

Report: Problém batohu

Specifikace úlohy	2
Vygenerované výsledky	2
Program	2
Popis algoritmů	3
Hardwarová konfigurace	3
Naměřené výsledky	4
Porovnání algoritmů SA/SAPenalty	4
Testování parametrů	5
Počáteční teplota	6
Koeficient ochlazování	7
Počet interních cyklů	8
Vývoje řešení	9
Závěr	10
Přiložené důležité soubory	10

Specifikace úlohy

Úkoly:

1. Řešení konstruktivního 0/1 [problému batohu](#) (dále *problém*).
 - a. Umí využít tyto algoritmy:
 - Simulated Annealing (SA)
2. Zjištění, jakým způsobem se nastavují parametry SA
3. Experimentální vyhodnocení SA na poskytnutých datasetech s různým nastavením parametrů

Výpočetní složitost je měřena počtem navštívených konfigurací, tedy vyhodnocených sestav věcí v batohu. Je zachycen i reálný čas.

Plné zadání úlohy [zde](#).

Vygenerované výsledky

Pomocí poskytnutého generátoru instancí *problému* byli vygenerovány různé datasety. Každý dataset obsahuje instance, kde právě jeden parametr generátoru se mění a zbytek parametrů je zafixován na konstantních hodnotách.

Dataset	Popis
Balance	Parametr určuje poměr lehkých a těžkých věcí v instanci.
Correlation	Parametr určuje korelaci ceny s hmotností.
GranularityHeavy	Parametr určuje exponent granularity k pro instance, které mají převahu těžkých věcí. Pravděpodobnost, že věc s váhou w bude v instanci zahrnuta, je $p = \frac{1}{(w_{max} - w)^k}$
GranularityLight	Parametr určuje exponent granularity k pro instance, které mají převahu lehkých věcí. Pravděpodobnost, že věc s váhou w bude v instanci zahrnuta, je $p = \frac{1}{w^k}$
MaxCost	Parametr určuje maximální cenu nejdražší věci v instanci.
MaxWeight	Parametr určuje maximální hmotnost nejtěžší věci v instanci.
Robust	Data obsahují permutaci jedné instance.
Things	Parametr určuje počet věcí v instanci.
WeightCapRatio	Parametr určuje poměr kapacity batohu k sumární váze.

Program

Program byl napsán v jazyce Python. Algoritmy jsou též napsány v jazyce Python s optimalizací pomocí jazyka Cython.

Pro vyřešení *problému* lze využít tyto **algoritmy**:

Soubor	Popis
<code>package_algorithm/sa.py</code>	Implementace algoritmu SA, který používá opravovací funkci pro opravu nevyhovujících řešení, pro která platí $\sum chosenWeights > capacity$
<code>package_algorithm/sa_penalty.py</code>	Implementace algoritmu SA, který používá penalizační funkci pro ohodnocení nevyhovujících řešení, pro která platí $\sum chosenWeights > capacity$

Popis algoritmů

Algoritmy **SA** a **SAPenalty** jsou oba implementací algoritmu Simulovaného ochlazování (Simulated Annealing), které se liší pouze v tom, jakým způsobem nakládají s nevyhovujícími řešeními.

Oba algoritmy přijímají tyto **parametry**:

Parametr	Typ	Hodnoty	Informace
Počáteční teplota	Celé číslo	<1, inf)	
Konečná teplota	Celé číslo	<1, inf)	Teplota, jejíž dosažení značí konec algoritmu.
Koeficient ochlazování	Desetinné číslo	(0, 1)	Určuje rychlost snižování teploty podle vztahu: $teplota_1 = teplota_0 * koeficient$
Počet interních cyklů	Celé číslo	(0, inf)	Kolikrát je hledáno a vyzkoušeno sousední řešení, než dojde ke snížení teploty.

Sousední řešení jsou vybírána přehozením bitu v původním řešení – přidání/odebrání věci v batohu.

