

RESOLVENDO O PROBLEMA PSAT COM O AUXÍLIO DA FERRAMENTA DE SOFTWARE LIVRE MINISAT

Aluno: Mikail Campos Freitas
Orientador: Prof. Marcelo Finger

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

INTRODUÇÃO - SAT

- Problema da Satisfatibilidade Booleana
- Importância teórica e prática:
 - Primeiro problema NP-completo (Cook, 1971)
 - Planejamento em inteligência artificial
 - Projeto de circuitos integrados
 - Verificação de software
 - Entre outras...
- Evolução das soluções
- Estabelecidas competições de SAT (SAT Competition, SAT-Race, SAT Challenge)
- Soluções atuais são mais de 30 vezes mais rápidas do que em 2001!

INTRODUÇÃO - SAT

- Decisão sobre a satisfatibilidade de uma fórmula booleana proposicional
- Muitas vezes apresentado na forma normal conjuntiva (conjunção de disjunções)
- Ex:
Dada a fórmula $F = (p \vee \neg q) \wedge (q \vee \neg r) \wedge (\neg p \vee \neg q \vee \neg r) \wedge r$.
É possível satisfazer F ?

INTRODUÇÃO - PSAT

- Problema da Satisfatibilidade Probabilística
- Extensão do SAT, combina lógica e probabilidade
- Mostrado NP-completo (Georgakopoulos, Kavvadias, Papadimitriou, 1988)
- Conjunto de cláusulas proposicionais com probabilidade associada
- Decidir sobre a consistência da atribuição de probabilidades ao conjunto de cláusulas
- Aplicações:
 - Aprendizado automático
 - Linguística computacional
 - Modelagem lógico-probabilística
 - Entre outras...

INTRODUÇÃO - PSAT

- Ex: (Problema dos Bêbados)

Três amigos costumam ir a um bar, de modo que em todas as noites pelo menos dois deles estão em sua mesa de costume. Entretanto, cada um afirma ir ao bar “apenas” 60% das noites.

Eles estão falando a verdade?

INTRODUÇÃO - PSAT

■ Ex: (Problema dos Bêbados)

Três amigos costumam ir a um bar, de modo que em todas as noites pelo menos dois deles estão em sua mesa de costume. Entretanto, cada um afirma ir ao bar “apenas” 60% das noites.

Eles estão falando a verdade?

Formulação:

- $(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3)$
- $P(x_1) = P(x_2) = P(x_3) = 0.6$

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

PSAT - FORMA NORMAL ATÔMICA

- Instância PSAT $\Sigma = \{P(\alpha_i) \bowtie_i p_i \mid 1 \leq i \leq k\}$
 - α_i é fórmula sobre $\{x_1, \dots, x_n\}$, $\bowtie_i \in \{=, \leq, \geq\}$
- Uma instância Σ está na forma normal atômica quando pode ser particionada em (Γ, Ψ) onde:
 - $\Gamma = \{P(\alpha_i) = 1 \mid 1 \leq i \leq m\}$
 - $\Psi = \{P(y_i) = p_i \mid y_i \text{ é átomo}, 1 \leq i \leq k\}$

PSAT - FORMA NORMAL ATÔMICA

- Instância PSAT $\Sigma = \{P(\alpha_i) \bowtie_i p_i \mid 1 \leq i \leq k\}$
 - α_i é fórmula sobre $\{x_1, \dots, x_n\}$, $\bowtie_i \in \{=, \leq, \geq\}$
- Uma instância Σ está na forma normal atômica quando pode ser particionada em (Γ, Ψ) onde:
 - $\Gamma = \{P(\alpha_i) = 1 \mid 1 \leq i \leq m\}$
 - $\Psi = \{P(y_i) = p_i \mid y_i \text{ é átomo, } 1 \leq i \leq k\}$
- Ex: (bêbados)
 - $(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3)$
 - $P(x_1) = P(x_2) = P(x_3) = 0.6$

PSAT - FORMA NORMAL ATÔMICA

- Instância PSAT $\Sigma = \{P(\alpha_i) \bowtie_i p_i \mid 1 \leq i \leq k\}$
 - α_i é fórmula sobre $\{x_1, \dots, x_n\}$, $\bowtie_i \in \{=, \leq, \geq\}$
- Uma instância Σ está na forma normal atômica quando pode ser particionada em (Γ, Ψ) onde:
 - $\Gamma = \{P(\alpha_i) = 1 \mid 1 \leq i \leq m\}$
 - $\Psi = \{P(y_i) = p_i \mid y_i \text{ é átomo}, 1 \leq i \leq k\}$
- Ex: (bêbados)
 - $(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3) = (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3)$
 - $P(x_1) = P(x_2) = P(x_3) = 0.6$

