Experimentální hodnocení kvality algoritmů

NI-KOP DÚ 3

Tomáš Kalabis

Konstruktivní problém batohu

Je dáno

- celé číslo *n* (počet věcí)
- celé číslo *M* (kapacita batohu)
- konečná množina $V = \{v_1, v_2, \dots, v_n\}$ (hmotnosti věcí)
- konečná množina $C = \{c_1, c_2, \dots, c_n\}$ (ceny věcí)

Je možné zkonstruovat množinu $X=\{x_1, x_2, \dots, x_n\}$, kde každé x_i je 0 nebo 1, tak, aby platilo

$v_1x_1 + v_2x_2 + \dots + v_nx_n \leq M$ (aby batoh nebyl přetížen).

a výraz

$c_1x_1 + c_2x_2 + \dots + c_nx_n$ byl maximální

Výsledkem tak je množina vybraných věcí v batohu, které jsou lehčí než kapacita batohu (číslo *M*) a hodnota všech těchto věcí je nejcennější.

Technologie a Program

Jako hlavní technologie byl vybrán jazyk *Python* verze 3.9.5 pro svou uživatelskou přívětivost, jednoduchou práci s moduly a jejich bohatou podporu jako je například *matplotlib* (modul pro vykreslování grafů) nebo *numpy* (modul pro matematické výpočty). Pro programování bylo použito prostředí *Jupyter Notebook*. Nevýhodou využití jazyka *Python* je jeho nižší rychlost oproti jazykům jako *C++*.

Program se spouští vyhodnocením všech buněk v Jupyter Notebooku v pragramu `./knapsackProblem-03.ipynb`.

Řešení a metody

Konstruktivní Problém batohu byl řešen několika metodami, které byly následně experimentálně testovány. Tato kapitola popisuje využité metody.

Metoda Hrubé síly

Tato metoda byla řešena pomocí for cyklu, kde bylo číslo iterace cyklu které v binární soustavě reprezentuje instanci jako konfigurační proměnná. Je tedy zřejmé, že tato metoda nebude nijak náchylná k výkyvům kvality. Její časová náročnost je $\Theta(2^n)$.

Metoda Větví a hranic (Branch and Bound)

Metoda je implementovaná jako stromová rekurze v jehož každém uzlu se kontroluje, zda podstrom tohoto uzlu je schopen naplnit batoh tak, aby dosavadní maximální cena plnění batohu byla překonána. V negativním případě se podstrom tohoto uzlu vůbec nezkouší, jelikož víme, že nová maximální cena batohu v tomto podstromě nebude nalezena. Časová náročnost této metody je $O(2^n)$.

Greedy Metoda

Tato metoda nejprve před samotným výpočtem seřadí předměty v batohu podle poměru cena/hmotnost v klesajícím pořadí. Nejvýhodnější předměty jsou tedy jako první. Následně se provede výpočet, kde algoritmus vkládá předměty do batohu v tomto pořadí a pokud se do batohu nevejde, předmět se vynechá. Metoda má značnou výhodu v její rychlosti, ale je velmi náchylná na chyby a výsledek tak ve většině případů není optimální, avšak únosný. Časová náročnost výpočtu je $O(n)$, ale náročnost metody jako celek je $O(n \cdot \log n)$ kvůli řazení předmětů na začátku.

Greedy Redux Metoda

Tato metoda opravuje jeden nedostatek Greedy metody. Při jejím použití může být řešení menší než jeden nejhodnotnější předmět, který se do batohu ještě vejde. Implementace této metody je tedy maximum z nejhodnotnějšího předmětu, který se do batohu vejde a řešení Greedy metodou. Časová náročnost je stejná jako u Greedy metody.

Dynamická heuristika podle ceny (DHBP)

Tato metoda využívá dynamického programování, konkrétně ukládání mezivýsledků za účelem zamezení opakování stejného podproblému. Aby však šlo využít dynamické programování podle ceny, je třeba přeformulovat problém.

Označme E(i, c) instanci 0/1 inverzního problému batohu se zadanou cenou c, která vznikne z řešené instance omezením na prvých i věcí. Označme dále W(i, c) sumární hmotnost věcí řešené instance. Pak platí

- W(0,0) = 0
- W(0,c) = ∞ pro všechna c > 0
- W(i+1, c) = min(W(i, c), W(i, c-ci+1)+wi+1) pro všechna i > 0.

Z výsledných řešení *W(n, c)* vybereme řešení, pro které je *W(n, c)* < *M* pro největší *c*. Je zřejmé, že řešení můžeme omezit na *c* menší nebo rovné součtu cen všech věcí.

Algoritmus zkouší jednotlivé možné c od maximální možné hodnoty (součet cen předmětů v batohu) po nižší hodnoty c, a jakmile narazí na řešení které je menší než *M*, je toto řešení zároveň maximální a tedy je řešením konstruktivního problému dané instance batohu.

Experimenty

Experimenty probíhaly na vygenerovaných sadách skrze generátor instancí a permutátor instancí. Zkoumali se následující výše zmíněné metody a jejich vlastnosti na nastavení následujících parametrů:

- poměr kapacity batohu k sumární váze
- korelace cena/váha
- rozložení vah a granularitě

Experimenty na metodě Hrubé síly byli vynechány, jelikož implementace této metody je zjevně imunní vůči těmto parametrům. Jednotlivé experimenty byly zkoumány na všech *n* <= *25* aby se potvrdili závěry. Zároveň všechny sady experimentů měli 100 instancí což poskytuje dostatečný statistický vzorek. Implicitní hodnoty generovaných sad byly *W=500 (maximální váha)*, *C=500 (maximální cena)*. Dále se zkoumala robustnost algoritmů.

