

Domácí úloha č. 6 – Problém vážené splnitelnosti booleovské formule

Specifikace úlohy

Cílem této domácí úlohy bylo najít ohodnocení booleovské formule F ve tvaru $Y=(y_1, y_2, \dots, y_n)$ proměnných x_1, x_2, \dots, x_n tak, aby $F(Y)=1$ a součet vah proměnných, které jsou ohodnoceny jedničkou, byl maximální. Pro řešení této úlohy byla zvolena heuristika **Simulované ochlazování**. Optimální řešení je ověřeno proti algoritmu **hrubé síly**.

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 64bit. 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()**.

Vstupní soubory jsem použil ze stránky <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>.

Popis zvoleného algoritmu

Simulované ochlazování je heuristický algoritmus, který pomůže algoritmu, který uváznuje 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 = novyStav;
        }
    }
    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 úlohy

Implementace byla provedena v jazyce Java. Volba algoritmu (bruteForce/simulatedCooling) 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\paasat\SimulatedCooling.java`.

Implementace logiky formule

Hlavní logika vyhodnocování formule je v souboru **Formula.java**, kde lze zjistit, jestli je formule splnitelná, nebo nikoli. Dále součet vah jedničkových proměnných a počet splnitelných klauzulí. Datová struktura pro ukládání formulí je dynamická (ArrayList), takže se úloha neomezuje pouze na 3SAT.

Implementace načítání souborů a generování vah

Implementaci jsem kompletně převzal z předchozí úlohy. Bylo však potřeba upravit načítání souborů, protože instance jsou uloženy ve více souborech, na rozdíl od batohu, kde bylo vše v jednom souboru. Rovněž bylo nutno naprogramovat generátor vah. Pro každý soubor s formulí se tak generuje soubor, ve kterém jsou uloženy váhy jednotlivých proměnných. Celý soubor je ukončen znakem 0, aby šlo použít stejný algoritmus načítání, jako při načítání jednotlivých klauzulí. Pokud již soubor existuje, dále se negeneruje, aby nebyl započítán do času výpočtu. Váhy jsou tedy pro všechna měření stejné. Implementace načítání a generování vah je v souborech **FileLoader.java** a **DirectoryLoader.java**.

Implementace simulovaného ochlazování

Implementaci heuristiky jsem rovněž převzal z minulé úlohy, bylo nutno jí ale trochu upravit. Klíčovým bodem celé heuristiky je vyhodnocování, jestli vedlejší vygenerovaný stav přijmout, nebo ne. Generování vedlejšího stavu jsem zachoval (změna jednoho náhodného bitu v ohodnocení). Musel jsem však pozměnit porovnávání dvou stavů, což se děje ve funkci **isBetter()**. Přijímány jsou i stavy, které nejsou splnitelné. Každý výpočet startuje ve stavu, který začíná jedničkou a pak jsou samé nuly (1000000...).

Implementace porovnávání dvou stavů

Cílem úlohy je maximalizovat váhu kladných proměnných. Zároveň je ale nutné mít celou formuli splnitelnou, takže maximalizovat celkový počet splněných klauzulí (**Max Sat**). Je potřeba hlídat oba cíle, ale počet splněných klauzulí je pro nás důležitější, protože to vede ke splnitelnosti celé formule.

Základní představu o vyhodnocování bych měl, ale rozhodl jsem se, že zkusím jednotlivé možnosti porovnat a vyhodnotit, jaký algoritmus porovnávání dvou stavů vyjde nejlépe. Pro každou sadu dat vybírám pouze prvních 20 souborů (20 formulí).

Porovnávání dvou stavů pro data 20/91

Následující tabulka ukazuje naměřené hodnoty pro vstupní soubor 20/91 (počet proměnných, počet klauzulí) a pro různé algoritmy srovnání. Teplotní koeficient 100, minimální teplota 1, zchlazení 0,95, equilibrium 100. Expandováno bylo vždy 370 000 stavů. Pro každý řádek byla provedena minimálně 3 měření a výsledek zprůměrován.

Algoritmus porovnávání dvou stavů – data 20/91	Splněných z 20ti	suma vah	čas [ms]	průměrná suma vah
A1) Podle součtu vah jedničkových stavů (suma vah)	13	8953	114124	517
A2) Podle počtu splněných klauzulí	13,6	9249	114426	507
B) Podle součtu vah a počtu splněných klauzulí (součet)	12,6	8498	114566	402
C) Porovnání součtu splněných a při shodě i váhy	13	8876	115831	411
C) Porovnání součtu splněných a při shodě i váhy + bonus 10	14	9520	116167	509
C) Porovnání součtu splněných a při shodě i váhy + bonus 20	15,3	10169	115642	499
C) Porovává sumu splnitelných a při shodě i váhy + bonus 50	15,6	10354	113577	493
D) Preference splněných formulí	14	8946	116314	478

Jednotlivé algoritmy jsou implementovány ve funkci **isBetter()** a zakomentovány. Jako nejlepší variantu jsem eliminoval posuzovat dva stavy dle počtu splněných klauzulí, ale i podle sumy vah (což je součet vah proměnných, které jsou ohodnocené jedničkou a zároveň to počítám pouze u formulí, které jsou splnitelné). Vzhledem k tomu, že větší prioritu má pro nás splnitelnost celé formule, takže při splnění celé formule je ke stavu přičtena navíc ještě hodnota 50.

Pro porovnání algoritmus hrubou silou expanduje při tomto nastavení 1 048 576 stavů (oproti 370 000 heuristiky).

Porovnávání dvou stavů pro data 50/80

Abych měl jistotu, že vybrané řešení poskytuje přijatelné výsledky pro celé spektrum vstupních instancí, rozhodl jsem se vyzkoušet některé algoritmy i na větších instancích, konkrétně 50/80 (počet proměnných 50 a počet klauzulí 80). Počet klauzulí byl schválně snížen, abych se dostal na přijatelný čas, ale zároveň nebude mít optimální řešení, které půjde dobře porovnat. Kvalita a rychlost řešení je totiž přímo úměrná poměru počtu proměnných a počtu klauzulí.

Níže tabulka pro data 50/80:

Algoritmus	splněných	suma vah	čas [ms]	průměrná suma vah
A1) Podle součtu vah jedničkových stavů	16,6	26724	58228	1204
A2) Podle počtu splněných klauzulí	16,6	27034	59936	1218
B) Podle součtu vah a počtu splněných klauzulí	17	26697	58264	941
C) Porovnání součtu splněných a při shodě i váhy + 50	17,3	27275	58422	1180
D) Preference splněných formulí	17	26673	58821	1176

I u této sady dat se podařilo najít nejlepších výsledků variantou C) s bonusem 50 při zachování stejné časové náročnosti a expanze stejného počtu stavů. Rozhodl jsem se u této implementace porovnání stavů zůstat.

Parametry simulovaného ochlazování

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. Je tedy potřeba určit výchozí stav, který bude vždy přibližně uprostřed a budu zkoušet hodnoty kolem něho. Zchlazovací koeficient je jasný, ten nastavím doprostřed, abych pak měl možnost hýbat s ním na každou stranu. Nejdůležitější bude správně nastavit teploty a hodnotu equilibra, které nejvíce určuje prohledávání stavového prostoru a mohu tím ovlivnit, jestli se podaří nalézt globální optimum, nebo nikoli.

