

Domácí úloha č. 5 – Knapsack problém IV.

Specifikace úlohy

Cílem této domácí úlohy bylo seznámit se s vybranou pokročilou iterativní technikou na řešení problému batohu. Bylo na výběr z několika algoritmů a já jsem si vybral hned ten první: **Simulované ochlazování**. Hlavním cílem bylo seznámit se s danou heuristikou, zejména se způsobem, jakým se nastavují její parametry (rozvrh ochlazování) a modifikace (zjištění počáteční teploty).

Nástroje k řešení

K implementaci jsem využil programovací jazyk **Java** pod prostředím **NetBeans**. Všechny výpočty běželi na procesoru Intel Core 2 Duo 3.00 GHz a pod operačním systémem Microsoft Windows 7. Výsledky byly zpracovány tabulkovým procesorem Microsoft Excel.

Výsledný zdrojový kód je spouštěn ze souboru **Main.java**, zbytek kódu je přehledně rozdělen do tříd.

K měření času jsem použil funkci **System.currentTimeMillis()**.

Popis zvoleného algoritmu

Simulované ochlazování je heuristický algoritmus, který pomůže algoritmu, který uváznul v lokálním minimu, dostat se do minima globálního. To zahrnuje připuštění tahů, které jsou horší (vedou k horšímu řešení), ale po několika krocích nám umožní dostat se přes „kopec“. Simulované ochlazování lze popsat pomocí pseudoalgoritmu, který vychází z přednášek:

```
while ( !frozen(teplota) ) {
    i = 0;
    while( equilibrium(i, pocetPolozek) ) {
        i++;
        novyStav = ziskejStav();
        if ( novyStav.price > staryStav.price ) {
            nejlepsiStav = nevystav;
        }
    }
    coolDown(teplota);
}
```

Parametry algoritmu je koeficient počáteční teploty (určuje velikost počáteční teploty), koeficient ochlazování (o kolik snížíme teplotu v každém kroku), minimální teplota (pod kterou algoritmus již nejde) a koeficient equilibra (určující počet stavů, neboli počet kroků v každém cyklu zchlazování). Všechny tyto parametry nám ovlivňují průběh algoritmu a určují, jaké stavy budou ještě přípustné při prohledávání okolí. Naším úkol je změřit, jaký vliv bude mít změna některého z těchto parametrů.

Implementace

Implementace byla provedena v jazyce Java. Volba algoritmu je provedena návrhovým vzorem Strategy. Veškeré nastavení, jak programu, tak i algoritmu se řeší v **main.java**. Simulované ochlazování je celé uloženo v souboru `src\batoh4\SimulatedCooling.java`.

Naměřené hodnoty a výsledky

Všechna měření jsem prováděl jak pro čas, tak i pro počet expandovaných stavů. V grafech ale uvádím pouze počet expandovaných stavů, což má lepší vypovídací vlastnost, než doba běhu algoritmu.

Měření provádím tak, že všechny parametry zafixuji a jeden z nich měním. Výchozí stav je:

Koeficient počáteční teploty = 500, T_{min} = 1, Zchlazování = 0.85, Equilibrium = 2, Instance 40

Závislost doby běhu algoritmu na počáteční teplotě

Pro tuto závislost jsem měnil koeficient počáteční teploty od 1 do 10, což se ukázalo jako nedostatečné, proto jsem déle pokračoval logaritmicky až do milionu. Pro tuto závislost jsem naměřil tyto hodnoty:

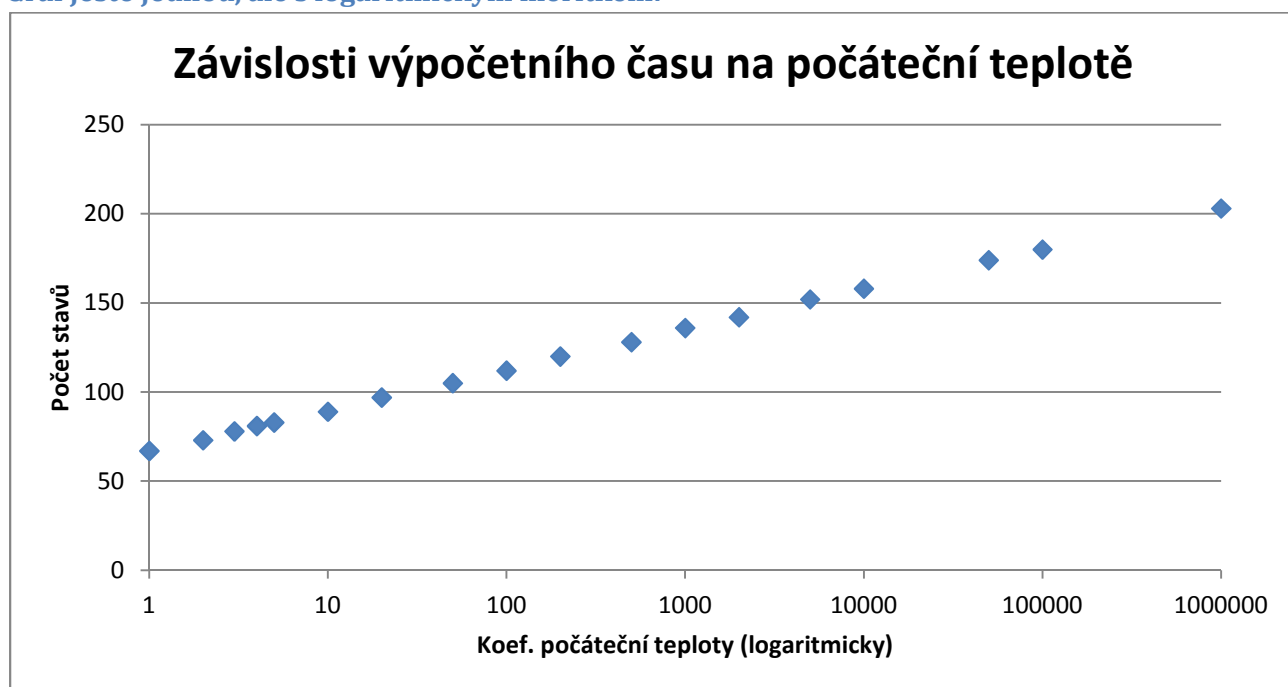
T	Čas	# stavů	Relativní chyba	Maximální chyba
1	22	67	0,003352999	0,167649951
2	22	73	0,003633235	0,18166175
3	24	78	0,003677483	0,18387414
4	23	81	0,002168142	0,10840708
5	22	83	0,004631268	0,231563422
10	37	89	0,003235005	0,161750246
20	36	97	0,003313668	0,165683382
50	35	105	0,002497542	0,124877089
100	26	112	0,003274336	0,163716814
200	26	120	0,00287119	0,143559489
500	27	128	0,001966568	0,098328417
1000	28	136	0,002320551	0,116027532
2000	33	142	0,003077679	0,153883972
5000	32	152	0,00359882	0,179941003
10000	32	158	0,003554572	0,177728614
50000	36	174	0,0023353	0,116764995
100000	35	180	0,003362832	0,168141593
1000000	40	203	0,003220256	0,161012783

Grafické znázornění závislosti výpočetního času na počáteční teplotě:



