



Resolvendo o problema PSAT com auxílio da ferramenta de software livre MiniSat

Aluno: Mikail Campos Freitas

Orientador: Prof. Dr. Marcelo Finger

Introdução

Um problema clássico e muito importante na Ciência da Computação é o Problema da Satisfatibilidade (SAT). É constituído em decidir se uma dada fórmula booleana pode ser satisfeita ou não. Foi o primeiro problema a ser mostrado ser NP-completo.

Por possuir diversas aplicações diretas e indiretas (muitos problemas são traduzidos para instâncias SAT e assim então resolvidos) algoritmos e softwares para resolver o SAT têm evoluído muito nos últimos anos: soluções atuais são mais de 30 vezes mais rápidas do que soluções em 2001!

Uma extensão do SAT é o Problema da Satisfatibilidade Probabilística (PSAT), onde temos um conjunto de cláusulas sobre variáveis booleanas e um conjunto de probabilidades associadas a essas cláusulas. O PSAT constitui-se em decidir se esse conjunto de restrições probabilísticas, aplicado ao conjunto de cláusulas em questão, é consistente ou não.

Soluções para o PSAT baseadas no SAT

Buscando utilizar o fato de que existem resolvidores SAT já muito eficientes, Marcelo Finger e Glauber de Bona propõem três soluções[1]:

(1) Redução Canônica de PSAT para SAT:

Tradução direta de uma instância PSAT para uma instância SAT, utilizando-se um resolvidor SAT para decidir a satisfatibilidade da instância traduzida.

(2) Geração de Colunas através de Redução de Turing para SAT:

Utilização da técnica de geração de colunas do algoritmo Simplex. O resolvidor SAT é utilizado para calcular o custo associado à solução inicial, e para verificar a consistência das soluções parciais em cada passo.

(3) Geração de Colunas através de Redução de Turing para MAXSAT ponderado:

Semelhante ao algoritmo anterior, mas baseado no uso de um resolvidor MAXSAT Ponderado para escolher a melhor solução parcial em cada passo.

Finger e de Bona criaram três softwares livres para implementar cada algoritmo citado, respectivamente: PSATtoSAT, PsatColGen, PSATtoMaxSat. Tais softwares são distribuídos no projeto de nome PSAT [2].

Os resolvidores desenvolvidos são baseados em alguns softwares externos como:

- zChaff: resolvidor SAT (licença própria)
- wmaxsatz2009: resolvidor MAXSAT Ponderado (licença GPLv3)
- LA: utilizado para realizar inversão de matrizes (licença GPLv3)

Modificações Propostas

O resolvidor SAT utilizado, zChaff, apesar de software livre possui licença restrita para uso em pesquisa e que não possibilita a sua redistribuição sem consenso da Universidade de Princeton. Isso por sua vez limita a utilização e distribuição do projeto PSAT, que tem como objetivo poder ser utilizado, modificado e possivelmente redistribuído sem restrições.

A primeira modificação proposta é a troca do resolvidor SAT zChaff pelo MiniSat. O resolvidor MiniSat mostrou desempenho excepcional, sendo premiado em cinco de sete competições SAT que participou. Além disso o MiniSat foi desenvolvido com o intuito de facilitar sua integração a outros projetos e é distribuído sob licença MIT, que é compatível com GPLv3.

