

A eficiência do zChaff e a transição de fase

Eduardo Menezes de Moraes

13 de abril de 2010

Resumo

Neste trabalho foram analisados alguns casos de famílias de fórmulas proposicionais que podem ser resolvidas eficientemente por provadores de teoremas modernos, mais especificamente o zChaff da Princeton University.

Exploramos o efeito da *transição de fase* que afeta o tempo gasto por muitos provadores de teoremas onde fórmulas que possuem uma certa razão de cláusulas por variável consomem um tempo consideravelmente maior para serem resolvidas.

1 Introdução

O problema da satisfazibilidade proposicional (SAT) é muito importante para diversas áreas e aplicações. Parte da sua importância vem do fato desse problema ser *NP-Completo*. Isto significa que qualquer problema na classe de complexidade *NP* pode ser reduzido ao problema SAT. Isto também o torna muito difícil de ser resolvido eficientemente. Atualmente não existem algoritmos capazes de resolver eficientemente um problema *NP-Completo* e acredita-se que talvez esses problemas nunca poderão ser resolvidos em tempo polinomial.

Porém isso não significa que não se possa resolver certas instâncias de fórmulas proposicionais eficientemente. Por exemplo, as instâncias formadas apenas por cláusulas de Horn podem ser resolvidas rapidamente, assim como o subproblema 2-SAT (uma explicação mais detalhada para esse subproblema será apresentada na seção 2.2).

Porém, mesmo fórmulas de forma geral podem ser resolvidas eficientemente para algumas instâncias. Alguns critérios que podem indicar uma instância que pode ser resolvida eficientemente serão discutidas nesse trabalho.

2 Fundamentação teórica

Aqui apresentaremos rapidamente alguns conceitos básicos necessários para a compreensão deste trabalho.

2.1 A lógica proposicional

A lógica proposicional é um sistema formal onde fórmulas são formadas combinando *variáveis* e *conectivos lógicos*. As variáveis podem assumir dois valores, *verdadeiro*(1) ou

falso(0), e podem ser combinadas utilizando os conectivos binários de *conjunção* (“E”, símbolo \wedge), *disjunção* (“OU”, símbolo \vee), *implicação* (símbolo, \rightarrow) e o conectivo unário *negação* (“NÃO”, símbolo \neg).

Exemplo 1

$$(A \vee B) \wedge \neg C$$

é um exemplo de fórmula na lógica proposicional.

$$(Chove \wedge (Chove \rightarrow GramaMolhada)) \rightarrow GramaMolhada$$

é outro exemplo de fórmula.

Uma atribuição de valores *verdadeiro* ou *falso* às variáveis é chamada uma **interpretação** da fórmula. Como cada variável pode receber um valor verdadeiro ou falso fica claro que se temos n variáveis, temos 2^n possíveis interpretações.

Após atribuir valores às variáveis, podemos calcular se para aquela interpretação a fórmula é verdadeira segundo essas propriedades dos conectivos lógicos expostas nas tabela 1.

	\wedge		\rightarrow
$0 \wedge 0$	0	$0 \rightarrow 0$	1
$0 \wedge 1$	0	$0 \rightarrow 1$	1
$1 \wedge 0$	0	$1 \rightarrow 0$	0
$1 \wedge 1$	1	$1 \rightarrow 1$	1

	\vee		\neg
$0 \vee 0$	0	0	1
$0 \vee 1$	1	1	0
$1 \vee 0$	1		
$1 \vee 1$	1		

Tabela 1: Resultados dos conectivos lógicos

Interpretações que tornam a fórmula verdadeira, dizemos que *satisfazem* a fórmula. Dependendo de quantas interpretações satisfazem uma fórmula a classificamos pelo seguinte parâmetro:

- Se **todas** as interpretações satisfazem uma fórmula, ela é uma tautologia (ou teorema)
- Se **algumas** interpretações satisfazem uma fórmula, ela é satisfazível
- Se **nenhuma** interpretação satisfaz uma fórmula, ela é insatisfazível

Definição 2 Chamamos de **literal** uma variável (A) ou sua negação ($\neg A$).

Definição 3 Chamamos de **clausa** uma disjunção de literais (por exemplo $A \vee \neg B \vee C$).

Definição 4 Dizemos que uma fórmula está na Forma Normal Conjuntiva (CNF) se a fórmula é uma conjunção de clausas.

