

Univerzitet u Beogradu
Matematički fakultet



SEMINARSKI RAD

Automatsko rezonovanje

Tema:

Poređenje CDCL i stohastičkih SAT rešavača na rešavanju zadovoljivih sudoku instanci

Profesor:
dr Filip Marić

Studenti:
Srđan Terzić, 1054/2012
Nikola Stanković, 1033/2012

Beograd, 8.6.2013.

SADRŽAJ

1 UVOD.....	3
2 SUDOKU	3
3 GENERISANJE INSTANCI	4
4 POKRETANJE REŠAVAČA	6
4.1 minisat2.....	6
4.1.1 Instance 3x3	7
4.1.2 Instance 4x4	8
4.1.3 Instance 5x5	9
4.1.4 Instance 6x6	10
4.1.5 Srednje vrednosti	11
4.2 ubcsat.....	12
4.2.1 Instance 3x3	13
4.2.2 Instance 4x4	14
5 UPOREĐIVANJE REZULTATA.....	15
7 ZAKLJUČAK	16
8 REFERENCE.....	17

1 UVOD

Jedan od glavnih problema u računarstvu trenutno predstavlja problem zadovoljivosti. Taj problem se svodi na pronalaženje interpretacije koja zadovoljava određenu buleansku formulu. Drugim rečima, potrebno je pronaći vrednosti promenljivih na takav način da celokupna formula bude tačna. Ovaj problem se još naziva i SAT problem (eng. satisfiability). SAT je prvi poznati primer NP-kompletnog problema. To znači da ne postoji poznati algoritam koji rešava sve instance SAT problema. Međutim, razvijeni su posebni SAT rešavači koji mogu rešiti dovoljno veliki skup problema. Oni su našli praktičnu primenu u raznim oblastima, kao na primer verifikaciji hardvera i softvera, automatskog dokazivanja teorema, ali i u svakodnevnim problemima poput pravljenja rasporeda časova ili rešavanja logičkih problema. Jedan od takvih logičkih problema je i problem rešavanja sudokua, na koji ćemo se fokusirati u daljem izlaganju.

Postoje dve vrste SAT rešavača, CDCL (eng. Conflict-Driven Clause Learning) i stohastički rešavači. CDCL rešavači su zasnovani na unapređenoj verziji DPLL algoritma, dok stohastički imaju veliki broj različitih algoritama koji se primenjuju. Osnovna razlika je i to što CDCL rešavač, osim što može da pokaže da je formula zadovoljiva, može da javi da je formula nezadovoljiva, dok stohastički može da prijavi samo zadovoljivost formule. U ovom radu ćemo uporediti dva SAT rešavača iz ove dve grupe, minisat2 i ubcsat.

2 SUDOKU

Sudoku je logička igra u kojoj se barata brojevima, a u nekim posebnim vrstama čak i slovima. Sastoji se od kvadratne table sa poljima u koja se unose brojevi. Tipičan sudoku se igra na tabli od 9x9 polja, a cela tabla se sastoji od devet manjih kvadrata 3x3. Analogno ovome, mogu se kreirati i veće table za igru, npr. 16x16, 25x25 i 36x36, ali su ovakvi problemi jako teški ako treba da ih rešava čovek. U nastavku teksta, radi jednostavnosti, sudokue ćemo obeležavati brojem unutrašnjih kvadrata. To znači da je sudoku 3x3 u stvari sudoku od ukupno 81 polja, a 4x4 sudoku od 256 polja.

Tipičan početak igre zasniva se na tabli na kojoj se nalazi određeni broj nasumično razbacanih cifara. Cifre su uvek od 1 do dimenzije sudokua. Na igraču je da popuni prazna polja pridržavajući se sledećih pravila:

1. U svakom redu moraju biti različite cifre
2. U svakoj koloni moraju biti različite cifre
3. U svakom unutrašnjem kvadratu moraju biti različite cifre

Bitno je da instanca bude rešiva, tj. da je moguće popuniti celu tablicu pridržavajući se ovih pravila.

3 GENERISANJE INSTANCI

Prva stvar koja je bitna prilikom generisanja instanci je da instanca mora imati jedinstveno rešenje. U skladu s tim, prvobitno zamišljeni princip nasumičnog popunjavanja tablice ne garantuje da će uslov rešivosti biti ispunjen, pa nam se kao rešenje nametnuo drugačiji pristup. Drugi pristup je bio generisanje popunjene tablice po poznatim uslovima, a zatim uklanjanje određenog broja cifara na slučajan način dok se ne dođe do zadovoljavajuće popounjenosti. Kada se kreira sudoku koji će rešavati čovek bitno je formirati ga tako da uvek ima jedinstveno rešenje. Zbog toga se koriste posebni algoritmi koji uklanjaju jedan po jedan broj i testiraju sudoku na jedinstvenost. Ukoliko u nekom trenutku sudoku postane nejedinstven, primenjuje se bektreking algoritam i pokušava se sa brisanjem drugog broja. Ipak, ovakva operacija može biti veoma zahtevna za formiranje sudokua čija je veličina veća od standardne. Zbog toga smo iz generisanih rešivih instanci slučajnim izborom eliminisali određeni broj cifara dok se ne dobije zahtevana popunjenost. Treba napomenuti da instance koje smo dobili ne moraju imati jedinstveno rešenje, već se može javiti i više različitih rešenja. Ovo pogotovu može doći do izražaja na slabo popunjenim instancama. Ipak smatramo da ovaj podatak nije relevantan jer je problem jedinstvonog rešenja potreban kada će sudoku rešavati čovek. Računar primenjuje drugačije strategije u rešavanju, pa smatramo da ovo pitanje nije ključno.

Takođe, odlučili smo da formiramo tri nivoa popunjenosti za svaku instancu, i to 10, 30 i 50 procenata. Formirali smo 5 različitih veličina sudokua od 3x3 do 7x7. Svaka veličina ima po četiri različite instance da bi dobijeni rezultati bili pouzdaniji.

