

Seznámení se metodou simulovaného ochlazování na problému batohu

NI-KOP DÚ 4

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.

Simulované ochlazování

Simulované ochlazování je jedna z pokročilých iterativních metod použitých pro řešení konstruktivního problému batohu. Metoda nezaručuje správný výsledek, zato je velmi rychlá. Principiálně se podobá metodě *Hill climbing*, avšak při náhodném prohledávání sousedů ve stavovém prostoru může přijmout i horší stav. To je dáno pravděpodobností přijmutí horšího stavu, která je také ovlivněna mírou zhoršení nového stavu oproti starému stavu. Výhodou této metody je také její parametrizovatelnost, kterými lze ovlivnit délku výpočtu / kvalitu výpočtu. Pseudokód mé implementace je následující:

```

stav = počáteční_stav
nejlepší_stav = None
while(not stav.frozen()) {
    while(not equilibrium()){
        stav = nový_stav()
        if (stav.lepší_než(nejlepší_stav)){
            nejlepší_stav = stav
        }
    }
    cool()
}

```

V průběhu výpočtu figuruje tzv. *teplota*, která ovlivňuje počet navštívených stavů. S klesající teplotou klesá i pravděpodobnost přijetí horších stavů. Teplota je vymezena konstantou *počáteční_teploata* a metodou *frozen()*, která rozhoduje, zda je teplota dostatečně nízká, aby byl výpočet ukončen. Teplota je také ovlivňována metodou *cool()*, která určuje rychlost snižování teploty. Teplota není snižována po každém vyzkoušeném stavu a snižuje se po dávkách vyzkoušených stavů. Počet těchto stavů určuje metoda *equilibrium()*.

Experimenty

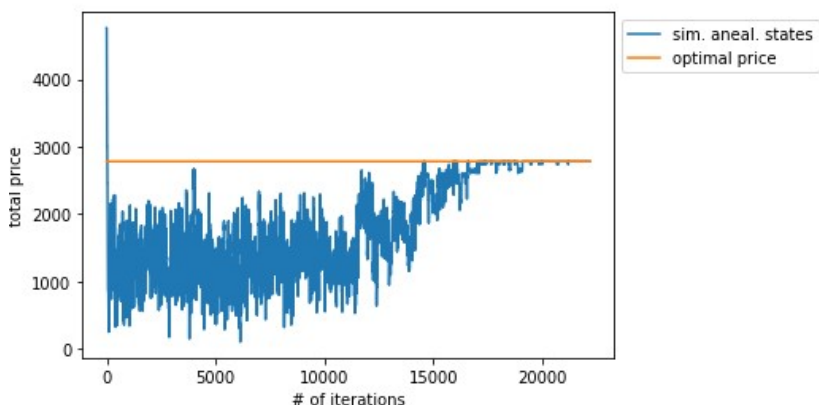
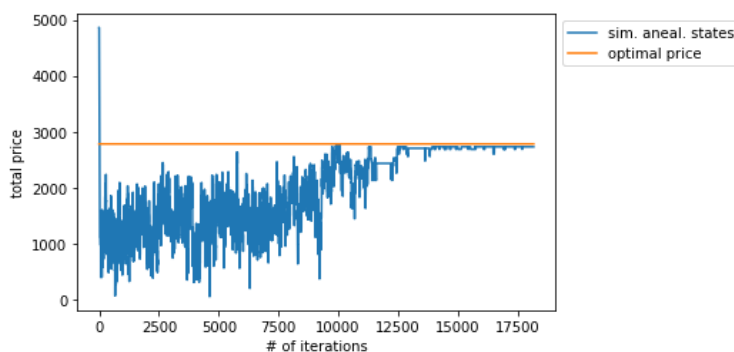
Experimenty byly prováděny na testovací sadě *NK* pro $n = 35$. Experimentovalo se s parametry *počáteční_teploata*, *equilibrium()*, *frozen()*, *cool()* a měřila se kvalita výsledků a rychlost výpočtu s těmito parametry.

Pozn.: Pro následující měření zachycené grafy bylo použito následující výchozí nastavení pokud není řečeno jinak: počáteční stav – náhodný, výchozí teplota – 10 000, konstanta frozen – 100/10, C – 0,95 a equilibrium – 150.