PSAT - FORMA NORMAL ATÔMICA

- Instância PSAT $\Sigma = \{P(\alpha_i) \bowtie_i p_i \mid 1 \leq i \leq k\}$
 - α_i é fórmula sobre $\{x_1, \dots, x_n\}$, $\bowtie_i \in \{=, \leq, \geq\}$
- Uma instância Σ está na forma normal atômica quando pode ser particionada em (Γ, Ψ) onde:
 - $\Gamma = \{P(\alpha_i) = 1 \mid 1 \leq i \leq m\}$
 - $\Psi = \{P(y_i) = p_i \mid y_i \text{ é átomo}, 1 \leq i \leq k\}$
- Ex: (bêbados)
 - $(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3) = (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3)$
 - $P(x_1) = P(x_2) = P(x_3) = 0.6$

$\Gamma = \{(x_1 \vee x_2), (x_2 \vee x_3), (x_1 \vee x_3)\}$

$\Psi = \{P(x_1) = P(x_2) = P(x_3) = 0.6\}$

PSAT - FORMA NORMAL ATÔMICA

- Toda instância PSAT tem uma instância em Forma Normal Atômica equivalente
- Valoração v sobre $y_1 \dots y_k$ é Γ -consistente se podemos estender v para $y_1 \dots y_k, x_1 \dots x_n$ de modo que $\Gamma(v) = 1$; assim, o vetor $[1 \ v(y_1) \dots v(y_k)]^T$ também é Γ -consistente
- Lema:

Uma instância $\langle \Gamma, \Psi \rangle$ na forma normal atômica é satisfazível sse existe uma matriz A_Ψ de dimensão $k + 1 \times k + 1$ onde todas as suas colunas são Γ -consistentes e $A_\Psi \pi = p, \sum \pi = 1$ tem solução $\pi \geq 0$.

$$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix}, a_{i,j} \in \{0, 1\}$$

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

SOLUÇÕES

- Usando resolvidor SAT, Marcelo Finger e Glauber de Bona propuseram e implementaram três soluções:
 - Redução Canônica de PSAT para SAT
 - Geração de Colunas através de Redução de Turing para SAT
 - Geração de Colunas através de Redução de Turing para MAXSAT Ponderado

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

- $$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix}$$

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

$$\begin{aligned} \text{■} \quad & \begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad \left\{ \begin{array}{l} b \text{ bits} \\ \pi_i = 0.\pi_i^b \dots \pi_i^1 \\ p_i = 0.p_i^b \dots p_i^1 \end{array} \right. \end{aligned}$$

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

$$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad \begin{cases} b \text{ bits} \\ \pi_i = 0.\pi_i^b \dots \pi_i^1 \\ p_i = 0.p_i^b \dots p_i^1 \end{cases}$$

- Multiplicação $a_{i,j} \cdot \pi_j$: conjunção dos bits de π_j com $a_{i,j}$
- Soma: soma binária
- Igualdade: equivalência binária

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

$$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad \begin{cases} b \text{ bits} \\ \pi_i = 0.\pi_i^b \dots \pi_i^1 \\ p_i = 0.p_i^b \dots p_i^1 \end{cases}$$

- Multiplicação $a_{i,j} \cdot \pi_j$: conjunção dos bits de π_j com $a_{i,j}$
- Soma: soma binária
- Igualdade: equivalência binária
- Número de variáveis na tradução é $O(k^3 \log k)$

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

$$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad \begin{cases} b \text{ bits} \\ \pi_i = 0.\pi_i^b \dots \pi_i^1 \\ p_i = 0.p_i^b \dots p_i^1 \end{cases}$$

- Multiplicação $a_{i,j} \cdot \pi_j$: conjunção dos bits de π_j com $a_{i,j}$
- Soma: soma binária
- Igualdade: equivalência binária
- Número de variáveis na tradução é $O(k^3 \log k)$
- Resolvedor SAT é chamado

SOLUÇÕES - REDUÇÃO CANÔNICA

- Tradução através da codificação de operações binárias

$$\begin{bmatrix} 1 & \dots & 1 \\ a_{1,1} & \dots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \dots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix} \quad \begin{cases} b \text{ bits} \\ \pi_i = 0.\pi_i^b \dots \pi_i^1 \\ p_i = 0.p_i^b \dots p_i^1 \end{cases}$$