Primer instance 3x3 sa različitim nivoima popunjenosti:

Popunjenost 100%	Popunjenost 50%	Popunjenost 30%	Popunjenost 10%
2 1 8 7 5 6 3 9 4	2 0 8 7 5 6 3 9 4	0 0 0 7 0 6 0 9 0	0 0 0 0 0 0 0 0 0
9 3 4 2 1 8 5 7 6	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	9 0 0 2 0 0 0 0 6
7 5 6 9 3 4 1 2 8	7 5 0 9 3 4 1 2 8	7 0 6 0 3 0 1 0 8	0 0 0 0 0 0 0 0 0
8 7 1 6 9 5 2 4 3	0 0 0 0 0 0 0 0 0	0 7 0 0 0 5 0 4 0	8 0 1 0 0 0 0 4 0
4 2 3 8 7 1 9 6 5	4 2 3 0 7 1 9 6 5	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
6 9 5 4 2 3 7 8 1	0 0 0 0 0 0 0 0 0	6 0 5 0 2 0 7 0 1	0 0 0 0 2 0 0 0 0
5 4 9 3 8 2 6 1 7	5 4 9 3 0 2 6 1 7	0 4 0 3 0 2 0 1 0	0 0 0 0 0 0 0 0 0
3 8 2 1 6 7 4 5 9	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0
1 6 7 5 4 9 8 3 2	1 6 7 5 4 0 8 3 2	1 0 7 0 4 0 0 0 2	1 0 0 0 0 0 0 0 0

Naredni problem predstavlja formiranje istinitosne formule koja opisuje svaku sudoku instancu. Pošto SAT rešavači barataju sa DIMACS formatom, napisan je program koji iz tekstualne datoteke sa sudoku tablicom formira istinitosnu formulu u DIMACS formatu.

Primer dimacs formule za gornju instancu popunjenosti 50%:

```
c Sudoku of size: 3
p cnf 729 11753
1 2 3 4 5 6 7 8 9 0
10 11 12 13 14 15 16 17 18 0
19 20 21 22 23 24 25 26 27 0
28 29 30 31 32 33 34 35 36 0
37 38 39 40 41 42 43 44 45 0
46 47 48 49 50 51 52 53 54 0
55 56 57 58 59 60 61 62 63 0
...
703 704 705 706 707 708 709 710 711 0
712 713 714 715 716 717 718 719 720 0
721 722 723 724 725 726 727 728 729 0
-1 -2 0
-1 -3 0
-1 -4 0
-1 -5 0
-1 -6 0
-1 -7 0
...
-711 -729 0
-720 -729 0
90 0
110 0
159 0
251 0
262 0
310 0
443 0
649 0
```

Treba napomenuti da gornji fajl ima preko 11 000 linija. Takođe treba uzeti u obzir da je ovo fajl za najjednostavniji sudoku 3x3. Pri kreiranju ovakvog fajla za instance 7x7, više nije realno govoriti o linijama fajla, već o njegovoj veličini u megabajtima. Naime, najveća obrađena instanca imala je 175 MB.

4 POKRETANJE REŠAVAČA

4.1 minisat2

Pošto je u predtestiranju primećeno da ovaj rešavač ne može da reši 7x7 instance sa manjom popunjenošću od 50%, te instance nisu ni razmatrane. Vrednosti u tablici izražene su u sekundama i dobijene kao srednja vrednost nakon tri pokretanja minisat2 rešavača za istu instancu. Za svaki nivo popunjenosti računata je i srednja vrednost radi lakšeg upoređivanja sa drugim rešavačem.

3x3	instanca 1	instanca 2	instanca 3	instanca 4	srednja vrednost
popunjenost 10%	0.020001	0.024001	0.020001	0.024001	0.022001
popunjenost 30%	0.012	0.020001	0.016001	0.016001	0.01600075
popunjenost 50%	0.016001	0.012	0.012	0.012	0.01300025

4x4					
popunjenost 10%	0.348021	0.356022	0.352022	0.344021	0.3500215
popunjenost 30%	0.264016	0.264016	0.256016	0.252016	0.259016
popunjenost 50%	0.232014	0.240015	0.240015	0.240015	0.23801475

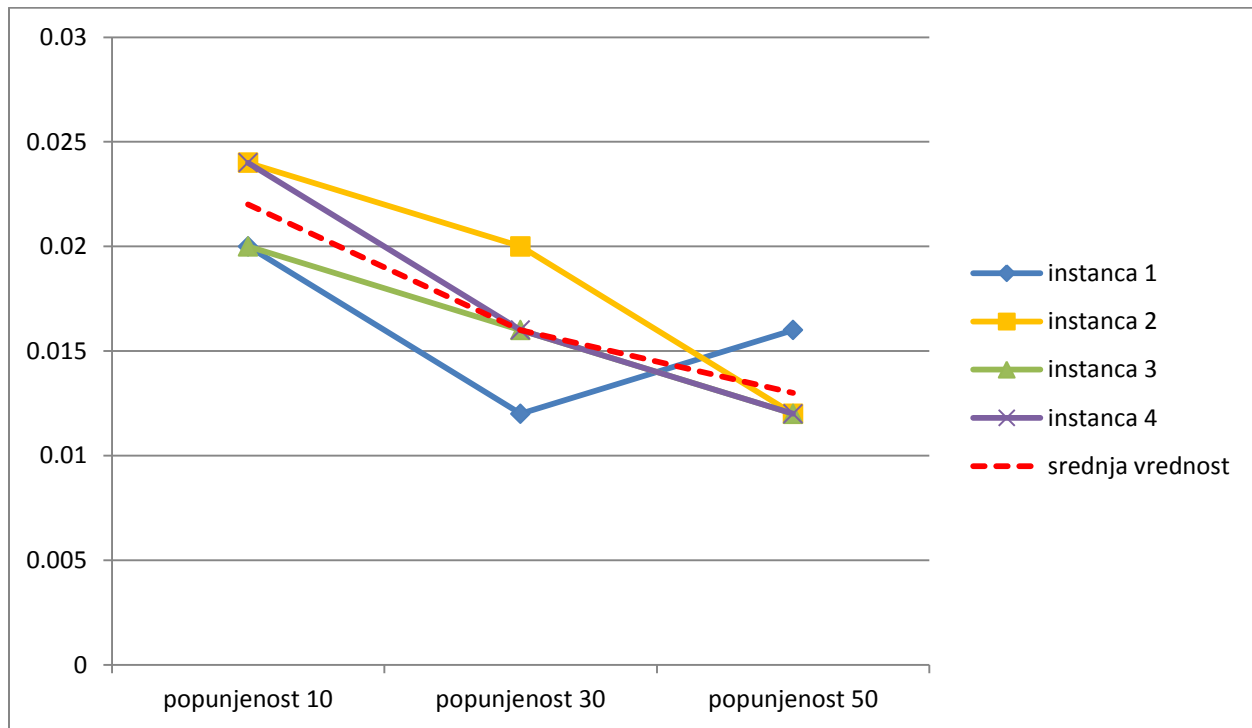
5x5					
popunjenost 10%	2.63616	2.50016	2.52016	2.75617	2.6031625
popunjenost 30%	2.28414	2.21614	2.00413	1.98012	2.1211325
popunjenost 50%	1.93612	1.78811	1.83611	2.23614	1.94912