Proposição 5 Dada uma fórmula qualquer A da lógica proposicional existe uma outra fórmula B na CNF com clausas de no máximo três literais tal que B é satisfazível $\Leftrightarrow A$ é satisfazível.

2.2 O problema

O problema SAT consiste em: dado uma fórmula, decidir se essa fórmula é satisfazível. Este foi o primeiro problema provado ser NP-Completo [Cook 1971, Trakhtenbrot 1984]. Um algoritmo que resolve o problema SAT é chamado um provador de teorema.

Um caso especial deste problema é o k -SAT que coloca a restrição extra de que a fórmula deve estar na CNF e suas clausas devem ter na máximo k literais. O problema 2-SAT pode ser resolvido em tempo polinomial ([Aspvall, Plass e Tarjan 1979]). Devido a proposição 5, todo problema SAT pode ser reduzido a um problema 3-SAT, portanto, o problema 3-SAT também é NP-Completo.

2.3 A transição de fase

Conforme será visto na seção 4, fórmulas com poucas clausas e muitas variáveis tendem a ser satisfazíveis. Por outro lado, fórmulas com muitas clausas e poucas variáveis tendem a ser insatisfazíveis. A passagem de um extremo para outro é chamado *transição de fase*.

O critério normalmente considerado é a razão entre o número de clausas e o número de variáveis ($\frac{m}{n}$). Segundo [Gent e Walsh 1994] e outros, para os valores de $\frac{m}{n}$ na *transição de fase*, ou seja, para os valores de $\frac{m}{n}$ onde nem todas as fórmulas são satisfazíveis, nem insatisfazíveis, existe um grande aumento no tempo necessário para se resolver o problema SAT. É exatamente esse fenômeno que será estudado.

Podemos observar que alguns resultados em [Morais 2008] e [Dubois 1991] sugerem que nem todos os provadores de teoremas estão sujeitos à *transição de fase*. Porém os provadores de teoremas apontados têm uma performance para qualquer fórmula muito inferior ao zChaff e a outros provadores de teoremas modernos.

2.4 O zChaff

O zChaff é uma implementação do algoritmo Chaff descrito em [Moskewicz et al. 2001]. O zChaff é um dos provadores de teoremas mais rápidos de hoje em dia, usando para isso uma combinação do algoritmo Davis-Putnam-Logemann-Loveland com estruturas de dados otimizadas, além de técnicas de aprendizagem e heurísticas.

Mais informações sobre os detalhes do Chaff podem ser encontradas em [Moskewicz et al. 2001]

3 Metodologia

Buscamos analisar a *transição de fase* com os seguintes experimentos:

1. Utilizando uma grande quantidade de problemas gerados aleatoriamente, medir qual é o tempo gasto em média para resolvê-los e qual a percentagem que é satisfazível para cada valor da razão $\frac{m}{n}$ até 10, com incrementos de 0,1.
2. Para certos valores de $\frac{m}{n}$ fixos, variar o número de variáveis de 100 a 1000 e verificar como evolui o tempo gasto por problemas gerados aleatoriamente.

Ambos os experimentos foram repetidos utilizando fórmulas na forma 2-SAT e 3-SAT.

Para medição do tempo gasto por cada fórmula foi utilizado o comando `time` do Linux e utilizado o tempo de usuário. Esta medida mostra quanto tempo foi gasto pelo CPU para executar este processo descontando o tempo gasto pelo Kernel ou por mudanças de contexto.

Todos os experimentos foram realizados em um computador Intel Dual Core de 1.8GHz com 1Gb de memória e sistema operacional Linux 2.6.31, distribuição Ubuntu 9.10 de 32 bits.

Para gerar as fórmulas foi implementado um gerador de fórmulas aleatórias em Perl. Para evitar literais repetidos, todos os literais gerados foram armazenados em hashes e enquanto um sorteio fosse gerando um literal igual, se repetia o sorteio. Estratégia igual foi utilizada para evitar clausas repetidas. Tais cuidados são necessários para evitar fórmulas mais curtas do que parecem.

Os experimentos do item 1 foram repetidos 1000 vezes para cada valor de $\frac{m}{n}$. Os experimentos do item 2 foram repetidos 1000 vezes por valor de $\frac{m}{n}$, por número de variáveis, no caso do 2-SAT. No caso do 3-SAT isso se mostrou impossível pois para certos valores de $\frac{m}{n}$ o tempo necessário mostrou-se demasiado grande. Nesse caso fizemos apenas 10 amostras e colocamos um limite de no máximo uma hora por problema. Quando esse limite impedia que a maioria dos problemas fosse resolvida, o resultado era descartado.

