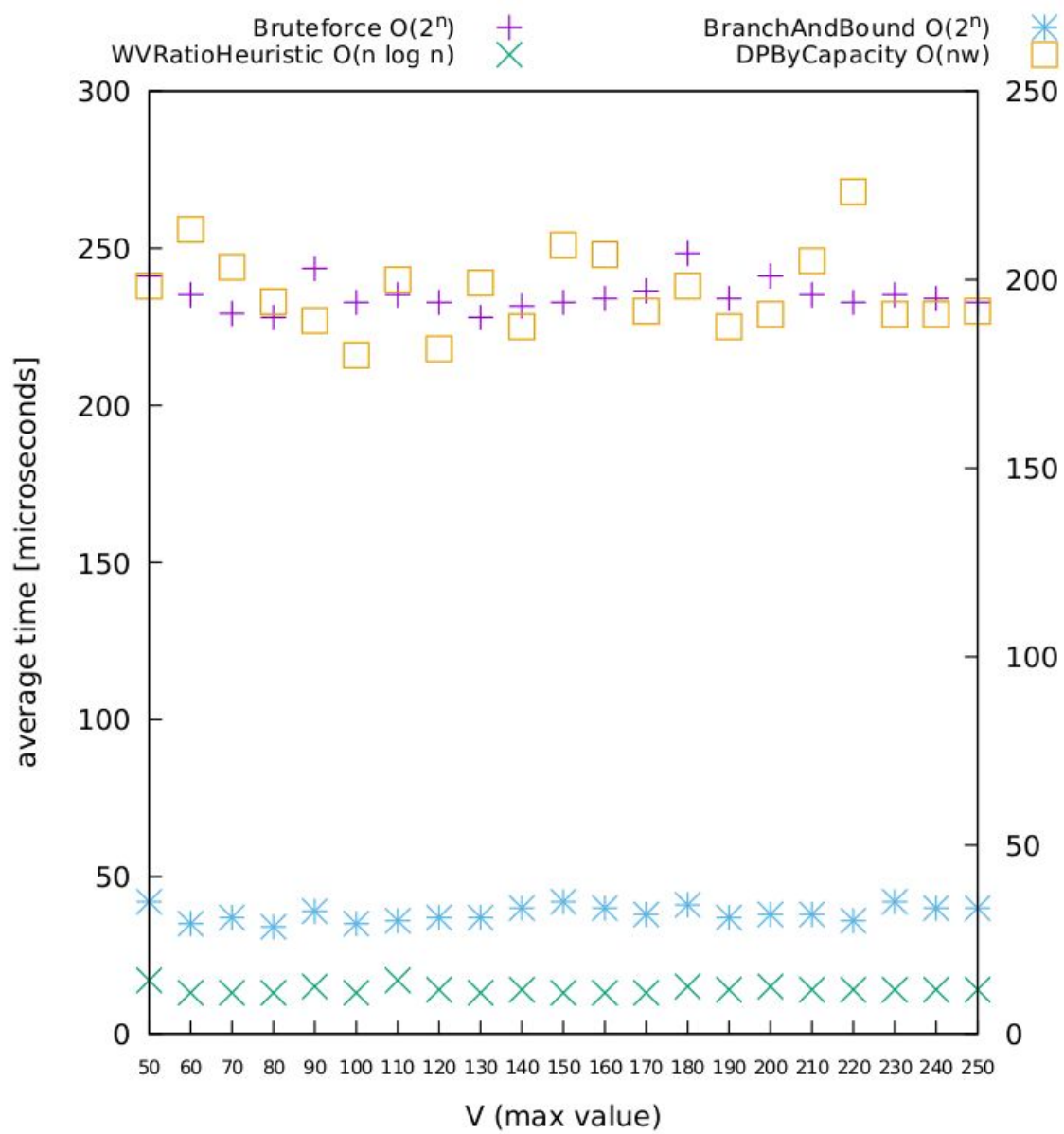
Poznámka k osám y: Pravá osa y náleží metodě Bruteforce, levá osa y všem ostatním.