- Multiplicação $a_{i,j} \cdot \pi_j$: conjunção dos bits de π_j com $a_{i,j}$
- Soma: soma binária
- Igualdade: equivalência binária
- Número de variáveis na tradução é $O(k^3 \log k)$
- Resolvedor SAT é chamado
- Implementação: PSATtoSAT

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:

- $$\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$$

- Sempre tem solução

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:
 - $\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$
 - Sempre tem solução **Mas pode ser Γ -inconsistente!**

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:
 - $\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$
 - Sempre tem solução **Mas pode ser Γ -inconsistente!**
- $k + 1$ chamadas iniciais ao resolvedor SAT para verificar Γ -consistência das colunas e inicializar vetor de custo

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:
 - $\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$
 - Sempre tem solução **Mas pode ser Γ -inconsistente!**
- $k + 1$ chamadas iniciais ao resolvedor SAT para verificar Γ -consistência das colunas e inicializar vetor de custo
- Função objetivo: combina número de colunas Γ -inconsistentes com a soma das suas probabilidades associadas

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:
 - $\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$
 - Sempre tem solução **Mas pode ser Γ -inconsistente!**
- $k + 1$ chamadas iniciais ao resolvedor SAT para verificar Γ -consistência das colunas e inicializar vetor de custo
- Função objetivo: combina número de colunas Γ -inconsistentes com a soma das suas probabilidades associadas
- A cada passo é usado o resolvedor SAT para gerar uma coluna Γ -consistente com custo associado negativo

SOLUÇÕES - GER. DE COLUNAS: SAT

- Problema da forma $A\pi = p$
- Começar com um problema relaxado:
 - $\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \pi' = p \Rightarrow A_{(0)}\pi_{(0)} = p$
 - Sempre tem solução **Mas pode ser Γ -inconsistente!**
- $k + 1$ chamadas iniciais ao resolvedor SAT para verificar Γ -consistência das colunas e inicializar vetor de custo
- Função objetivo: combina número de colunas Γ -inconsistentes com a soma das suas probabilidades associadas
- A cada passo é usado o resolvedor SAT para gerar uma coluna Γ -consistente com custo associado negativo
- Implementação: PsatColGen

SOLUÇÕES - GER. DE COLUNAS: MAXSAT

- Semelhante ao método anterior
- Inicialização igual a antes (problema relaxado, resolvidor SAT para verificar Γ -consistência)

SOLUÇÕES - GER. DE COLUNAS: MAXSAT

- Semelhante ao método anterior
- Inicialização igual a antes (problema relaxado, resolvidor SAT para verificar Γ -consistência)
- Diferença:
 - Antes: gerar uma coluna com custo associado negativo

SOLUÇÕES - GER. DE COLUNAS: MAXSAT

- Semelhante ao método anterior
- Inicialização igual a antes (problema relaxado, resolvidor SAT para verificar Γ -consistência)
- Diferença:
 - Antes: gerar uma coluna com custo associado negativo
 - Agora: gerar a coluna com o menor custo associado

SOLUÇÕES - GER. DE COLUNAS: MAXSAT

- Semelhante ao método anterior
- Inicialização igual a antes (problema relaxado, resolvidor SAT para verificar Γ -consistência)
- Diferença:
 - Antes: gerar uma coluna com custo associado negativo
 - Agora: gerar a coluna com o menor custo associado
- A cada passo usar resolvidor MAXSAT Ponderado para gerar a coluna Γ -consistente com menor custo

SOLUÇÕES - GER. DE COLUNAS: MAXSAT

- Semelhante ao método anterior
- Inicialização igual a antes (problema relaxado, resolvidor SAT para verificar Γ -consistência)
- Diferença:
 - Antes: gerar uma coluna com custo associado negativo
 - Agora: gerar a coluna com o menor custo associado
- A cada passo usar resolvidor MAXSAT Ponderado para gerar a coluna Γ -consistente com menor custo
- Implementação: PSATtoMaxSat

SOLUÇÕES - RESOLVEDORES

- Baseados no uso de softwares externos:
 - zChaff
 - wmaxsatz2009
 - LA

SOLUÇÕES - RESOLVEDORES

- Baseados no uso de softwares externos:
 - zChaff (licença própria)
 - wmaxsatz2009 (GPLv3)
 - LA (GPLv3)

SOLUÇÕES - RESOLVEDORES

- Baseados no uso de softwares externos:
 - zChaff (licença própria)
 - Uso restrito a pesquisa, não comercial
 - wmaxsatz2009 (GPLv3)
 - LA (GPLv3)