6x6					
popunjenost 10%	69.1643	214.065	38.9304	12.5648	83.681125
popunjenost 30%	56.4875	19.2732	15.497	15.701	26.739675
popunjenost 50%	11.6527	10.6967	11.0847	10.6087	11.0107

7x7					
popunjenost 50%	70.1044	133.108	48.475	81.8651	83.388125

Dobijene vrednosti izražavamo preko grafikona zavisnosti vremena izvršavanja od popunjenosti instanci.

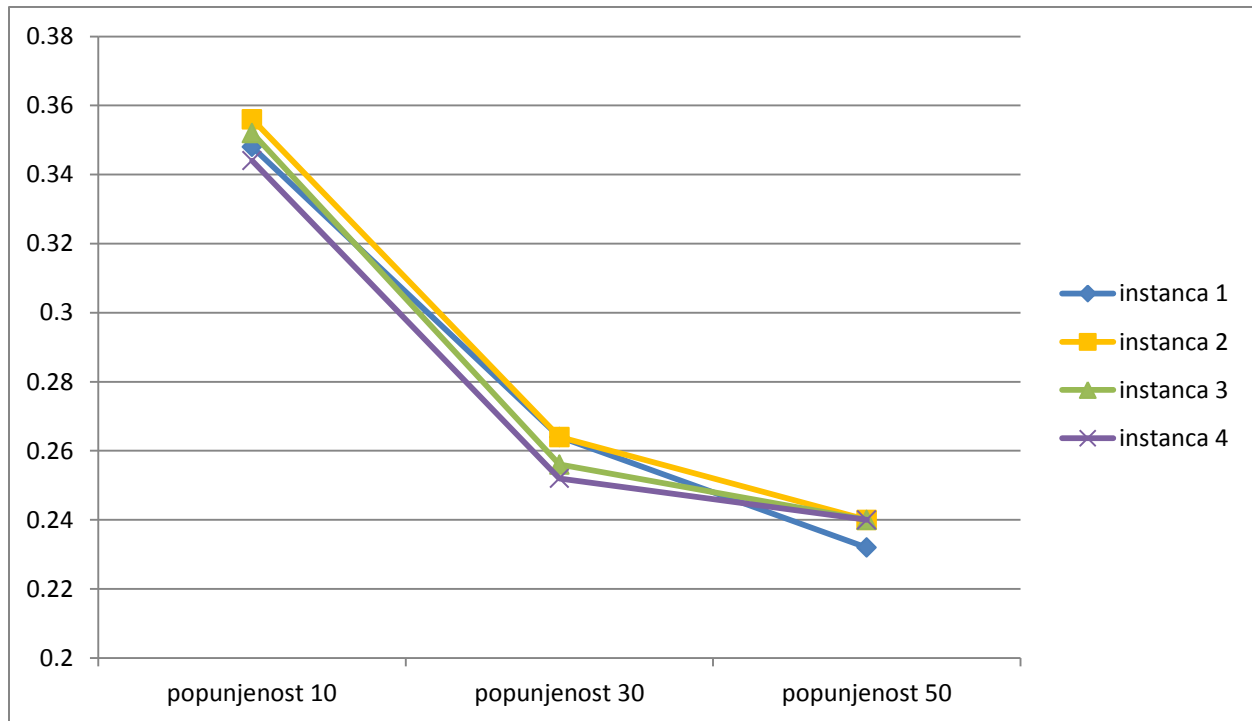
4.1.1 Instance 3x3



Grafik 1: sudoku 3x3, minisat2

Pošto se radi o veoma malim vrednostima, odstupanja na ovom grafiku se ne mogu smatrati ozbiljnim. Ipak, primećena je mala nepravilnost kod izvršavanja instance 1, gde je vreme izvršavanja za popunjenost od 50% veće od vremena izvršavanja za 30%. Iako je uočena ova nepravilnost, možemo zaključiti da vreme izvršavanja opada sa povećavanjem procenta popunjenosti instanci, što se jasno može videti iz linije koja označava srednju vrednost. Vrednosti dobijene za instancu 1 mogu se smatrati izuzetkom.