Výchozí stav jsem stanovil na následující hodnoty, které byly již upraveny dle měření provedených dále:

Koeficient počáteční teploty = 500, Tmin = 1, Zchlazování = 0.85, Equilibrium = 100

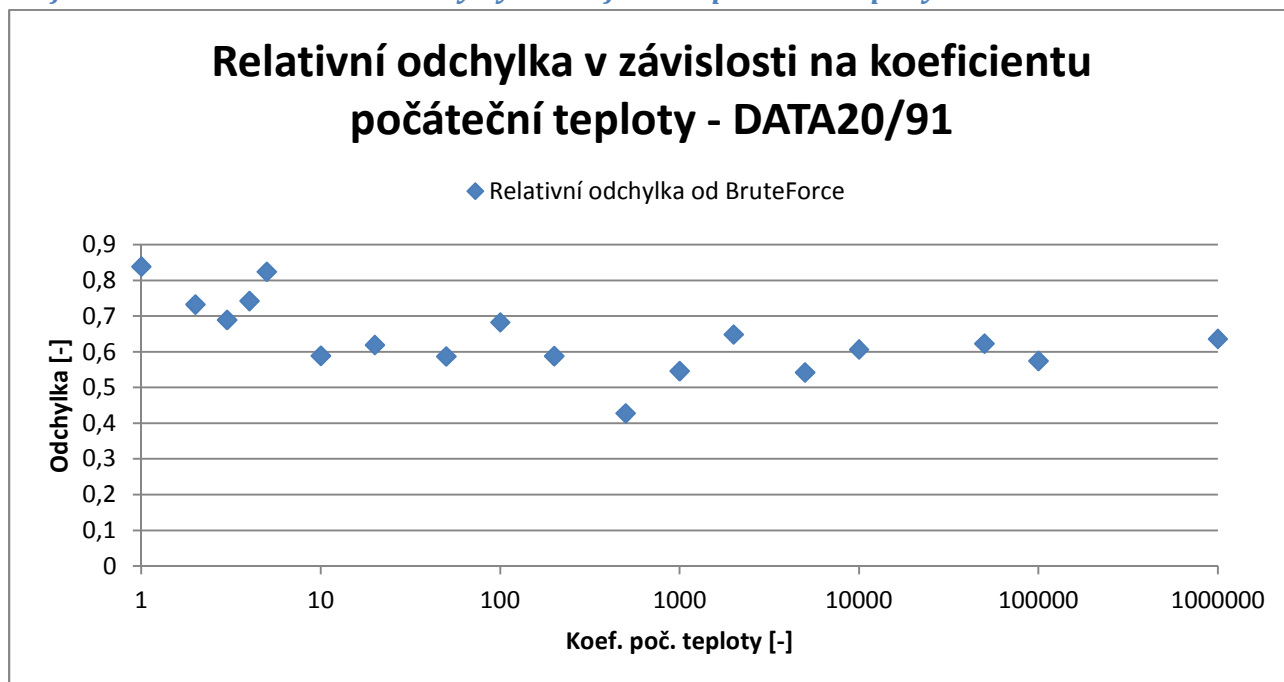
Závislost doby běhu algoritmu na počáteční teplotě

Pro tuto závislost jsem naměřil tyto hodnoty, pro datový soubor 20/91:

	# stavů	suma vah	čas	exp. stavů	váhy/stavy	stavy/exp.	odchylka
BruteForce	20	15606	109205	20971520	0,0007442	9,537E-07	0
1	4	2504	19063	124000	0,0201935	3,226E-05	0,8395489
2	6	4162	15646	1400000	0,0029729	4,286E-06	0,7333077
3	7	4840	16494	1480000	0,0032703	4,73E-06	0,6898629
4	7	4009	17316	1560000	0,0025699	4,487E-06	0,7431116
5	4	2731	17756	1600000	0,0017069	0,0000025	0,8250032
10	9	6405	20055	1800000	0,0035583	0,0000005	0,5895809
20	9	5934	21852	1960000	0,0030276	4,592E-06	0,6197616
50	10	6432	24558	2200000	0,0029236	4,545E-06	0,5878508
100	9	4943	26192	2360000	0,0020945	3,814E-06	0,6832628
200	9	6417	27931	2520000	0,0025464	3,571E-06	0,588812
500	14	8921	30608	2760000	0,0032322	5,072E-06	0,4283609
1000	10	7072	32999	2920000	0,0024219	3,425E-06	0,546841
2000	8	5476	35185	3080000	0,0017779	2,597E-06	0,6491093
5000	11	7132	37707	3320000	0,0021482	3,313E-06	0,5429963
10000	9	6129	39251	3480000	0,0017612	2,586E-06	0,6072664
50000	10	5871	43055	3880000	0,0015131	2,577E-06	0,6237985
100000	10	6632	44856	4040000	0,0016416	2,475E-06	0,5750352
1000000	9	5667	51480	4640000	0,0012213	1,94E-06	0,6368704

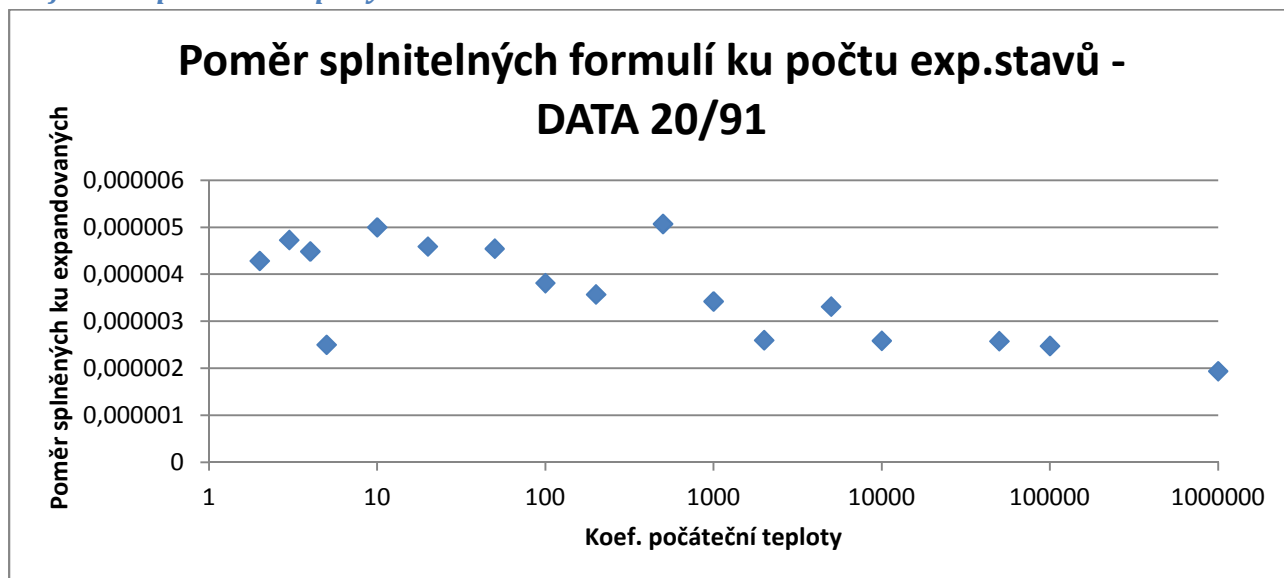
Vzhledem k tomu, že mám změřená data i hrubou silou, mohu je srovnat a zjistit nejmenší odchylku, která se u vstupní datové sady 20/91 nacházela kolem koeficientu vstupní teploty 500. Grafy přiloženy níže.

Grafické znázornění relativní odchylky na koeficientu počáteční teploty:



Graf je znázorněn s logaritmickou osou x, která nejlépe reprezentuje změnu teplotního koeficientu při měření. Z grafu je vidět, že nejmenší odchylka vůči algoritmu hrubou silou je kolem teplotního koeficientu 500. Vyšší teploty nám pak akorát zvyšují časovou náročnost algoritmu.

Grafické znázornění závislosti poměru splnitelných formulí a expandovaných stavů na koeficientu počáteční teploty:



Na tomto grafu je znázorněno, poměr počtu nalezených řešení (splnitelných formulí) ku počtu celkově procházených stavů. Opět u teplotního koeficientu 500 je hodnota největší. Tzn. že z určitého

počtu stavů byl největší podíl těch splnitelných, což je důležité, protože se snažíme najít nejlepší řešení za co nejkratší čas.