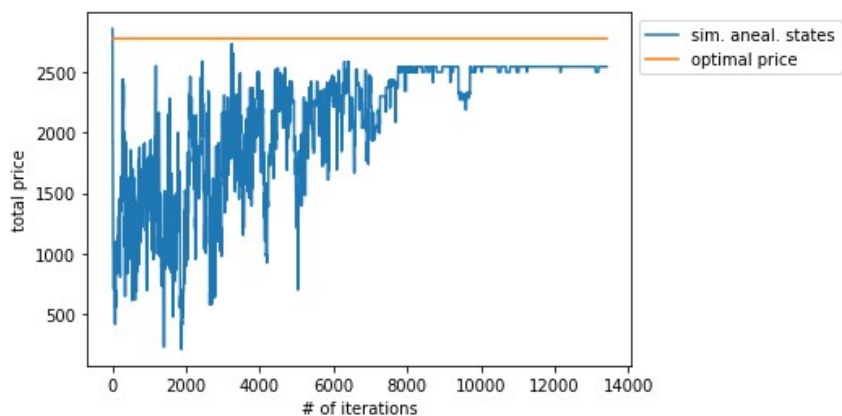
počáteční teplota

Při nedostatečné teplotě je méně pravděpodobné nalezení globálního maxima a nalezení optimálního řešení. Teplota totiž ovlivňuje pravděpodobnost přijetí horšího stavu. Příliš vysoká hodnota je přebytná a hodnoty velmi oscilují po příliš dlouhou dobu. Při experimentování byly ideální hodnoty mezi 5000 a 15000.

Obrázek 1: počáteční_teplota = 5000



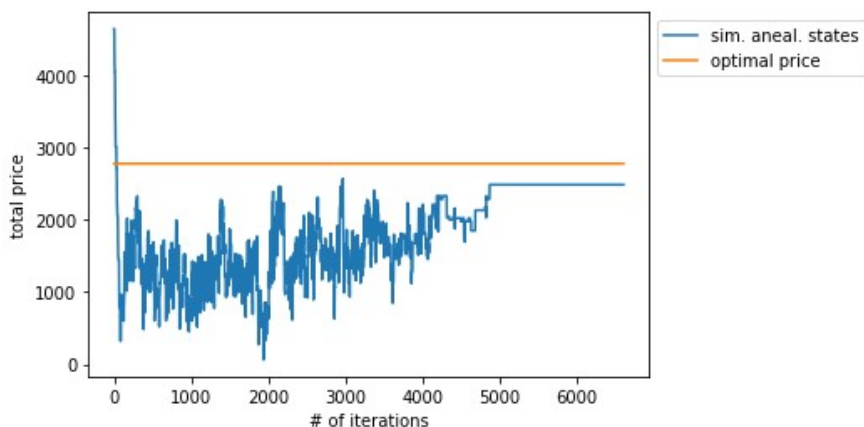
Obrázek 2: počáteční_teplota = 20000



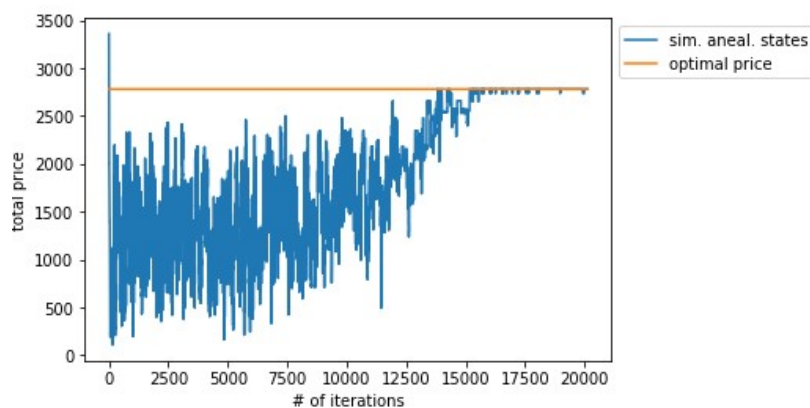
Obrázek 3: počáteční_teplota = 1000

equilibrium()

