

SAT Benchmarks from Circuits with Random Gates

Modoacă Iulia Todoca Ioana Șapcă Miruna Bogdan Jude

Universitatea de Vest din Timișoara
Facultatea de Matematica și Informatică
Program de master: Securitate cibernetică

Cordonator Științific: : Conf. Dr. Madalina Erascu

Monday 27th January, 2025



Cuprins

Introducere

Prezentare generală MiniSat

Instalare MiniSat

Analiza Codului MiniSat

Benchmarks - Descrierea Experimentului

Configurațiile Hardware

Detalii Benchmark - PC 1

Detalii Benchmark - PC 2

MiniSat în Debug Mode

Analiza Rezultatelor

Concluzii

Bibliografie

Introducere

Satisfiabilitatea booleană (SAT) a fost studiată de decenii, permițând astfel solverelor SAT să abordeze sarcini precum validarea proiectelor digitale. În ciuda progreselor, performanța lor rămâne limitată și sunt adesea folosite ca cutii negre,” procesând problemele în formă normală conjunctivă (CNF) fără interacțiune. Solvere SAT bazate pe conflict îmbunătățesc eficiența folosind metoda DPLL, backtracking cu analiză a conflictelor și învățare de clauze, propagare a constrângerilor booleene (BCP) și literalii urmăriți pentru optimizare.

Aceste solvere pot fi împărțite conceptual în trei componente principale::

1. Reprezentare. Instanța SAT trebuie să fie reprezentată intern folosind structuri de date eficiente, împreună cu orice informație derivată.
2. Inferență: Deși căutarea prin forță brută nu este de obicei suficientă, solvele se bazează pe mecanisme pentru a calcula și propaga implicațiile directe ale stării curente a informațiilor.
3. Căutare. Căutarea poate fi văzută ca o altă metodă de derivare a informațiilor.

Prezentare generală MiniSat

MiniSat este un solver SAT foarte eficient și extensibil, utilizat pe scară largă în mediul academic și în industrie datorită simplității, robusteții și performanței sale. Folosește algoritmul de învățare a clauzelor bazate pe conflicte (CDCL), care învață din conflicte pentru a preveni greșelile repetate, folosind First-UIP pentru eficiență și o schemă de urmărire a doi literali pentru propagarea rapidă a clauzelor. Strategii avansate, precum heuristicile bazate pe activitate (de exemplu, VSIDS), mecanismele de reluare și tehnicile de simplificare, îmbunătățesc și mai mult performanța acestuia.

Ce este satisfacibilitatea booleană (SAT)?

1. SAT este problema de a decide dacă există o atribuire de valori pentru variabilele unei formule propoziționale astfel încât formula să fie adevărată.
2. Este o problemă clasică în informatică și are aplicabilități largi în diverse domenii (verificare formală, inteligență artificială, satisfacția constrângerilor, optimizare).

MiniSat este un solver SAT de versatil, utilizat pe scară largă pentru evaluarea și testarea noilor algoritmi și euristici. Codul său compact și performanța robustă îl fac o bază excelentă pentru evaluarea euristicilor alternative de decizie, metodelor de preprocesare și strategiilor de analiză a conflictelor. Simplitatea MiniSat asigură ușurința modificării și integrării funcțiilor experimentale, susținând adoptarea sa atât în cercetarea academică, cât și în aplicațiile practice. De asemenea, a jucat un rol semnificativ în competițiile SAT, oferind un standard pentru compararea solverelor în condiții consecvente.

Instalare MiniSat

Cum se instalează MiniSat pe Ubuntu cu WSL (Windows Subsystem for Linux)?

1. Actualizarea pachetelor sistemului prin utilizarea comenzilor `sudo apt update` si `sudo apt upgrade`
2. Instalarea MiniSat cu executarea comenzi `sudo apt install minisat`
3. Verificarea instalarii prin executarea comenzii `minisat` in terminal .

Analiza Codului MiniSat

MiniSat: Arhitectură și Performanță

- ▶ Codul MiniSat este scris în C++ și este destul de compact, având mai puțin de 10.000 de linii de cod.
- ▶ Arhitectura codului este ușor de înțeles și modificat, oferind o bună bază pentru cercetători și dezvoltatori care doresc să experimenteze și să îmbunătățească performanța solverului.

Principalele Componente ale Codului MiniSat:

- ▶ Funcția solver: Este punctul de intrare pentru rezolvarea unei probleme SAT. Coordonează întregul ciclu de viață al solverului, de la preprocesare până la returnarea rezultatului.