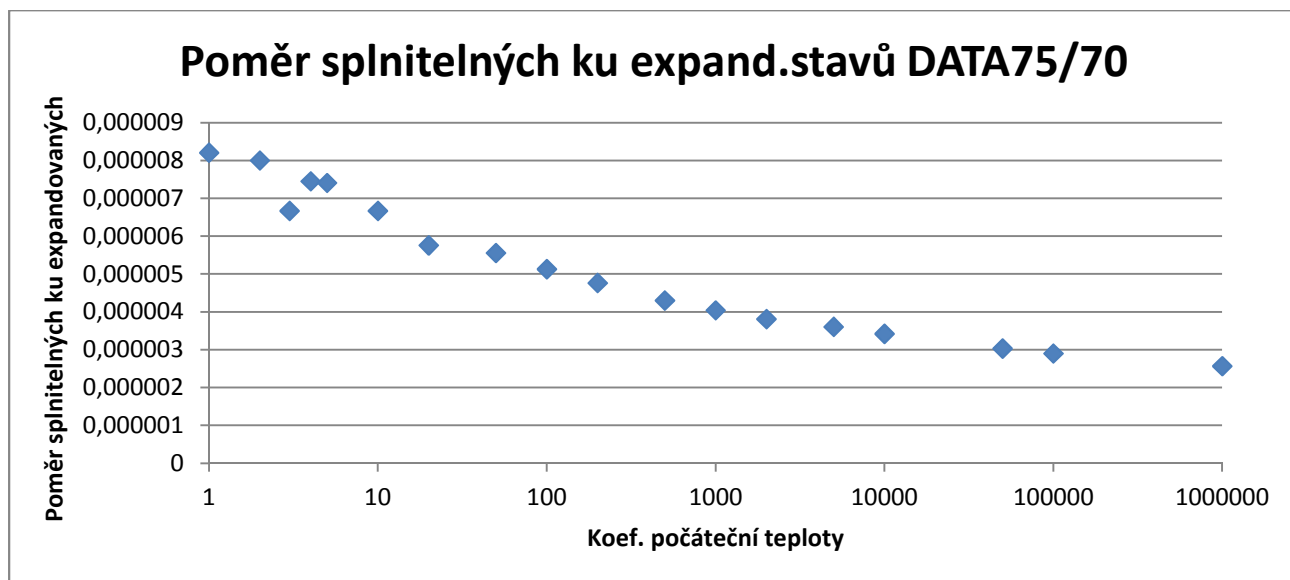
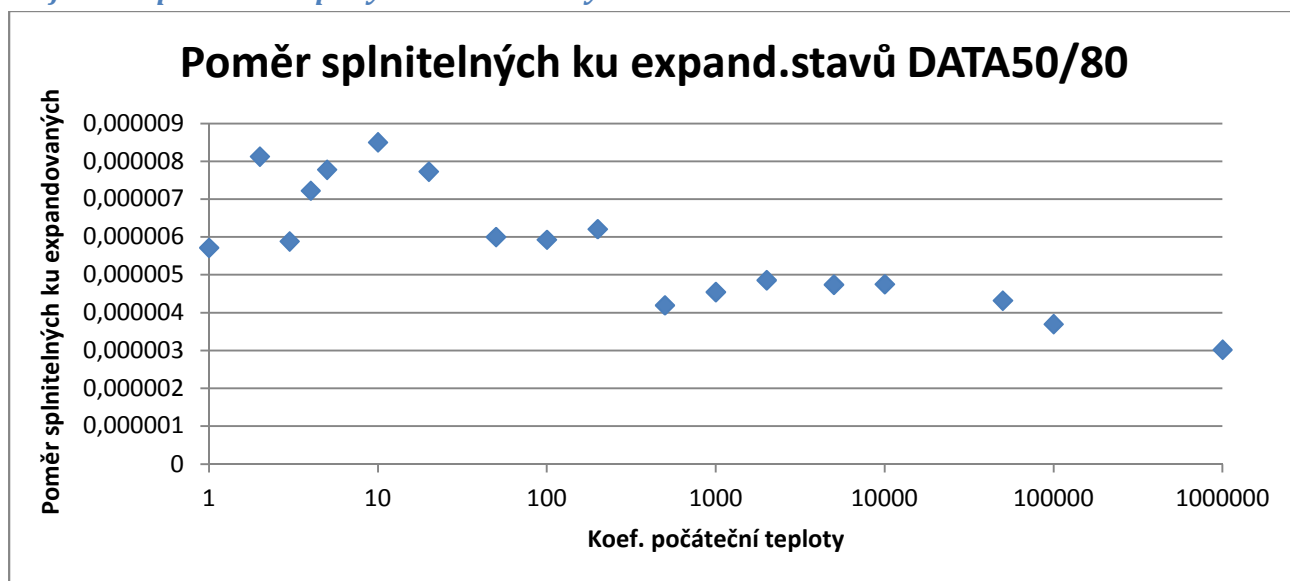
Měření pro datový soubor 50/80 a 75/40 dohromady:

Měření jsem provedl rovněž pro větší sady dat, abych viděl, jak se algoritmus chová v rámci celého spektra vstupních dat.

	DATA 50/80 (ochlaz 0.7)					DATA 75/70 (ochlaz 0.7)				
	# stavů	suma vah	čas	exp. stavů	stavy / exp.	# stavů	suma vah	čas	exp. stavů	stavy / exp.
BruteForce	20	15606	109205	20971520	9,537E-07	20	15606	109205	20971520	9,537E-07
1	8	12866	19633	1400000	5,714E-06	16	38049	24304	1950000	8,205E-06
2	13	22132	16020	1600000	8,125E-06	18	41270	28535	2250000	0,000008
3	10	16627	16853	1700000	5,882E-06	17	43783	31957	2550000	6,667E-06
4	13	21410	18022	1800000	7,222E-06	19	43708	31914	2550000	7,451E-06
5	14	21308	17857	1800000	7,778E-06	20	48132	33717	2700000	7,407E-06
10	17	27654	19845	2000000	0,0000085	20	46352	37859	3000000	6,667E-06
20	17	27909	21859	2200000	7,727E-06	19	45483	41196	3300000	5,758E-06
50	15	23754	24937	2500000	0,000006	20	46715	44628	3600000	5,556E-06
100	16	26045	26733	2700000	5,926E-06	20	49595	48568	3900000	5,128E-06
200	18	26496	28780	2900000	6,207E-06	20	49326	52136	4200000	4,762E-06
500	13	20297	30607	3100000	4,194E-06	20	49208	57705	4650000	4,301E-06
1000	15	24635	32688	3300000	4,545E-06	20	49908	61407	4950000	4,04E-06
2000	17	27083	34956	3500000	4,857E-06	20	52567	65226	5250000	3,81E-06
5000	18	28233	37727	3800000	4,737E-06	20	48257	68576	5550000	3,604E-06
10000	19	30620	39832	4000000	4,75E-06	20	57998	72255	5850000	3,419E-06
50000	19	29373	43648	4400000	4,318E-06	20	58534	82075	6600000	3,03E-06
100000	17	29041	45415	4600000	3,696E-06	20	48489	86653	6900000	2,899E-06
1000000	16	25989	52797	5300000	3,019E-06	20	51148	96452	7800000	2,564E-06

Pro datové sady 50/80 a 75/70 nemám naměřená data hrubou silou, takže zkusím alespoň zjistit, kolik se podařilo najít splnitelných formulí v závislosti na počtu procházených stavů:

Grafické znázornění závislosti počtu splnitelných formulí ku počtu expandovaných stavů na koeficientu počáteční teploty u větších datových sad:



U vstupních dat 20/91, kde je poměr počtu proměnných a počtu klauzulí přibližně 4,5, tak ke splnění většiny formulí bylo potřeba dojít až k teplotnímu koeficientu 500. U vstupních dat s výrazně menším poměrem, který byl blízko jedničky, stačilo mít teplotní koeficient nízko (což se odráží i v počtu procházených stavů a časové náročnosti).

Je vidět, že výpočetní náročnost přímo závisí na poměru počtu proměnných a počtu klauzulí.

Toto pozorování rovněž odpovídá grafu na straně 19 v PDF přiloženém k zadání úlohy (https://edux.fit.cvut.cz/courses/MI-PAA/_media/homeworks/06/ai-phys1.pdf).

Měření závislosti poměru počtu proměnných a počtu klauzulí na datech 20/91:

klauzulí	poměr	splnitelných z 20ti	suma nejlepších vah	čas ms	expandováno
91	4,55	9	5983	45556	2920000
80	4	14	10450	36976	2880000
75	3,75	18	13123	36627	2880000
70	3,5	20	13331	33977	2840000
65	3,25	20	13113	30722	2840000

Z tabulky je vidět, že s klesajícím poměrem roste úspěšnost nalezení řešení při přibližném zachování počtu expandovaných stavů.