Algoritmus umožňuje přijmout i horší řešení s jistou pravděpodobností. Tato pravděpodobnost je dána vztahem $e^x > random(0..1); x = \frac{worseFitness - betterFitness}{currTemperature}$.

SAPenalty penalizuje nevyhovující řešení při výpočtu fitness. Fitness se počítá pomocí vzorce $(capacity - \sum chosenWeights + 1) * \sum chosenCosts$.

SA opravuje nevyhovující řešení, než vypočítává jejich fitness. Při opravě jsou z batohu vybírány věci v pořadí od věci s nejnižší hodnotou $\frac{cost}{weight}$, dokud neplatí

$\sum chosenWeights < capacity$. Fitness hodnota je pak rovna $\sum chosenCosts$.

Protože algoritmus SA je průměrně mnohem přesnější, byl vybrán jako hlavní algoritmus pro testování a byl optimalizován pomocí jazyka Cython.

Hardwarová konfigurace

Výsledná data byla vygenerována na operačním systému Linux, procesor 2,3 GHz Intel Core i5, RAM paměť 8 GB 2133 MHz LPDDR3.

Naměřené výsledky

Všechny ukazované výsledky byli naměřeny na zadaných a některých vygenerovaných datasetech.

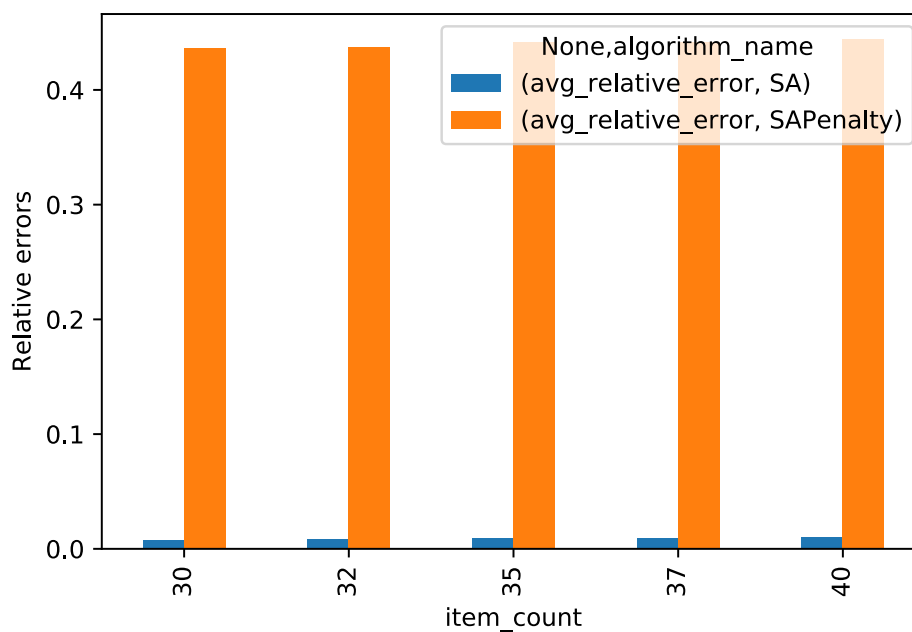
Porovnání algoritmů SA/SAPenalty

Z následujících výsledků relativních chyb při stejném nastavení a na stejných datech je jasné, že algoritmus SA je výhodnější používat.

SA/SAPenalty relative error comparison

algorithm_name	max_relative_error		avg_relative_error	
	SA	SAPenalty	SA	SAPenalty
item_count				
30	0,121016	0,742442	0,007142	0,436356
32	0,094969	0,836483	0,007968	0,437133
35	0,106762	0,764768	0,009115	0,441838
37	0,122474	0,828992	0,008819	0,441753
40	0,10881	0,835176	0,009708	0,443942

SA/SAPenalty relative error comparison



Testování parametrů

Následující měření byla provedena na stejných datech. Vždy byla měněna hodnota pouze jednoho parametru.