Pași principali:

1. Setarea stării solverului: nivele de decizie, atribuirea variabilelor, baza de clauze.
 2. Preprocesarea: Reducerea complexității înainte de a începe căutarea efectivă.
 3. Propagarea unității: Deducerea valorilor variabilelor pe baza clauzelor unitare.
 4. Analiza conflictelor și învățarea clauzelor: Dacă apare un conflict, se analizează și se adaugă o clauză învățată.
- ▶ Funcția propagate: Se ocupă de propagarea unității, proces esențial pentru reducerea complexității problemei SAT.
 1. Tehnica "Two-Literal Watching": În loc să urmărească toate literele dintr-o clauză, MiniSat monitorizează doar două, ceea ce reduce overhead-ul computațional.

- ▶ Funcția analyze: Este responsabilă pentru diagnosticarea conflictelor și generarea de clauze învățate. Aceasta minimizează redundanța și îmbunătățește performanța în iterațiile viitoare.

First-UIP (Unique Implication Point): Un punct cheie în analiza conflictului, care ajută la minimizarea traseelor de căutare greșite și la îmbunătățirea performanței.

- ▶ Funcția simplify: Este utilizată pentru eliminarea clauzelor redundante și pentru optimizarea memoriei.

Reducerea bazei de clauze: Acest proces ajută la îmbunătățirea performanței și la economisirea resurselor.

- ▶ Optimizări ale Codului:

1. Euristicile de selecție a variabilelor (VSIDS): Prioritizează variabilele care sunt relevante pentru conflictele recente.
2. Restarturi periodice: Permite explorarea unor noi zone ale spațiului de soluție, ajutând la evitarea blocajelor în căutare.
3. Eliminarea clauzelor inutile: Asigură că baza de clauze rămâne relevantă și nu se aglomerează cu informații inutile care ar putea degrada performanța.

Benchmarks - Descrierea Experimentului

Obiectivele experimentului

1. Compararea performanței MiniSat în două medii hardware diferite
2. Măsurarea timpilor de rulare pentru fișiere CNF din benchmark-ul random-circuits

În benchmark-ul circuitelor aleatorii, circuitele booleene sunt modelate ca formule CNF, iar problemele SAT sunt generate prin construirea circuitelor din porți alese aleatoriu. Fiecare poartă mapează bitii de intrare la bitii de ieșire, iar circuitul este construit prin conectarea mai multor porți, controlând relațiile de intrare-ieșire. Pentru a crea o problemă SAT, se aplică alocări aleatorii ale intrărilor, se calculează bitii de ieșire rezultați și aceste ieșiri sunt codificate în formule CNF. Această abordare evaluează capacitatea solverelor SAT de a procesa eficient probleme bazate pe circuite, valorificând tehnici specializate pentru identificarea și gestionarea porților logice.

Configurațiile Hardware

PC 1 :

1. CPU: Intel i7-8565U @ 1.99 GHz.
2. RAM: 16 GB
3. Sistem de operare: Windows 11

PC 2 :

1. CPU: CPU: Intel i7-8550U @ 1.80 GHz.
2. RAM: 16 GB
3. Sistem de operare: Windows 10

Detalii Benchmark - PC 1

Fișiere benchmark și timpi de rulare

1. Timp de rulare pentru fiecare fișier CNF.
2. Rezultate obținute: "SAT" pentru toate fișierele

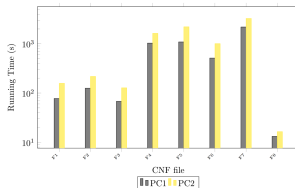
Benchmark Name	Running Time	File Size	MiniSAT Result
7ac7fabd8c078aea420087a0c80e5563-circuit_32in32out_with_400gates_6in6out_dist64_seed1	77.6646 s	98.635 kb	SAT
6f7a0e1cf94b6b26eafc08a827a692ce-circuit_64in64out_with_64gates_8in5out_dist256_seed1	125.385 s	27.676 kb	SAT
37ca184832fc6fa43a22ae900f1756a2-circuit_32in32out_with_350gates_6in6out_dist64_seed1	67.9899 s	85.275 kb	SAT
71ec94c233016219e12d671594dc88e5-circuit_32in32out_with_70gates_7in7out_dist128_seed1	1032.41 s	68.994 kb	SAT
170b13af977e962321c493544b2bd0a9-circuit_48in64out_with_800gates_4in4out_dist128_seed4	1088.31 s	8.123 kb	SAT
3966d56f3fd7b55afbb4254eaf6746c-circuit_32in32out_with_80gates_7in7out_dist128_seed1	512.985 s	79.170 kb	SAT
8942dca5dc0876fc3f723f738d72de1c-circuit_32in32out_with_64gates_8in6out_dist128_seed2	2186.12 s	61.300 kb	SAT
303480ca7e8322d771c94caf4ebd4e95-circuit_48in64out_with_700gates_4in4out_dist128_seed1	13.2318 s	7.046 kb	SAT

Detalii Benchmark - PC 2

Fișiere benchmark și timpi de rulare

1. Timp de rulare pentru fiecare fișier CNF - PC2
2. Timpul mai lung pentru unele fișiere față de PC1.
3. Compararea performanței pe PC1 vs. PC2.