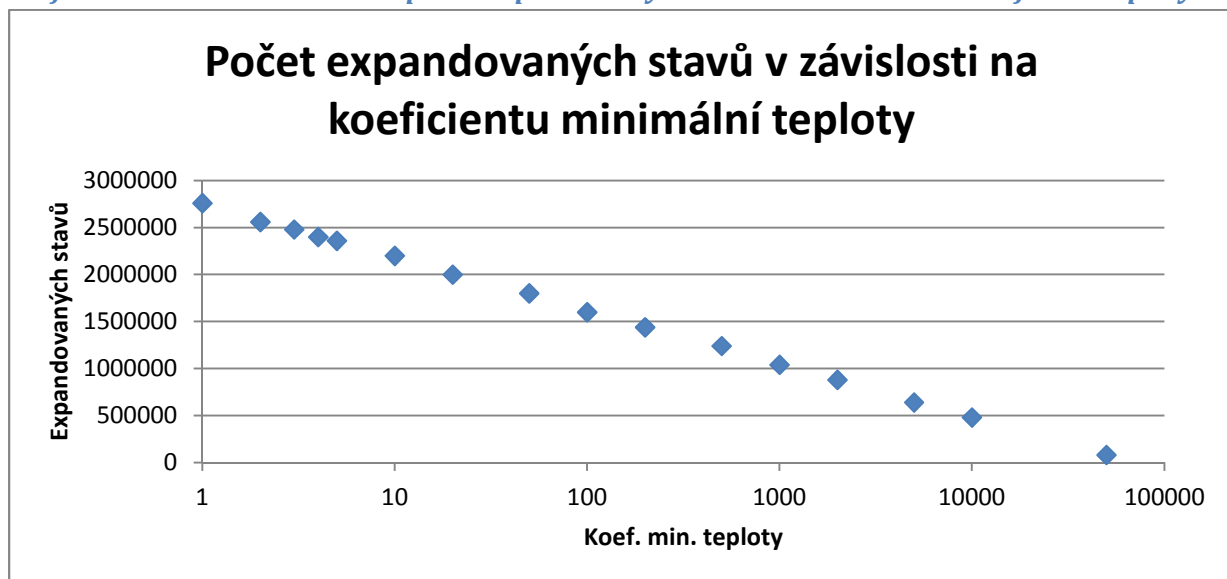
Závislost výpočetního času na minimální teplotě

Pro měření této závislosti jsem měnil minimální cenu od jedné až do situace, kdy byl výpočet časově velice náročný. Obecně řečeno - se zvyšující se minimální teplotou má algoritmus daleko menší šanci vyskočit z lokálního minima.

Naměřené hodnoty závislosti minimální teploty pro DATA 20-91:

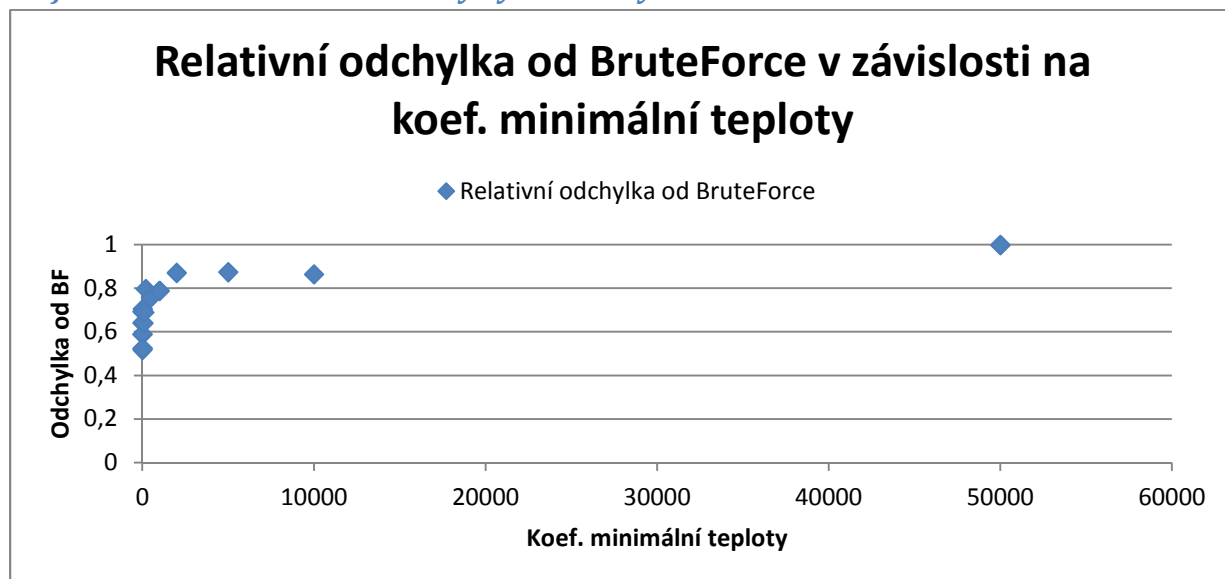
	# stavů	suma vah	čas	exp. stavů	váhy/stavy	stavy/exp.	odchylka
BruteForce	20	15606	109205	20971520	0,000744	9,54E-07	0
1	9	6402	42177	2760000	0,00232	3,26E-06	0,589773
2	9	5583	39207	2560000	0,002181	3,52E-06	0,642253
3	10	6389	38148	2480000	0,002576	4,03E-06	0,590606
4	12	7524	36873	2400000	0,003135	0,000005	0,517878
5	8	4775	36234	2360000	0,002023	3,39E-06	0,694028
10	12	7418	34351	2200000	0,003372	5,45E-06	0,52467
20	6	4595	31085	2000000	0,002298	0,000003	0,705562
50	8	5603	27488	1800000	0,003113	4,44E-06	0,640971
100	7	4836	24761	1600000	0,003023	4,38E-06	0,690119
200	4	3160	22079	1440000	0,002194	2,78E-06	0,797514
500	6	3754	19545	1240000	0,003027	4,84E-06	0,759451
1000	5	3288	15991	1040000	0,003162	4,81E-06	0,789312
2000	4	2000	13476	880000	0,002273	4,55E-06	0,871844
5000	3	1948	9788	640000	0,003044	4,69E-06	0,875176
10000	3	2102	7328	480000	0,004379	6,25E-06	0,865308
50000	0	0	1230	80000	0	0	1
100000	0	0	11	0			1
1000000	0	0	11	0			1

Grafické znázornění závislosti počtu expandovaných stavů v závislosti na koef. min. teploty:



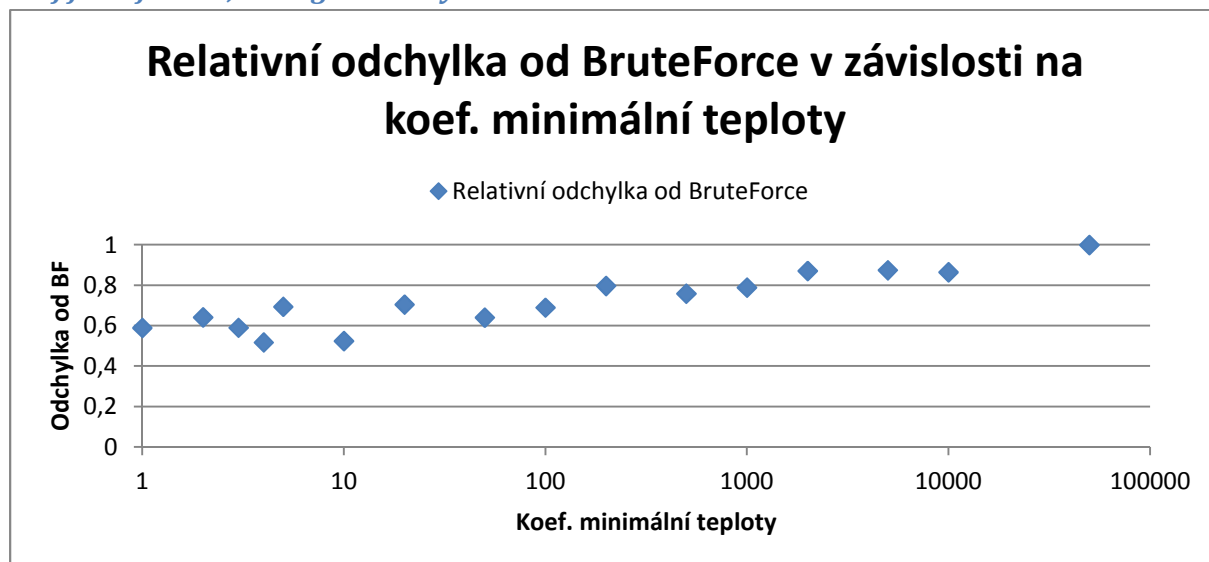
Počet stavů klesá s rostoucí minimální teplotou, což je stejné chování jako u batohu a lze to očekávat. Při rostoucí minimální teplotě klesá počet stavů, které budeme expandovat, protože klesá počet zchlazovacích cyklů.

