Řešení problému vážené splnitelnosti booleovské formule pokročilou iterativní metodou

1 SPECIFIKACE ÚLOHY

Cílem této úlohy bylo použít vybranou pokročilou iterativní metodou pro řešení problému vážené splnitelnosti boolovské formule v konjunktivní normální formě (SAT resp. 3SAT problém). Předmětem zprávy je popis zvoleného algoritmu, měření pro různá nastavení konfiguračních proměnných a experimenty s nimi. Více informací viz odpovídající stránka na Eduxu.

2 ROZBOR MOŽNÝCH VARIANT ŘEŠENÍ

Bylo třeba se rozhodnout pro jeden z těchto algoritmů:

- Simulované ochlazování
- Genetické algoritmy
- Tabu search

Vybral jsem simulované ochlazování, což bylo určené už z úkolu č. 4, kde jsem se stejným algoritmem experimentoval na problému batohu.

Pro zjednodušení implementace jsem problém omezil na tzv. 3SAT, který je však z pohledu teorie NP stejně těžký. Jde o variantu problému, kdy každá klauzule obsahuje právě 3 literály.

Dále jsem se z důvodů (které jsou popisovány u konkrétních měření) rozhodl výpočet zrelaxovat uživatelsky definovaným koeficientem (0 > k <= 1.0), určující minimální počet splněných klauzulí. Za tuto cenu jsem získal algoritmus, který dokáže alespoň částečně vyřešit i velké instance (při snížení koeficientu), ale při řešení dostatečně malých instancí stále přináší úplný výsledek (tedy splnění všech klauzulí).

3 POPIS POSTUPU ŘEŠENÍ A ALGORITMU

Algoritmus Simulované ochlazování je iterativní heuristika, což znamená, že se snaží z určitého počátečního řešení postupně dopracovat k řešení co nejlepšímu. V případě SAT je získání výchozího řešení problém, protože neexistuje triviální řešení a generování náhodného řešení je u velkých instancí neúnosně zdlouhavé. V mém případě program nejdříve ověří, zda k zadání neexistuje v přidruženém **„.sol“** souboru výchozí řešení. V případě že ne, pokusí se jej vygenerovat. Rychlost generování i samotného výpočtu je značně ovlivněna nastavením relaxačního koeficientu.

Základní parametry algoritmu jsou:

- Equilibrium (rovnovážná poloha)
- Počáteční teplota
- Teplota tuhnutí
- Rychlost chlazení
- Koeficient relaxace (jak velká část klauzulí musí být splněna, aby bylo ohodnocení považováno za řešení)

Algoritmus pracuje tak dlouho, dokud z počáteční teploty nedosáhne teploty tuhnutí, ke které se dostává postupně násobením aktuální teploty koeficientem rychlosti chlazení, což musí být číslo menší než 1, aby docházelo ke zmenšování teploty. Toto zajišťuje vnější smyčka algoritmu.

Uvnitř je další smyčka, jejíž počet opakování je dán hodnotou Equilibria. Ve vnitřní smyčce algoritmus nalezne souseda (který je řešením) inverzí jednoho náhodného bitu aktuálního stavu. Zde narážíme na problém, kdy mezi sousedy řešení vůbec nemusí existovat a opět se uplatňuje použití relaxačního koeficientu.

Následně dojde k ověření, zda je tento náhodný soused lepší nebo horší. Pokud je lepší, vždy se přijme. Pokud je horší, přijme se s určitou pravděpodobností, která je dána aktuální teplotou a vzdáleností od aktuálně nejlepšího řešení. Čím horší řešení je, tím menší pravděpodobnost, že bude přijato. Ale čím vyšší je teplota, tím je větší šance, že i poměrně o hodně horší řešení bude přijato. Smysl tohoto je vymanit se z lokálních minim. Pravděpodobnost přijetí horšího řešení je počítána na základě vztahu:

$$e^{-rozdíl\_ceny/aktuální\_teplota}$$

Je nutno dodat, že přijetím řešením není myšleno, že je považováno za aktuálně nejlepší, i když nám ve skutečnosti nejlepší řešení zhoršuje. Pouze je přijato jako aktuální stav do další iterace algoritmu.

Po ukončení vnitřní smyčky je snížena teplota, a pokud nedošlo k teplotě tuhnutí, opakujeme vnitřní smyčku znovu s novou teplotou.

3.1 Důležitá modifikace oproti problému batohu

Je možné definovat již zmíněný koeficient relaxace, kterým lze řídit minimální počet splněných klauzulí. V závislosti na tom je při určování lepšího řešení v průběhu algoritmu v první řade zohledňováno, zda nové řešení splňuje více klauzulí – pokud ano, je vždy přijato. Navíc je penalizována váha těch řešení, které nejsou úplné a to násobkem jejich procentuálního plnění (např. 74% => 0.74). Naopak řešení, která jsou úplná, zvýhodňujeme násobkem 2. Tímto způsobem tlačíme algoritmus ke splnění co nejvíce klauzulí.

4 NAMĚŘENÉ VÝSLEDKY

4.1 HW / SW KONFIGURACE TESTOVACÍHO SYSTÉMU

- CPU Intel® Core™2 Duo; 2.26 GHz, 2.27 GHz
- 4 GB RAM
- OS Windows 8.1, 64-bit
- Java 7

4.2 ZPŮSOB MĚŘENÍ

Čas byl měřen pomocí třídy **ThreadMXBean** a byly použity jak instance ze serveru http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html, tak instance vygenerované SAT řešičem (zde včetně iniciálních řešení).

Relativní chyby z důvodu absence optimálních řešení nebylo možné počítat. Provedl jsem však alespoň několik měření, kde sleduji vývoj výsledné ceny (váhy) řešení.

Pro omezení vlivu náhody algoritmu je čas měřen až po stanovení výchozí teploty. Navíc v případě, kdy nemáme výchozí řešení, ani získání náhodného řešení není do výsledného času započítáváno. Je tedy měřen čistý běh iterativní metody po získání všech vstupních parametrů.