Após os valores terem sido produzidos, utilizou-se o `gnuplot` para tentar aproximar os valores experimentais a funções matemáticas no caso do item 2.

4 Resultados

4.1 Experimento 1

Na figura 1 vemos o resultados do experimento 1 para fórmulas 3-SAT. Podemos ver claramente um aumento no tempo de resolução dos problemas para valores de $\frac{m}{n}$ entre 4 e 6. Este é também o intervalo em que a percentagem migra de 1 para 0, principalmente considerando a curva suavizada. O pico do tempo é encontrado entre 3 e 4, coincidindo com o valor onde a percentagem satisfazível está mais próxima de 0,5 (4,3 é esse valor de $\frac{m}{n}$ segundo [Gent e Walsh 1994]). Para valores de $\frac{m}{n}$ menores que 4 e maiores que 6 o tempo é aproximadamente constante, apesar de ser maior para valores de $\frac{m}{n}$ maiores que 6.

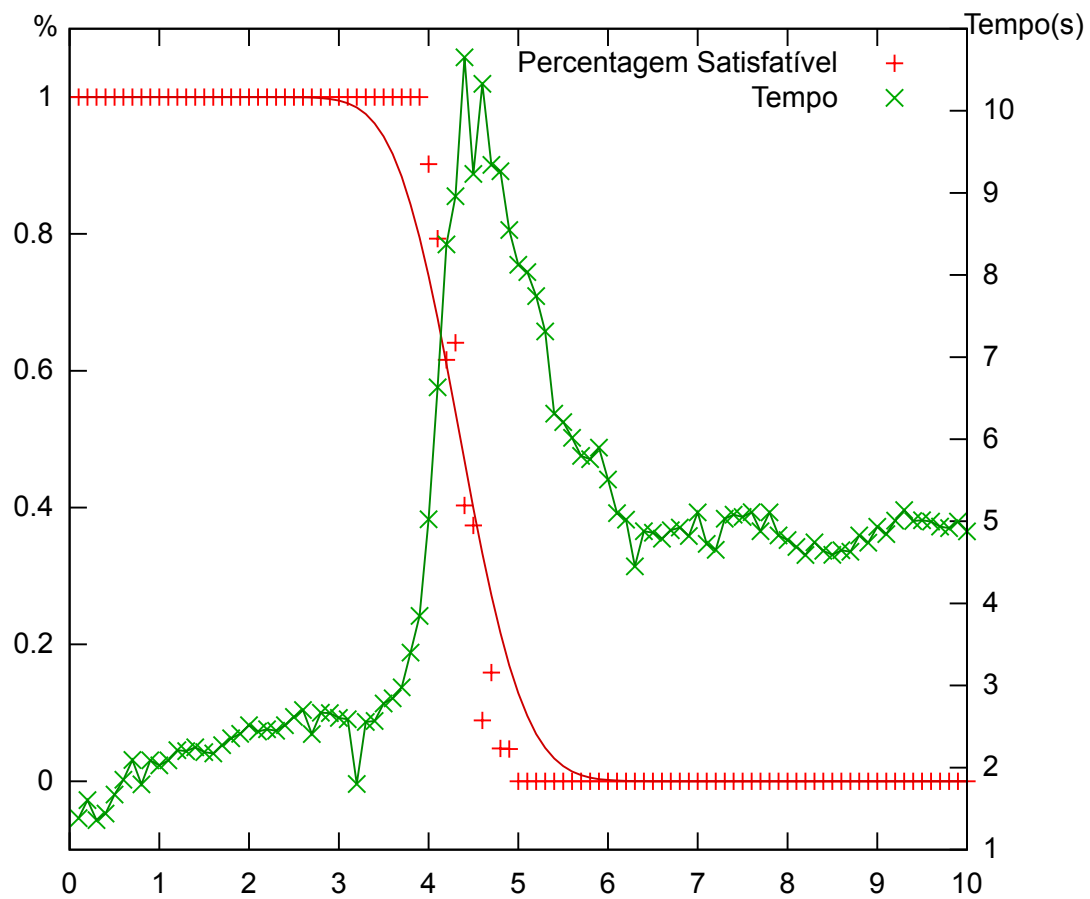


Figura 1: Experimento 1 para fórmulas 3-SAT

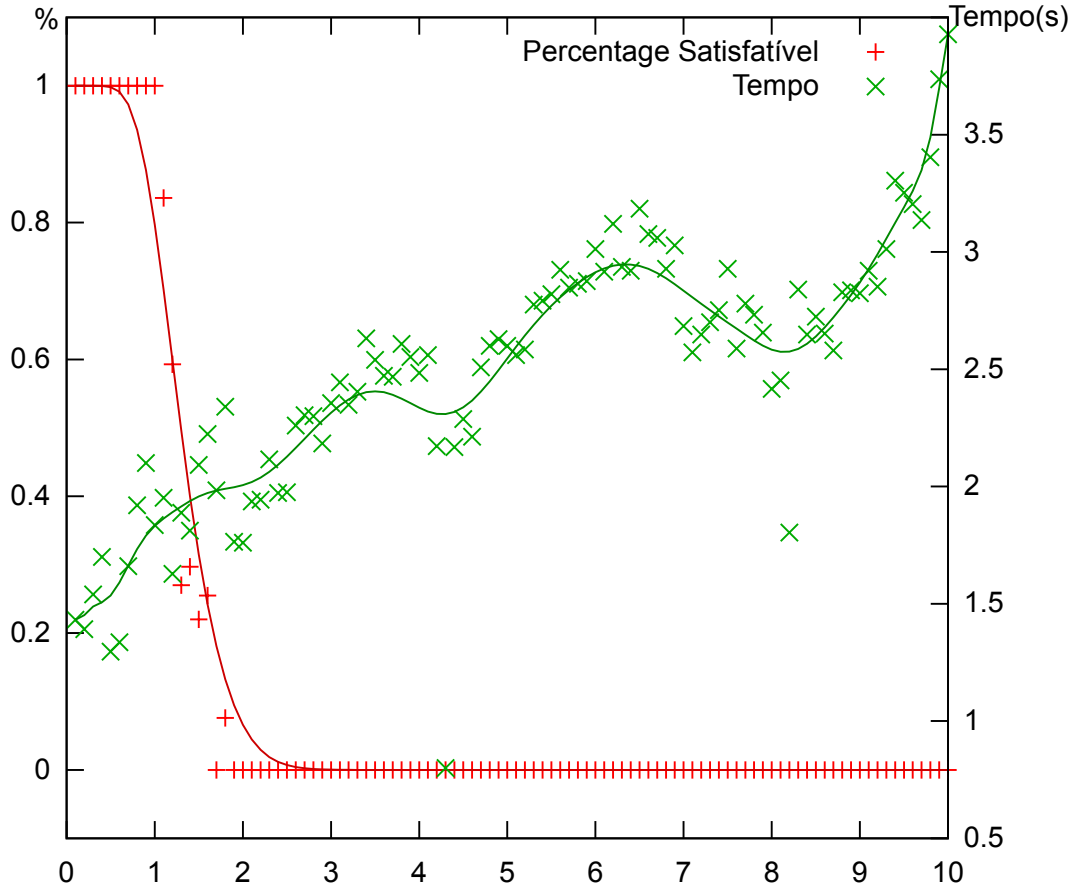


Figura 2: Experimento 1 para fórmulas 2-SAT

O gráfico 1 é um ótimo exemplo de gráfico mostrando o aumento de tempo na *transição de fase*.

Na figura 2 vemos o gráfico resultante do experimento 1 para fórmulas 2-SAT. A principal diferença que vemos em comparação com o gráfico 1 é que a transição de fase acontece muito mais cedo, com $\frac{m}{n}$ em valores entre 1 e 2. Isto está de acordo com resultados encontrados em [Wikipedia 2010]. Além disso, em [Gent e Walsh 1994] encontramos que a *transição de fase* no problema 4-SAT ocorre entre 8 e 16, sugerindo uma tendência à *transição de fase* ocorrer em valores maiores de $\frac{m}{n}$ conforme se aumenta o k de um problema k -SAT. Em relação ao tempo de resolução das instâncias, existe bastante flutuação, porém não existe um aumento significativo próximo à *transição de fase*. A tendência é ascendente, mas sem picos. De fato, o 2-SAT pode ser resolvido em tempo polinomial e o zChaff faz isso.