Testování probíhalo na notebooku s následujícími parametry:

- Operační systém: Ubuntu 21
- RAM: 8GB DDR4
- Procesor: i5-6300HQ 2.3GHz

Výsledky

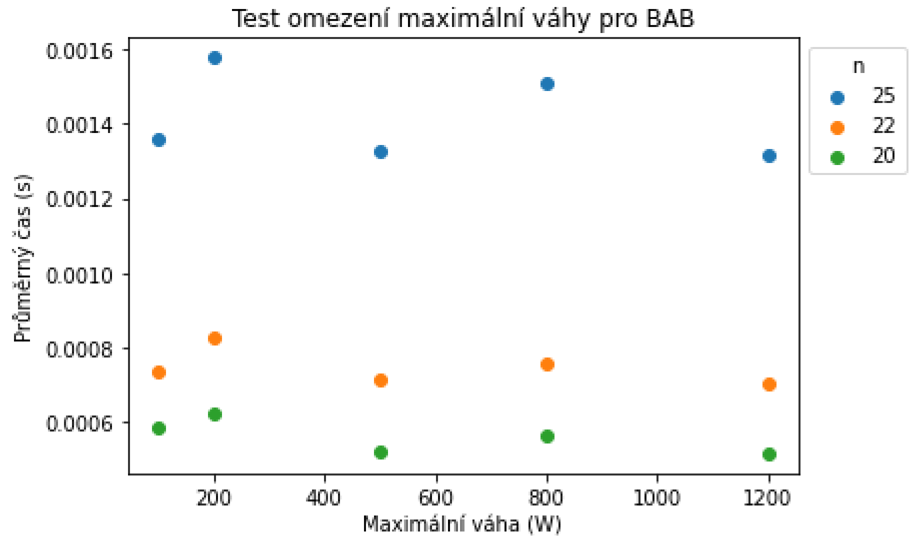
V následující sekci jsou popsané charakteristické vlastnosti jednotlivých metod, zejména jejich závislosti na různých parametrech generovaných instancí. U všech experimentů byli měřeny i závislosti na velikosti instance (*n*) pro ověření, které až na výjimky byli podle očekávání s narůstajícím *n* přibývající čas výpočtu / relativní chyby. Výsledky měřených hodnot jsou uloženy v souboru output.json.

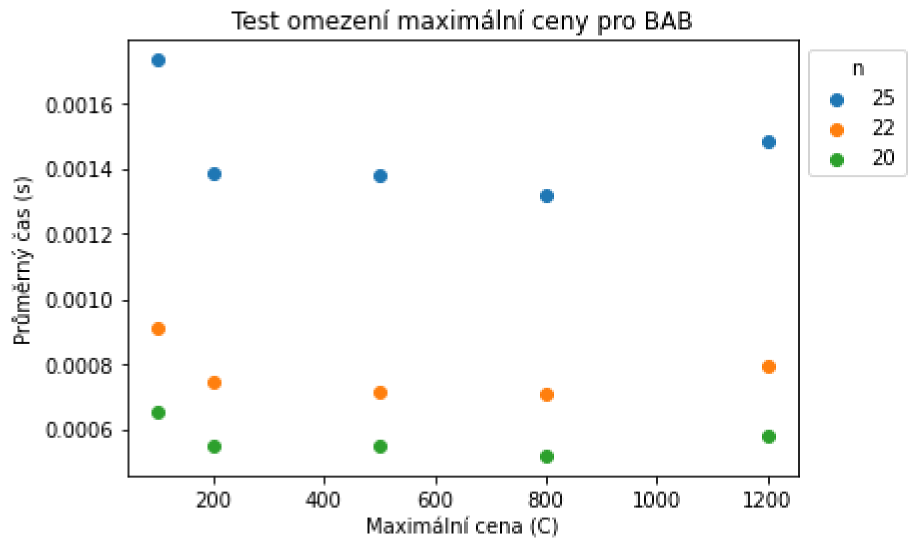
Poznámka: *V grafech jsou metody popsané zkráceně. Zkratky a jejich význam: BAB(Branch and Bound metoda), G (Greedy metoda), R (Redux metoda), DHBP(Dynamická heuristika podle ceny).*

Branch & Bound

Experiment zkoumající průměrný čas v závislosti na omezení maximální ceny a váhy

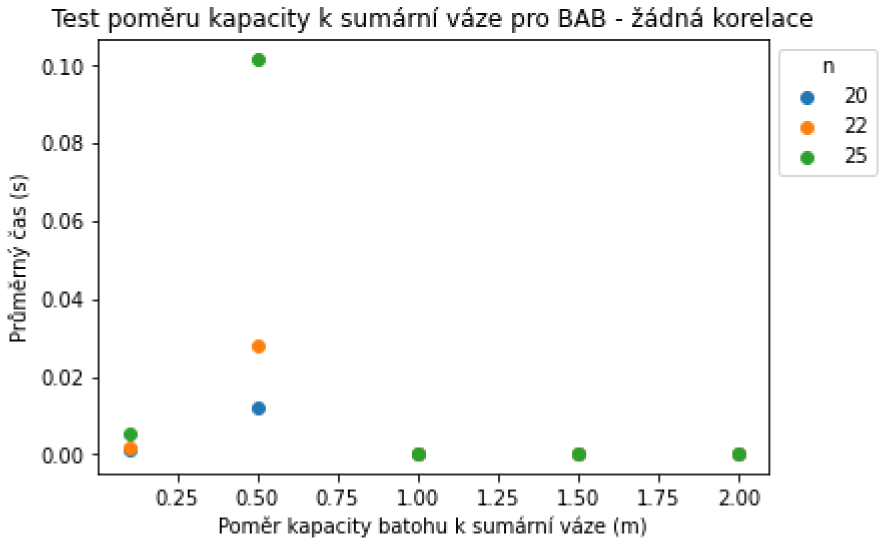
Tento experiment byl měřen s implicitním nastavením a *W/C* náleží prvkům z množiny *{100, 200, 500, 800, 1200}* a *C/W=500.* Z následujících grafů lze soudit, že závislost času na omezení maximální ceny/váhy není patrná na první pohled a jeví se náhodně v obou případech.