4.3 STANOVENÍ PARAMETRŮ

Při stanovení parametrů algoritmu jsem v první řadě vycházel z předchozí úlohy, kde jsem se pokusil na základě měření co nejpřesněji a co nejobecněji určit hodnoty těchto parametrů, které jsem stanovil takto:

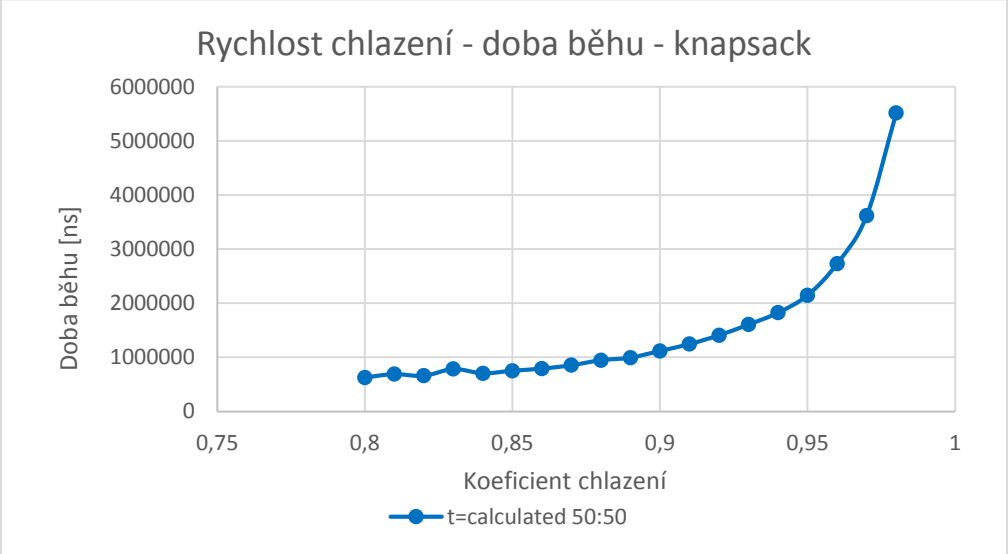
- Equilibrium (rovnovážná poloha): Velikost instance
- Počáteční teplota: Vypočtena dle pravidla 50:50
- Teplota tuhnutí: 1-2 stupně
- Rychlost chlazení: koeficient 0,95 – 0,98

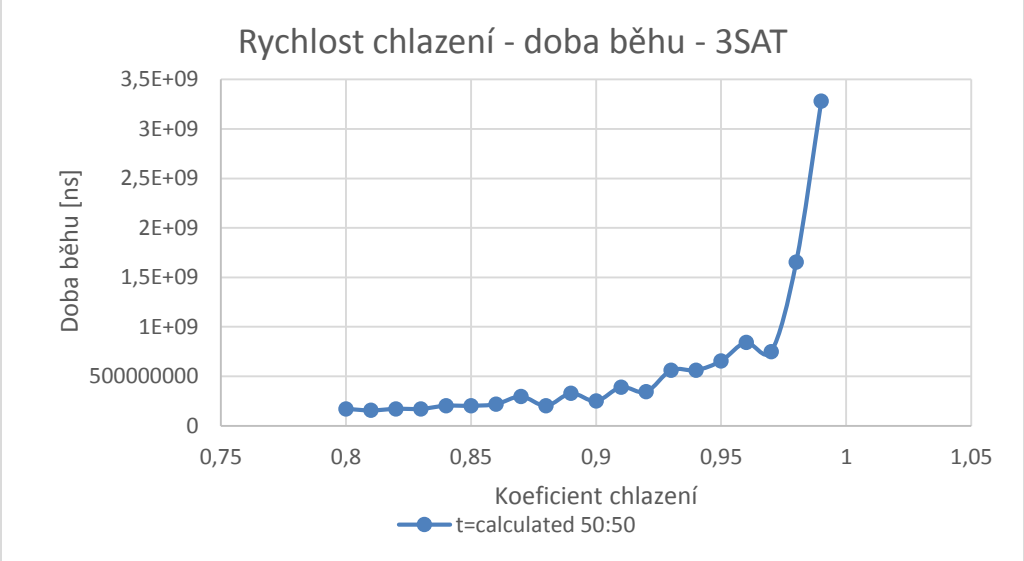
Tyto hodnoty (určené na problému batohu o velikosti instance **40**) jsem ověřil na instanci 3SAT problému **50vars150clauses** se známým řešením (přiloženo). Zjistil jsem, že až na drobné změny dané prvkem náhody v algoritmu je křivka časové složitosti pro problém 3SAT obdobná. Určené parametry jsou tedy vhodné i pro tento problém. Pouze jsem upravil koeficient equilibria.

Pro srovnání přikládám měření jak pro problém batohu, tak pro 3SAT.

4.3.1 Rychlost chlazení

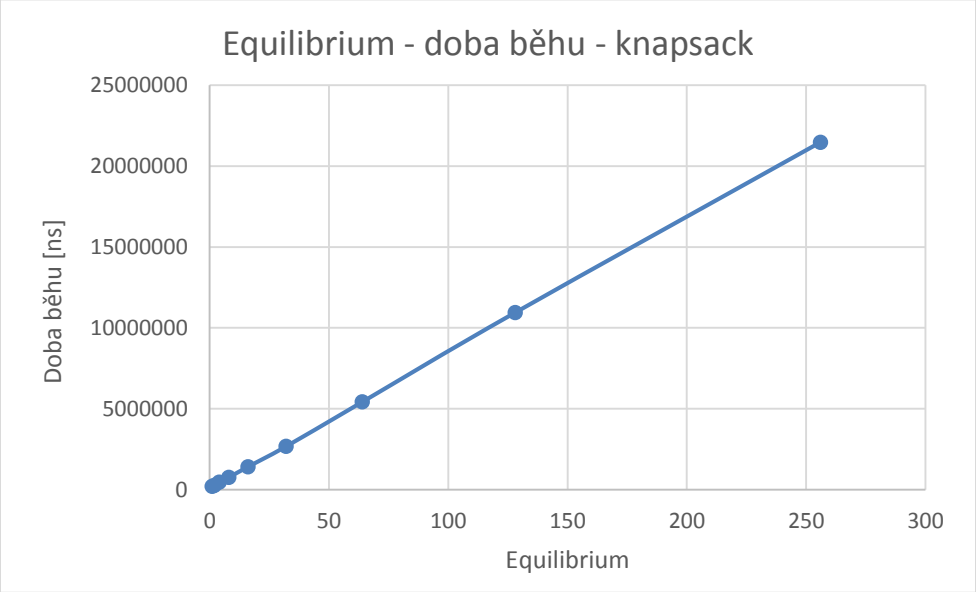
parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	calculated
Teplota tuhnutí	5
Rychlost chlazení	**0,8 – 0,99**
Velikost instance	50
Koeficient relaxace	0,95