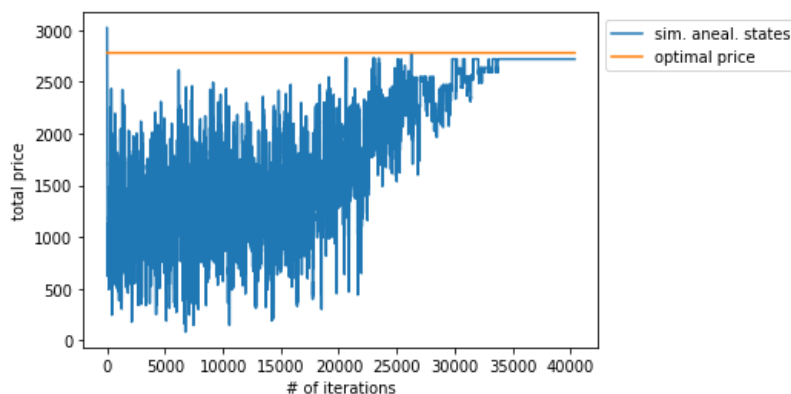
Metoda je implementovaná jako pevná konstanta a nemění se tak v průběhu výpočtu. Při vyšším *equilibriu* je i vyšší doba výpočtů. Zároveň však při nízkém *equilibriu* je větší tendence zůstat v lokálním maximu. Z grafů je patrné, že s rostoucím *equilibriem* roste také počet iterací a rozsah oscilací stavů. Optimální hodnotou byl dle experimentů hodnota cca 150.



Obrázek 4: *equilibrium* = 50



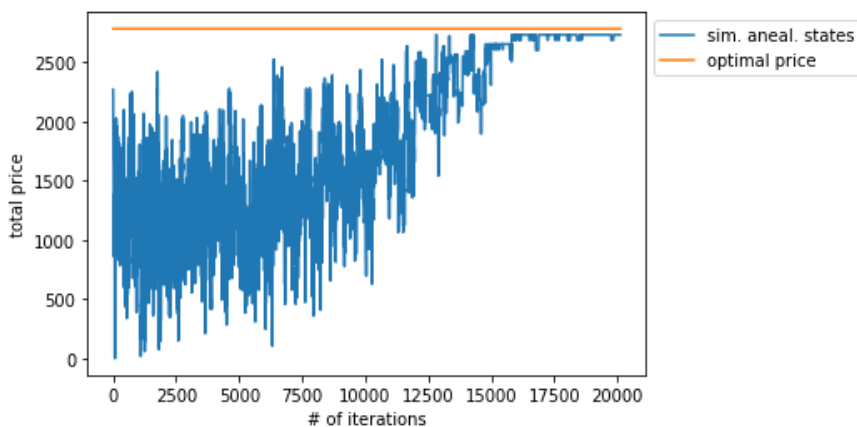
Obrázek 5: *equilibrium* = 150



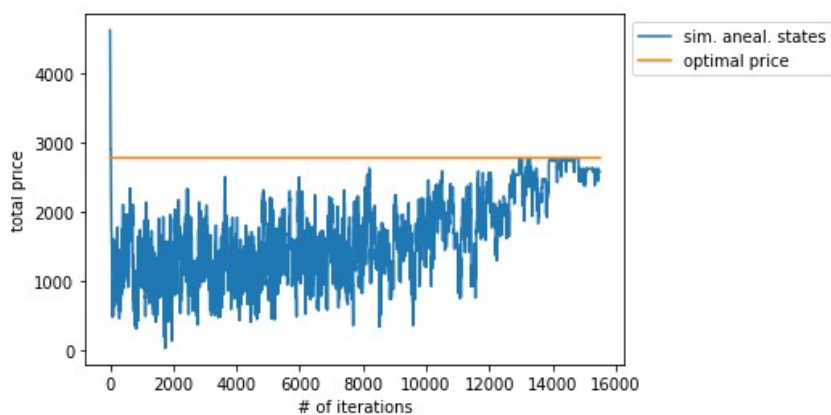
Obrázek 6: *equilibrium* = 300

frozen()