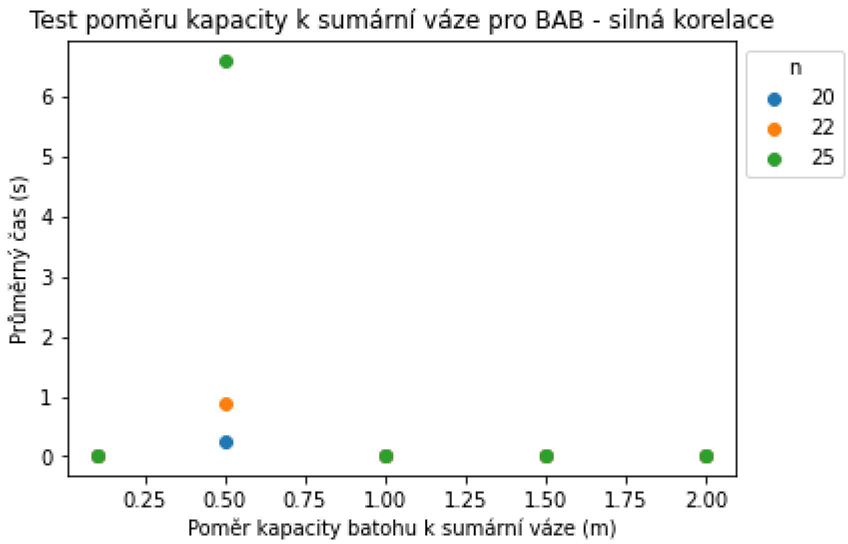


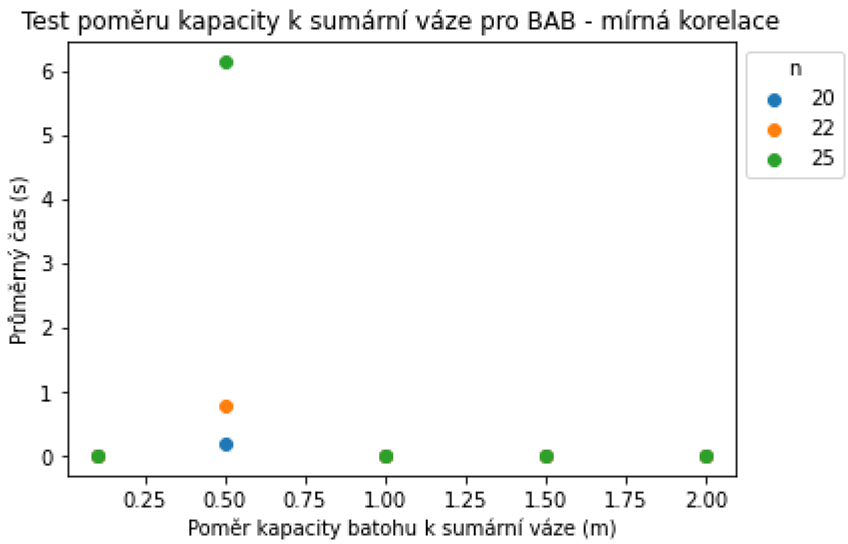


Experiment zkoumající průměrný výpočetní čas v závislosti na poměru kapacity k sumární váze a korelaci

U tohoto experimentu se zkoumaly důsledky na čas jednotlivých hodnotách *m* (poměr kapacity batohu k sumární váze) ve třech stavech korelace – žádná, mírná a silná. Proměnná *m* nabývala hodnot z množiny *{0.1, 0.5, 1, 1.5, 2}*. Všechny 3 stavy mají vrchol při hodnotě *m*=0,5. Lze také zpozorovat, že i oproti stavu s žádnou korelací jsou stavy s mírnou a silnou korelací až řádovým časovým zhoršením (pro případ n =25).

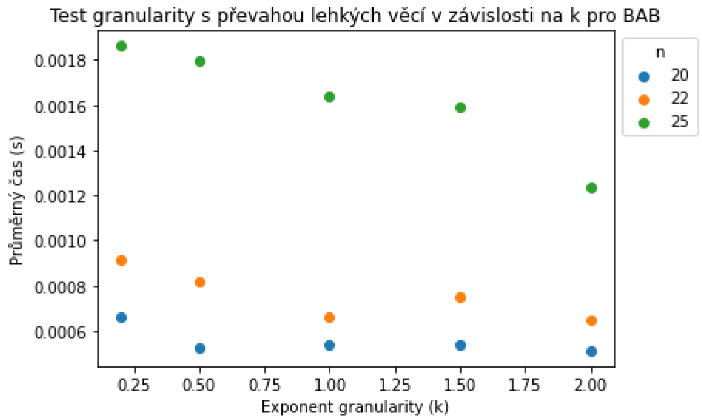


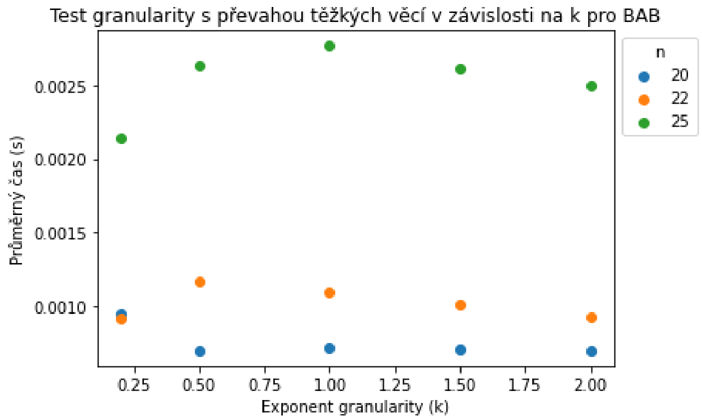


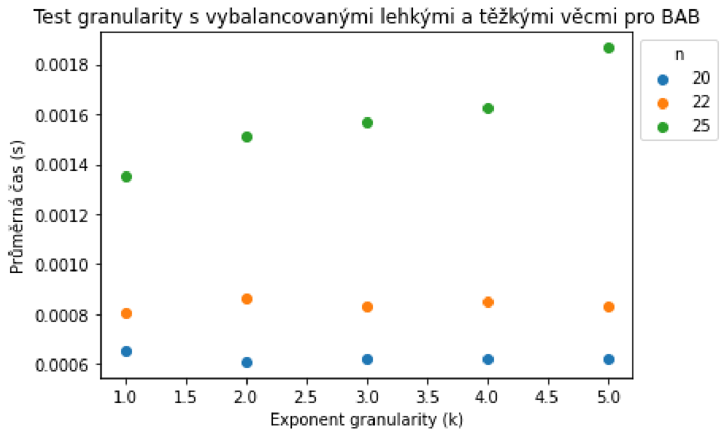


Experiment zkoumající průměrný výpočetní čas v závislosti na granularitě a jejím exponentu

Tento experiment byl měřen pro proměnou *k* nabývající hodnot z množiny *{0.2, 0.5, 1, 1.5, 2}* pro obě typy převah – lehkých i těžkých věcí. U grafu popisující granularitu s převahou lehkých věcí si lze všimnout klesajícího trendu. U grafu s granularitou s převahou těžkých věcí je možno pozorovat také klesání, avšak až od *k=1* kde byl také nejvyšší průměrný čas. Také můžeme říci, že granularita s převážně těžšími předměty je výpočetně náročnější podle vyšších časových nároků. Pro vybalancované váhy věcí je naopak zvyšující se čas s přibývajícím *k*. Granularita tedy ovlivňuje výpočetní sílu metody Branch and Bound.