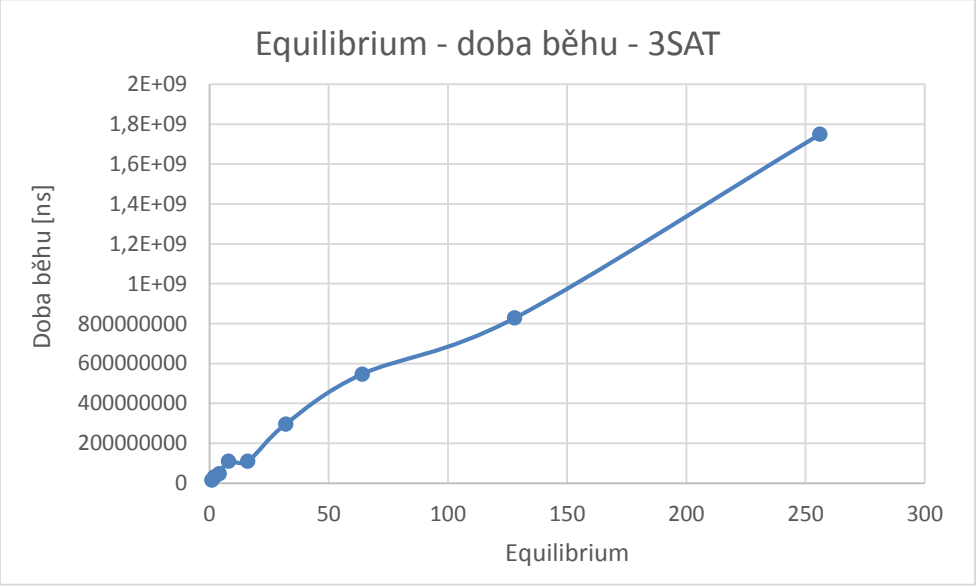




4.3.2 Equilibrium

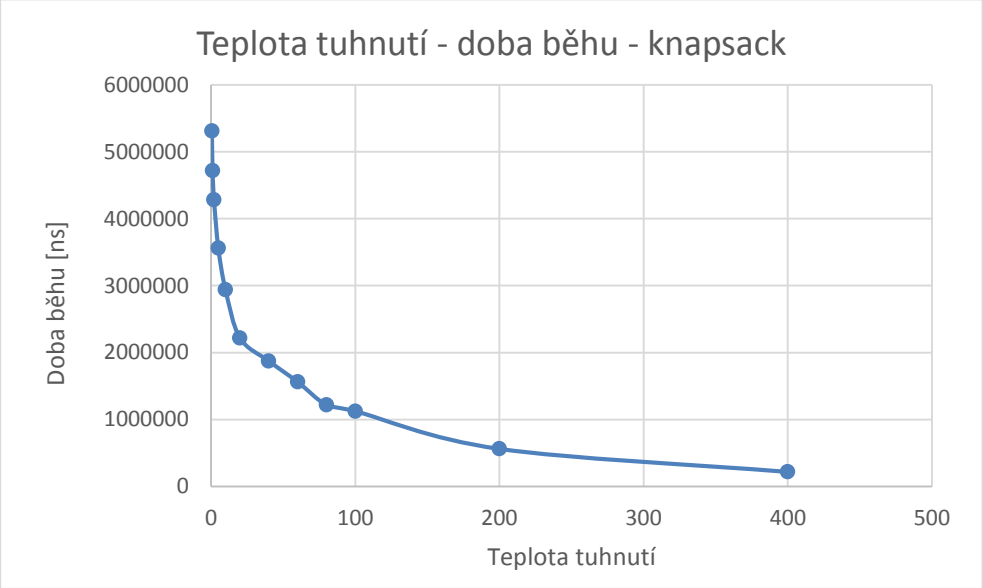
parametr	hodnota
Equilibrium	**1 – 256**
Počáteční teplota	calculated
Teplota tuhnutí	5
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	0,95