SOLUÇÕES - RESOLVEDORES

- Baseados no uso de softwares externos:
 - zChaff (licença própria)
 - Uso restrito a pesquisa, não comercial
 - Distribuição apenas com o consenso da Universidade de Princeton
 - wmaxsatz2009 (GPLv3)
 - LA (GPLv3)

SOLUÇÕES - RESOLVEDORES

- Baseados no uso de softwares externos:
 - zChaff (licença própria)
 - Uso restrito a pesquisa, não comercial
 - Distribuição apenas com o consenso da Universidade de Princeton
 - wmaxsatz2009 (GPLv3)
 - LA (GPLv3)
- Interface e troca de dados feita por arquivos
- Overhead na criação de processos

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

MODIFICAÇÕES - OBJETIVOS

MODIFICAÇÕES - OBJETIVOS

- Melhorar o desempenho do resolvidores PSAT
- Facilitar a distribuição, uso e modificação do projeto

MODIFICAÇÕES - MINISAT

MODIFICAÇÕES - MINISAT

- Resolvedor SAT desenvolvido com o intuito de fácil integração a outros projetos
- Foi premiado em duas de quatro SAT Competition que participou e em todas as SAT-Race
- Licença MIT

MODIFICAÇÕES - MINISAT

- Resolvedor SAT desenvolvido com o intuito de fácil integração a outros projetos
- Foi premiado em duas de quatro SAT Competition que participou e em todas as SAT-Race
- Licença MIT **Compatível com GPLv3!**

MODIFICAÇÕES - PROPOSTA

MODIFICAÇÕES - PROPOSTA

- Trocar a utilização do zChaff pelo MiniSat
- Incluir o código de todos os softwares externos aos resolvedores

INTRODUÇÃO

PSAT

SOLUÇÕES

MODIFICAÇÕES

RESULTADOS

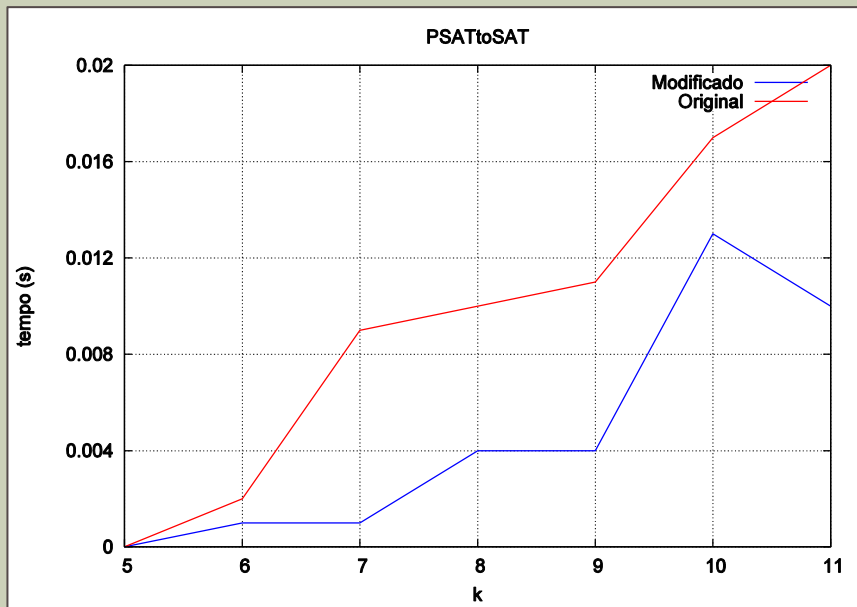
RESULTADOS - TESTES

- Instâncias satisfazíveis
 - Variar k (junto com n e m)
 - 50 instâncias para cada k

- Instâncias aleatórias
 - Fixar k e n , variar m (3-SAT)
 - 50 instâncias para cada valor de m