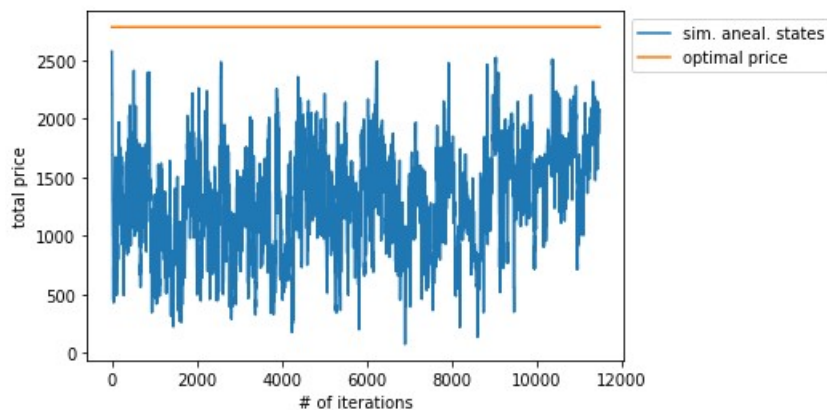
Frozen určuje ukončení výpočtu a je implementována jako konstanta. Je na místě odhadnout ideální konstantu. Nízká konstanta zapříčiňuje neměnný stav kdy je metoda „jistá svým výsledkem“ a příliš vysoká konstanta ukončí výpočet když je teplota stále vysoká a bere se tak ještě horší stavy. Z grafů je tento trend vidět. Při experimentování vyšla jako optimální mezi 50 až 100.



Obrázek 7: konstanta *frozen* = 10



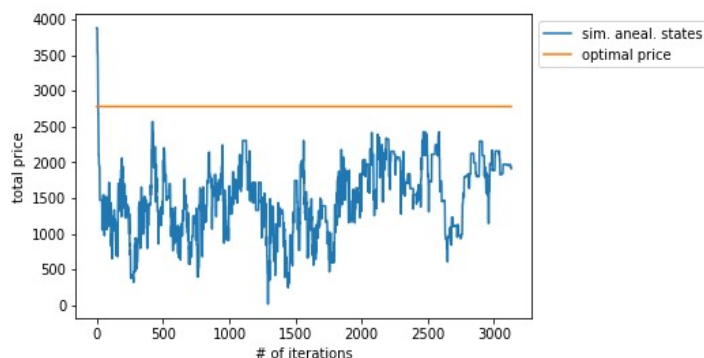
Obrázek 8: konstant *frozen* = 50



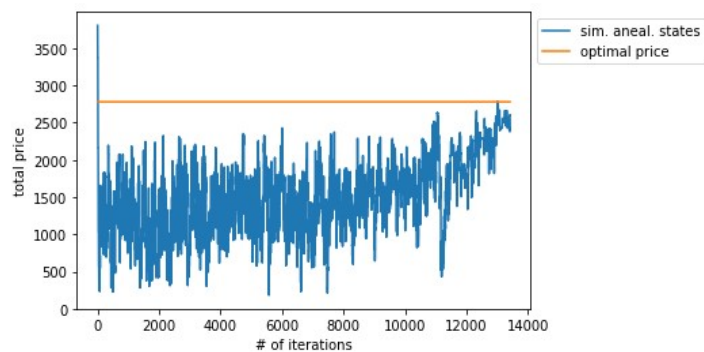
Obrázek 9: konstanta *frozen* = 200

cool()