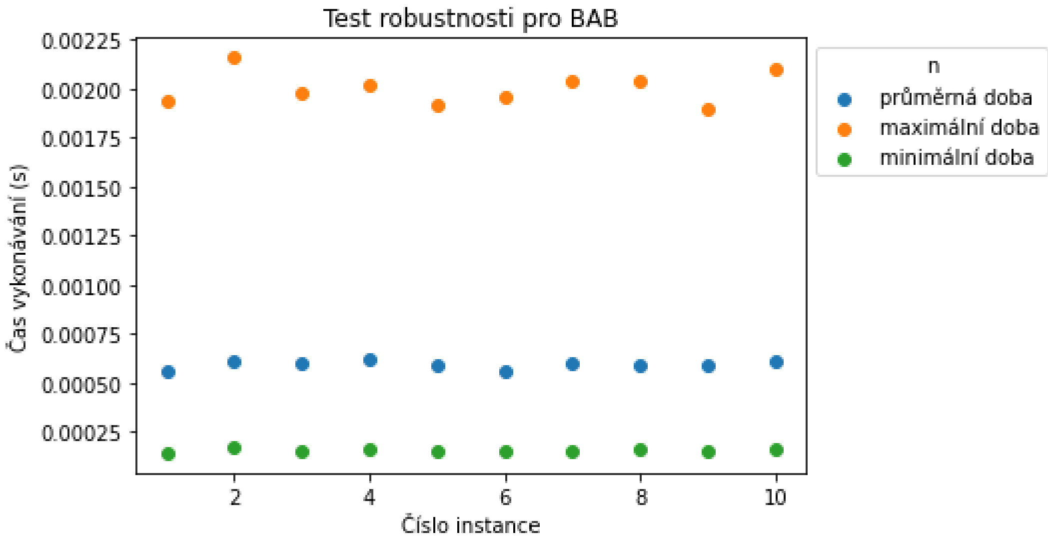






Experiment zkoumající čas výpočtu instance v závislosti na permutaci předmětů v batohu – test robustnosti

Tento experiment měřil průměrnou, maximální a minimální dobu výpočtu 10 instancí zapsaných 100 různými způsoby zápisu instance (100 permutací předmětů v batohu). V grafu je zaznamená průměrná, maximální i minimální doba výpočtu 100 různých zápisů jedné instance. Každá instance má měřené hodnoty na vertikální ose. Z grafu pak lze vyčíst, že má implementace metody není robustní a záleží tak na způsobu zápisu instance. Rozdíly mezi maximální dobou běhu a průměrnou dobou běhu jsou až čtyřnásobné a mezi maximální a minimální až téměř desetinásobné. Ověřil jsem tak hypotézu, že tato metoda není robustní.



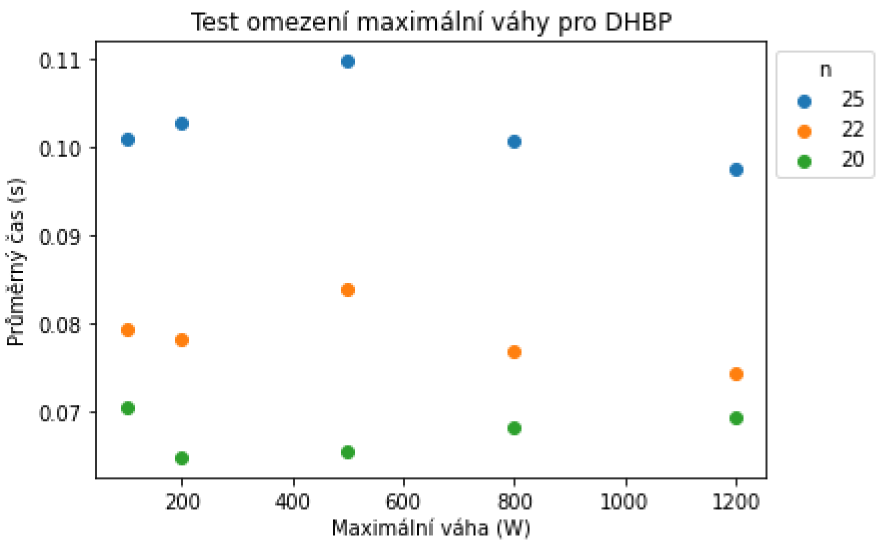
Závislosti parametrů na relativní chybě nebyly zkoumány, jelikož tato metoda vrací bezchybné výstupy a experiment by tak postrádal smysl.

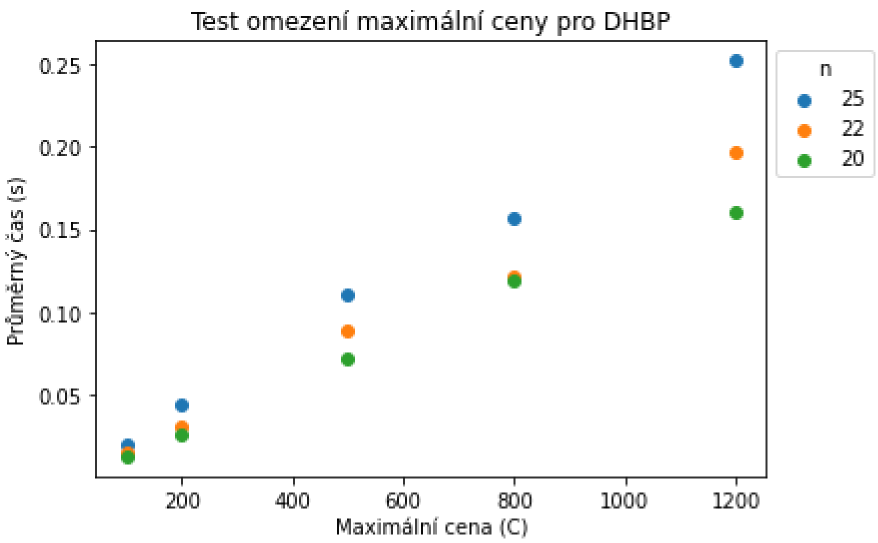
DHBP (Dynamické programování)

U této metody opět postrádá smysl zkoumat relativní chybu.