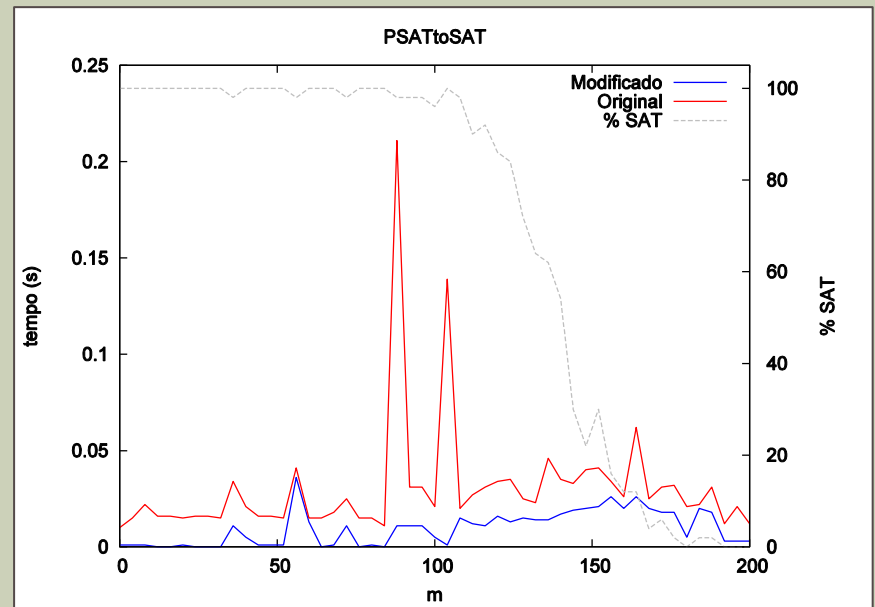
RESULTADOS - PSATtoSAT

Instâncias satisfazíveis



$$(n = 2k + 1)$$

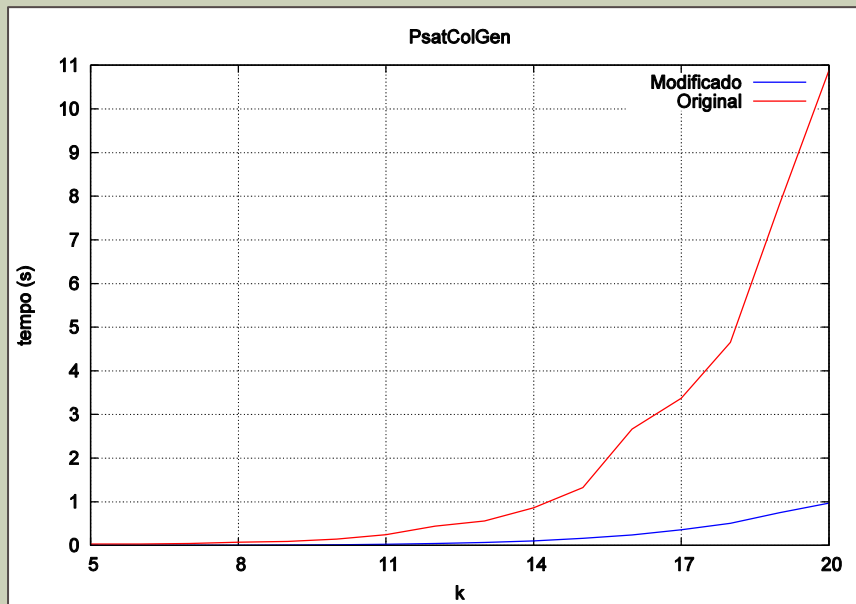
Instâncias aleatórias



$$(n = 40, k = 4)$$

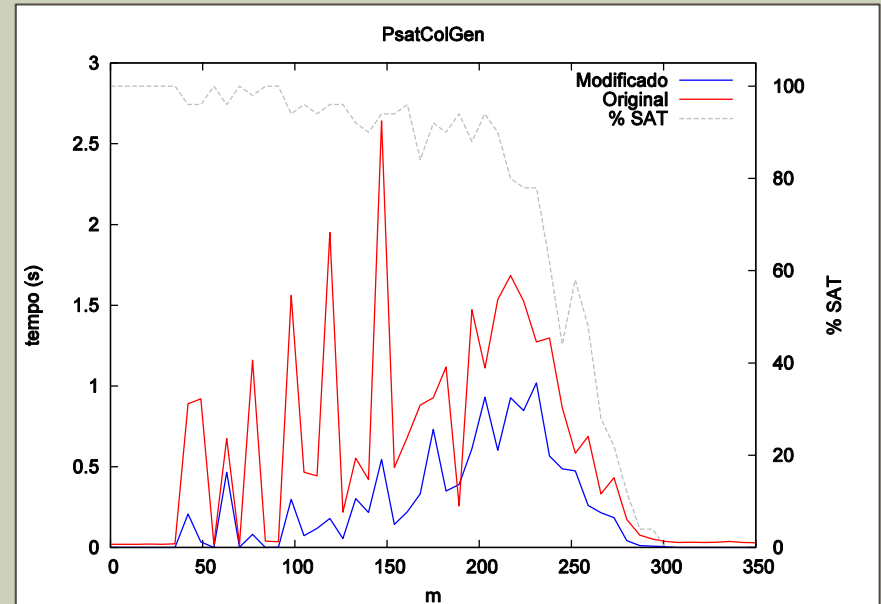
RESULTADOS - PsatColGen

Instâncias satisfazíveis