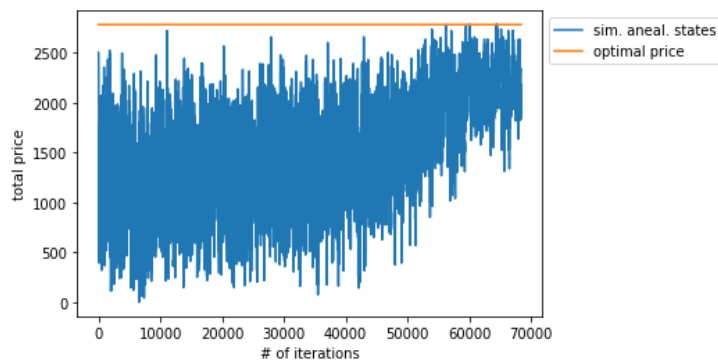
Metoda cool() určuje míru snížení teploty. Ta je ovlivněna konstantou C , která typicky nabývá hodnot od 0,8 do 0,99. Je implementována tak, že násobí aktuální teplotu touto konstantou. Teplota tak klesá rychleji při vyšších teplotách a zároveň klesá výrazně rychleji při nižším C . To přímo ovlivňuje celkový počet iterací a tedy i dobu výpočtu (a nejvíce ji ovlivňuje). Na druhou stranu při vyšší hodnotě C je i vyšší pravděpodobnost nalezení optimálního řešení. Nejlepší hodnota se zdá být 0,95, jelikož 0,99 dává přesnější výsledky, zato rychlost výpočtu je při této hodnotě příliš dlouhá.



Obrázek 10: $C = 0,8$



Obrázek 11: $C = 0,95$



Obrázek 12: $C = 0,99$

Výsledky měření

Měření probíhalo na sadě NK pro $n=35$ s náhodným počátečním stavem a vybraným počátečním stavem (viz další odstavec). Měřeny byly následující hodnoty: průměrná relativní chyba, průměrný čas, maximální čas a maximální relativní chyba. Byly zkoumány jejich závislosti na parametrech popsanych v předchozí kapitole. Je patrné, že nejvíce prodlužuje čas a zvětšuje kvalitu konstanta C. Také konstanty frozen a počáteční teplota prodlužují čas, pokud jsou nastaveny příliš nízké/vysoké.

Při opakovaném spuštění téže instance je velmi pravděpodobné, že optimální výsledek bude jedním z řešení. Když je počáteční stav vybírán náhodně, může být velmi vzdálený optimálnímu řešení a záleží tak na počátečním stavu. Pokud se vybere jedno z řešení vypočítaných například greedy heuristikou a na to se aplikuje simulované ochlazování, počáteční stav bude mnohem blíže optimálnímu řešení. To zachycují rozdvojené buňky v následující tabulce, kde levá podbuňka zachycuje náhodný počáteční stav a pravá podbuňka zachycuje počáteční stav dle greedy heuristiky.

Testování probíhalo na notebooku s následujícími parametry:

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

Počáteční teplota	C	Equil.	frozen	Průměrný čas		Průměrná relativní chyba		Max. Čas		Max. relativní chyba	
15000	0,99	180	10	5,9 s		0,3 %		7,4 s		7 %	
15000	0,9	180	10	0,6 s		2 %		0,8 s		12 %	
5000	0,99	150	100	2,8 s	2,7 s	0,3 %	0,1 %	3,9 s	2,9 s	4 %	2 %
5000	0,95	150	100	0,5 s	0,5 s	1,4 %	0,2 %	0,8 s	0,6 s	10 %	5 %
5000	0,98	150	100	1,4 s		0,6 %		1,9 s		6 %	
5000	0,95	150	10		0,8 s		0,2 %		1,1 s		4 %
1000	0,9	150	100		0,1 s		0,3 %		0,2 s		6 %

nejlepší co se týče kvality je tak nastavení: *počáteční teplota – 5000, C – 0,99, equilibrium – 150, frozen konstanta – 100*. Toto nastavení vyniká jak pro náhodný tak pro vybraný počáteční stav. Pokud se vezme v potaz i výpočetní rychlost, optimální nastavení se zdá být: *počáteční teplota – 5000, C – 0,95, equilibrium – 150, frozen konstanta – 100* pro

náhodný poč. stav a nastavení pro vybraný poč. stav: *počáteční teplota – 1000, C – 0,9, equilibrium – 150, frozen konstanta – 100*

Závěr

Pro řešení konstruktivního problému batohu jsem použil metodu simulovaného ochlazování, která je velmi parametrizovatelná. Provedl jsem experimenty zkoumající vliv parametrů na kvalitu a rychlost metody. Lze také říci, že vybráním vhodného počátečního stavu může pomoci metodě například zvolením za počáteční stav řešení greedy heuristikou.