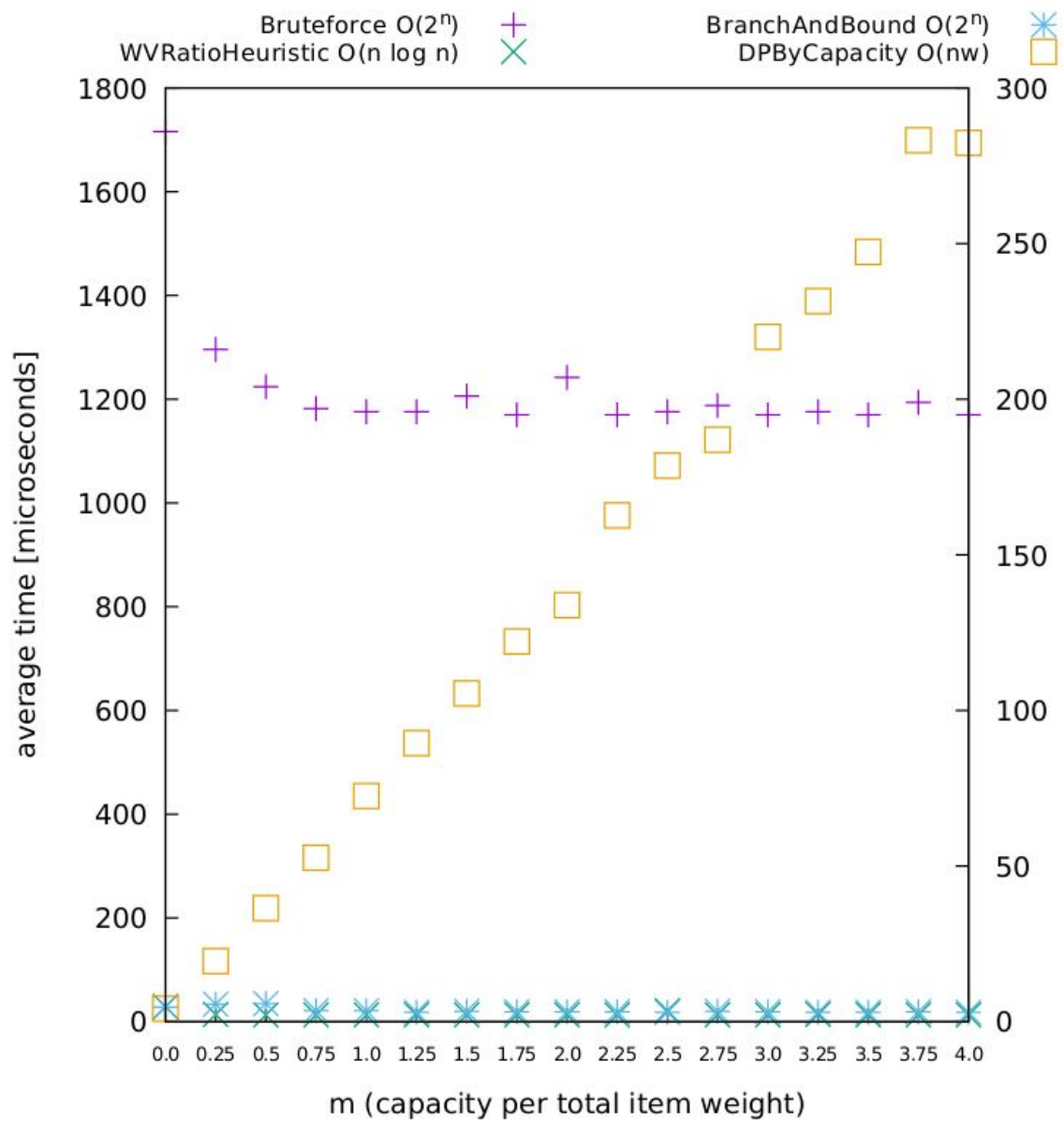


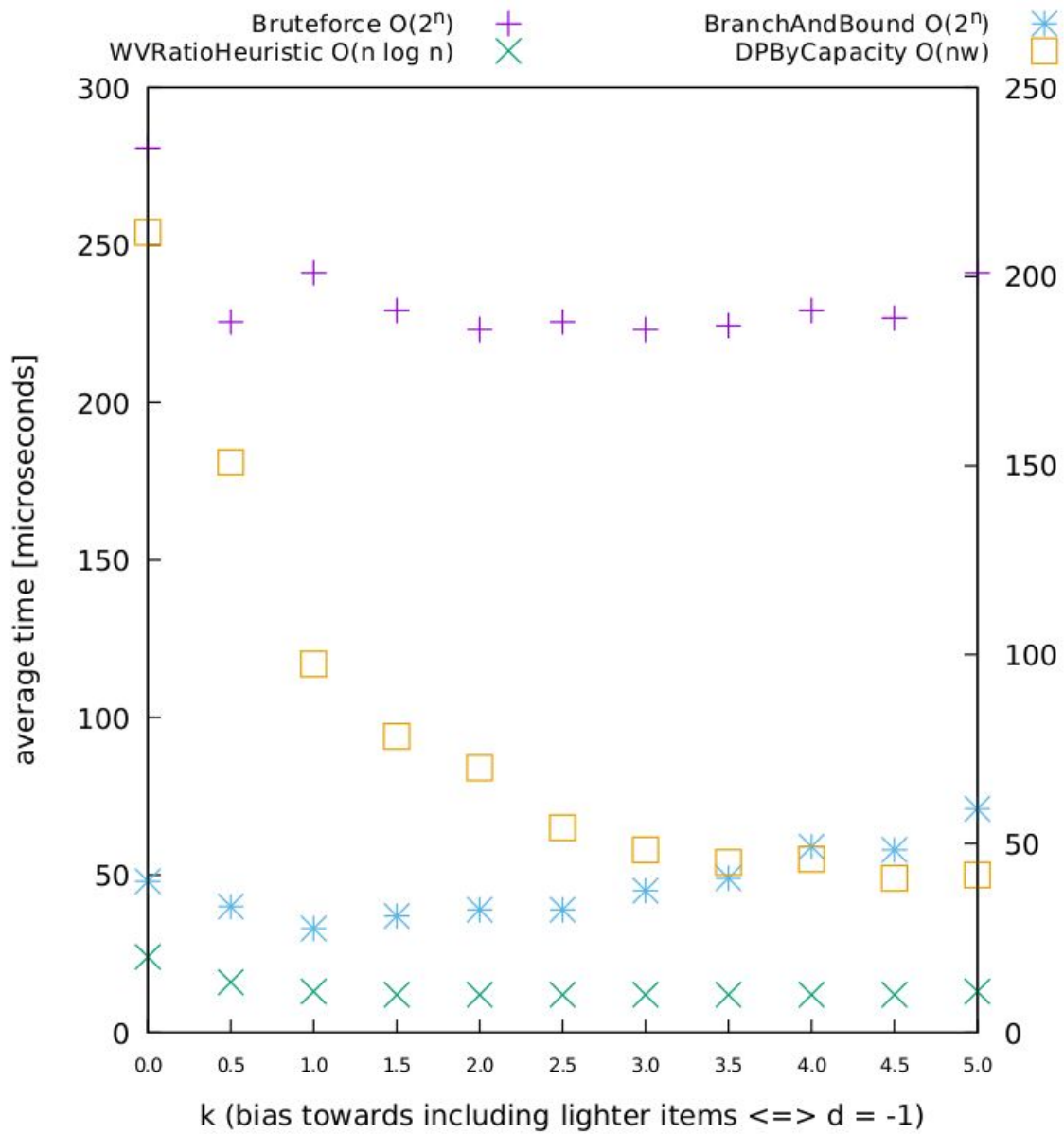
Defaultní parametry( $W=50, V=250, m=0.5, k=0.5, d=0$ )

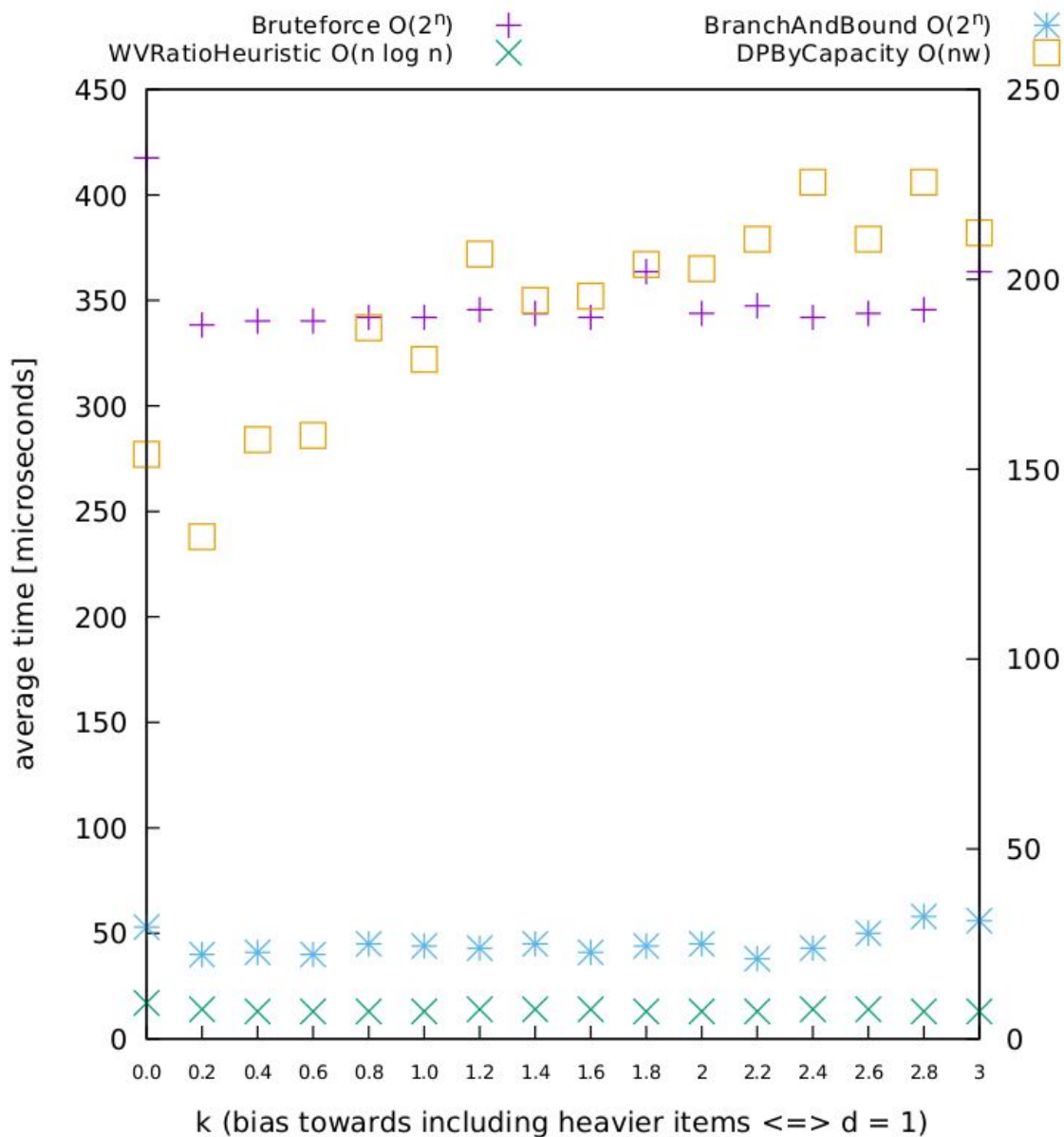
Tento graf slouží jako reference pro porovnání s ostatními, kde se liší daný parametr.











## Interpretace výsledků

- Metoda větví a hranic bývá rychlejší, než DP
- Výpočetní náročnost DP s dekompozicí podle kapacity není citlivá na max. cenu
- Výpočetní náročnost DP s dekompozicí podle kapacity je citlivá na max. váhu
- Tvzení, že výkon metod, které vycházejí ze stavu prázdný batoh, se může lišit od metod, které vycházejí ze stavu plný batoh podle poměru sumární váha ku kapacitě, se nepodařilo ověřit, jelikož žádné metody, které vycházejí ze stavu plný batoh, nebyly testovány.
- Vysoká granularita instance (mnoho lehkých předmětů) snižuje výpočetní náročnost DP s dekompozicí podle kapacity. To dává smysl, jelikož mnoho malých předmětů má nižší sumární váhu, než mnoho velkých předmětů.
- Vysoká granularita instance (mnoho lehkých předmětů) lehce zvyšuje výpočetní náročnost algoritmu BranchAndBound. Jako vysvětlení se nabízí skutečnost, že u

mnoha lehkých předmětů nebudeme tak často ořezávat konfigurace přesahující kapacitu batohu.