Experiment zkoumající průměrný čas v závislosti na omezení maximální ceny a váhy

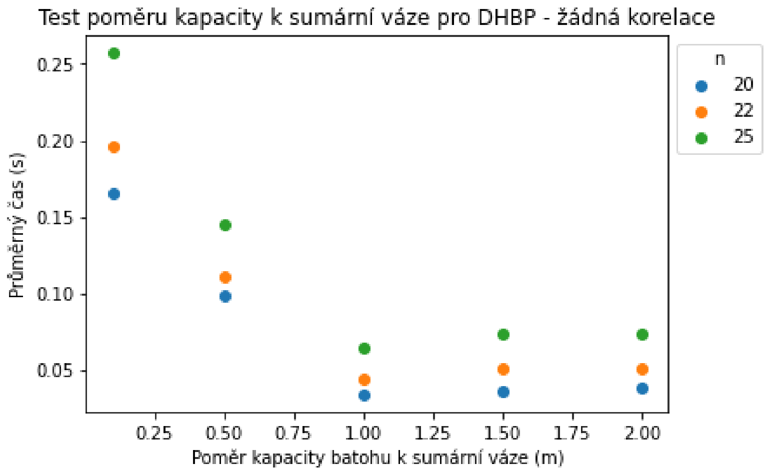
Tento experiment byl měřen s implicitním nastavením a *W/C* náleží prvkům z množiny *{100, 200, 500, 800, 1200}* a *C/W=500.* U grafu popisující závislost na omezení maximální váhy si lze všimnout vrcholu při *W=500*, což je zároveň hodnota, která je nastavená jako maximální cena. Lze tak usoudit, že algoritmus je časově nejvytíženější, pokud *W=C*. Pro ověření hypotézy je však zapotřebí podrobnější zkoumání jelikož trend je opravdu mírný. V grafu s omezením ceny je na první pohled patrná závislost. S rostoucím *C* se lineárně zvětšuje časová náročnost.

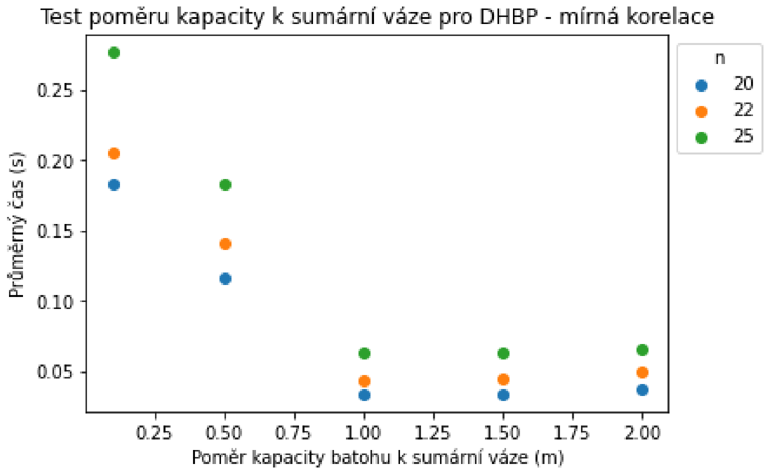


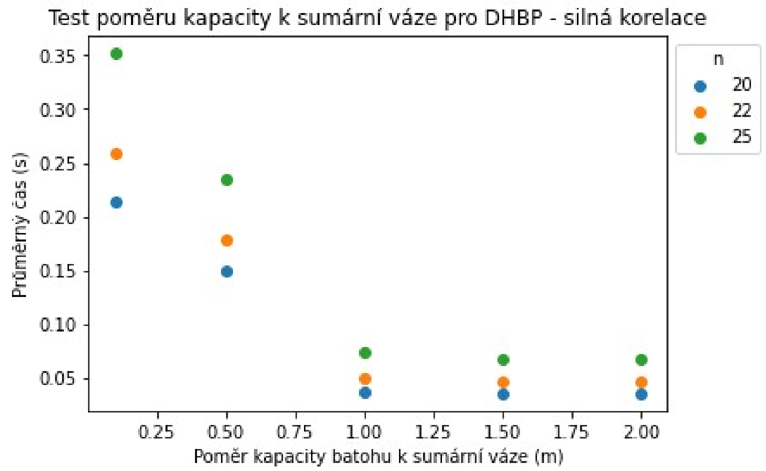


Experiment zkoumající průměrný výpočetní čas v závislosti na poměru kapacity k sumární váze a korelaci

U tohoto experimentu se zkoumaly důsledky na čas jednotlivých hodnotách *m* (poměr kapacity batohu k sumární váze) ve třech stavech korelace – žádná, mírná a silná. Proměnná *m* nabývala hodnot z množiny *{0.1, 0.5, 1, 1.5, 2}*. Ve všech grafech je trend klesající do hodnoty k=1 a poté křivka velmi mírně stoupá. Při hodnotě blízké 0 je tedy náročnost nejvyšší. Je nutno také poznamenat, že při silné korelaci je průměrný čas vyšší o cca 30% než u měření s žádnou a mírnou korelací.