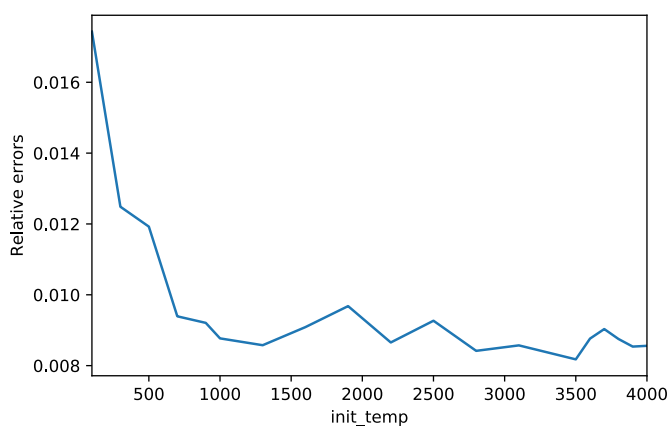
Základní nastavení parametrů při spuštění:

Parametr	Nastavení
Počáteční teplota	500
Konečná teplota	1
Koeficient ochlazování	0,995
Počet interních cyklů	50

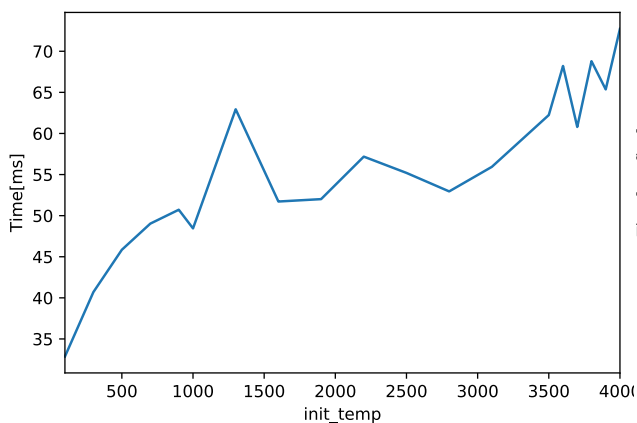
Počáteční teplota

init_temp	max_relative_error	avg_relative_error	elapsed_time	elapsed_configs
100	0,598615	0,017429	32,86	45950
300	0,598615	0,012486	40,7	56900
500	0,598615	0,011925	45,85	62000
700	0,262642	0,009392	49,03	65350
900	0,262642	0,009207	50,72	67900
1000	0,238636	0,008769	48,45	68950
1300	0,233935	0,008577	62,94	71550
1600	0,279835	0,009087	51,72	73600
1900	0,238636	0,00968	52,01	75350
2200	0,233935	0,008654	57,18	76800
2500	0,262642	0,009268	55,2	78050
2800	0,233935	0,008417	52,95	79200
3100	0,233935	0,008572	55,94	80200
3500	0,233935	0,008177	62,24	81450
3600	0,238636	0,008763	68,21	81700
3700	0,238636	0,009033	60,79	82000
3800	0,238636	0,008752	68,79	82250
3900	0,238636	0,008538	65,36	82500
4000	0,238636	0,008559	72,73	82750