Grafické znázornění relativní odchylky naměřených hodnot od Brute Force:



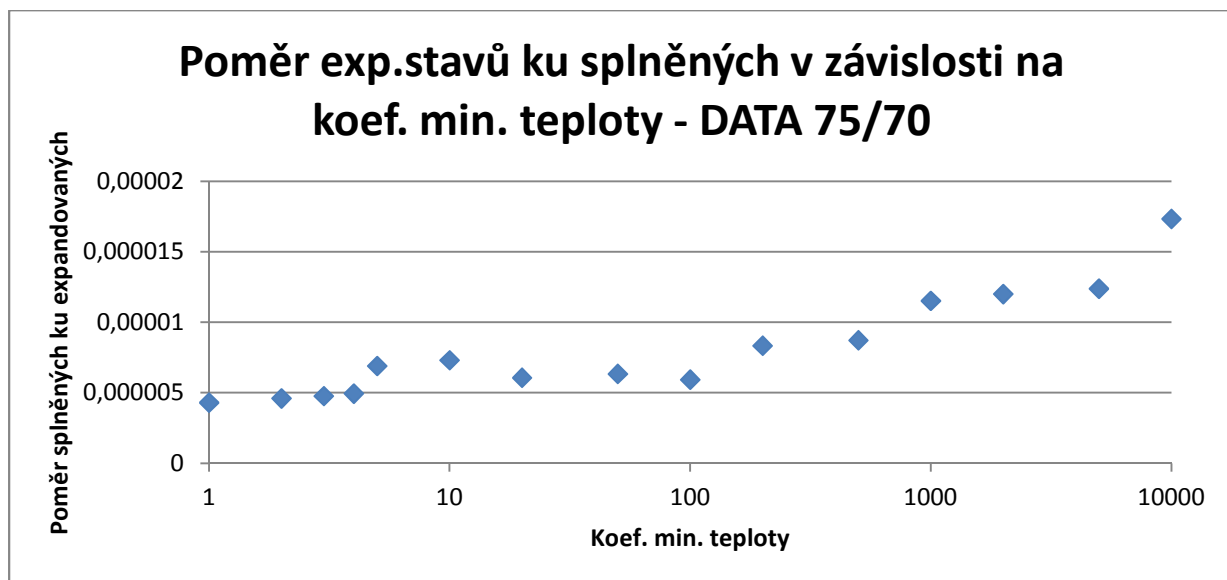
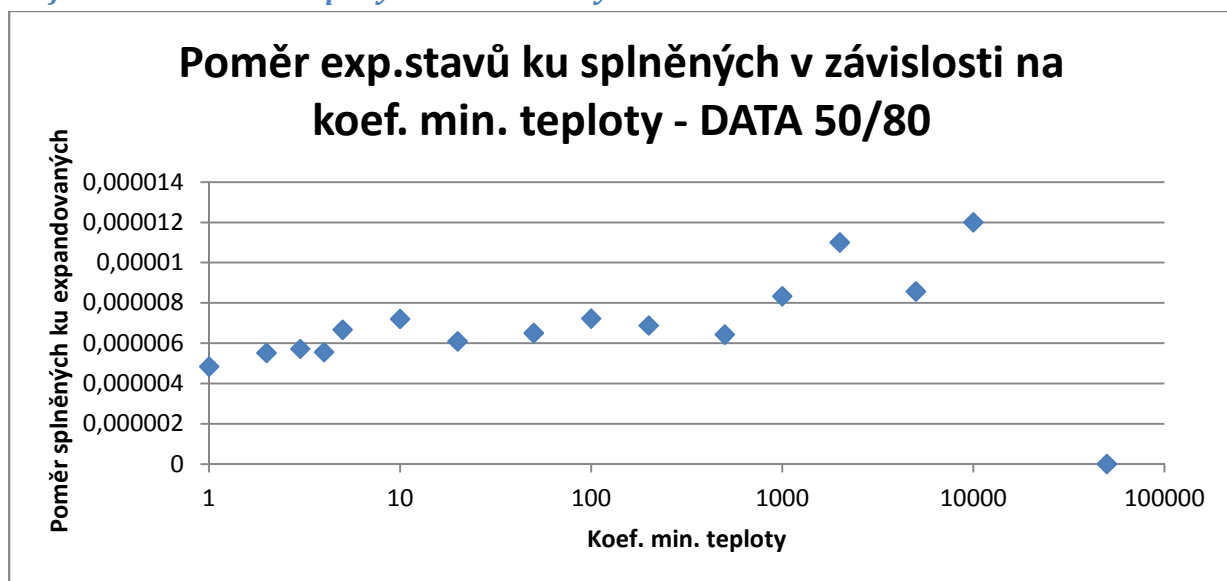
Nejmenší odchylka je zhruba někde u koeficientu minimální teploty velikosti 4. S klesajícím počtem expandovaných stavů také dochází ke snížení možnosti, že najdeme optimální řešení a relativní odchylka roste.

Graf ještě jednou, ale logaritmicky:



Na grafu s logaritmickým měřítkem je lépe vidět trend růstu odchylky. Pro další měření jsem tedy nechal nastavenou výchozí hodnotu koeficientu minimální teploty na hodnotě 1.

Grafické znázornění závislosti počtu splnitelných formulí ku počtu expandovaných stavů na koeficientu minimální teploty u větších datových sad:



Tento poměr nám akorát ukazuje, že když vzrůstá koeficient minimální teploty, tak se snižuje počet procházených stavů. Při hodnotě 1000 se nám ale pořád dařilo najít polovinu správných řešení při expanzi třetiny stavů, proto je výsledný poměr vyšší jak u teplot nižších.

Minimální teplotu jsem ponechal na hodnotě 1 vzhledem k nejmenší relativní chybě, která byla kolem hodnoty 4.

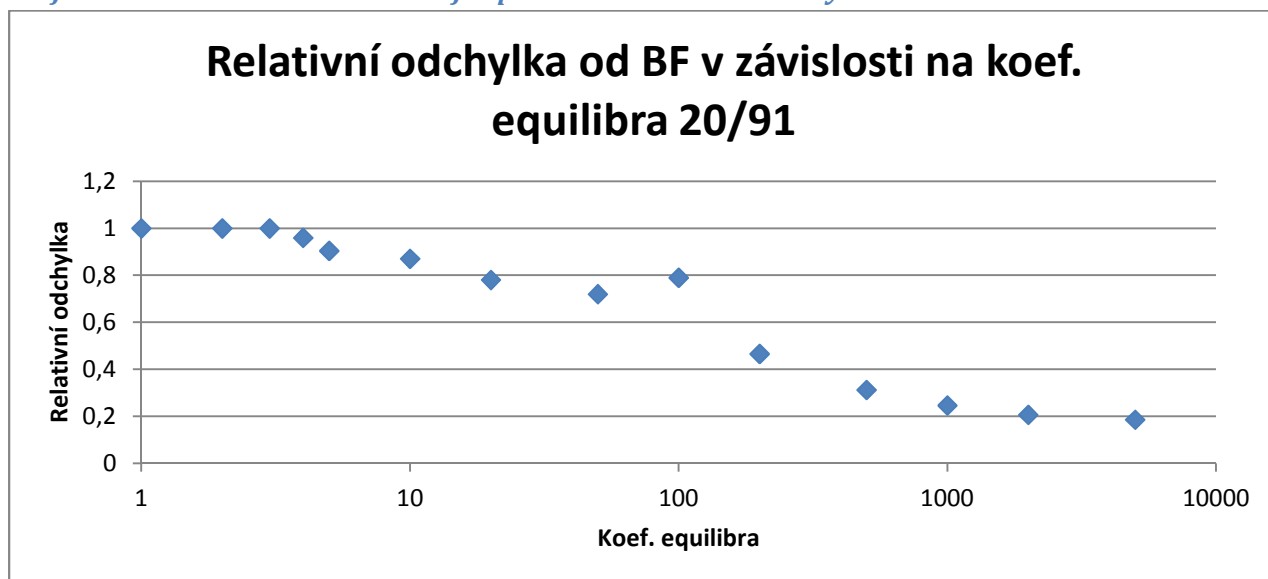
Závislost změny koeficientu equilibra (počtu kroků)

Koeficient equilibra jsem měnil od jedničky až po 5000, kde již byl výpočet neúnosný. Počet expandovaných stavů rostl úměrně se zvoleným počtem kroků.