Benchmark Name	Running Time	File Size	MiniSAT Result
170b13a977e962321e493544b2bd0a9-circuit_48in64out_with_800gates_4in4out_dist128_seed4	2207.79 s	8.123 kb	SAT
303480ca7e8322d771c94caf4eb4e95-circuit_48in64out_with_700gates_4in4out_dist128_seed1	16.4813 s	7.046 kb	SAT
37ca184832fc6a43a22ae900f1756a2-circuit_32in32out_with_350gates_6in6out_dist64_seed1	127.865 s	85.275 kb	SAT
396fd56f3d7185aafbba425e4e6f746c-circuit_32in32out_with_80gates_7in7out_dist128_seed1	1005.2 s	79.170 kb	SAT
511a2d81661a185c2115dae42270dec-circuit_32in32out_with_500gates_6in6out_dist64_seed1	661.589 s	125952 kb	SAT
6f7a0e1c94b6b26cafc08a827a692ce-circuit_64in64out_with_64gates_8in5out_dist256_seed1	217.443 s	27.676 kb	SAT
70bf0d054d4c9f3394fab32845b49243-circuit_32in32out_with_100gates_7in7out_dist64_seed2	624.875 s	101376 kb	SAT
71ce94c233016219e12d671594dc88e5-circuit_32in32out_with_70gates_7in7out_dist128_seed1	1627.2 s	68.994 kb	SAT
7ae7fabd8c078aea420087a0c80e5563-circuit_32in32out_with_400gates_6in6out_dist64_seed1	157.582 s	98.635 kb	SAT
8942ca5dc0876fc3f723f738d72de1c-circuit_32in32out_with_64gates_8in6out_dist128_seed2	3263.72 s	61.300 kb	SAT



MiniSat în Debug Mode

Ce este Debug Mode?

1. MiniSat în Debug Mode este folosit pentru a observa pașii interni ai solverului și pentru a înțelege mai bine cum sunt procesate fișierele CNF.
2. Utilizat pentru a depista erori sau pentru a înțelege performanțele în detaliu.

Cum am folosit Debug Mode:

1. Setarea unui benchmark simplu: Am creat un fișier CNF mic (6 variabile și 11 clauze) pentru a observa comportamentul solverului.

Pași observați:

1. MiniSat parsează fișierul CNF fără erori.
2. A fost invocată funcția de simplificare, dar nu s-au eliminat clauze, deoarece problema era simplă.
3. Funcția de propagare a fost invocată pentru deducerea valorilor variabilelor.
4. Funcția de backtracking: Nu a fost necesară, deoarece nu au apărut conflicte.
5. Rezultatul: MiniSat a determinat că problema este satisfiabilă fără a fi nevoie de învățarea clauzelor sau backtracking.

Observatii importante!

- ▶ DPLL și CDCL în practică: DPLL a permis reducerea problemei prin propagarea implicațiilor logice. CDCL a fost de asemenea observat, dar nu a fost necesar să învățăm clauze deoarece nu au apărut conflicte.
- ▶ Impactul UIP (Unique Implication Point): Procesul de identificare și minimizare a conflictelor, precum și evitarea căutării traseelor greșite, au fost foarte clare în debug mode.

Analiza Rezultatelor

Factorii care influențează timpul de rulare:

1. Numărul de variabile și clauze din fișierele CNF..
2. Impactul complexității problemei asupra timpului de execuție.
3. Algoritmii DPLL și CDCL.
4. Performanța depinde mult de frecvența procesorului și de gestionarea resurselor.
5. Impactul complexității problemelor asupra timpilor de execuție

Concluzii

1. MiniSat demonstrează eficiență în rezolvarea problemelor SAT de complexitate variabilă.
2. Factorii hardware și software influențează semnificativ performanța solverului.

Bibliografie

1. Aloul et al. - "Solving Difficult Instances of Boolean Satisfiability in the Presence of Symmetry," IEEE Transactions on Computer-Aided Design, 22(9), pp. 1117-1137, 2003
2. Benchmark Problems - This resource provides a collection of satisfiable and un- satisfiable SAT benchmark instances.
3. Documentation - "Benchmarking Circuits," which includes information on random gate sequences for evaluating error rates.
4. , T., Gableske, O., Sinz, C. (2004). Restoring circuit structure from SAT instances. International Workshop on Logic Synthesis (IWLS).
5. Xu et al. - "SAT Benchmarks for Randomized Urquhart Problems," which provides a collection of CNF problems for SAT solvers