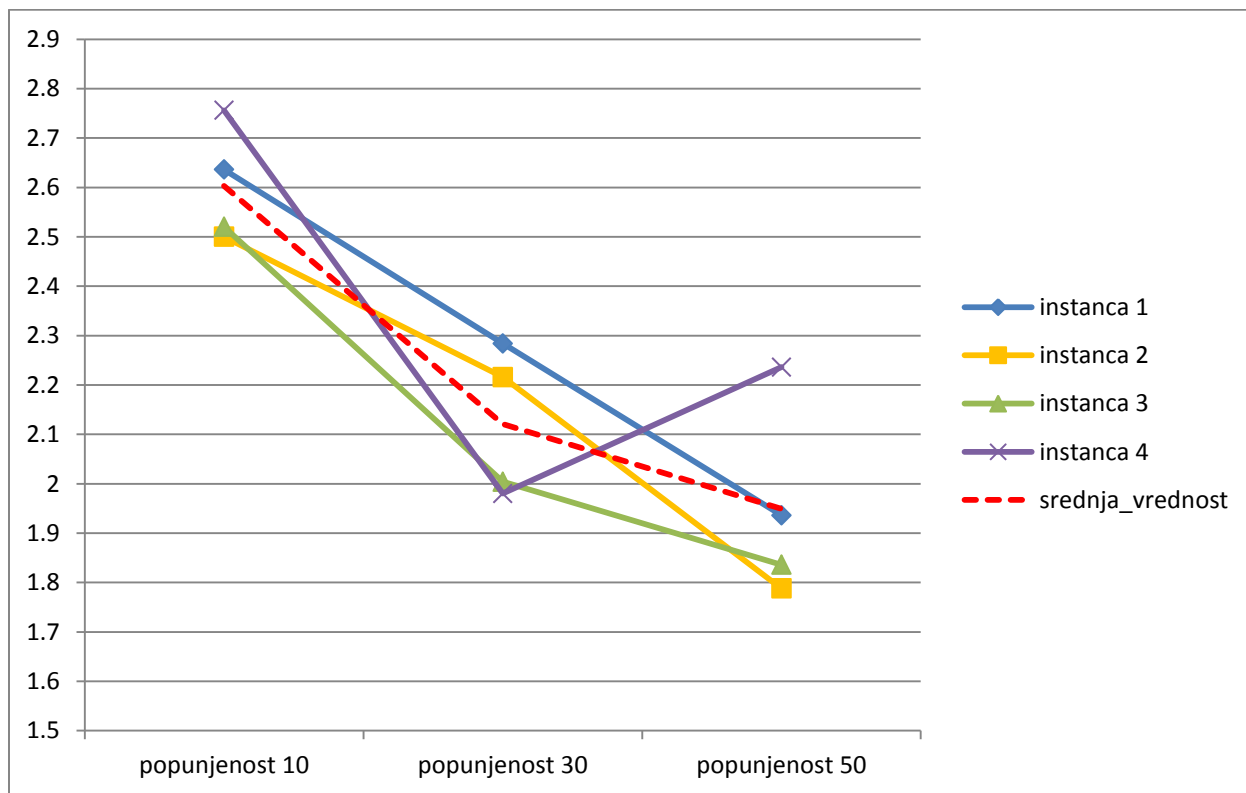
4.1.2 Instance 4x4



Grafik 2: sudoku 4x4, minisat2

Što se tiče instance 4x4, nisu primećena nikakva odstupanja, a vremena izvršavanja su neznatno veća nego za instance 3x3. Ovaj rešavač sa lakoćom rešava i najmanje popunjene instance za manje od jedne sekunde. Srednja vrednost ovde nije uzeta u obzir zbog preglednosti grafika, a takođe i ne doprinosi efikasnom tumačenju podataka jer su dobijeni rezultati suviše slični.