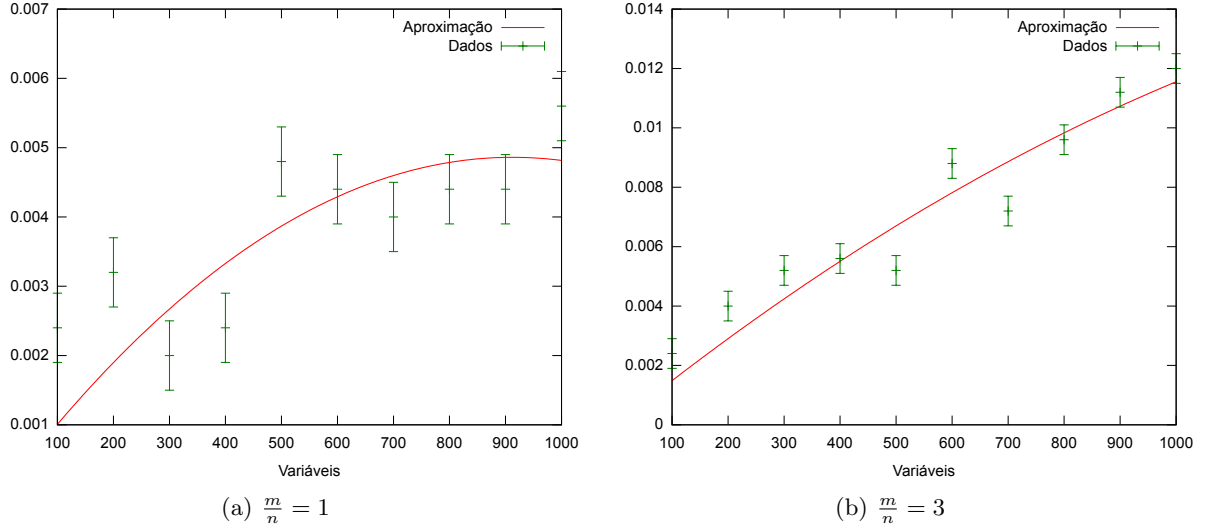


Figura 3: Experimento 2 para fórmulas 3-SAT

4.2 Experimento 2

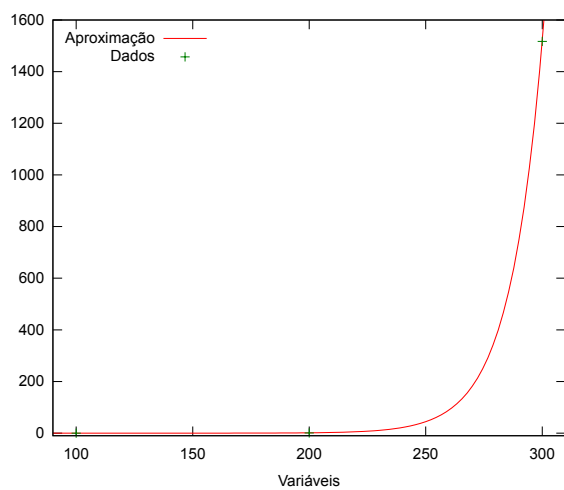
Na figura 3a temos os resultados de se variar o número de variáveis mantendo a razão $\frac{m}{n}$ fixo em 1. Conforme explicado na seção 3 para tornar possível a obtenção dos dados, foi necessário diminuir o número de repetições para apenas 10. Isso faz com que exista uma grande incerteza no resultado, principalmente para valores pequenos e próximos como estes. Perceba que o eixo das ordenadas deste gráfico possui uma escala muito pequena.

A melhor aproximação que conseguimos (que ainda assim é uma aproximação ruim) é o polinômio de segundo grau $ax^2 + bx + c$ com os coeficientes $a = -5,8415 \times 10^{-9}$, $b = 1,0657 \times 10^{-5}$ e $c = 0$. Portanto uma aproximação polinomial.