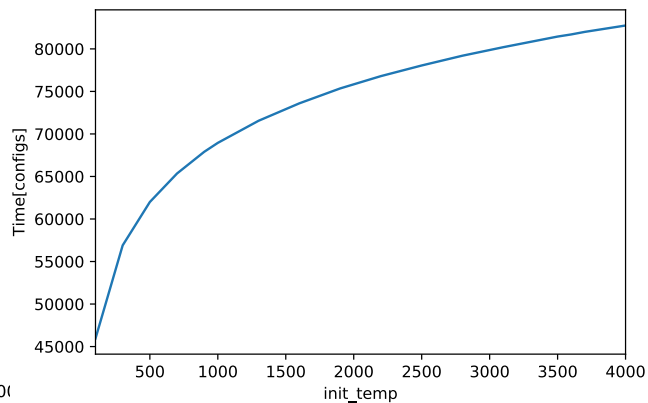
Avg error - init temperatures



Avg speed - init temperatures



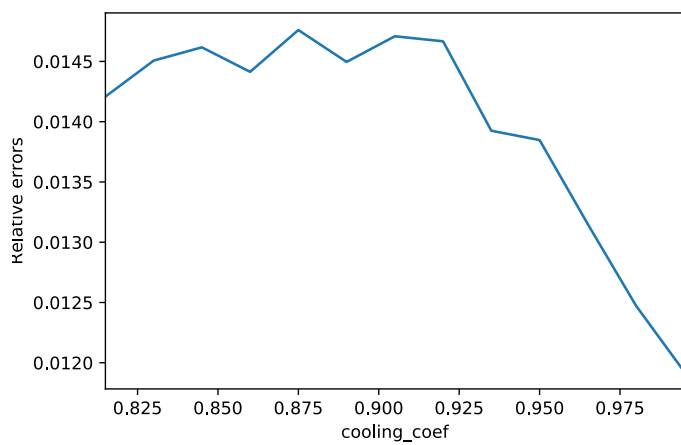
Avg speed - init temperatures



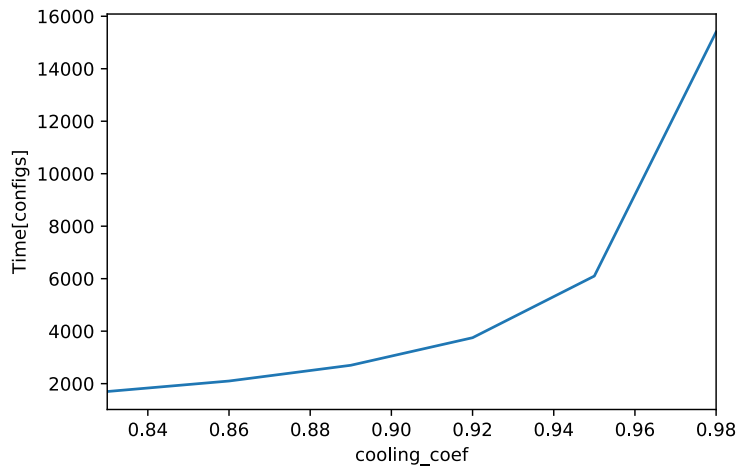
Koeficient ochlazování

cooling_coef	max_relative_error	avg_relative_error	elapsed_time	elapsed_configs
0,83	0,598615	0,014507	1,32	1700
0,86	0,598615	0,014414	1,6	2100
0,89	0,598615	0,014496	2,16	2700
0,92	0,598615	0,014667	3,01	3750
0,95	0,598615	0,013848	4,27	6100
0,98	0,598615	0,012473	10,75	15400