Naměřená data pro DATA 20/91:

	# stavů	suma vah	čas	exp. stavů	váhy/stavy	stavy/exp.	odchylka
BruteForce	20	15606	109205	20971520	0,000744	9,54E-07	0
1	0	0	513	27600	0	0	1
2	0	0	863	55200	0	0	1
3	0	0	1284	82800	0	0	1
4	1	630	1714	110400	0,005707	9,06E-06	0,959631
5	2	1497	1969	138000	0,010848	1,45E-05	0,904075
10	3	2018	3170	276000	0,007312	1,09E-05	0,870691
20	6	3422	6255	552000	0,006199	1,09E-05	0,780725
50	6	4370	15322	1380000	0,003167	4,35E-06	0,719979
100	5	3283	31501	2760000	0,001189	1,81E-06	0,789632
200	13	8338	62877	5520000	0,001511	2,36E-06	0,465718
500	16	10733	153350	13800000	0,000778	1,16E-06	0,312252
1000	18	11767	306888	27600000	0,000426	6,52E-07	0,245995
2000	19	12388	613860	55200000	0,000224	3,44E-07	0,206203
5000	20	12708	1529166	138000000	9,21E-05	1,45E-07	0,185698
10000							1
50000							1
100000							1
1000000							1

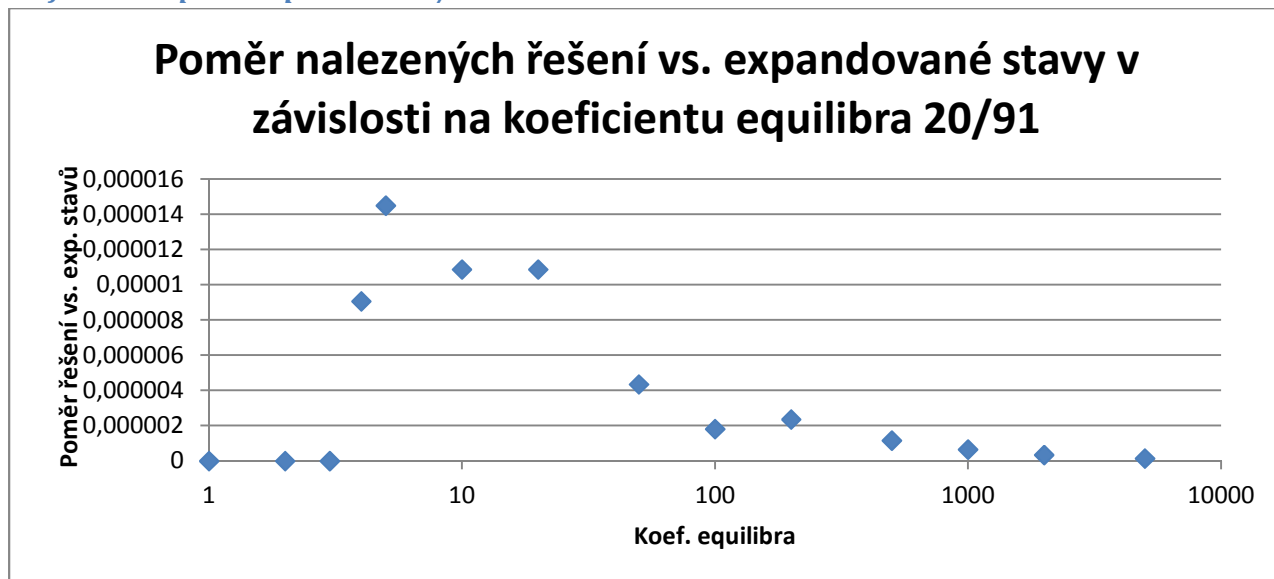
Grafické znázornění závislosti koef. equilibra na relativní odchylce od BruteForce:



Pokud zvyšujeme koeficient equilibra, roste také počet procházených stavů. To má za důsledek ten, že se zvyšuje šance na nalezení řešení a snižuje se tím odchylka. Zároveň však roste počet

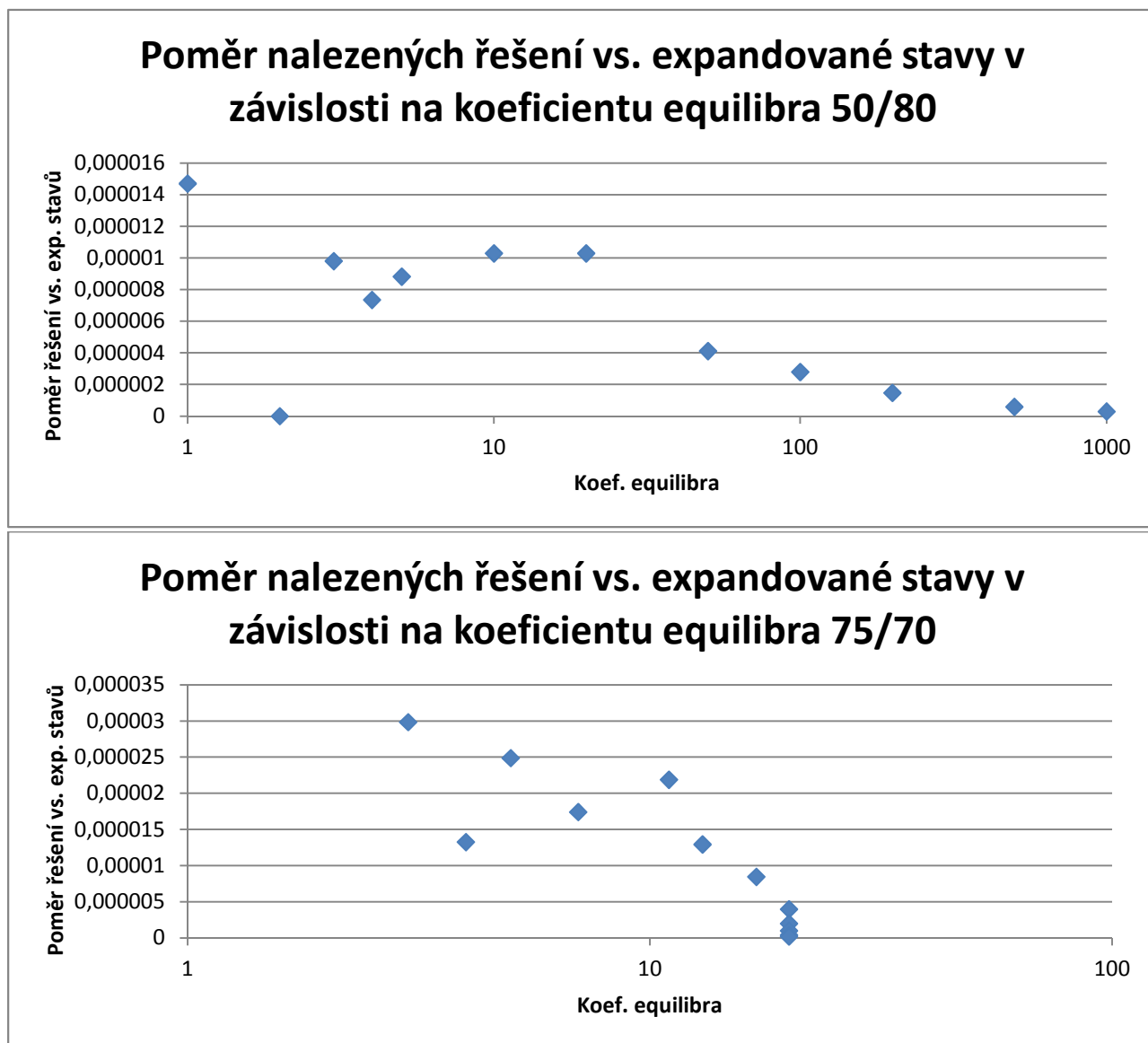
procházených stavů! Bylo by dobré do závislosti zakomponovat i časovou složku, resp. počet procházených stavů v závislosti na počtu nalezených řešení.

Grafické znázornění poměru počtu procházených stavů a nalezených řešení v závislosti na koeficientu equilibra pro data 20/91:



Z grafu je krásně vidět, že optimální nastavení equilibra je kolem hodnoty 5-10. S rostoucím equilibrem roste počet vnitřních kroků výpočtu a roste počet procházených stavů, ale již tolik neroste počet nalezených řešení.