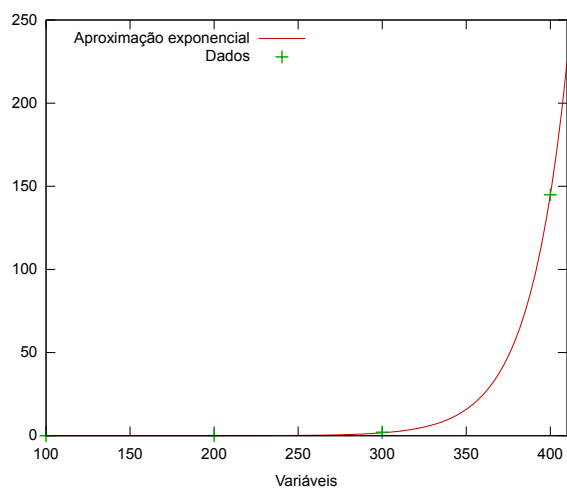
Na figura 3b, a razão $\frac{m}{n}$ é 3. A incerteza se torna proporcionalmente um pouco menor, mas ainda significativa. A melhor aproximação também é uma aproximação polinomial $ax^2 + bx + c$ com coeficientes $a = -3,6874 \times 10^{-9}$, $b = 1,5240 \times 10^{-5}$ e $c = 1,0040 \times 10^{-30}$ (mais próximo a uma função linear).

Já na figura 4a temos um gigantesco aumento nos valores do eixo das ordenadas, indo de 0,014 para 1600. As incertezas passam a ser desprezíveis. Perceba porém que o eixo das abscissas vai apenas até 300. Isso se deve ao fato do tempo para execução em valores maiores que 300 ter ultrapassado 1 hora em 9 das 10 execuções, fazendo com o que o resultado fosse descartado, conforme explicado na seção 3.

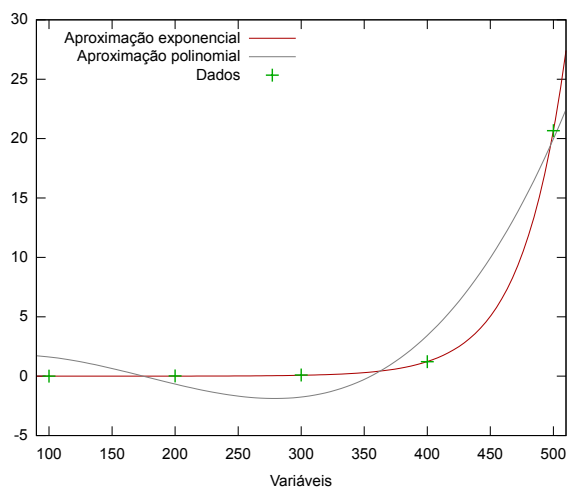
A curva que melhor se aproxima dos dados da figura 4a é uma curva exponencial da forma $a \times b^{(c \times x)}$ com coeficientes $a = 9,380 \times 10^{-7}$, $b = 1,098$ e $c = 0,7531$. Perceba que apesar de essa função exponencial ter uma base bem baixa (próxima de 1,1), ainda assim é uma função exponencial e para $x = 400$ atinge o valor $1,8 \times 10^6$, de fato indicando ser impossível a conclusão da resolução de fórmulas com esse tamanho em apenas 1 hora.



(a) $\frac{m}{n} = 4,3$



(b) $\frac{m}{n} = 6$



(c) $\frac{m}{n} = 8$

Figura 4: Experimento 2 para fórmulas 3-SAT (continuação)

$\frac{m}{n}$	Comportamento	Ordem
1	Polinomial	$-5,8415 \times 10^{-9}x^2$
3	Polinomial	$-3,6874 \times 10^{-9}x^2$
4,3	Exponencial	$1,098^x$
6	Exponencial	$1,060^x$
8	Exponencial	1.026^x

Tabela 2: Comportamento do 3-SAT

Estes dados nos indicam que os efeitos da *transição de fase* no tempo são ainda mais graves, pois aumentam o tempo necessário em um fator exponencial. De fato a posição 4,3 é exatamente o ápice da *transição de fase*.

Os dados mostrados em 4b repetem as características daqueles mostrados em 4a, porém mais atenuados: com $\frac{m}{n} = 6$ estamos próximos do fim da *transição de fase*. A melhor aproximação também é uma função exponencial de forma $a \times b^{(c \times x)}$ com coeficientes $a = 2,970 \times 10^{-6}$, $b = 1,060$ e $c = 0,7529$. Os coeficientes são bem próximos dos conseguidos na figura 4a, porém a mudança do b de 1,09 para 1,06 faz uma diferença realmente grande.

Era esperado que houvesse um crescimento exponencial pelo menos para alguns valores de $\frac{m}{n}$, caso contrário estaríamos resolvendo o problema SAT em tempo polinomial e teríamos que $P = NP$.