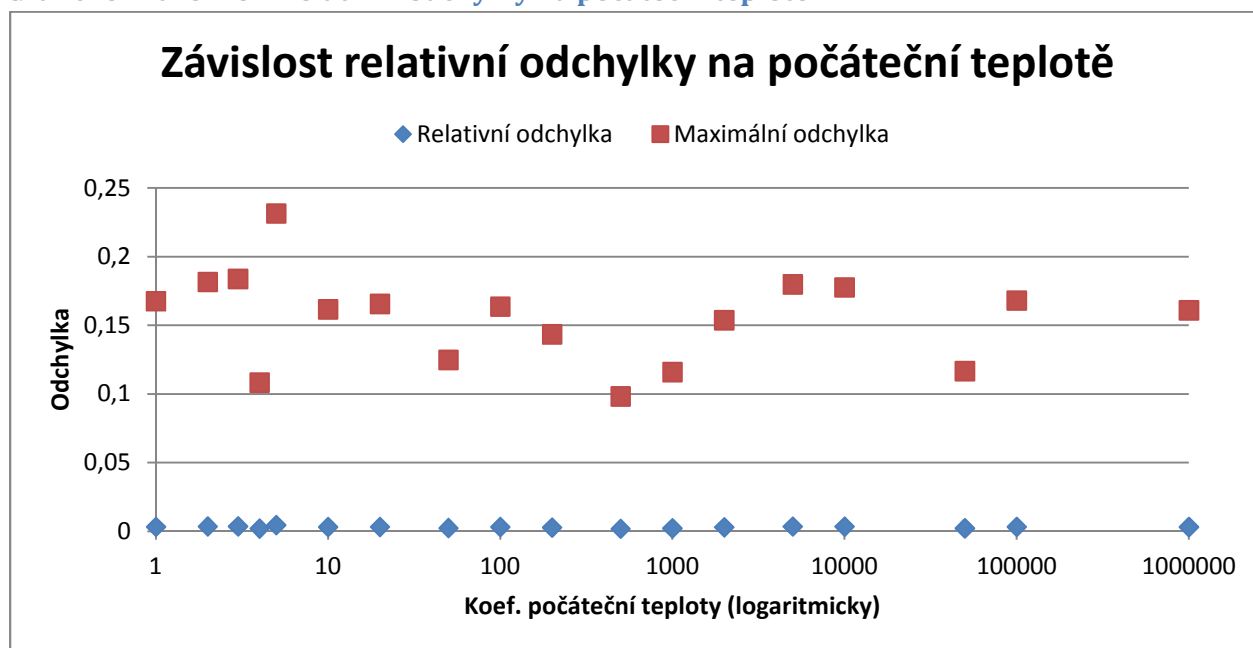
Z grafu je vidět, že nejdříve je růst významný, ale po čase se uklidní a roste jen pozvolna. Při zvyšování počáteční teploty roste počet prohledávaných stavů, protože zchlazovací koeficient zůstává stejný a roste tak počet kroků nutných pro to, abychom se dostali na minimální teplotu.

Graf ještě jednou, ale s logaritmickým měřítkem:



Na grafu s logaritmickým měřítkem je trend daleko lépe vidět.

Grafické znázornění relativní odchylky na počáteční teplotě:



Je vidět, že relativní, ani maximální chyba se nemění. Nejnižší odchylka byla pro teplotní koeficient 500. Hodnotu 500 jsem pak použil rovněž do výchozí konfigurace, z které vycházím pro každé měření.

Závislost výpočetního času na minimální teplotě

Pro měření této závislosti jsem měnil minimální cenu tak, abych se dostal až na počet expandovaných stavů roven jedné. Se zvyšující se minimální teplotou má algoritmus daleko menší šanci vyskočit z lokálního minima.

Naměřené hodnoty závislost minimální teploty:

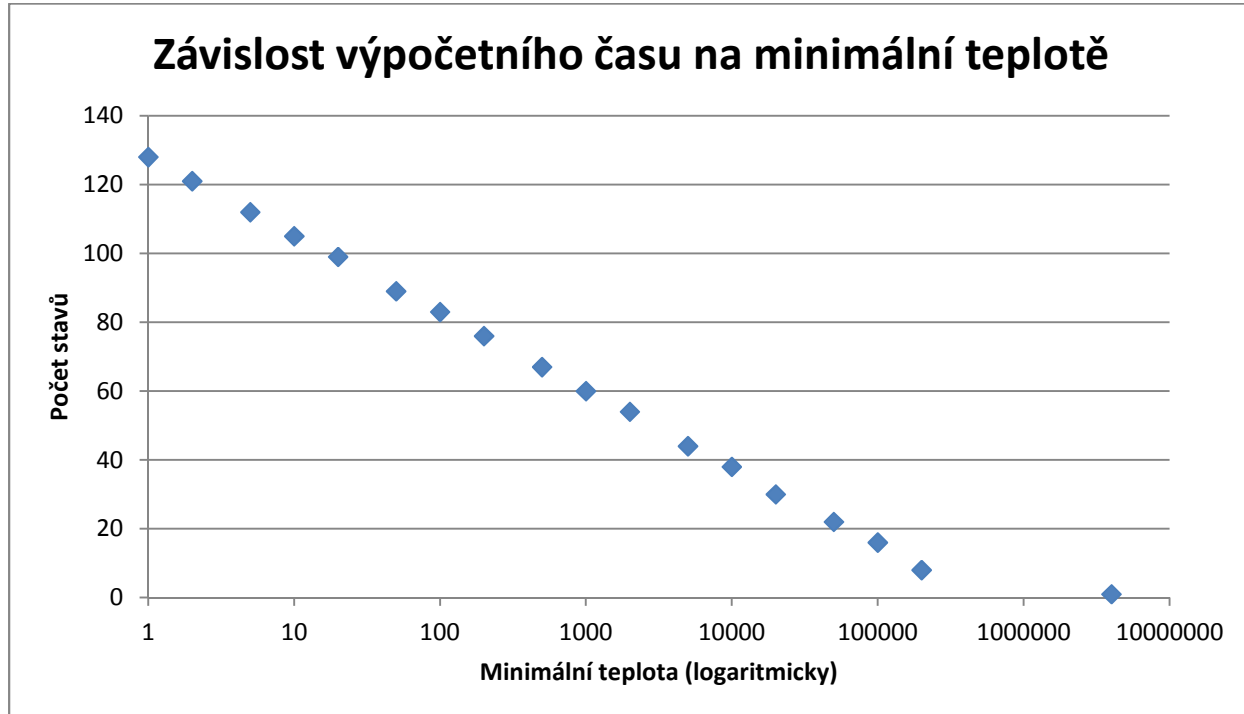
Tmin	Čas	Expandováno	Relativní chyba	Maximální chyba
1	30	128	0,002767945	0,138397247
2	28	121	0,003230088	0,161504425
5	27	112	0,003082596	0,154129794
10	26	105	0,003308751	0,165437561
20	24	99	0,002792527	0,139626352
50	22	89	0,003352999	0,167649951
100	22	83	0,003372665	0,168633235
200	21	76	0,003652901	0,182645034
500	21	67	0,003692232	0,184611603
1000	18	60	0,003308751	0,165437561
2000	17	54	0,00320059	0,160029499
5000	14	44	0,003456244	0,172812193
10000	14	38	0,003987217	0,199360865
20000	15	30	0,003333333	0,166666667
50000	11	22	0,004754179	0,237708948
100000	9	16	0,00519174	0,259587021
200000	7	8	0,005722714	0,286135693
4000000	5	1	0,010781711	0,539085546

Grafické znázornění naměřených hodnot:



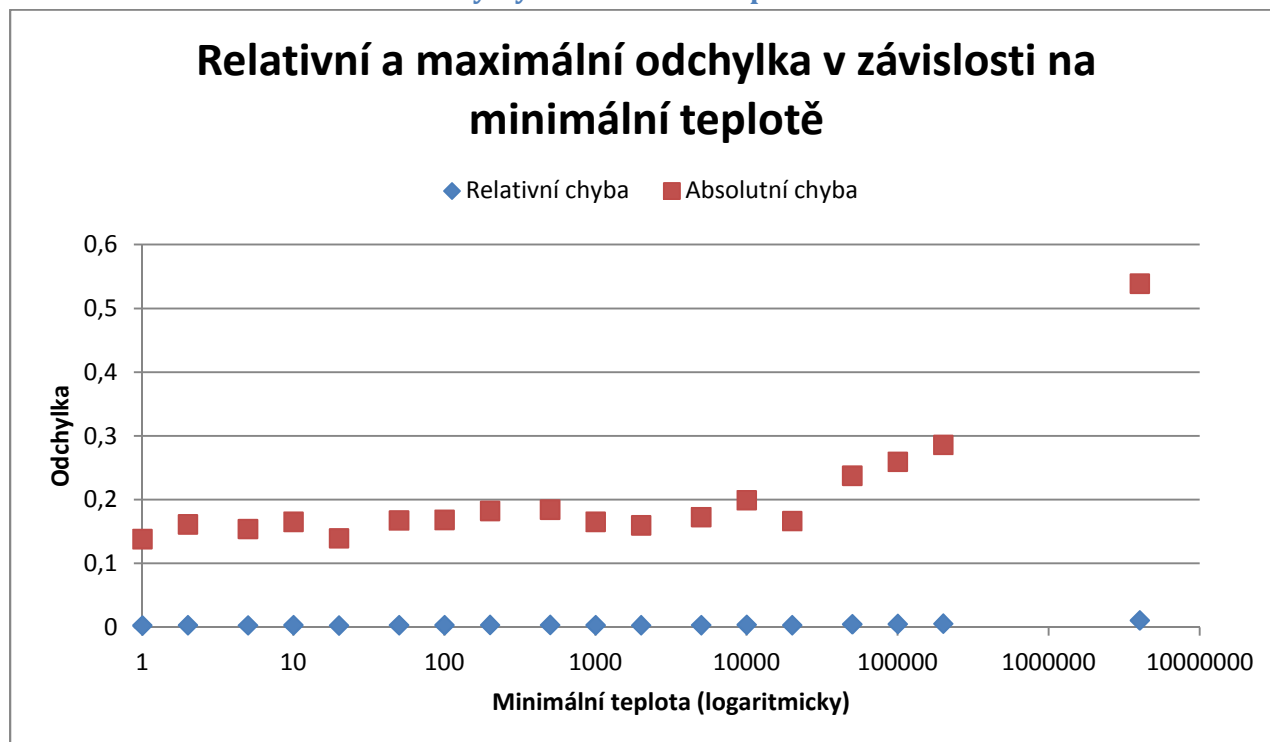
Počet expandovaných stavů klesá na začátku velice strmě s rostoucí minimální teplotou, poté se klesání uklidní. Je to přesně opačná závislost než růst počáteční teploty. Při rostoucí minimální teplotě klesá počet stavů, které budeme expandovat, protože klesá počet zchlazovacích cyklů.

Graf ještě jednou, ale logaritmicky:



Na grafu s logaritmickým měřítkem je opět lépe vidět trend poklesu.

Grafické znázornění relativní odchylky na minimální teplotě:



Relativní i maximální chyba s klesajícím počtem procházených stavů roste, protože algoritmus má daleko menší rozsah a nemá šanci opustit lokální minimum.

Závislost výpočetního času na koeficientu zchlazování

Koeficient zchlazování jsem nastavoval dle přednášek v rozsahu 0,5 až 0,999 (okolí 0,85).