Grafické znázornění poměru počtu procházených stavů a nalezených řešení v závislosti na koeficientu equilibra pro větší datové sady:



I pro větší datové sady je vidět, že je optimum vnitřních cyklů někde kolem hodnoty 5-10. Pokud bych měl více času na měření, bylo by asi vhodné změnit výchozí hodnotu equilibra ze stávajících 100 na hodnotu 10 a přeměřit stávající hodnoty pro počáteční a minimální teplotu.

Závislost výpočetního času na koeficientu zchlazování

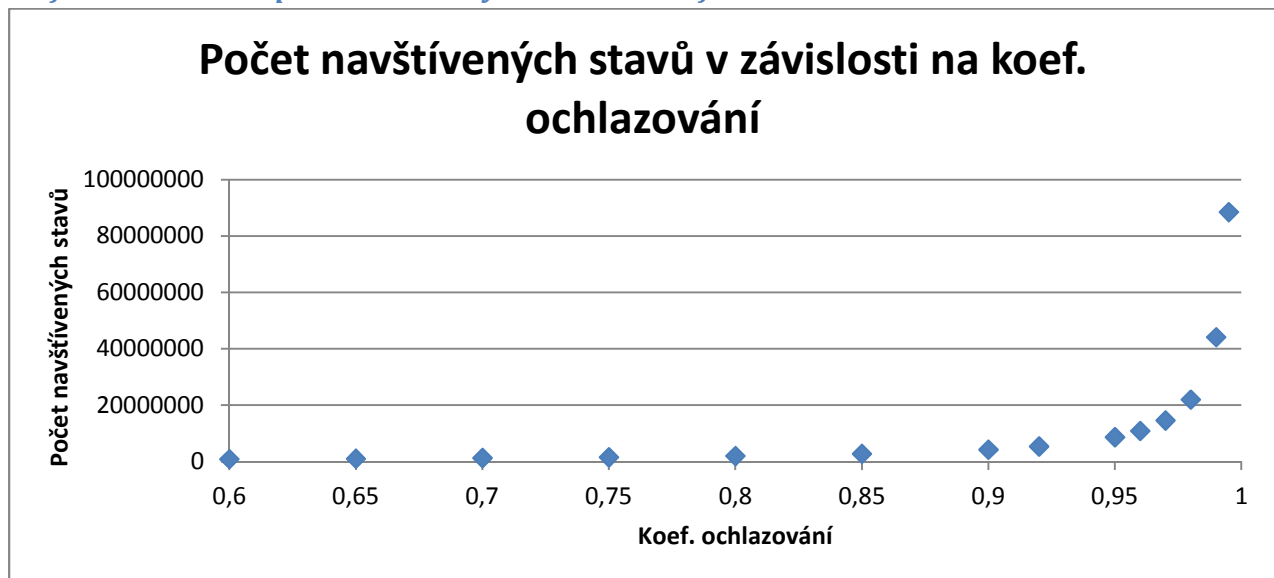
Koeficient zchlazování jsem nastavoval dle přednášek v rozsahu 0,5 až 0,995 (okolí bodu 0,85).

Naměřené hodnoty pro datovou sadu 20/81:

Pro tuto datovou sadu mám naměřený i algoritmus hrubou silou, takže bylo s čím srovnávat.

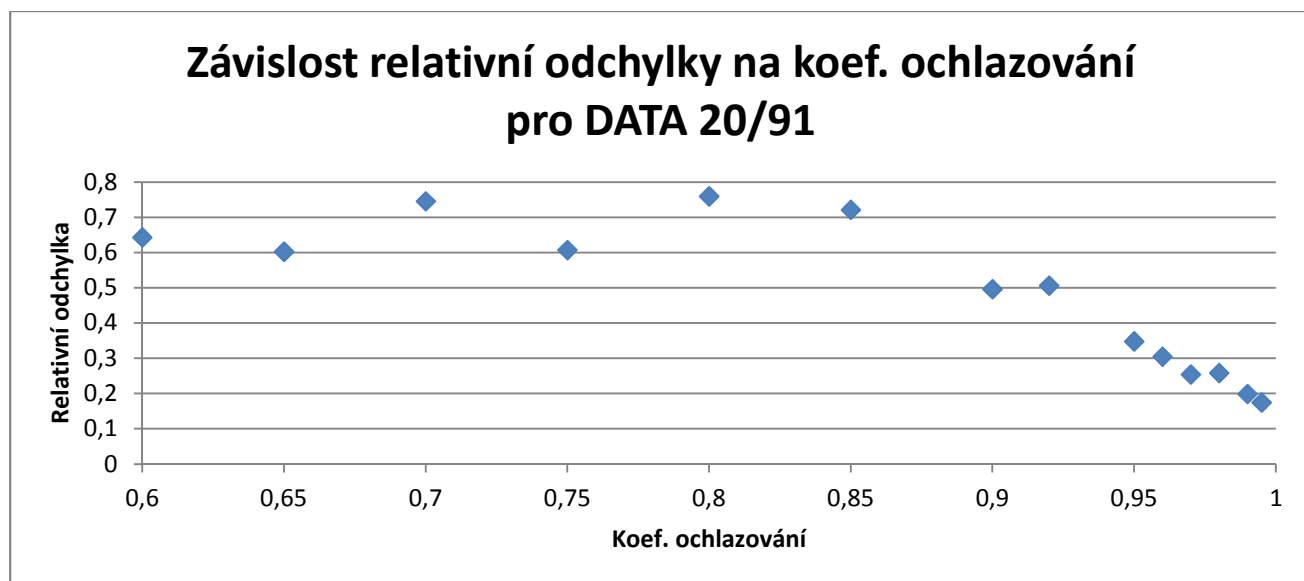
	# stavů	suma vah	čas	exp. stavů	váhy/stavy	stavy/exp.	odchylka
BruteForce	20	15606	109205	20971520	0,000744	9,54E-07	0
0,6	8	5572	13617	880000	0,006332	9,09E-06	0,642958
0,65	9	6209	12788	1040000	0,00597	8,65E-06	0,60214
0,7	6	3974	14537	1280000	0,003105	4,69E-06	0,745354
0,75	8	6132	17613	1560000	0,003931	5,13E-06	0,607074
0,8	7	3753	22357	2000000	0,001877	3,5E-06	0,759516
0,85	7	4353	30801	2760000	0,001577	2,54E-06	0,721069
0,9	12	7866	47453	4240000	0,001855	2,83E-06	0,495963
0,92	12	7710	60410	5360000	0,001438	2,24E-06	0,505959
0,95	15	10180	96677	8680000	0,001173	1,73E-06	0,347687
0,96	16	10852	121478	10880000	0,000997	1,47E-06	0,304626
0,97	17	11642	163836	14600000	0,000797	1,16E-06	0,254005
0,98	17	11578	246966	22000000	0,000526	7,73E-07	0,258106
0,99	19	12508	492726	44160000	0,000283	4,3E-07	0,198513
0,995	20	12880	990567	88520000	0,000146	2,26E-07	0,174676

Grafické znázornění počtu navštívených stavů na koeficientu zchlazování:



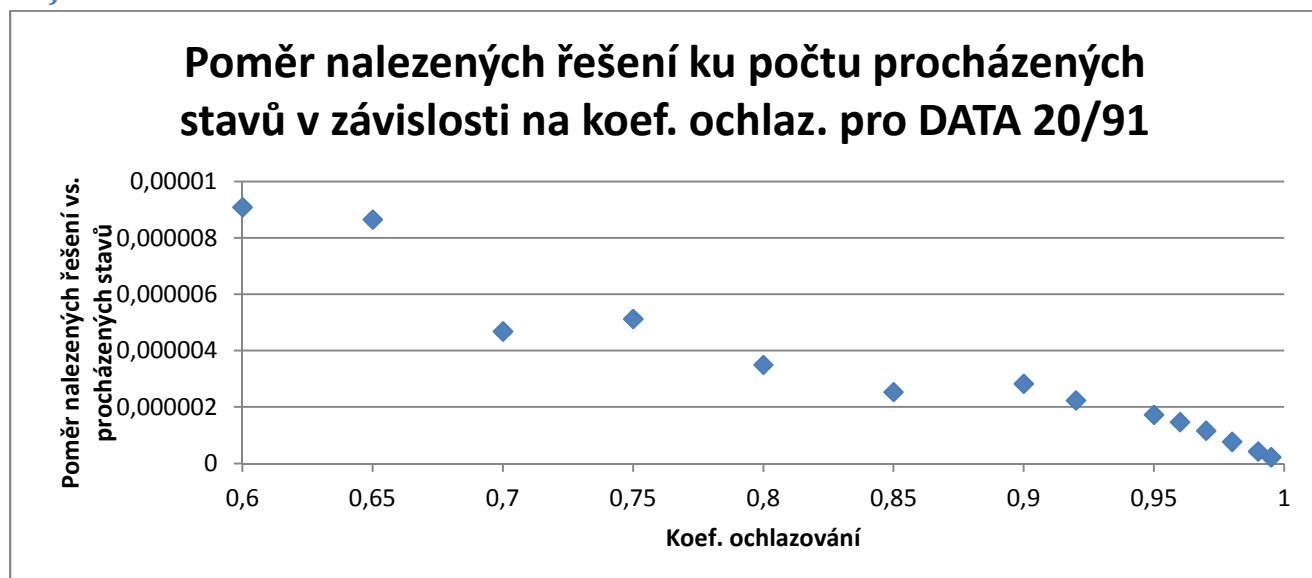
Z grafu je vidět, že s rostoucím koeficientem zchlazování roste počet prozkoumaných stavů a tudíž i časová náročnost algoritmu. Je to stejné chování jako u batohu a dalo se to předpokládat.