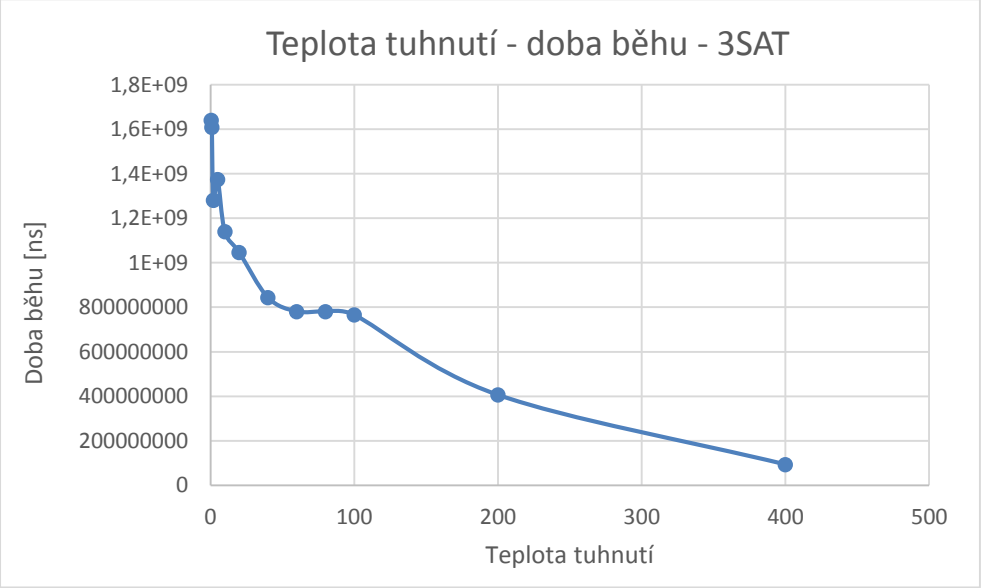




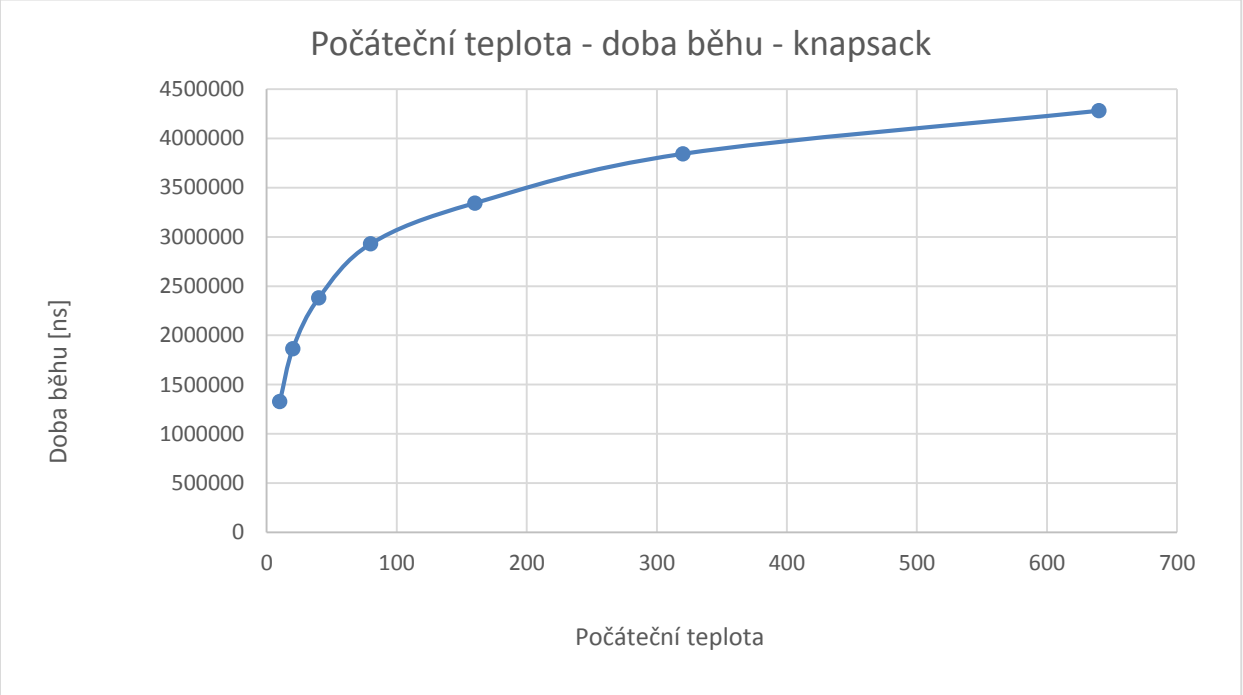
4.3.3 Teplota tuhnutí

parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	calculated
Teplota tuhnutí	**0,5 – 400**
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	0,95