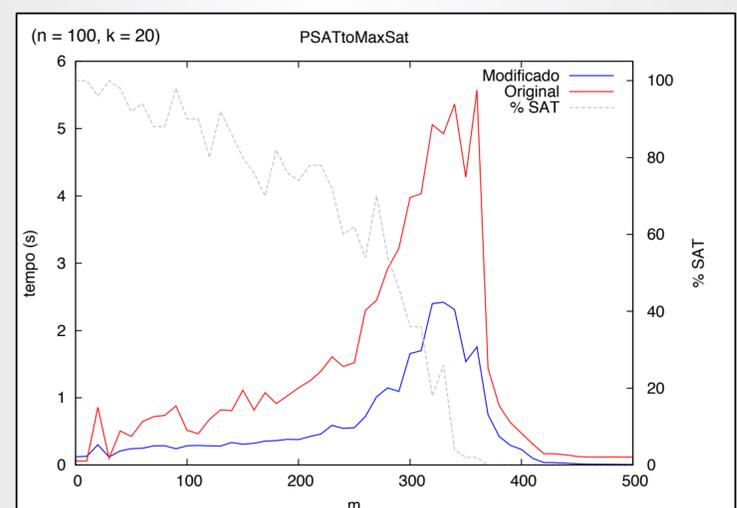
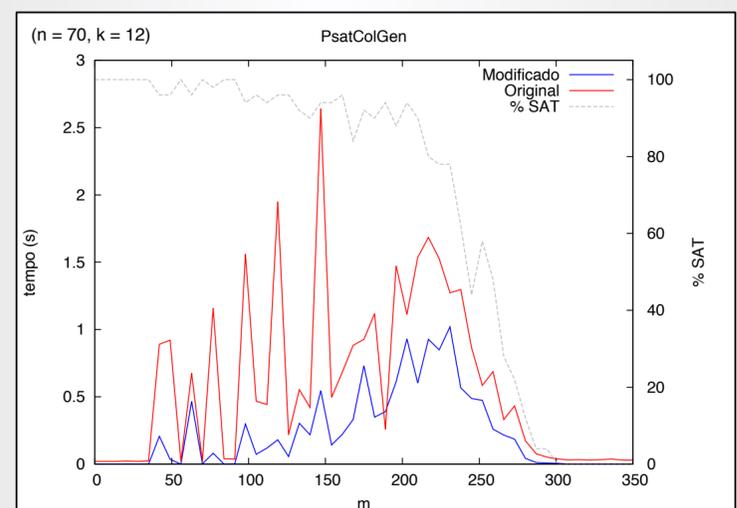
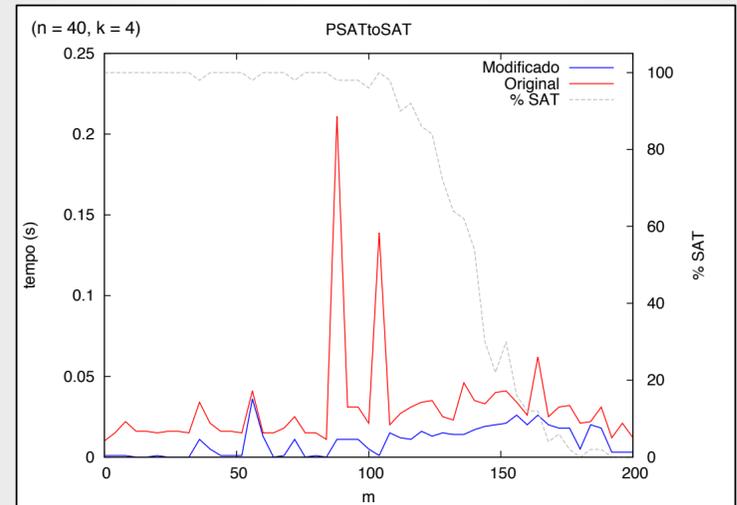
A segunda modificação proposta é integrar o código de todos os softwares auxiliares ao código dos resolvidores PSAT. Assim a troca de dados entre os resolvidores PSAT e os softwares auxiliares ocorre em memória RAM ao invés de ocorrer via leitura/escrita de arquivos, como é o caso atual. Além disso, transformando os softwares auxiliares em módulos do executável principal não há mais o overhead da criação de novos processos cada vez que cada um desses softwares é necessário.

Atualmente o projeto é distribuído sem o zChaff, que precisa ser obtido com a Universidade de Princeton [3].

Com as modificações propostas espera-se melhorar o desempenho dos resolvidores PSAT citados, assim como facilitar a utilização, modificação e distribuição deles, que passarão a ser distribuídos completos, sob licença GPLv3.

Resultados

Aqui são apresentados alguns resultados de propósito geral, comparando-se as versões originais e modificadas de cada resolvidor:



(Legenda: **k** - número de variáveis com probabilidade associada; **n** - número de variáveis sem probabilidade associada; **m** - número de cláusulas 3-SAT, geradas aleatoriamente, que devem ser satisfeitas; **%SAT** - porcentagem de instâncias satisfazíveis para cada valor de m)

Referências

[1] Finger, M. and de Bona, G. 2011. Probabilistic Satisfiability: Algorithms with the presence and absence of a phase transition. Manuscript in preparation, 2012.

[2] <http://sourceforge.net/projects/psat/files/>

[3] <http://www.princeton.edu/~chaff/zchaff.html>