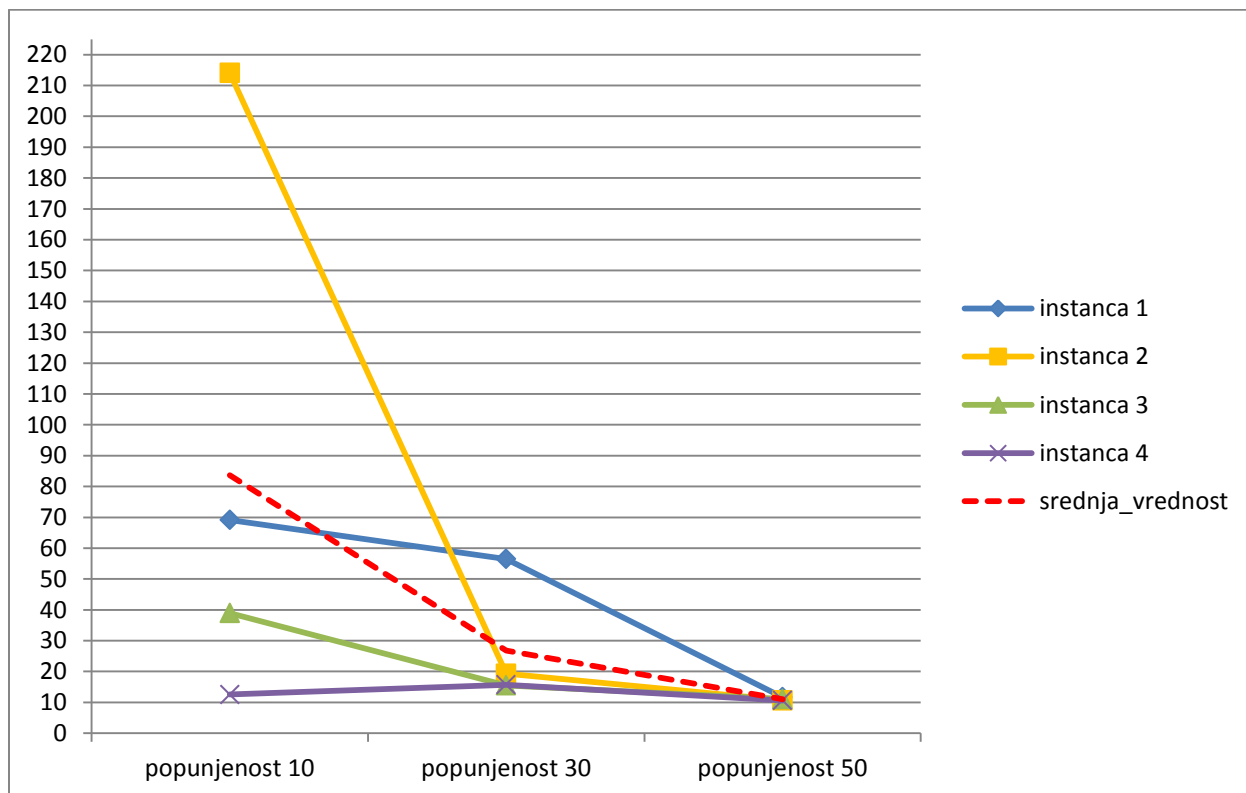
4.1.3 Instance 5x5



Grafik 3: sudoku 5x5, minisat2

Tek sa instancama 5x5 stvari počinju da budu zanimljive. Vreme izvršavanja počinje eksponencijalno da raste i sada je rešavaču potrebno mnogo više vremena nego pri rešavanju 4x4 instanci. Procentualno gledano, ova razlika je mnogo veća nego u slučaju prelaska između instanci 3x3 i 4x4. Opet je primećeno odstupanje gde je za veću popunjenost potrebno više vremena pri izvršavanju (instanca 4, popunjenosti 30% i 50%). Ovakvo odstupanje je ipak značajno jer u ovom slučaju ukupno vreme izvršavanja ide čak i do skoro 3 sekunde. Ipak postojanje ovog odstupanja ne utiče na prosečno vreme izvršavanja koje je u granicama normale. Treba primetiti i da je grafik srednje vrednosti znatno strmiji na prelazu između popunjenosti 10% i 30% nego između 30% i 50% (to se može primetiti i na prethodnim graficima, ali su vremenski intervali mnogo mali pa to nema veliki uticaj). Odavde izvodimo zaključak da ovom rešavaču odgovara povećavanje popunjenosti, ali samo do određene granice. Već za popunjenosti od 30 i 50% on ne dobija značajno ubrzanje povećavanjem popunjenosti. Ako razmotrimo ponašanje instance 1, vidimo da kod nje vreme izvršavanja linearno opada, ali se ovo ipak može smatrati specijalnim slučajem.

4.1.4 Instance 6x6

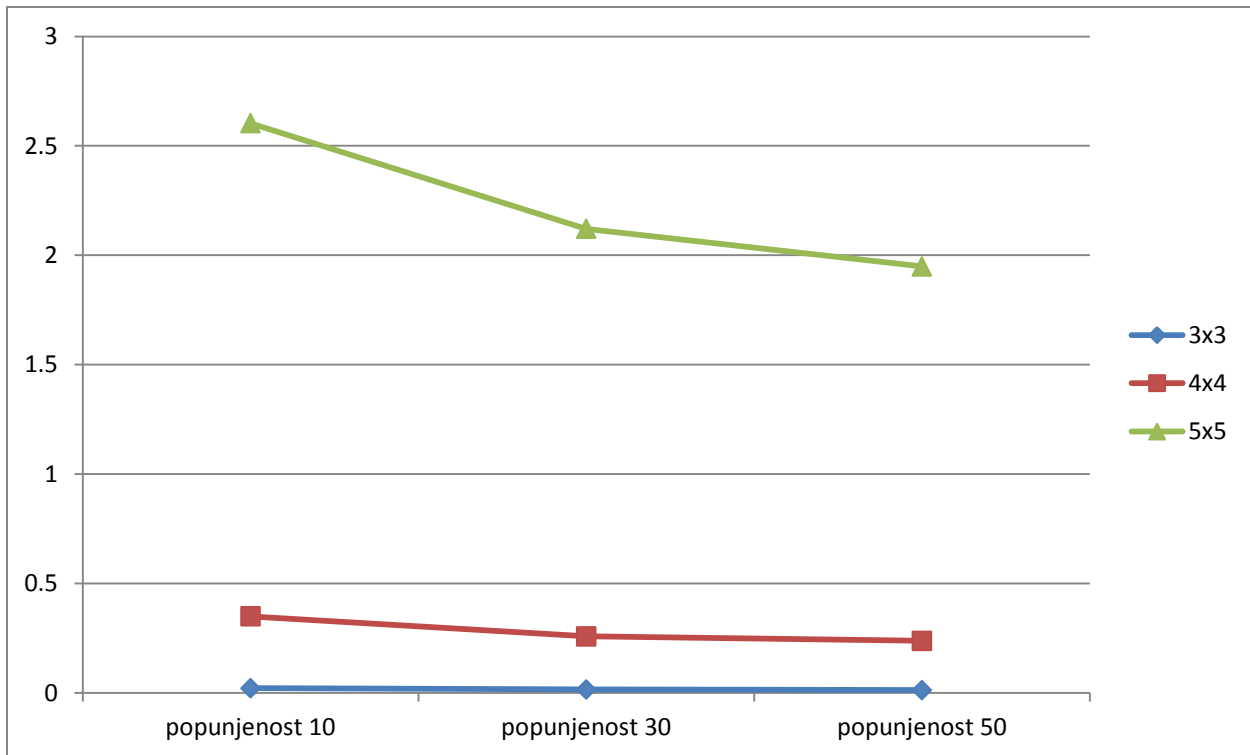


Grafik 4: sudoku 6x6, minisat2

Kod izvršavanja 6x6 instanci dobijeni rezultati su prilično interesantni. Sve instance popunjenosti 50% su rešene u približno istom vremenu od oko 10 sekundi, ali kako nivo popunjenosti opada tako se javljaju varijacije u vremenu izvršavanja. Ako posmatramo segment popunjenosti 30% možemo primetiti da samo instanca 1 beleži veće odstupanje koje u ovom slučaju iznosi oko 40 sekundi, što nikako nije mala vrednost. Ipak, najveću raznolikost imamo kod popunjenosti od 10%, gde neke instance rešavač reši i za nešto više od 10 sekundi, a sa nekim se muči i do 215 sekundi. Ovo nam pokazuje da je sudoku 6x6 sa popunjenošću 10% veoma zahtevan za rešavanje, što donekle opravdava varijacije u dobijenim rezultatima. Ipak, kada se razmotri srednja vrednost ona ostaje u granicama normale, a oblik njenog grafika je vrlo sličan prethodnim graficima.

Kod izvršavanja instanci 7x7 za popunjenost 50% dobijeni su određeni rezultati, koji kao i u slučaju instanci 6x6 popunjenosti 10% međusobno prilično variraju. Ispostavlja se da je ova popunjenost za instance 7x7 krajnji domet minisat2 rešavača jer popunjenost od 30% ne može da reši. Zbog toga su u tablici 1 prikazane samo vrednosti dobijene za instance sa 50% popunjenosti.