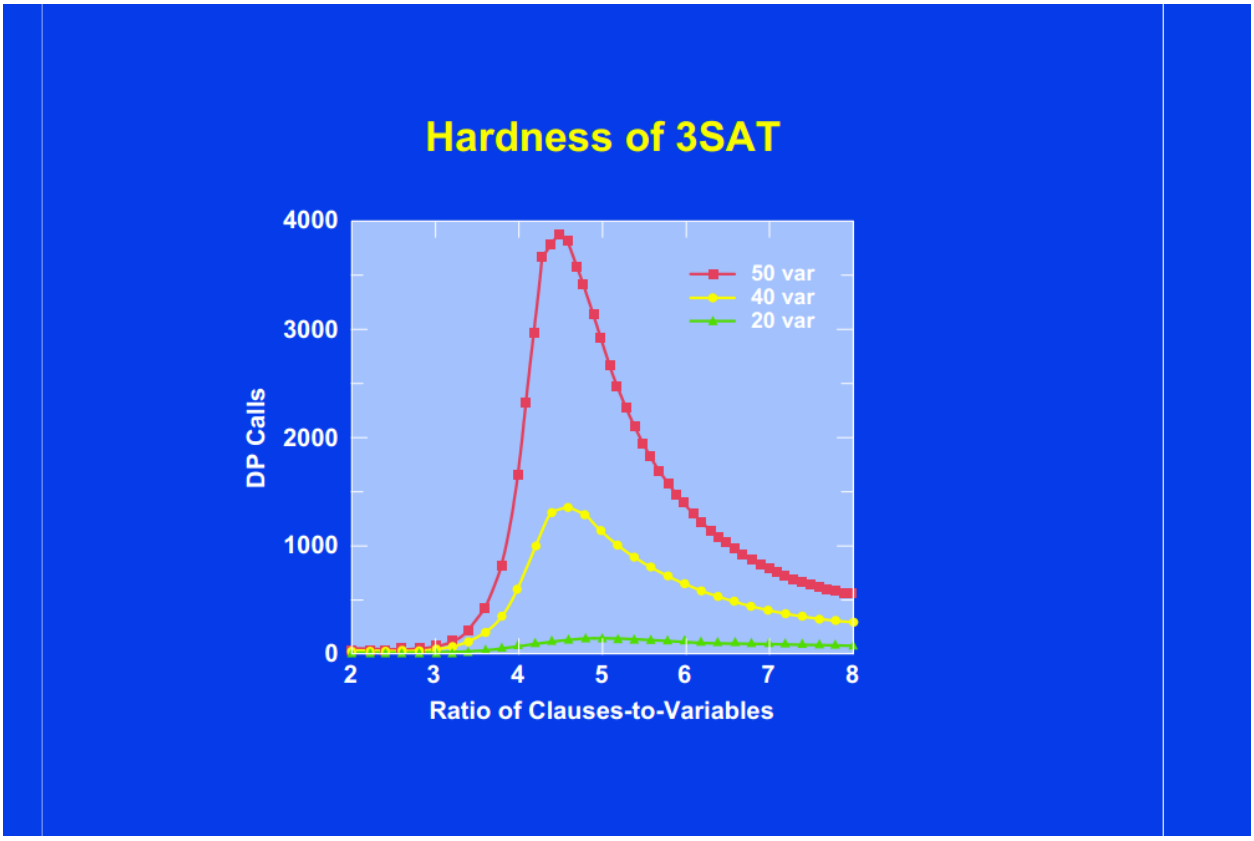


4.3.4 Počáteční teplota



4.4 TEST NA ŠÍŘI INSTANCÍ

Pro účely tohoto testu jsem použil DIMACS instance o velikosti 20 – 250 proměnných. Tyto instance mají poměr počtu klauzulí k počtu proměnných přibližně 4,3, což bylo vyhodnoceno jako nejtěžší instance.



(Zdroj: https://edux.fit.cvut.cz/courses/MI-PAA/_media/homeworks/06/ai-phys1.pdf).

4.4.1 DIMACS: 20 proměnných

Program našel ohodnocení pokrývající všechny klauzule (počet splněných klauzulí vidíme na předposledním řádku) s váhou **499**, při relaxačním koeficientu 0,99 za přibližně 1 vteřinu.

```
Solution file uf20-01.sol is missing.
Generating random solution
Random solution generated, it took (46875000 ns)
Calculating initial temperature
Initial temperature calculated
= Starting calculation
----- SimulatedAnnealing3SATSolver parameters -----
----- RELAXATION_COEFICIENT=0.99 (at least 90/91 clauses must be satisfied)
----- EQUILIBRIUM=60
----- FROZEN_TEMPERATURE=2.0
----- INITIAL_TEMPERATURE=10.0
----- COOLING_DELTA=0.98
-----------------------------------------------------
= Calculation finished
SATSolution [satisfiedClauses=91, solutionWeight=499,
instance=SATInstance [variablesCount=20, clausesCount=91, clauses=[4, -18, 19]
Solution: [1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1]
Program found evaluation to cover 91/91 clauses
Calculation took (265625000 ns)
```

4.4.2 DIMACS: 50 proměnných

Při této velikosti instance už bychom pro koeficienty splnitelnosti blížící se 1,0 čekali příliš dlouho na vygenerování náhodného řešení i nalezení dalších řešení mezi sousedy. Je proto třeba koeficient snížit. Při tomto běhu bylo nalezeno pokrytí 214 klauzulí z 218.

```
Solution file uf50-01.sol is missing.
Generating random solution
Random solution generated, it took (92421875000 ns)
Calculating initial temperature
Initial temperature calculated
= Starting calculation
----- SimulatedAnnealing3SATSolver parameters -----
----- RELAXATION_COEFICIENT=0.98 (at least 213/218 clauses must be satisfied)
----- EQUILIBRIUM=150
----- FROZEN_TEMPERATURE=2.0
----- INITIAL_TEMPERATURE=80.0
----- COOLING_DELTA=0.98
-------------------------------------------------
= Calculation finished
SATSolution [satisfiedClauses=214, solutionWeight=1993,
instance=SATInstance [variablesCount=50, clausesCount=218, clauses=[-3, 36, 7]
Solution: [0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1
Program found evaluation to cover 214/218 clauses
Calculation took (1343750000 ns)
```

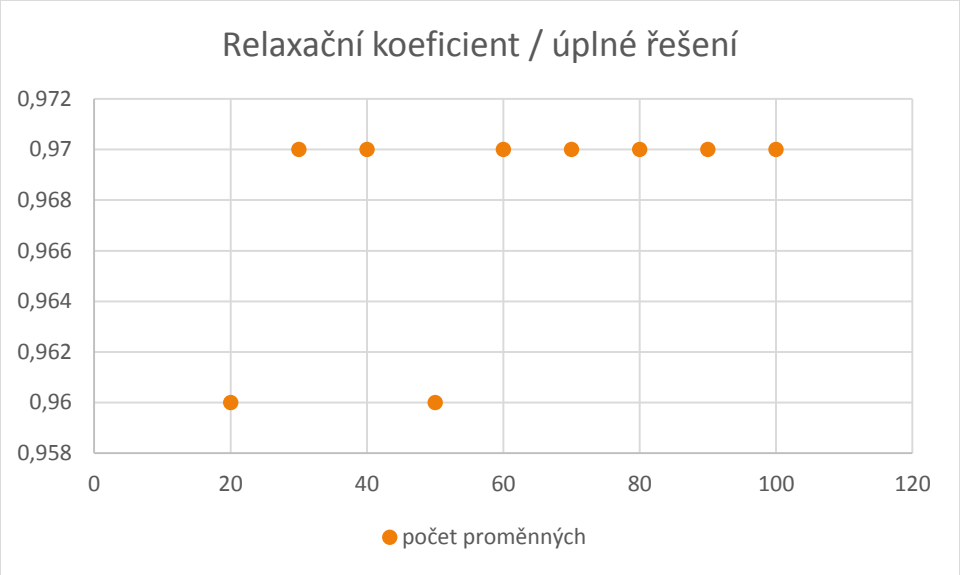
4.4.3 DIMACS: 100 a více proměnných

- 100 proměnných: koeficient: 0,95 běh: 3734375000 ns splněné klauzule: 409/430
- 150 proměnných: koeficient: 0,94 běh: 8625000000 ns splněné klauzule: 607/645
- 200 proměnných: koeficient: 0,93 běh: 15437500000 ns splněné klauzule: 800/860
- 250 proměnných: koeficient: 0,92 běh: 22546875000 ns splněné klauzule: 979/1065

4.5 Test vlivu koeficientu relaxace na počet splněných klauzulí

Pokud disponujeme výchozím (úplným) řešením, počet splněných klauzulí bude vždy maximální, protože algoritmus díky omezující podmínce nikdy nepřistoupí k řešení s menším počtem splněných klauzulí (stejný počet klauzulí s větší váhou možný už je). I pro instance, kde známe řešení je však třeba povolit určitou úroveň relaxace, abychom byli schopni nalézt sousední řešení.