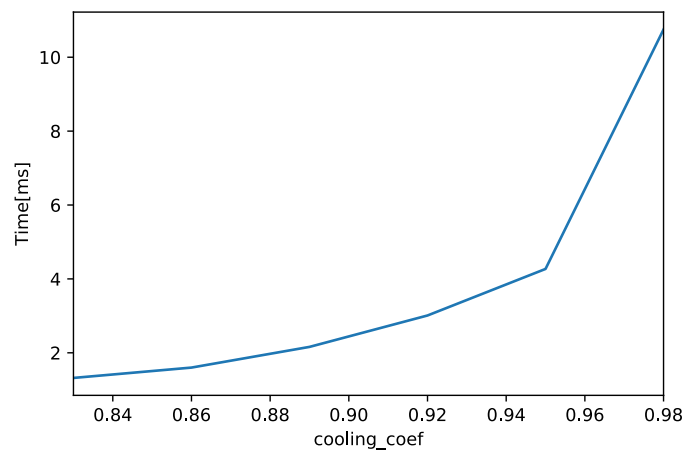
Avg error - cooling



Avg speed - cooling



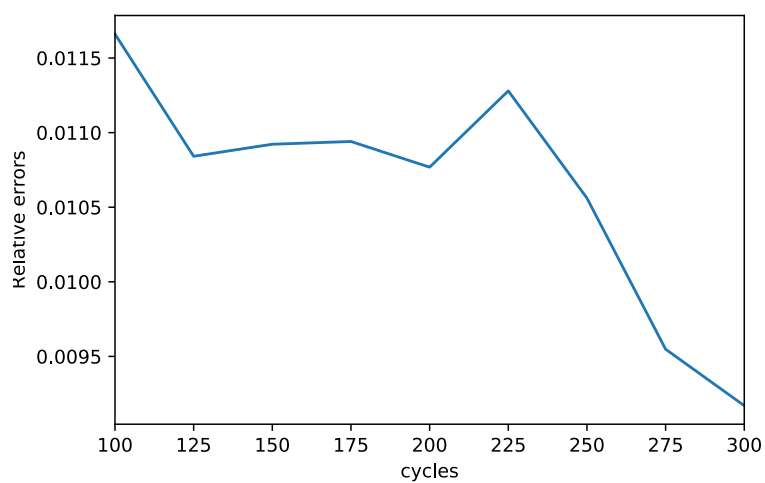
Avg speed - cooling



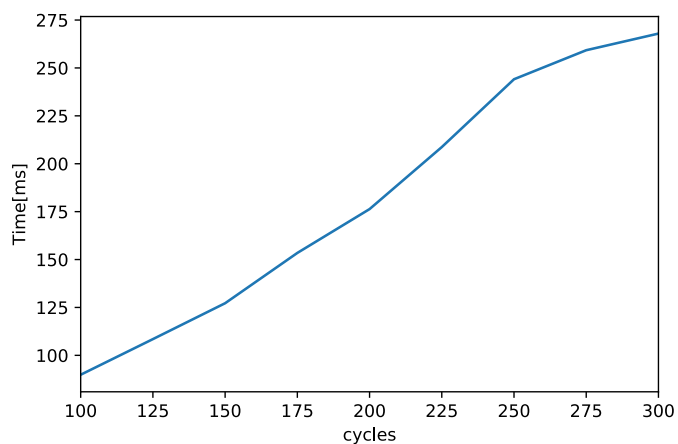
Počet interních cyklů

cycles	max_relative_error	avg_relative_error	elapsed_time	elapsed_configs
100	0,598615	0,01166	89,88	124000
125	0,598615	0,010841	108,47	155000
150	0,598615	0,010922	127,2	186000
175	0,598615	0,010941	153,45	217000
200	0,598615	0,010768	176,33	248000
225	0,598615	0,01128	208,7	279000
250	0,598615	0,010561	244,1	310000
275	0,327719	0,009548	259,25	341000
300	0,262642	0,00917	267,93	372000