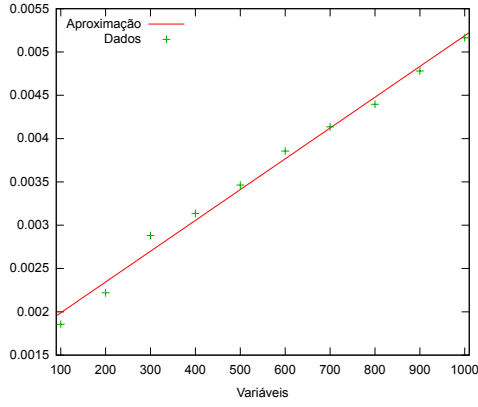
Porém mais surpreendente é o gráfico 4c. Com $\frac{m}{n} = 8$ estamos já longe da *transição de fase*, mas mesmo assim foi possível ir apenas até um número de variáveis igual a 500, com 600 variáveis estourando o limite de tempo para todas as instâncias.

No gráfico 4c temos duas aproximações possíveis, uma aproximação polinomial e uma aproximação exponencial. A aproximação polinomial é da forma $ax^3 + bx^2 + cx + d$ com coeficientes $a = 8.4825 \times 10^{-7}$, $b = -0.000449$, $c = 0.0526476$ e $d = 0$, enquanto a aproximação exponencial foi da forma $a \times b^{(c \times x)}$ com coeficientes $a = 1.5269 \times 10^{-5}$, $b = 1.026$ e $c = 1.10004$. A soma dos quadrados dos residuais da aproximação polinomial é 11,871 enquanto da aproximação exponencial é 0,00104, indicando que a aproximação exponencial é muito melhor que a polinomial.

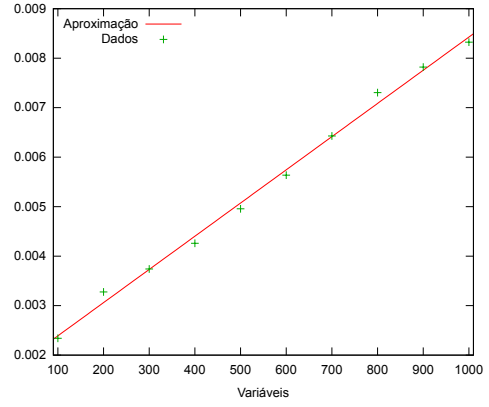
Isto indica que mesmo fora da *transição de fase*, para uma valor de $\frac{m}{n} = 8$ temos um crescimento exponencial do tempo necessário.

Podemos resumir o comportamento do 3-SAT nos diversos valores de $\frac{m}{n}$ na tabela 2.

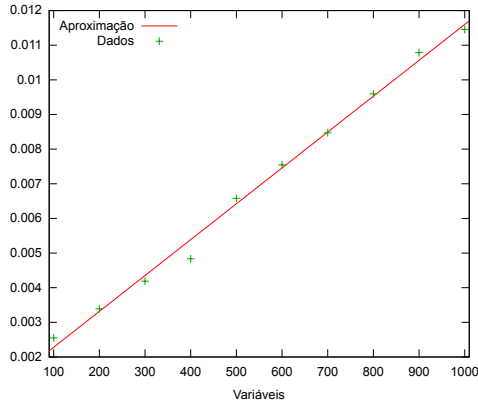
O experimento 2 para fórmulas 2-SAT pôde ser feito com mais precisão, utilizando 1000 repetições. Como pode ser visto na figura 5, todos demonstraram um comportamento muito próximo do linear, apesar do tempo em cada experimento aumentar constantemente, assim como os coeficientes das aproximações. As aproximações utilizadas nos gráficos da 5 são apresentados na tabela 3.