4.1.5 Srednje vrednosti



Grafik 5: Srednje vrednosti za sudokue 3x3, 4x4 i 5x5

4.2 ubcsat

Ubcsat, kao i ostali stohastički rešavači zasniva se na slučajnom izboru i ne može da prijavi nezadovoljivost formule. Pošto za ovaj rešavač postoji veliki broj algoritama za koje se pokreće (kao što su saps, novelty, novelty++, gsat...), izvršeno je predtestiranje i ispitivanje koji se algoritam najbolje ponaša pri rešavanju sudoku instanci. Došli smo do zaključka da je saps algoritam trenutno najpodobniji za rešavanje ovakve vrste problema. Tekođe je odlučeno da se program pokreće u 1000 iteracija svaki put, a svaki rezultat je dobijen kao aritmetička sredina šest pokretanja (jer je primećeno da rezultati više variraju nego kod izvršavanja minisat2 algoritma). U skladu sa tim rešavač ubcsat je pozivan na sledeći način:

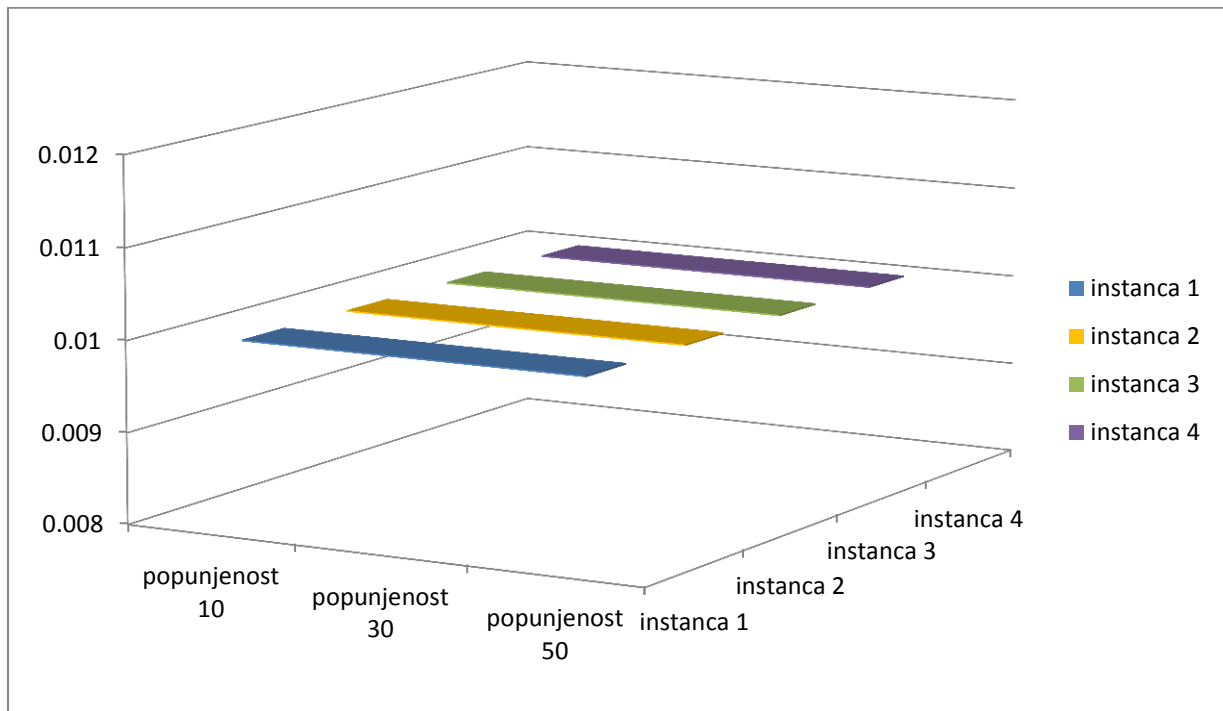
```
ubcsat -alg saps -inst 3x3instance1-50%.cnf -runs 1000 -findunique 1
```

Parametar „-alg“ označava algoritam koji će se koristiti (u našem slučaju saps), parametar „-inst“ instancu koja se pokreće, „-runs“ broj iteracija, a „-findunique 1“ označava da se rešavač zaustavlja čim pronađe model.

3x3	instanca 1	instanca 2	instanca 3	instanca 4	srednja vrednost
popunjenost 10	0.01	0.01	0.01	0.01	0.01
popunjenost 30	0.01	0.01	0.01	0.01	0.01
popunjenost 50	0.01	0.01	0.01	0.01	0.01

4x4	instanca 1	instanca 2	instanca 3	instanca 4	srednja vrednost
popunjenost 10	0.12	0.11	0.07	0.15	0.1125
popunjenost 30	0.59	0.49	0.43	0.61	0.53
popunjenost 50	0.08	0.09	0.07	0.08	0.08

