Exaktní metody řešení rozhodovacího problému batohu

NI-KOP DÚ 1

Tomáš Kalabis

Zadání

Je dáno

- celé číslo *n* (počet věcí)
- celé číslo *M* (kapacita batohu)
- celé kladné číslo *B*
- 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

$c_1x_1 + c_2x_2 + \dots + c_nx_n \geq B$

tj. nabýval hodnoty alespoň *B* (cena věcí v batohu byla alespoň vyžadovaná)?

Výsledkem tak je množina vybraných věcí v batohu, které jsou lehčí než kapacita batohu (číslo *M*) a cennější než minimální požadovaná hodnota věcí (číslo *B*).

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*.

Program se spouští vyhodnocením všech buněk v Jupyter Notebooku. Aby program správně fungoval, musí být vytvořená složka `results/` v adresáři s programem a složky `NR/` a `ZR/` by měly být také ve stejné složce jako program (případně upravit volání funkce).

Řešení a metody

Metoda Hrubou silou (Brute Force)

Metoda Hrubou silou je nejpomalejší z metod řešení tohoto problému. Metoda projde všechny případy uložených věcí v batohu a zkontroluje, zda instance plnění batohu splňuje uvedené podmínky. Značnou nevýhodou této metody je, že prochází instance plnění batohu který nedokáže splnit minimální hodnotu v batohu. Časová složitost tohoto problému je $\Theta(2^n)$.

Metoda Větví a hranic (Branch and Bound)

Tato metoda řeší problém zmíněný v předchozím odstavci. Metoda je implementovaná jako stromová rekurze v jehož každém uzlu se kontroluje, zda podstrom tohoto uzlu je schopen naplnit minimální požadavek na celkovou cenu plnění batohu. V negativním případě se podstrom tohoto uzlu vůbec nezkouší, jelikož víme, že nemá šanci na úspěch. Časová náročnost této metody je $O(2^n)$.

Experimenty

Experimenty probíhaly na sadách *NR* a *ZR*, které byly dodány se zadáním. Zkoumali se počty navštívených uzlů při jednotlivých instancích, jednotlivých hodnotách *n* a jednotlivých metodách.

Testování probíhalo na notebooku s následujícími parametry:

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

Výsledky

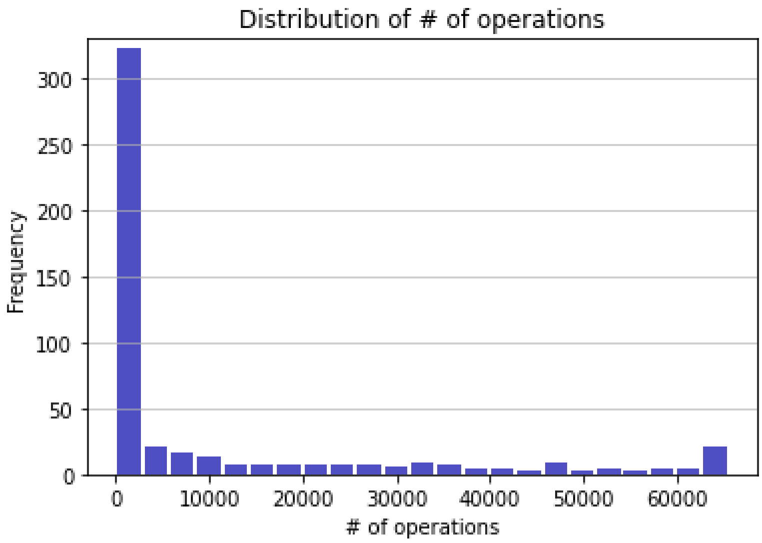
Poznámka: v následujících tabulkách jsou zaznamenány počet navštívených uzlů Hrubou silou. Od n > 15 však již hrubá síla nebyla testována a hodnoty tak byly doplněny o své očekávané hodnoty.

	Průměrný počet vyzkoušených konfigurací		
n	Hrubá síla	NR – metoda větví	ZR – metoda větví
4	2^4 = 16	3.066	3.16
10	2^{10} = 1 024	136.6	77.368
15	2^{15} = 32 768	4 412.048	1 288.946
20	2^{20} = 1 048 576	154 225.908	26 976.58
22	2^{22} = 4 194 304	652 626.246	82 840.82

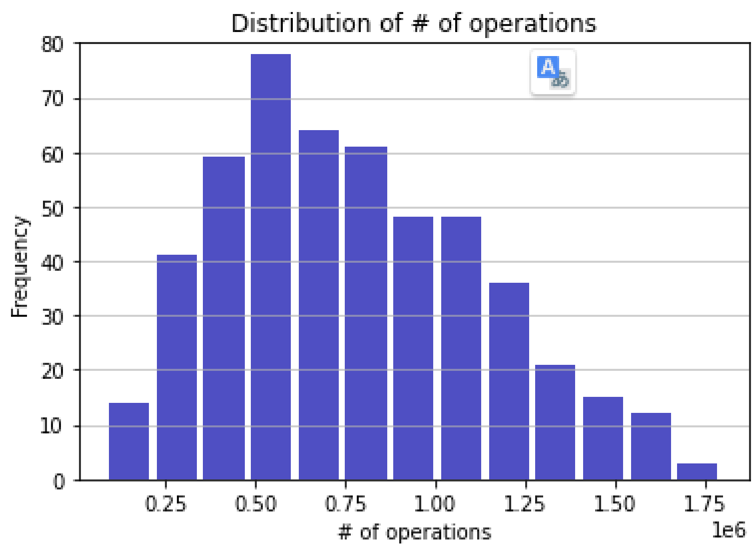
	Maximální počet vyzkoušených konfigurací

n	Hrubá síla	NR – metoda větví	ZR – metoda větví
4	$2^4 = 16$	15	8
10	$2^{10} = 1\,024$	1 010	205
15	$2^{15} = 32\,768$	32 760	4271
20	$2^{20} = 1\,048\,576$	1 048 519	98 689
22	$2^{22} = 4\,194\,304$	4 194 267	350 659

Předchozí 2 tabulky zobrazují vyzkoušený počet různých konfigurací při jednotlivých metodách a sadách (hrubá síla je vůči rozdílným sadám stejná). Patrné je zlepšení oproti metodě Hrubé síly a maximální počet vyzkoušených konfigurací je vždy menší než počet při použití metody Hrubé síly.



Na grafu rozdělení sady *NR* pro *n = 15* je patrné, že velká většina rozložení je rovna 0. Mnoho instancí tedy nemá řešení pro rozhodovací problém, jelikož se algoritmus zastavil již v počátku – součet cen všech možných věcí v batohu je menší než minimální požadovaná cena (konstanta *B*). Z dosavadních zjištění se tak jeví, že sada byla generována náhodně.



Rozdělení sady *ZR* pro $n = 22$ připomíná klasickou Gaussovu křivku s největším shlukem uprostřed, a neexistují žádné příklady, kde nebyly žádné operace – všechny instance měli součet cen věcí větší nebo rovno minimální požadované hodnotě ceny (konstanta *B*).

Závěr

Ověřili jsme rapidní zrychlení při použití metody větví a hranic oproti metodě hrubé síly. Po zkoumání sad NR a ZR jsme zjistili, že v sadě ZR se téměř nevyskytuje instance, jejíž minimální cenu *B* není možné překročit a naopak sada NR obsahuje mnoho takových instancí.