Na následujícím grafu vycházejícího z měření úloh (vygenerovaných generátorem) o 150 klauzulích a počtu proměnných od 20 do 100 vidíme, že úroveň koeficientu pro dosažení plně splněné formule není zvláštně závislá na počtu proměnných.



4.6 TEST VLIVU PARAMETRŮ NA VÝSLEDNOU VÁHU

Váhy W=(w1, w2, … , wn) jednotlivých proměnných jsou načítány z přidruženého souboru „**.weight**“. Pokud soubor neexistuje, jsou váhy vygenerovány náhodně s maximální váhou proměnné 100. Po prvním vygenerování jsem během měření již vždy používal ty samé váhy, jinak by měření nemělo žádnou relevanci. Váha řešení je suma všech vah těch proměnných, které mají hodnotu 1.

Váha neúplných měření je penalizována koeficientem poměru splněných klauzulí a váha úplných řešení je zvýhodňována násobkem 2. Během těchto měření jsem však uvažoval pouze úplná řešení, penalizace tak není na výsledných hodnotách uplatňována (až na měření právě koeficientu relaxace).

Měření probíhá na instanci o velikosti 50 proměnných / 150 klauzulí vygenerované generátorem.

4.6.1 Náhodný prvek algoritmu a relaxace

Některé výsledky v tomto měření nevyšly až tak ideálně, jako např. v případě problému batohu. Jedním důvodem je samotný prvek náhody metody simulovaného ochlazování. Druhým a důležitějším prvek jsou určité vlastnosti problému SAT, které mě přinutili zaintegrovat do algoritmu další omezující opatření, které se samozřejmě na měřeních projevují.

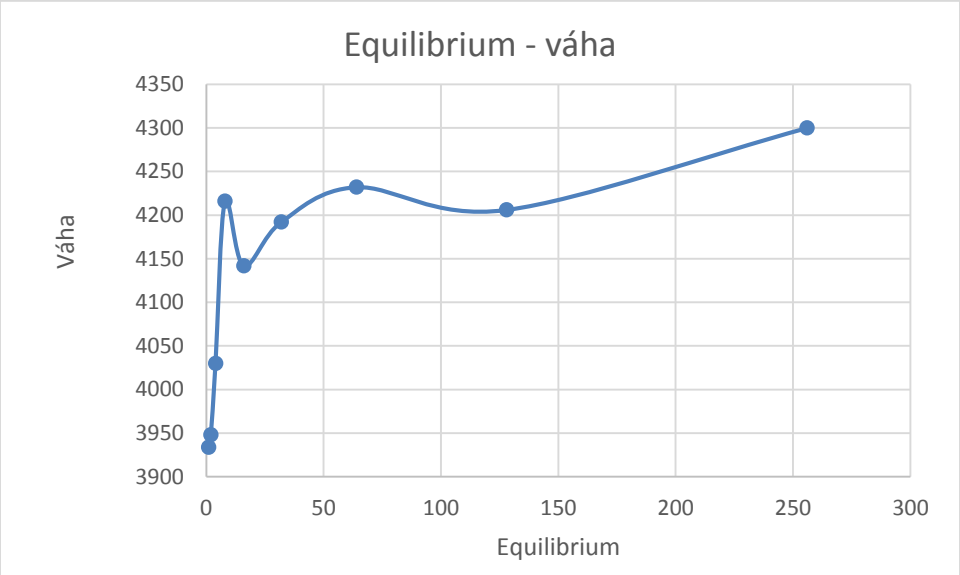
Jednou takovou vlastností je, že 3SAT problém nemá triviální řešení a vygenerování náhodného řešení je složitější, než u problému batohu. Důvodem je, že je řešeních ve stavovém prostoru méně (samozřejmě záleží na konkrétních instancích, ale bavme se o průměru). Souvisejícím problémem je pak fakt, že mezi sousedy aktuálního řešení vůbec nemusí existovat řešení další. Toto mě donutilo do algoritmu zakomponovat již zmiňovanou relaxaci, kdy je algoritmus schopen pracovat i s neúplnými řešeními. Navíc pokud ani s použitím relaxace není mezi sousedy nalezeno dostatečně dobré řešení (např. z důvodu příliš striktně nastaveného koeficientu relaxace), je po určitém počtu pokusů vrácena zcela náhodná kombinace. Tato situace nastává při rozumném nastavení koeficientu zřídka (záleží na velikosti i konfiguraci instance, ale zpravidla < 0.99), ale i to se v měřeních projevuje.

Pozn. Naopak v případě batohu vždy nějaké další řešení mezi sousedy bylo alespoň odebráním jednoho z předmětů nebo v případě prázdného batohu přidáním libovolného jednoho předmětu (a pokud instance je řešitelná, vždy musí jít do prázdného batohu alespoň jeden předmět přidat).

4.6.2 Equilibrium

parametr	hodnota
Equilibrium	**1 – 256**
Počáteční teplota	calculated
Teplota tuhnutí	2
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	0,97

Jak vidíme na následujícím grafu, určitá zlomová hranice equilibria se nachází okolo hodnoty 50, což je právě velikost instance. V předchozích měřeních jsme zjistili, že časová složitost s velikostí equlibria roste lineárně (viz výše) – tedy únosně, a proto jsem se rozhodl pro finální nastavení equlibria na hodnotu 3x velikost instance.