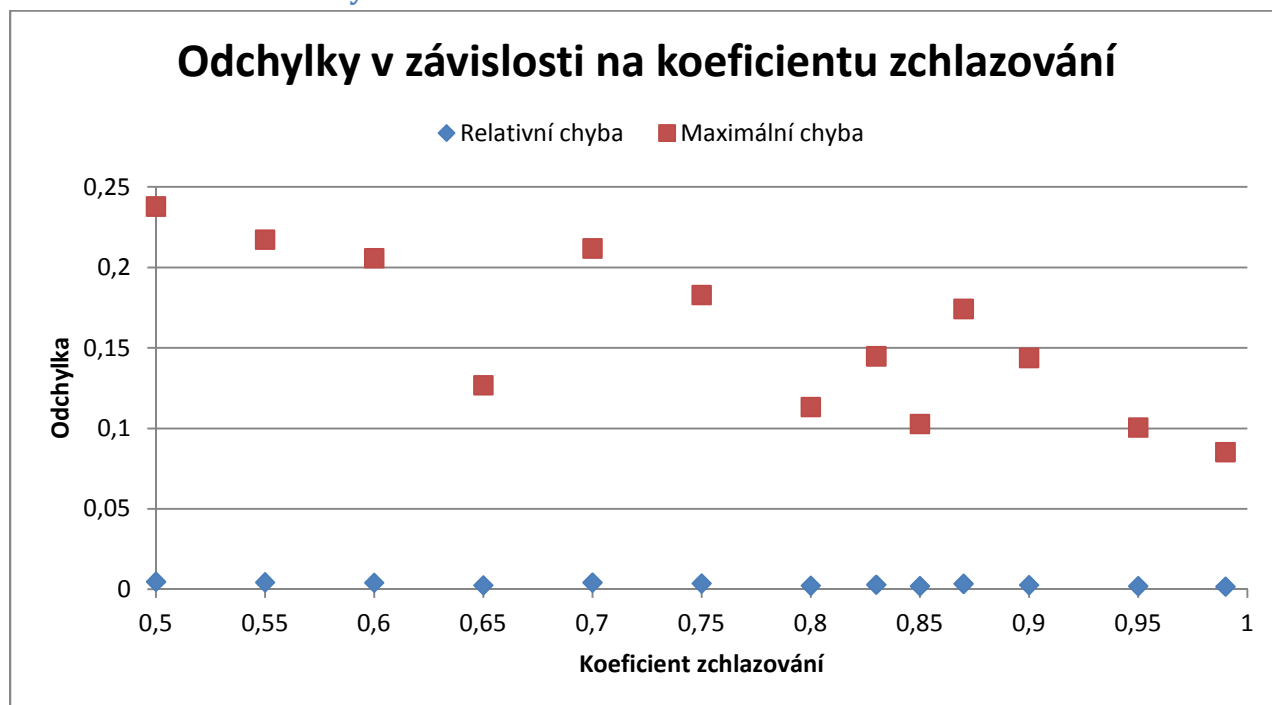
Zchlazování	Doba výpočtu	Expandováno	Relativní chyba	Maximální chyba
0,5	13	30	0,004759095	0,237954769
0,55	13	35	0,004346116	0,217305801
0,6	15	41	0,004115044	0,205752212
0,65	16	49	0,002536873	0,126843658
0,7	17	59	0,004237955	0,211897738
0,75	24	73	0,003657817	0,182890855
0,8	25	94	0,00226647	0,1133235
0,83	28	112	0,002895772	0,144788594
0,85	29	128	0,002055064	0,102753196
0,87	35	150	0,003485742	0,174287119
0,9	43	198	0,0026765	0,143824975
0,95	80	406	0,002010816	0,100540806
0,99	516	2067	0,001705998	0,085299902

Grafické znázornění výpočetního času na koeficientu zchlazování:



Z grafu je vidět, že s rostoucím koeficientem zchlazování roste počet prozkoumaných stavů a tudíž i časová náročnost algoritmu.

Grafické znázornění odchylek v závislosti na koeficientu zchlazování:



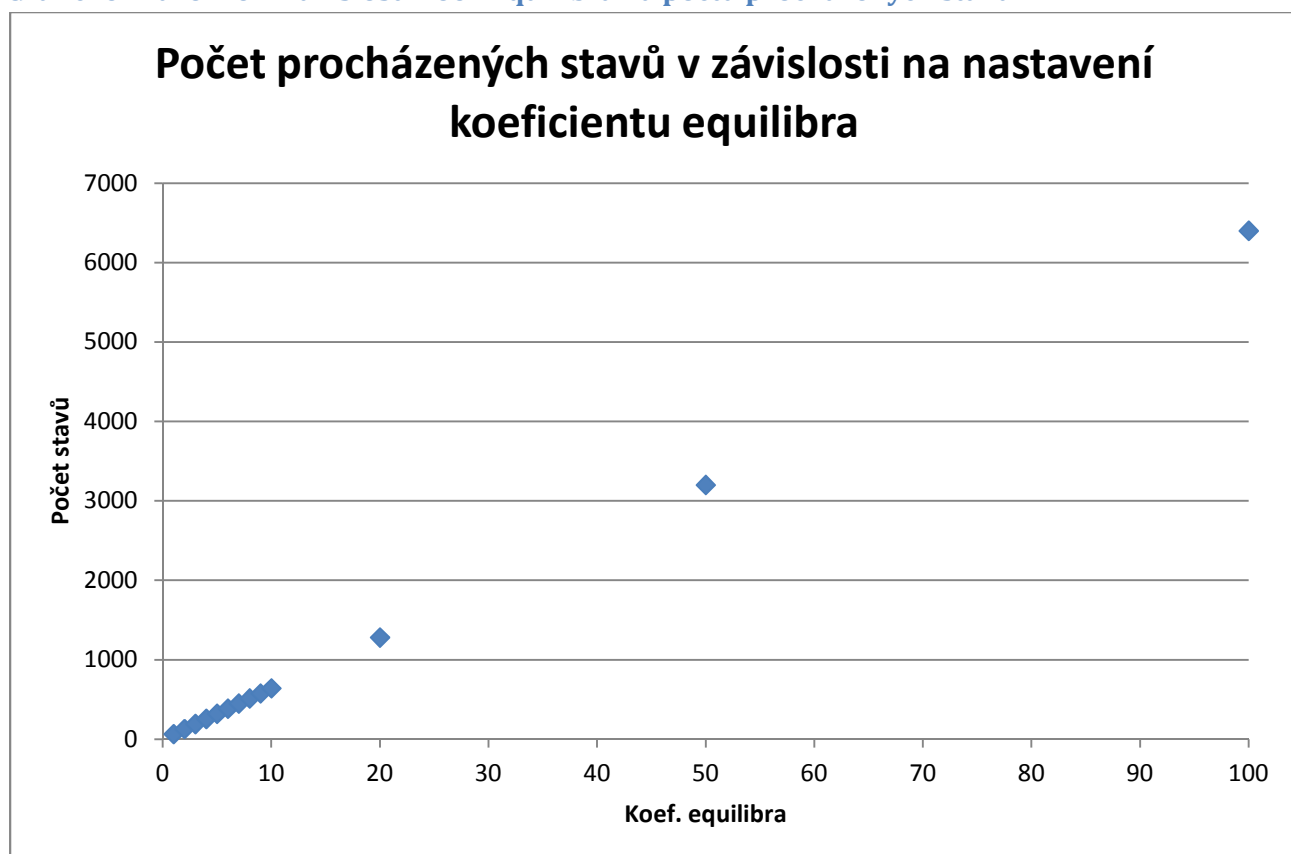
Relativní i maximální odchylka klesá s rostoucím koeficientem zchlazování a s rostoucím počtem procházených stavů.

Závislost změny koeficientu equilibra (počtu kroků)

Koeficient equilibra jsem měnil od jedničky až po stovku. Počet expandovaných stavů rostl úměrně se zvoleným počtem kroků.