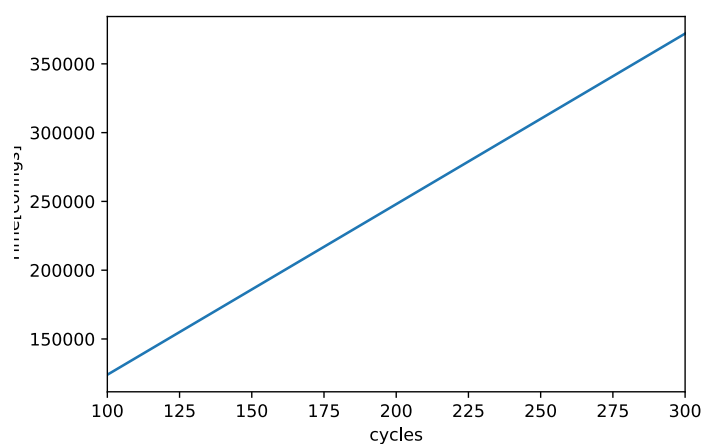
Avg error - cycles



Avg speed - cycles



Avg speed - cycles

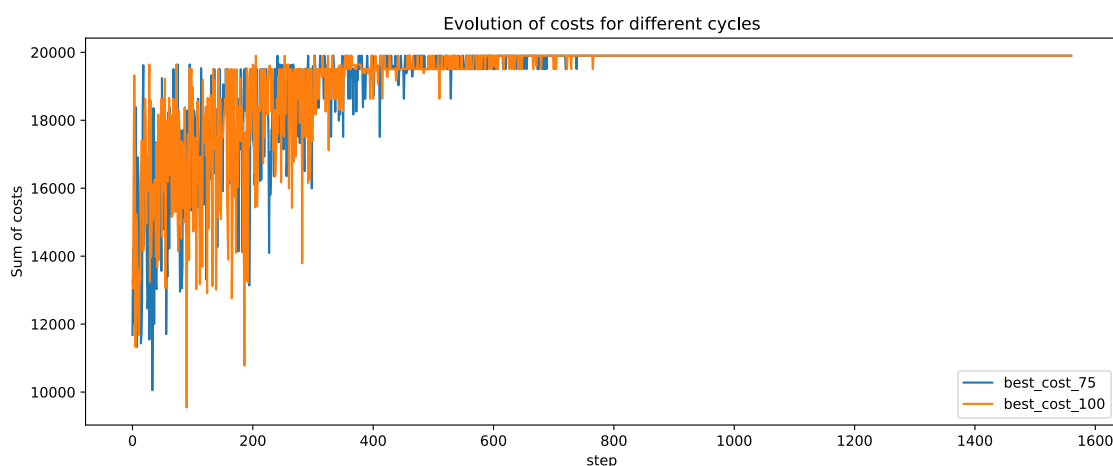
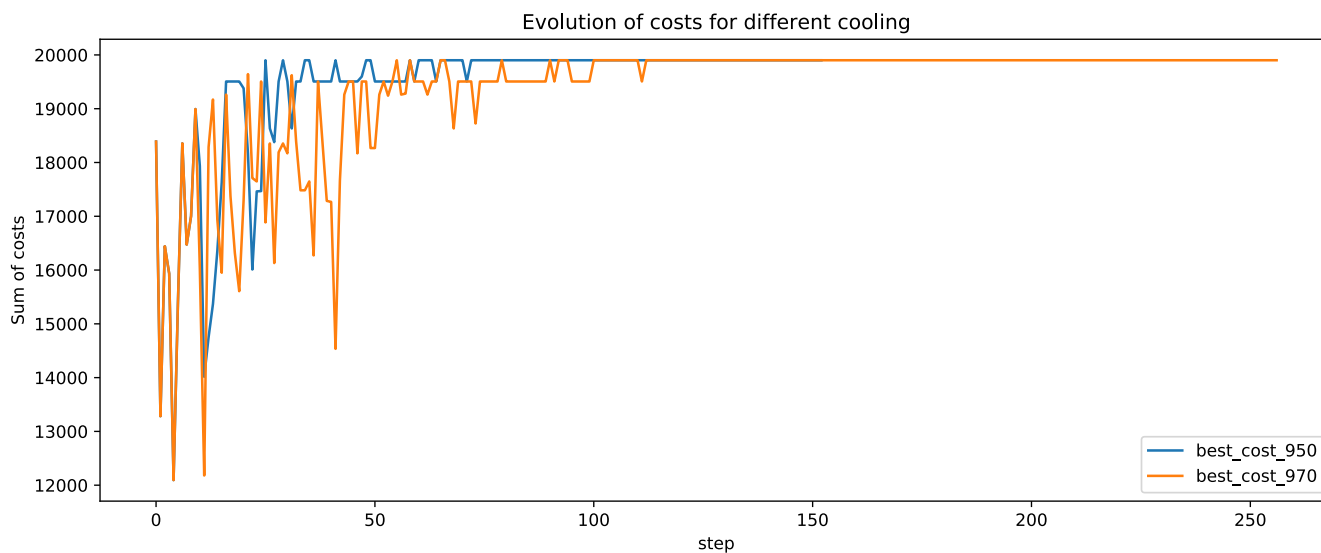
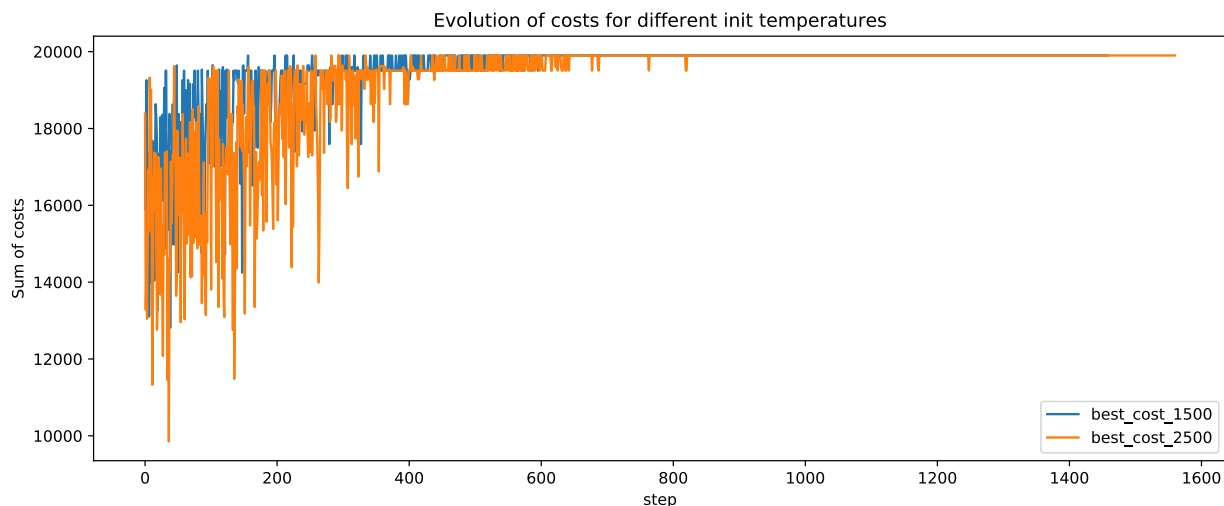