Grafické znázornění relativní odchylky od BruteForce v závislosti na koeficientu zchlazování:



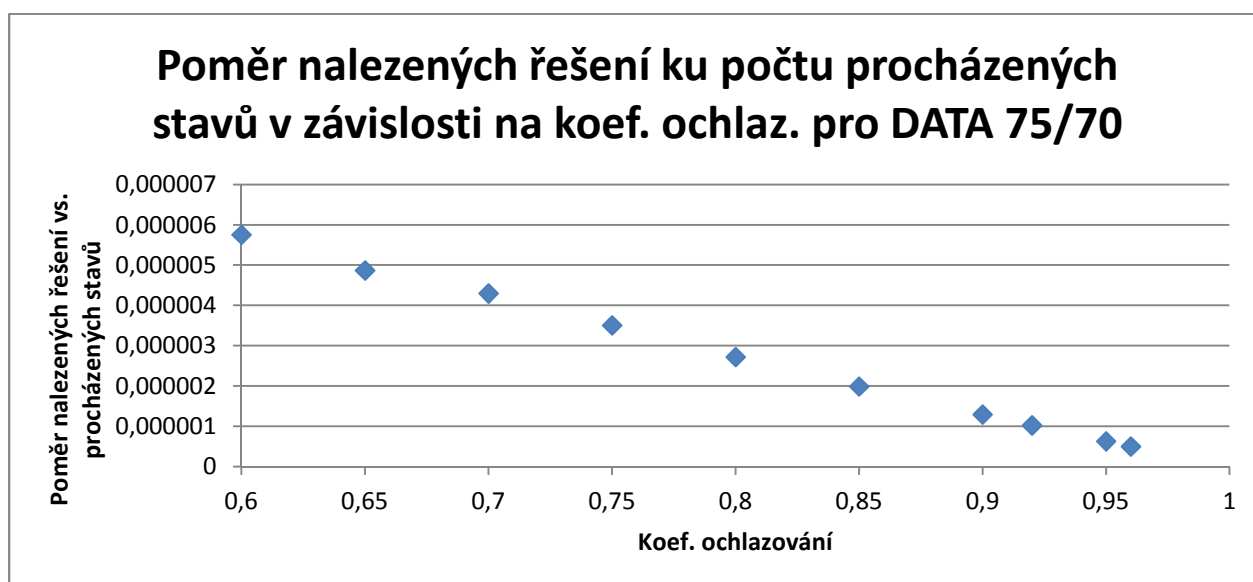
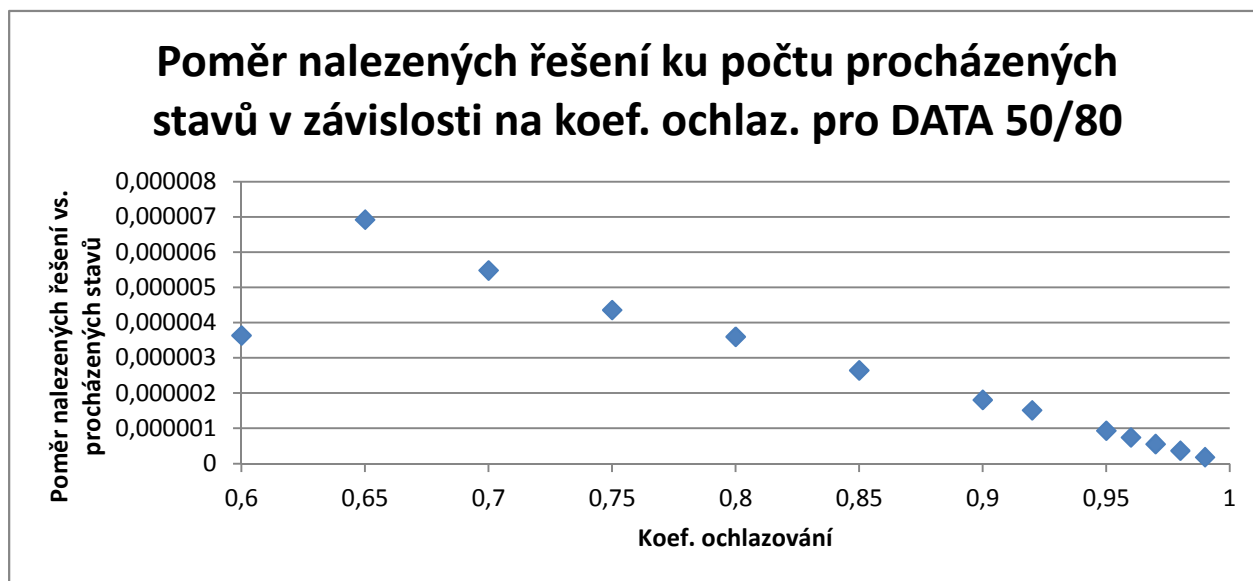
Relativní odchylka klesá s rostoucím koeficientem zchlazování, protože rostoucím počtem procházených stavů. Rovněž ale roste časová náročnost, takže by bylo dobré zapracovat do grafu i počet procházených stavů, viz. níže.

Grafické znázornění poměru nalezených řešení ku počtu procházených stavů v závislosti na koef. ochl.



Graf nám vlastně ukazuje, jak velká část procházených stavů byla splnitelná. Je vidět, že nejlepší poměr je u 0,65 což znamená, že odchylka je sice velká, ale projde se relativně málo stavů, takže určité řešení je známo brzo. Jak jde koeficient k jedničce roste počet procházených stavů natolik, že se to již nevyplatí procházet, protože počet nalezených stavů bude jenom o trochu větší a hlavně celkový počet stavů, které musíme projít začne převyšovat i algoritmus hrubé síly.

Grafické znázornění poměru nalezených řešení ku počtu procházených stavů v závislosti na koef. ochl. Pro větší datové sady:



Na grafech je vidět podobný poměr jako u vstupních dat 20/91.

Pokud změním koeficient equilibra na hodnotu 10 (což je hodnota, která mi vyšla jako optimální u měření výše), dostávám se na ideální hodnotu koef. ochlazování někde k hodnotě 0,8. Vzhledem k tomu, že jsem chtěl všechny parametry zachovat pro všechna měření a měnit pouze jeden, hodnoty (a grafy) jsou zobrazeny pro hodnotu equilibra 100, což není optimální hodnota.

Závěr

Problém SAT byl podobný problému batohu a šlo použít i stejně napsaný algoritmus. Bylo však potřeba vyřešit dva zásadní problémy:

- a) Jak porovnat dva stavy a určit který je lepší, abychom věděli, z kterého pokračovat dále ve stavovém prostoru.
- b) Jak porovnat dva stavy a určit který je lepší při hledání ideálních parametru ochlazování, abychom věděli, pro který parametr je řešení optimálnější (nutno zahrnout počet stavů které procházíme a počet řešení, které jsme našli).

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 roste počet procházených stavů a zároveň klesá odchylka od optimálního řešení.

Zajímavé bylo sledovat závislost poměru počtu proměnných a počtu klauzulí, což bylo viditelné hlavně u měření počáteční teploty. Celkově jsem se snažil nastavovat parametry ochlazování tak, aby fungovali jak pro poměr počtu proměnných a klauzulí 4.3, tak i pro poměr nižší, kde by řešení mělo být jednodušší.