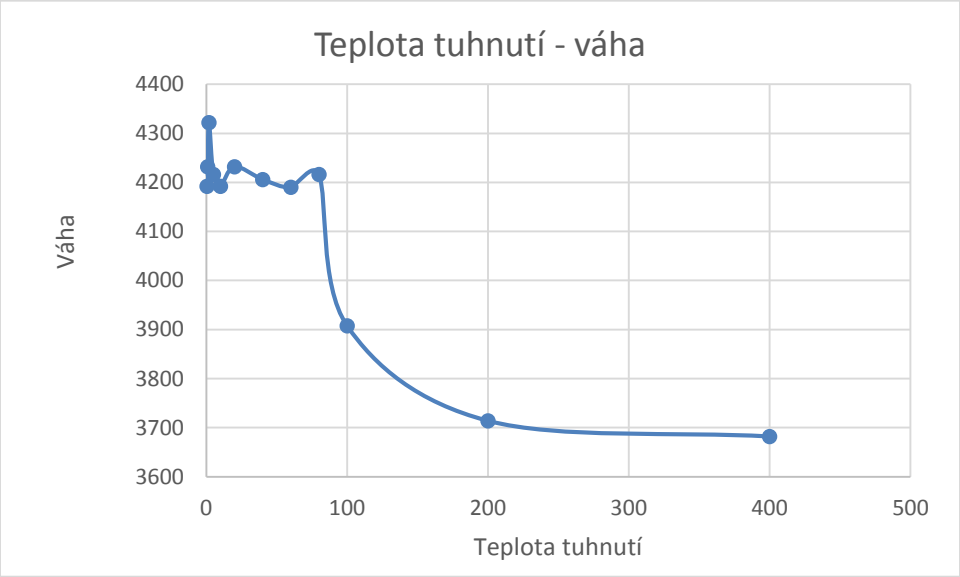
4.6.3 Teplota tuhnutí

parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	500
Teplota tuhnutí	**0,5 – 400**
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	0,97

Vzhledem k tomu, že měření teploty tuhnutí je velmi závislé také na hodnotě počáteční teploty, rozhodl jsem se tuto hodnotu pro toto měření zafixovat na hodnotě 500 (jinak je počítána dle pravidla 50:50).

Na grafu vidíme, že až do hodnoty 80 má algoritmus dostatek času vymanit se z lokálního minima a především se ustálit v prostoru řešení lepších vah. Při zatuhnutí u teplot blížících se počáteční teplotě zastavíme algoritmus ve fázi, kdy jsou často přijímána horší řešení a také neproběhne dostatek iterací, pro nalezení lepších řešení.

Finální nastavení teploty tuhnutí bych nastavil na hodnotu přibližně 2-5 stupňů a to z důvodu, abychom nebyli nuceni nastavovat příliš vysokou počáteční teplotu za účelem získání dostatečného počtu iterací vnější smyčky.

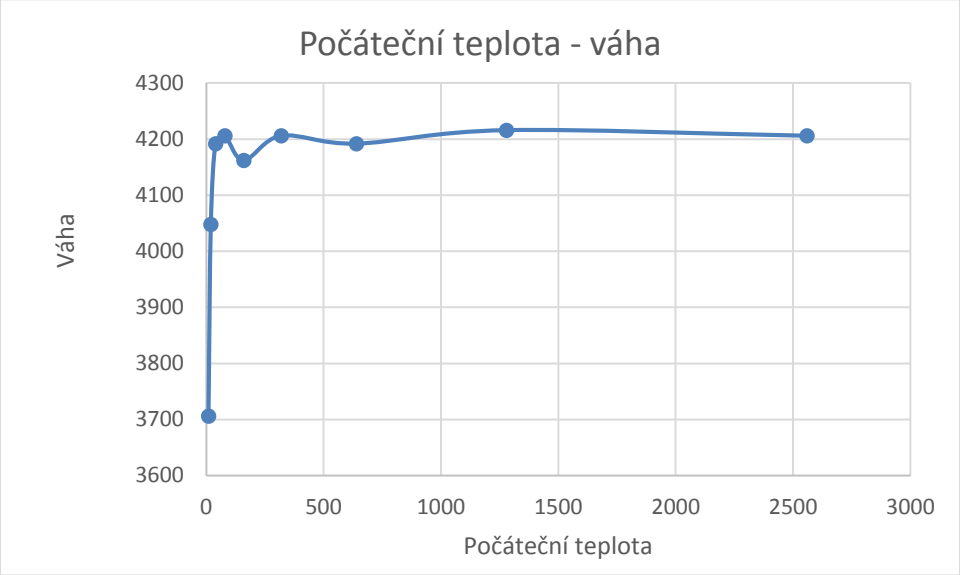


4.6.4 Počáteční teplota

parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	**10 – 2560**
Teplota tuhnutí	2
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	0,97

Počáteční teplota má podobný vliv na výsledek, jako teplota tuhnutí, protože společně s ní určuje počet iterací vnějšího cyklu. Rozdíl je ale v tom, že vysoká počáteční teplota lépe umožňuje opustit lokální minimum, protože zvyšuje pravděpodobnost přijetí horšího řešení.