Vývoje řešení

Následující grafy znázorňují průběhy evoluce řešení při změně parametrů. Základní nastavení parametrů:

- počáteční teplota = 2500
- počet cyklů = 50
- koeficient chlazení = 0.995

Jak je viditelné na výsledných grafech, při vysokých teplotách se hodnota řešení často mění, při střední teplotě se ustaluje a při nízkých teplotách se téměř nemění. Toto je chování, které je od algoritmu očekáváno.



Závěr

Nastavení parametrů:

Parametr	Význam
Počáteční teplota	Určuje počet prohledaných řešení a pravděpodobnost výběru horšího řešení. Pravděpodobnost přijetí horšího řešení je závislá i na rozdílu maximální a minimální ceny dané instance problému. Z naměřených hodnot se zdá, že je vhodné tuto hodnotu nastavit blízko rozdílu maximální a minimální hodnoty ceny dané instance problému.
Konečná teplota	Konstantně nechána na hodnotě 1.0.
Koeficient ochlazování	Určuje počet prohledaných řešení. Z naměřených hodnot se zdá, že je vhodné tuto hodnotu nastavit velmi blízko 1.
Počet interních cyklů	Určuje počet prohledaných řešení. Navýšení tohoto parametru podstatně zvyšuje počet prohledaných řešení a snižuje rychlost výpočtu. Z naměřených hodnot se zdá, že je vhodné tuto hodnotu nastavit na hodnotu okolo 100 cyklů a upravovat ostatní parametry.

Přiložené důležité soubory

Název	Popis
analysis	Složka s analýzou algoritmů. Obsahuje jupyter notebooky a výstupní hodnoty instancí, které byli algoritmy zpracovány.
data	Sady instancí řešeného <i>problému</i> .
algorithm_tester	Python modul testeru algoritmů.
tests	Unit testy programu.
analysis.ipynb	Jupyter notebook použitý pro zpracování výstupních dat.
package_algorithms	Balíček obsahující všechny implementované algoritmy.