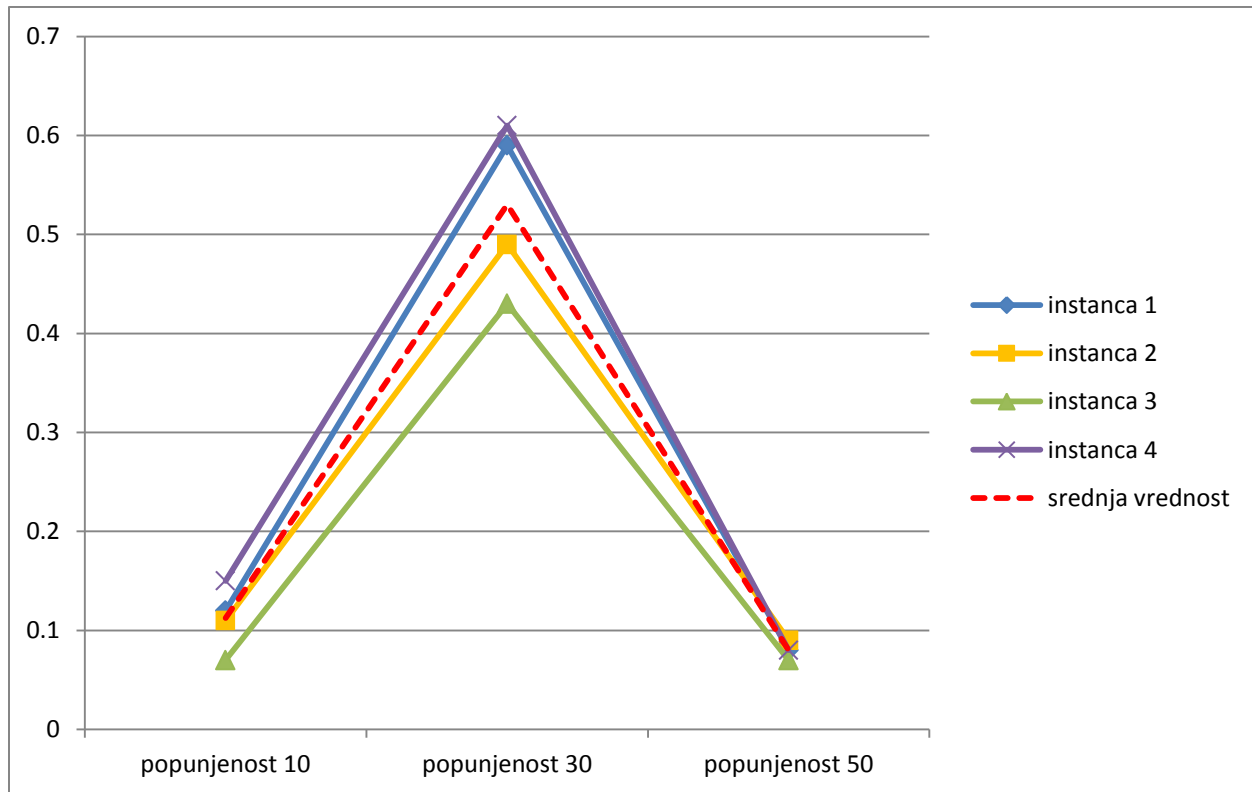
4.2.1 Instance 3x3



Grafik 6: sudoku 3x3, ubcsat

Kada razmotrimo pokretanje ubcsat rešavača za instance 3x3 možemo primetiti da su sve rešene u identičnom vremenu, bez obzira na popunjenost. Takođe, možemo primetiti da je brzina izvršavanja na zavidnom nivou, jer rešavač daje rešenje za stoti deo sekunde.

4.2.2 Instance 4x4



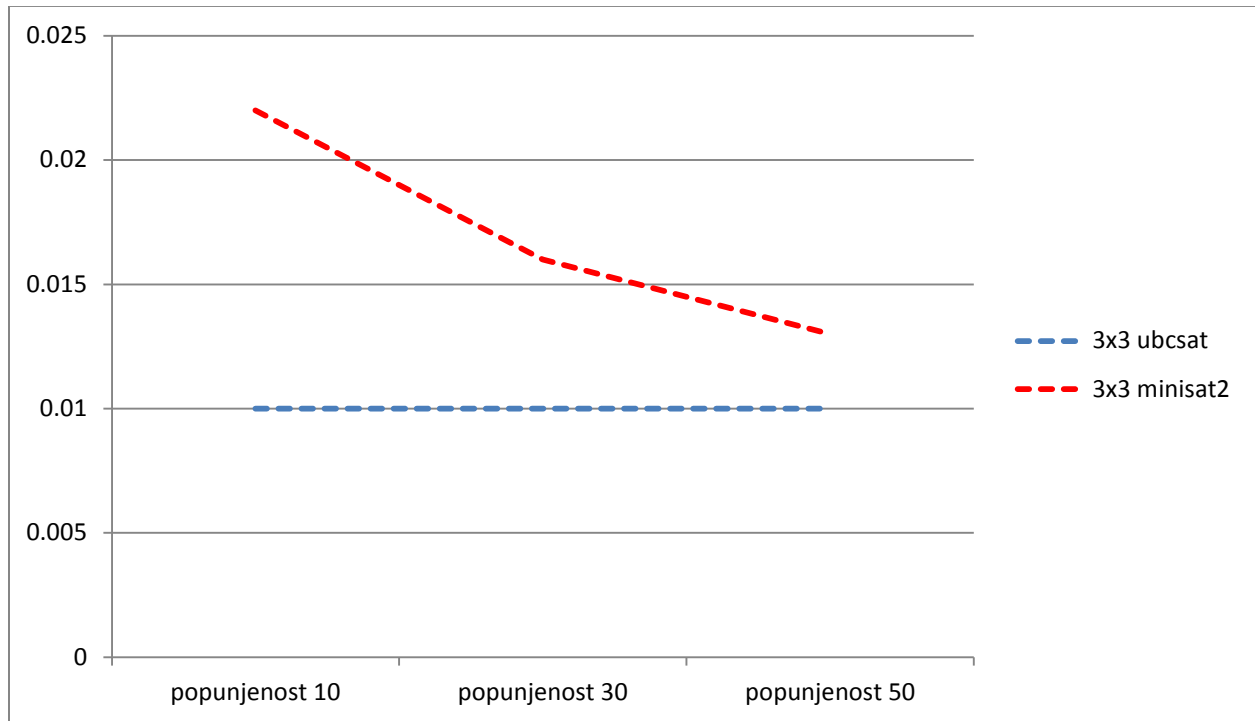
Grafik 6: sudoku 4x4, ubcsat

Ono što nas je iznenadilo bilo je pokretanje instanci 4x4. Primećuje se da je vreme izvršavanja za popunjenosti od 10 i 50% veoma slično i jako malo, dok je popunjenost od 30% iznenađujuće najzahtevnija. Ovako dobijene podatke ne možemo smatrati anomalijom, jer se isti obrazac ponavlja za pokretanje svake instance. Dolazimo do zaključka da je ovom rešavaču najteže da reši srednje popunjenu tablu, što je verovatno posledica njegovog nasumičnog pristupa rešavanju problema.

Ipak, ubcsat nastavlja da pokazuje čudne rezultate i u nastavku. Naime, on nije bio u stanju da reši ni 5x5 instancu popunjenu 50%. Čak i pokrenut za 1000 iteracija, ne uspeva da nađe model sa 0 nezadovoljivih klauza. Ta vrednost u iteracijama varira između 2 i 20, ali nikada 0. Ako se rešavač pokrene za manje popunjenu instancu vrednost nezadovoljivih klauza se penje čak i do 70. Respektivno, sa povećavanjem dimenzije problema i ove vrednosti se povećavaju i na taj način pokazuju nemoć ovog rešavača pri rešavanju većih problema.