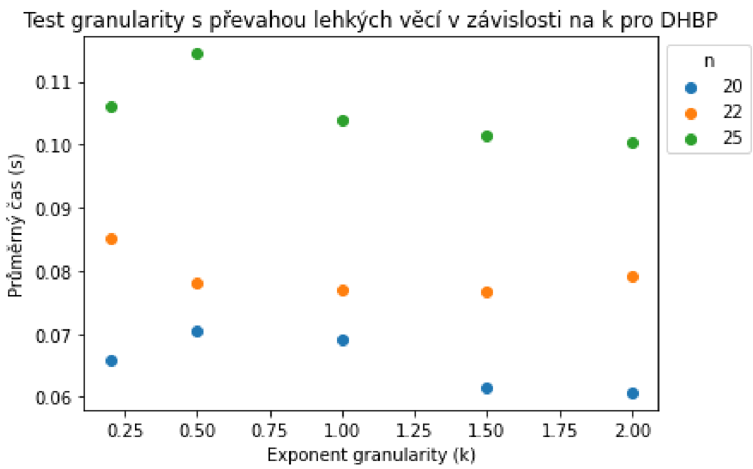


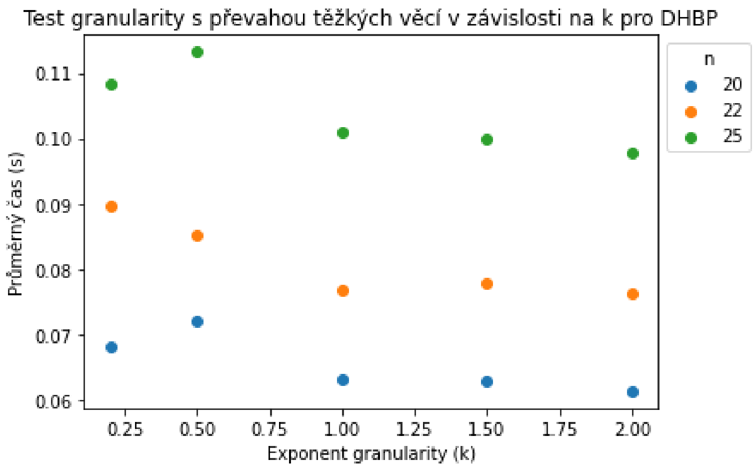


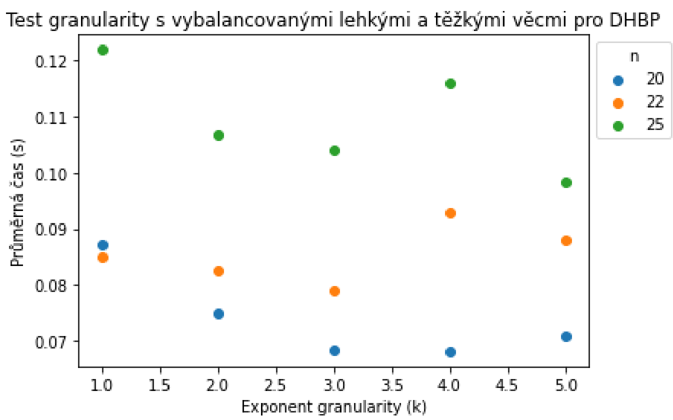


Experiment zkoumající průměrný výpočetní čas v závislosti na granularitě a jejím exponentu

Tento experiment byl měřen pro proměnou *k* nabývající hodnot z množiny *{0.2, 0.5, 1, 1.5, 2}* pro obě typy převah – lehkých i těžkých věcí. Opět lze pozorovat vrchol v k=0,5 v obou grafech, avšak časově je rozdíl nejvýše 10% (při n=25) a nikoliv řádově zhoršení.

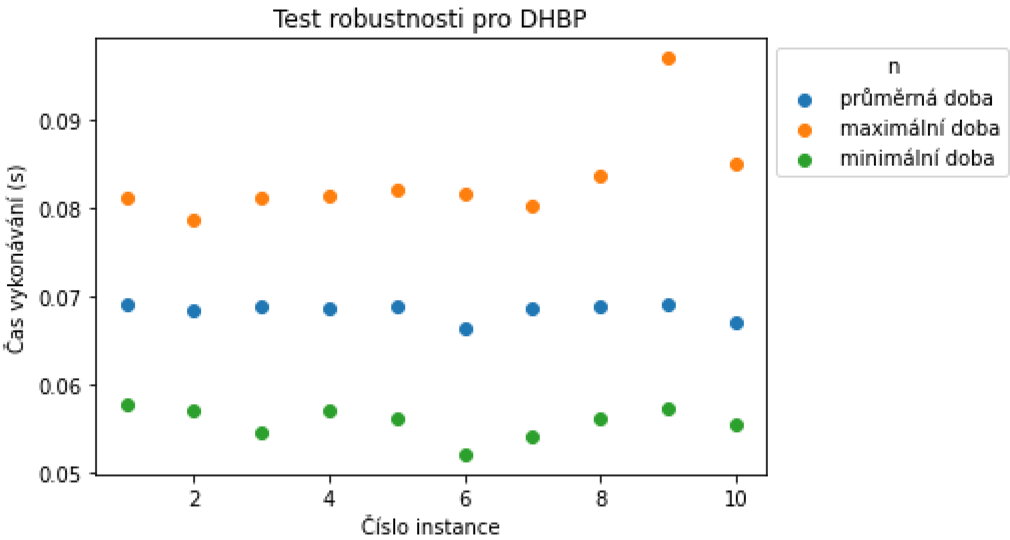






Experiment zkoumající čas výpočtu instance v závislosti na permutaci předmětů v batohu – test robustnosti

Tento experiment měřil průměrnou, maximální a minimální dobu výpočtu 10 instancí zapsaných 100 různými způsoby zápisu instance (100 permutací předmětů v batohu). V grafu je zaznamená průměrná, maximální i minimální doba výpočtu 100 různých zápisů jedné instance. Každá instance má měřené hodnoty na vertikální ose. Z grafu pak lze vyčíst, že pro metoda DHBP není robustní, ale nerobustnost je tolerovatelná (odklon od průměru cca 15%). Oproti metodě BAB je tato metoda mnohem více robustní.