$$(n = 2k + 1)$$

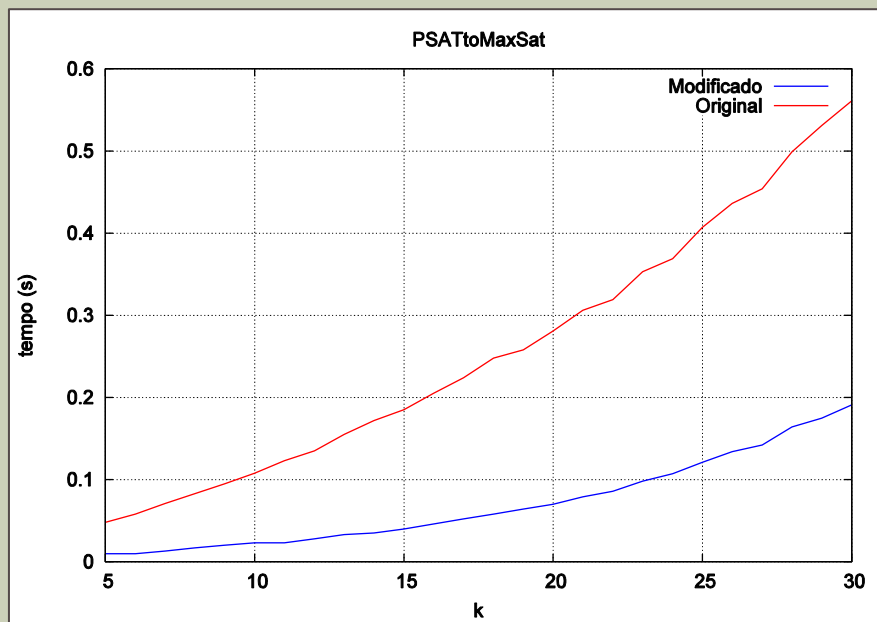
Instâncias aleatórias



$$(n = 70, k = 12)$$

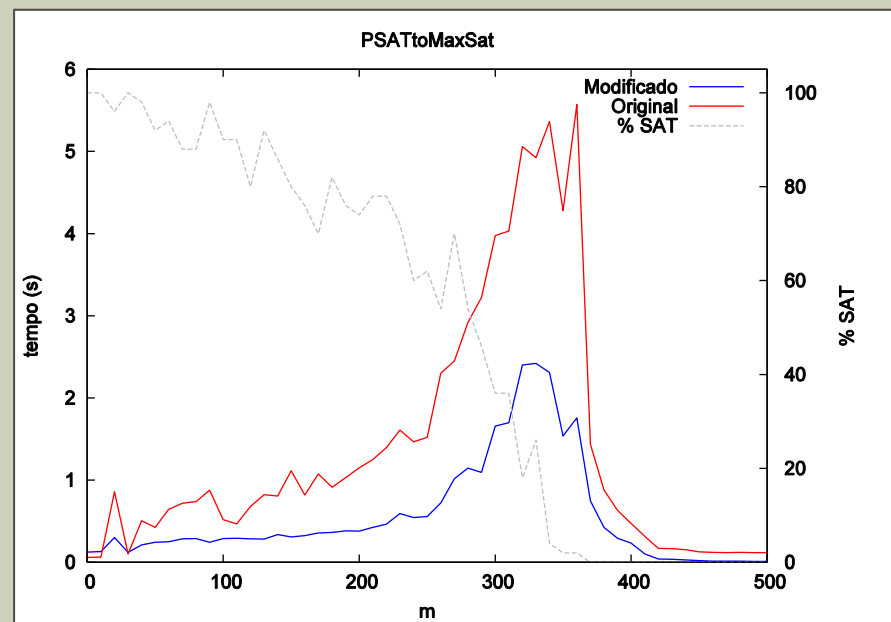
RESULTADOS - PSATtoMaxSat

Instâncias satisfazíveis



$$(n = 2k + 1)$$

Instâncias aleatórias



$$(n = 100, k = 20)$$

DÚVIDAS?



Obrigado!