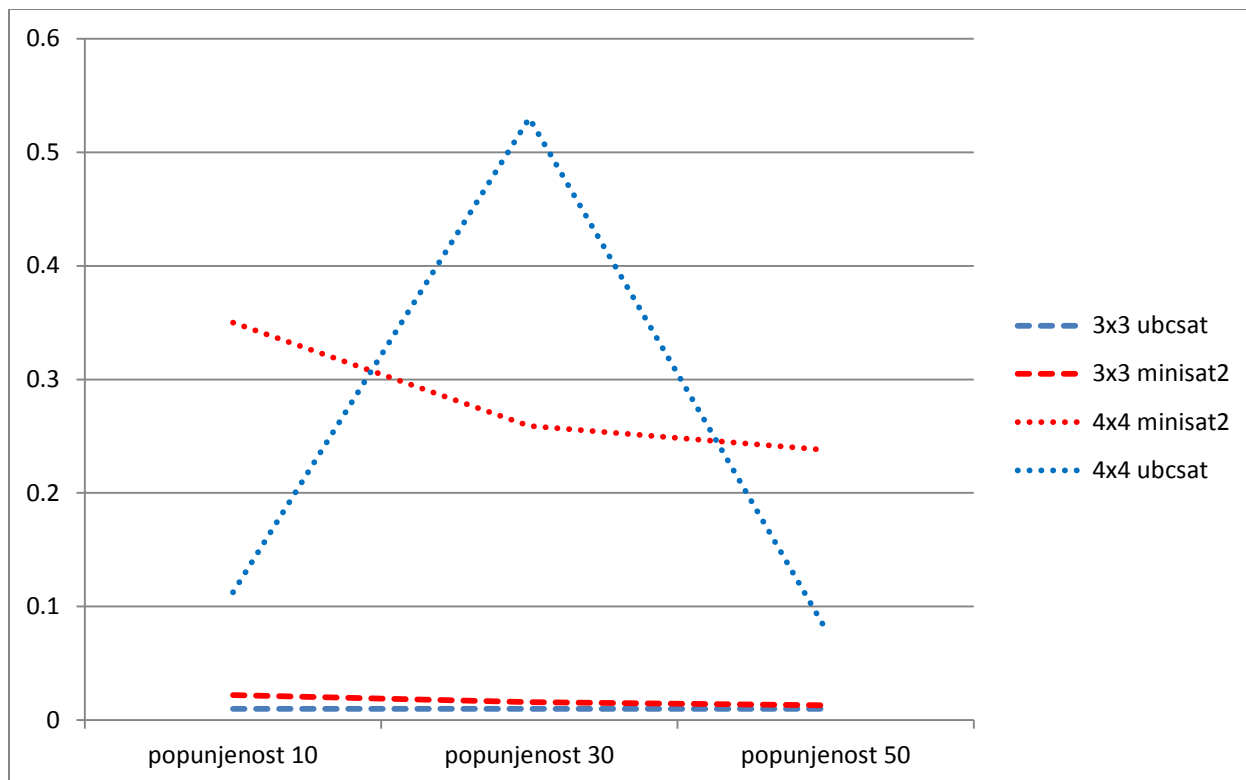
5 UPOREĐIVANJE REZULTATA

Pošto rešavači variraju u kvalitetu rešenja, upoređivaćemo instance koje su rešila oba rešavača, što konkretno obuhvata instance 3x3 i 4x4.



Grafik 7: Srednje vrednosti pri rešavanju 3x3 instanci za minisat2 i ubcsat rešavač

Primećujemo da je ubcsat brži i konstantniji u izvršavanju 3x3 instanci, ali je na ovom nivou takva dobit prilično zanemarljiva pošto se radi o veoma malim vremenskim intervalima.



Grafik 8: Srednje vrednosti za instance 3x3 i 4x4 za minisat2 i ubcsat rešavač

Ako razmotrimo ponašanje rešavača kod 4x4 instanci vidimo da je ubcsat znatno brži prilikom rešavanja instanci popunjenosti 10 i 50% i te vrednosti su vrlo bliske vrednostima dobijenim pri rešavanju instanci 3x3, dok kod minisat2 rešavača to ipak nije slučaj. Loša strana ubcsat rešavača je što pokazuje značajno usporenje kada se posmatra popunjenost od 30%. Vreme potrebno za izvršavanje je čak duplo veće u odnosu na vreme potrebno minisat2 rešavaču.

Iako ubcsat pokazuje bolje rezultate za instance 3x3 i 4x4 u odnosu na minisat2, on je ipak nemoćan kada se barata problemima veće dimenzije. Već za instance 5x5 ubcsat nije sposoban da izbaci rešenje, dok minisat2 to rešava bez problema za vreme manje od 3 sekunde. Takođe, minisat2 daje rešenje i za instance 7x7 popunjene 50%, što predstavlja nemoguć zadatak za ubcsat rešavač.

7 ZAKLJUČAK

Na osnovu dobijenih rezultata, generalno gledano, možemo zaključiti da je pristup rešavanju problema koji je implementiran u CDCL rešavačima znatno pogodniji za rešavanje sudoku instanci od stohastičkih rešavača. Stohastički rešavači imaju svoje prednosti koje se ogledaju u rešavanju problema manje dimenzije, ali pošto se radi o veoma malim vremenskim intervalima ta prednost je zanemarljiva. Na osnovu prethodnog tvrđenja i sposobnosti minisat2 rešavača da reši probleme velikih dimenzija zaključujemo da su CDCL rešavači bolji izbor prilikom rešavanja sudoku instanci.

8 REFERENCE

- Xiang-Sun Zhang, *Sudoku Puzzles Generating: from Easy to Evil*
- UBCSAT homepage: <http://ubcsat.dtopkins.com/>
- MiniSat homepage: <http://minisat.se/MiniSat.html>
- Wikipedia Sudoku page: <http://en.wikipedia.org/wiki/Sudoku>
- Milan Banković, Automatsko rezonovanje: <http://poincare.matf.bg.ac.rs/~milan/?content=ar>