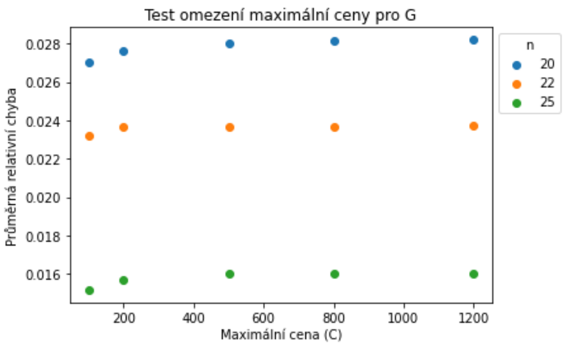


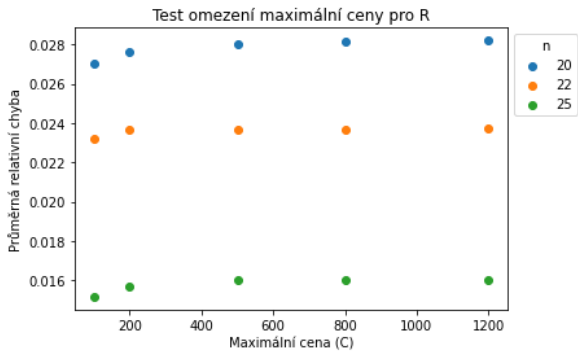
Greedy a Redux metody

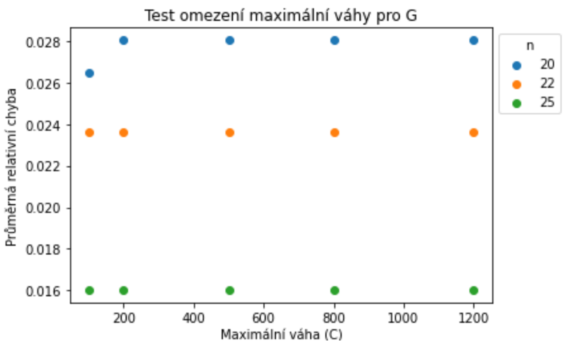
Tyto metody jsou oproti předchozím metodám výsledkově nepřesná a velmi rychlá (téměř lineární složitost). Nemá tedy smysl zkoumat jejich závislosti na čase. Zkoumání robustnosti také postrádá smysl, jelikož metody předměty řadí před samotným výpočtem. Naopak má smysl zkoumat závislosti relativní chyby.

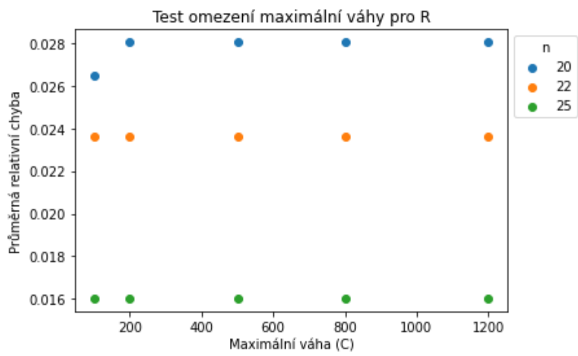
Experiment zkoumající průměrný čas v závislosti na omezení maximální ceny a váhy

Tento experiment byl měřen s implicitním nastavením a *W/C* náleží prvkům z množiny *{100, 200, 500, 800, 1200}* a *C/W=500.* Z výsledků lze vyčíst, že omezení ceny/váhy má téměř nepatrnou stoupající tendenci v závislosti na omezení maximální ceny/váhy. Stojí za poznamenání, že relativní chyba se zmenšuje se zvyšující se hodnotou *n*.



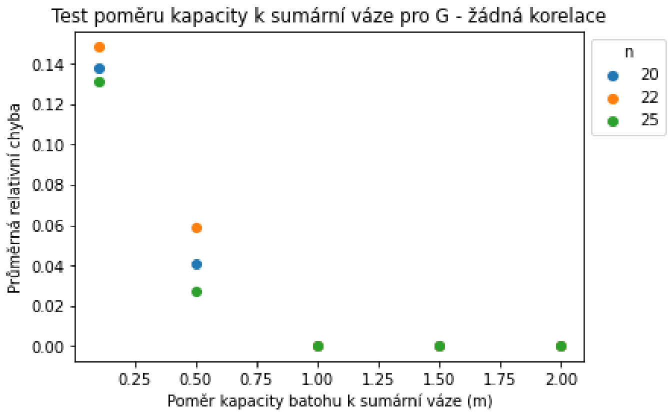


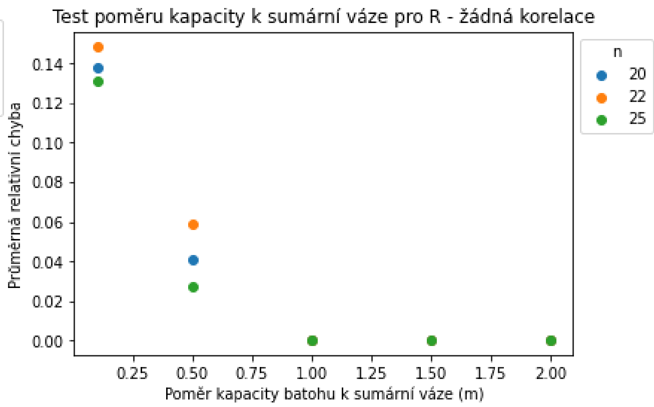


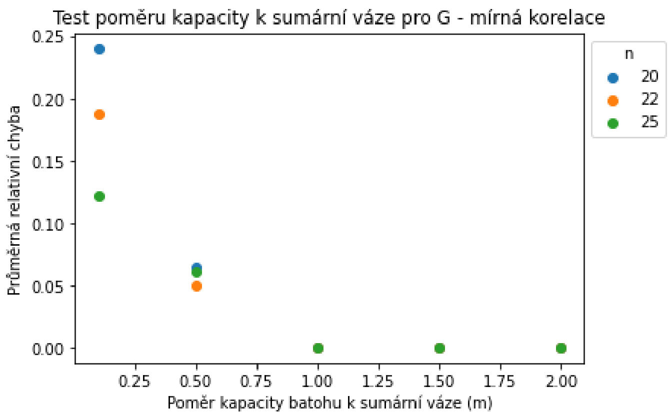


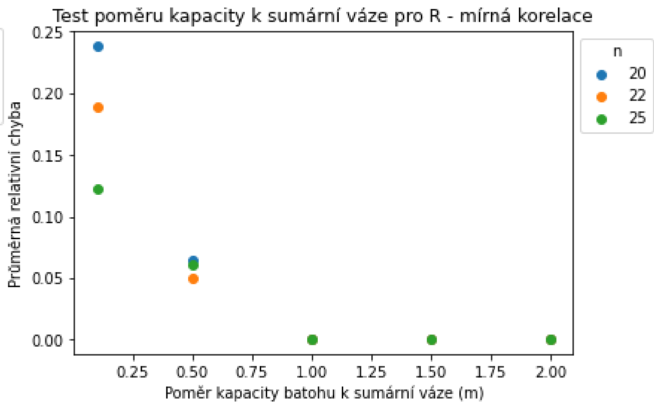
Experiment zkoumající průměrný výpočetní čas v závislosti na poměru kapacity k sumární váze a korelaci