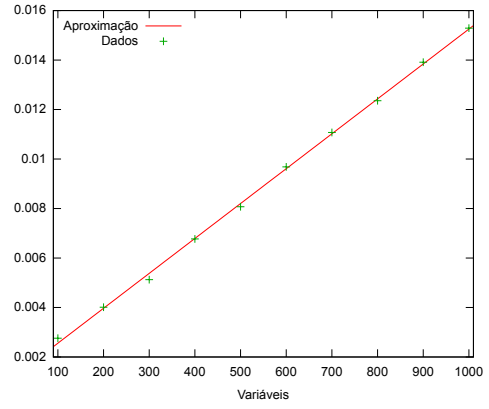
(a) $\frac{m}{n} = 1$



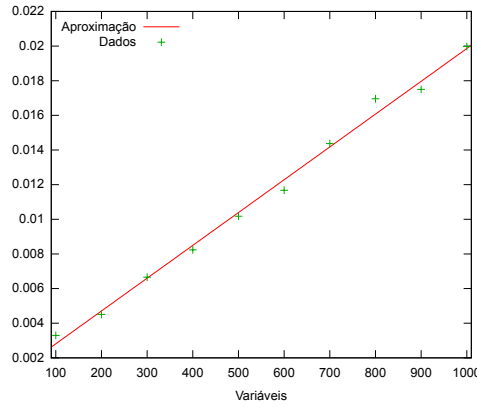
(b) $\frac{m}{n} = 3$



(c) $\frac{m}{n} = 4,3$



(d) $\frac{m}{n} = 6$



(e) $\frac{m}{n} = 8$

Figura 5: Experimento 2 para fórmulas 2-SAT

$\frac{m}{n}$	Comportamento	Aproximação
1	Linear	$3,555 \times 10^{-6}x + 0,0016$
3	Linear	$6,707 \times 10^{-6}x + 0,0017$
4,3	Linear	$1,035 \times 10^{-5}x + 0,0012$
6	Linear	$1,410 \times 10^{-5}x + 0,0011$
8	Linear	$1,893 \times 10^{-5}x + 0,0009$

Tabela 3: Comportamento do 2-SAT

5 Conclusão

Apesar do problema SAT ser um problema NP-Completo, é possível resolver eficientemente várias instâncias e a razão $\frac{m}{n}$ é um bom indicador se uma certa instância aleatória pode ser resolvida eficientemente.

Referências

- [Aspvall, Plass e Tarjan 1979]ASPVALL, B.; PLASS, M. F.; TARJAN, R. E. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, v. 8, n. 3, p. 121 – 123, 1979. ISSN 0020-0190. Disponível em: <<http://www.sciencedirect.com/science/article/B6V0F-45FCF92-2/2/5309e3f9271fa947b1af0f7ea9922c32>>.
- [Cook 2003]COOK, S. The importance of the p versus np question. *J. ACM*, ACM, New York, NY, USA, v. 50, n. 1, p. 27–29, 2003. ISSN 0004-5411.
- [Cook 1971]COOK, S. A. The complexity of theorem-proving procedures. In: *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1971. p. 151–158.
- [Dubois 1991]DUBOIS, O. Counting the number of solutions for instances of satisfiability. *Theor. Comput. Sci.*, v. 81, n. 1, p. 49–64, 1991.
- [Finger, Silva e Melo 2006]FINGER, M.; SILVA, F. S. C. da; MELO, A. C. V. de. *Lógica Para Computação*. [S.l.]: THOMSON LEARNING, 2006.
- [Gent e Walsh 1994]GENT, I. P.; WALSH, T. The sat phase transition. In: *Proceedings of 11th ECAI*. [S.l.: s.n.], 1994. p. 105–109.
- [Morais 2008]MORAIS, E. M. de. Provedores de teoremas baseados em contagem. 2008. Disponível em: <<http://www.ime.usp.br/~cef/mac499-08/monografias/rec/eduardo/monografia.pdf>>.
- [Moskewicz et al. 2001]MOSKEWICZ, M. W. et al. Chaff: Engineering an efficient sat solver. In: . [S.l.: s.n.], 2001. p. 530–535.

[Trakhtenbrot 1984]TRAKHTENBROT, B. A. A survey of russian approaches to perebor (brute-force searches) algorithms. *IEEE Ann. Hist. Comput.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 6, n. 4, p. 384–400, 1984. ISSN 1058-6180.

[Wikipedia 2008]WIKIPEDIA. *Boolean satisfiability problem* — *Wikipedia, The Free Encyclopedia*. 2008. [Online; acessado 16 de junho de 2008]. Disponível em: <http://en.wikipedia.org/wiki/Boolean_satisfiability_problem>.

[Wikipedia 2010]WIKIPEDIA. *2-satisfiability* — *Wikipedia, The Free Encyclopedia*. 2010. [Online; accessed 12-April-2010]. Disponível em: <<http://en.wikipedia.org/w/index.php?title=2-satisfiability&oldid=355578833>>.