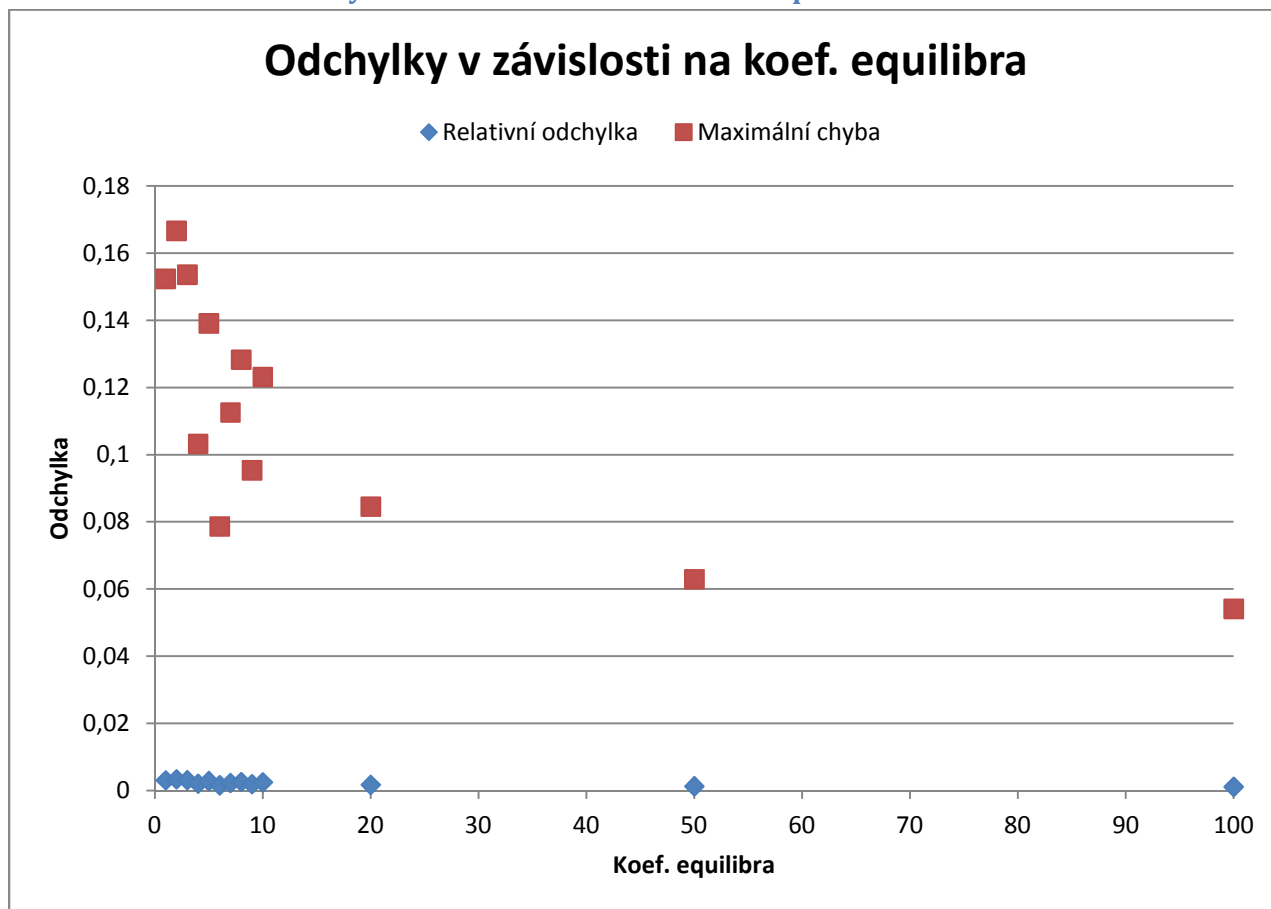
Koef.equil.	Doba výpočtu	Expandováno	Relativní chyba	Maximální chyba
1	21	64	0,003048181	0,152409046
2	34	128	0,003333333	0,166666667
3	41	192	0,003072763	0,153638151
4	55	256	0,002064897	0,103244838
5	61	320	0,002782694	0,13913471
6	72	384	0,001573255	0,078662734
7	78	448	0,002251721	0,112586037
8	90	512	0,002566372	0,128318584
9	97	576	0,001907571	0,095378564
10	106	640	0,002463127	0,123156342
20	233	1280	0,001691249	0,084562439
50	525	3200	0,001258604	0,062930187
100	985	6400	0,001081613	0,054080629

Grafické znázornění závislosti koef. Equilibra na počtu procházených stavů:



Závislost počtu stavů je lineární na nastavení koeficientu equilibra což by se dalo očekávat, protože dosažení equilibra je úměrné počtu stavů.

Grafické znázornění odchylek v závislosti na koeficientu equilibra:



Z grafu je vidět, že s rostoucím počtem procházených stavů klesá relativní i maximální odchylka.

Závěr

Z naměřených hodnot lze usoudit chování algoritmu, respektive vliv jeho parametrů na počet procházených stavů a s tím související výpočetní náročnosti a odchylky. Zvětšováním rozdílu teplot (buď snižováním minimální teploty, nebo zvyšováním počáteční teploty) dochází k zvětšování počtu procházených stavů. Při změně koeficientu equilibra rovněž rostl počet procházených stavů a zároveň klesla odchylka od optimálního řešení.