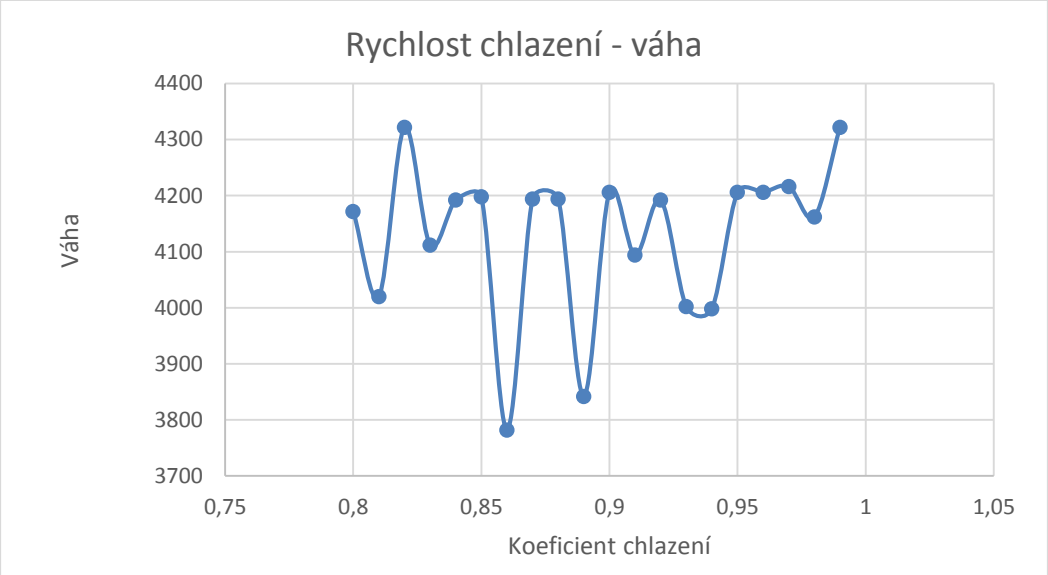
Na grafu vidíme, že už na přibližně hodnotě 80 dochází k získávání lepších řešení. Ve výsledku jsem tuto hodnotu nechal na výpočtu dle pravidla 50:50 (přibližně 50% horších řešení je přijato). Takto vypočtená hodnota se liší dle konkrétních instancí (+ prvek náhody), avšak v případě této instance se jednalo o hodnoty 100 – 200. Výpočet počáteční teploty dle pravidla 50:50 považuji za ideální a je navíc přenositelný na další problémy (v případě batohu též výpočet fungoval dobře).



4.6.5 Rychlost chlazení

parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	calculated
Teplota tuhnutí	2
Rychlost chlazení	**0,8 – 0,99**
Velikost instance	50
Koeficient relaxace	0,97

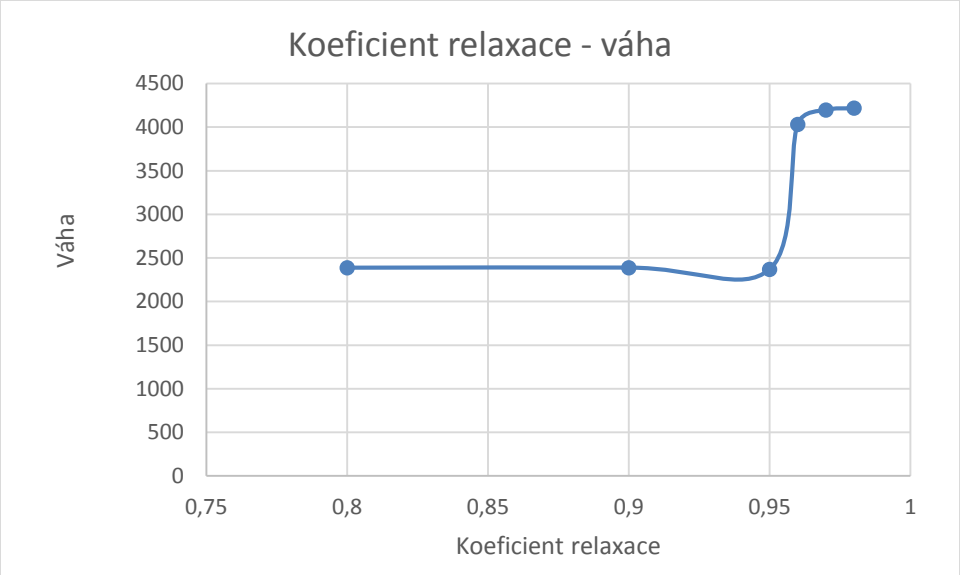
Následující graf bohužel není příliš vypovídající a výsledky jsou velmi náhodné. Jako důvod vidím především omezený prostor řešení (různých řešení není tolik) v kombinaci s prvky náhody algoritmu.



4.6.6 Koeficient relaxace

parametr	hodnota
Equilibrium	3*n = 150
Počáteční teplota	calculated
Teplota tuhnutí	2
Rychlost chlazení	0,97
Velikost instance	50
Koeficient relaxace	**0,8 – 0,98**

Zde se projevuje penalizace a zvýhodňování neúplných a úplných řešení (respectively). Dokud nám algoritmus vrací neúplná řešení, je jejich váha explicitně snížena, což se láme ve chvíli, kdy je koeficient dostatečně velký a vrací úplná řešení. Tento koeficient nelze obecně určit, neb je závislý na obtížnosti dané instance (množství řešení, které pro ní existují). Pokud instance obsahuje mnoho různých řešení, můžeme si dovolit jej zvýšit, v opačném případě musíme řešení více relaxovat (snížit koeficient).



5 ZÁVĚR

Na základě měření bylo ověřeno, že parametry algoritmu simulovaného ochlazování pro problém 3SAT jsou řádově stejné, jako pro problém batohu. Vzhledem k tomu, že princip algoritmu zůstává stejný, je toto očekávané chování.

Konkrétně je třeba parametry přizpůsobit velikosti instance, ale za vhodný startovní bod považuji:

- Equilibrium (rovnovážná poloha): Velikost instance * 3
- Počáteční teplota: Vypočtena dle pravidla 50:50
- Teplota tuhnutí: 2-5 stupňů
- Rychlost chlazení: koeficient 0,95 – 0,98
- Koeficient relaxace 0,97 pro splnění všech klauzulí

Výsledkem práce je algoritmus, který v případě, že nemáme požadavek na splnění všech klauzulí, dokáže efektivně řešit i velké instance o velikosti 200 proměnných a více. Problém nastává, kdy chceme u velké instance splnit všechny klauzule. Zde metoda selhává na faktu, že sousední řešení neobsahují vždy další řešení.

Měření vah řešení naznačují iterativní sílu algoritmu. Při zlepšování jednotlivých parametrů algoritmu dochází ke zlepšování výsledné váhy. Toto nebylo prokázáno pouze u parametru „rychlost chlazení“.