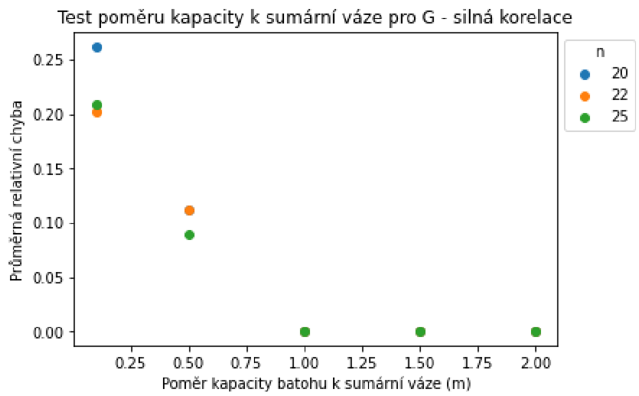
U tohoto experimentu se zkoumaly důsledky na relativní chybu jednotlivých hodnot *m* (poměr kapacity batohu k sumární váze) ve třech stavech korelace – žádná, mírná a silná. Proměnná *m* nabývala hodnot z množiny *{0.1, 0.5, 1, 1.5, 2}*. Obdobně jako v grafu zkoumající závislost výpočetního času na granularitě u ostatních metod má i tato klesající průběh na intervalu od <0, 1>. Také jsou relativní chyby větší u grafů s mírnou a silnou korelací oproti žádné korelaci.

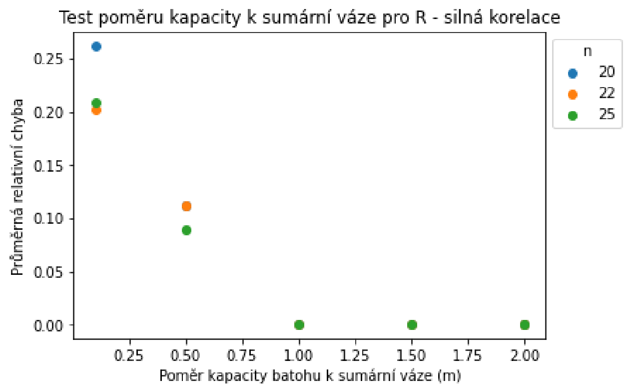






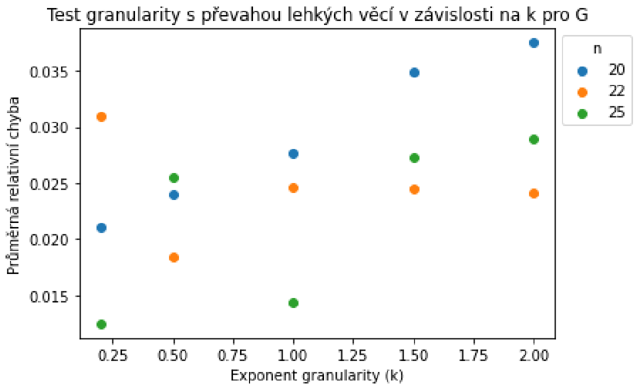


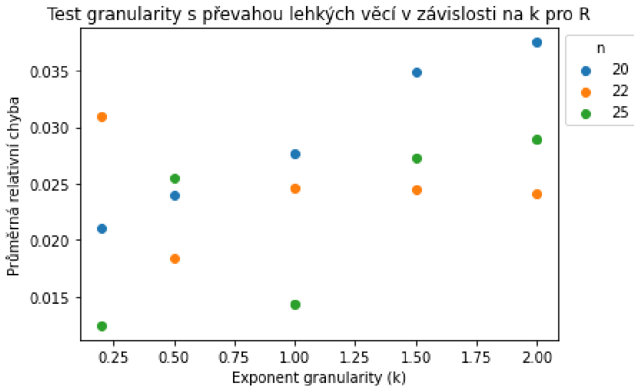


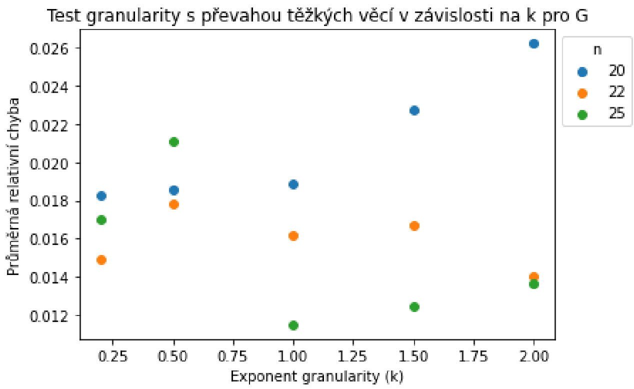


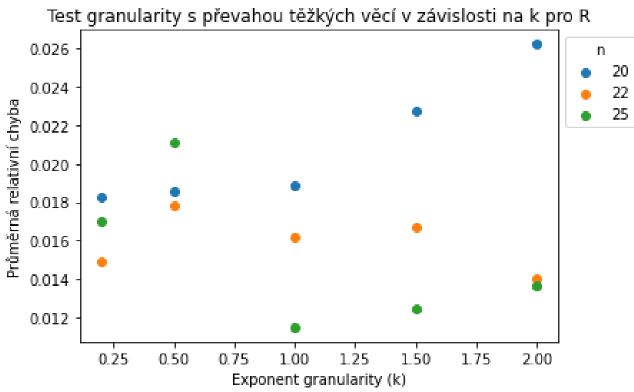
Experiment zkoumající průměrný výpočetní čas v závislosti na granularitě a jejím exponentu

Tento experiment byl měřen pro proměnou *k* nabývající hodnot z množiny *{0.2, 0.5, 1, 1.5, 2}* pro obě typy převah – lehkých i těžkých věcí. Z grafů lze vyčíst, že při převaze lehkých věcí byla průměrná relativní chyba obecně větší než u převahy těžších věcí. Závislost na exponentu granularity nelze odvodit, jelikož naměřené hodnoty jsou příliš neuspořádané.









Závěr

Potvrdil jsem, že výpočetní náročnost dynamického programování je citlivá na maximální cenu. Také jsem zjistil, že metoda Branch and Bound je citlivá na granularitu a také na vybalancovaný poměr kapacity batohu k sumární váze všech předmětů (zejména u silné korelace). Také se potvrdili očekávání o metodě Branch and Bound, že není robustní. Metoda DHBP také není robustní, ale v mnohem menší míře oproti Branch and Bound.