Universidade de São Paulo
Instituto de Matemática e Estatística
MAC 5789 - Laboratório de Inteligência Artificial

Exercício Programa 1: MaxSAT

Autor:

Walter Perez Urcia

São Paulo

Abril 2015

Resumo

Neste trabalho o objetivo foi explorar o problema Partial Weighted Max-SAT com diferentes parámetros e comparar seus resultados, para isso foi usado solvers oficiais de [1]. A primeira parte do trabalho consiste em fazer um gerador de dados com o padrão oficial wcnf (weighted conjunction normal form). As seguintes partes são experimentos feitos para o tamanho de clausulas dois e tres variando seus parámetros como o número de clausulas, tamanho de clausulas e número de átomos. A continuação presentamos os experimentos, resultados e conclusões.

Sumário

Lista de Figuras

Lista de Tabelas

1 Problema

1.1 Definições previas

Para entender o problema, é necessário primero definir algumas coisas como:

Átomo Variável lógica, ou seja que só pode ter como possíveis valores a Verdadeiro e Falso. O número total de átomos para este trabalho será representado com N.

Cláusula Conjunto de átomos numa disjunção. São da seguinte forma: $x_1 \vee x_2 \vee \ldots \vee x_K$, onde K é o tamanho da cláusula. O número total de cláusulas para este trabalho será representado com M.

Satisfiability Um átomo p é satisfasível se $v(p) = 1$ (verdadeiro). Uma cláusula só vai ser satisfasível se pelo menos menos um de seus átomos é satisfasível [5].

1.2 SAT: Satisfiability problem

Dado um conjunto de cláusulas (M cláusulas), determinar se existe uma valoração dos átomos que faça possível que a conjunção de todas as cláusulas, ou seja $C_1 \wedge C_2 \wedge \ldots C_M$, seja verdadeiro [3].

1.3 Max-SAT: Maximum Satisfiability problem

Dado um conjunto de cláusulas, determinar o conjunto de tamanho máximo (maior número de cláusulas) que seja satisfasível com alguma valoração dos átomos [3].

1.4 Partial Max-SAT

Neste problema é um caso especial de Max-SAT onde temos um conjunto de cláusulas que é obrigatório conhecidas como **cláusulas hard** e os outras cláusulas como **cláusulas soft**. Da mesma forma que o anterior problema, tem que determina-se o conjunto de tamanho máximo tendo em consideração o conjunto de cláusulas hard.

1.5 Weighted Max-SAT

Neste caso especial de Max-SAT cada cláusula tem um peso w_i e deve determina-se o conjunto de peso máximo que pode ser satisfasível com alguma valoração dos atomos.

1.6 Partial Weighted Max-SAT

É uma combinação dos dois problemas anteriores onde cada cláusula tem um peso e também se tem um conjunto de cláusulas hard (com peso máximo) que são obrigatórias e um conjunto de cláusulas soft. Se deve determinar o conjunto de peso máximo tendo en consideração as cláusulas hard [3]. Este é o problema com o que se vai experimentar neste trabalho.

2 Dados de entrada

Implementar um gerador aleatório de problemas no formato padrão cnf, com N átomos e M cláusulas, e as cláusulas todas possuem K-literais, com pesos entre 1 e o número total de cláusulas M. K, N e M são parâmetros a serem passados ao gerador. Esta implementação pode ser feita em qualquer linguagem de programação para rodar em qualquer sistema operacional. Não entregue a implementacão, apenas descreva seus pontos mais importantes, por exemplo, como lida com repetição de literais e cláusulas.

2.1 Restrições

No archivo de regras dado em [2] se explica o padrão wcnf da seguinte forma:
c
c Comments
c
p wcnf N M W_{max}
w_1 a_1 a_2 $\ldots a_K$
w_2 b_1 b_2 $\ldots b_K$
.
.
.
w_M z_1 z_2 $\ldots z_K$
Las linhas que começam com c são tomadas como comentarios e não adicionam informação numérica para a resolução do problema. A seguinte linha contem os valores de N (número de átomos), M (número de cláusulas) e W_{max} que é o máximo peso das cláusulas, que vai ser também o peso das cláusulas hard. Por útlimo tem M linhas descrevendo cada cláusula, primeiro tem seu peso e logo seus respetivos literais (numerados de 1 a N). Em caso seja necessário colocar alguma negação tão só tem que colocá-se o número negativo do identificador do átomo (ou seja, de -1 a -N). Além, cada cláusula vai ter o mesmo número de literais K em cada geração de dados. Em quanto as restrições numéricas temos:

- Os pesos das cláusulas tem que ser maiores ou iguais a 1
- A suma dos pesos das cláusulas soft tem que ser menor a 2^{63}
- Cláusulas hard tem peso W_{max} e cláusulas soft peso menor a W_{max}
- W_{max} sempre é maior que a suma dos pesos das cláusulas soft

2.2 Gerador aleatório de dados

O gerador de dados está escrito na linguagem de programação Python e recebe 4 parâmetros: N, M, K e o nome para o archivo gerado. Ao momento de fazer que os dados sejam gerados totalmente aleatórios surgem dois problemas e se tem que garantir que:

1. Literais numa mesma cláusula tem que ser diferentes, ou seja uma cláusula C não deve ser da seguinte forma: $a \vee b \vee \ldots b \vee \ldots z$
2. Não devem existir cláusulas iguais (considerando só seus literais e não seu peso)
3. Deve gerar cláusulas hard e cláusulas soft

A função principal que satisfaz ambas condições é a seguinte:

```
1  def generateClause( self ) :
2    clause = []
3    ishard = randint( 0 , 1 )
4    weight = ( self.M * self.M if ishard == 1 else randint( 1 , self.M ) )
5    self.top = max( self.top , weight )
6    while True :
7      clause = []
8      currentLiterals = {}
9      for p in range( self.K ) :
10       while True :
11         sign = randint( 0 , 1 )
12         atom = randint( 1 , self.N )
13         atom = ( 1 if sign == 0 else -1 ) * atom
14         if atom in currentLiterals : continue
15         else : break
16       currentLiterals [ atom ] = True
17       clause . append( atom )
18     if self . hashcode( clause ) in self . currentClauses : continue
19     else : break
20   self .addClause( clause )
21   clause . insert ( 0 , weight )
22   return clause
```

Para satisfazer a primeira condição, o gerador tem um dicionário *currentLiterals* que armazena os literais que existem nessa cláusula até esse momento. De esta forma, na linha 14 verifica se o novo literal gerado já existe nesse dicionário e o armazena se não existe, caso contrario gera outro literal. Da mesma forma para satisfazer a segunda condição, o gerador tem um dicionário *currentClauses* que armazena os valores hash das cláusulas geradas até esse momento, e na linha 18 verifica se a novo valor já existe ou não. Mas o principal problema com a verificação de existência das cláusulas geradas é que depende K e M, então para reduzir o tempo de execução neste passo, cada cláusula foi convertida a seu valor hash. Para esto se tem os seguintes parâmetros:

- Aos literais na cláusula foi adicionado N para fazer positivos aqueles com valores negativos
- A base para o valor hash é $B = 2N$ porque se está adicionando N a cada literal

- Para evitar transbordamento nos tipos de dados cada valor hash foi calculado módulo 1000000007 ($10^9 + 7$)
- O valor hash para uma cláusula C será da seguinte forma: $hash(C) = (a_1 * B^{K-1} + a_2 * B^{K-2} + \ldots + a_K * B^0) \mod 1000000007$

Fazendo essa conversão às cláusulas o tempo de execução já não depende de K porque somente estão sendo comparados números. Por último, para satisfazer a terceira condição, é gerado aleatoriamente um valor 0 ou 1 que diz se uma cláusula vai ser hard ou não (soft em caso seja 0, hard em caso seja 1). Além, tendo em conta as restricões em 2.1, as cláusulas soft têm peso entre 1 e M, mas as cláusulas hard têm peso M^2 porque o peso de um cláusula hard sempre deve ter um valor maior que a suma das cláusulas soft que como máximo poderiam ser $M - 1$ cláusulas.

3 Experimentos e resultados

Para poder fazer cada experimento foi utilizado o programa toysat [4] e um script para ter varios threads em paralelo tão para gerar os datos como para executar toysat solver. Além, os tempos mostrados nos gráficos são os tempos promedios de execução para cada configuração de parâmetros.

3.1 Experimento 1

3.1.1 Descrição

Para $K = 3$ (ou seja, 3-SAT) e $N = 100$, levantar a curva de resposta de tempo, e apresentá-la sobreposta à curva de percentagem de problemas satisfazíveis. Cada ponto deve ser obtido a partir de pelo menos 100 instâncias geradas aleatoriamente. Apresentar e discutir o formato do gráfico.

3.1.2 Resultados

A figura 1 mostra que para valores menores a 430 cláusulas (relação $M/N = 4.3$) o tempo de execução é menor a 1 segundo, mas para valores de M a partir de 430 o tempo crece muito, levando quase 10 minutos em promedio seu execução para 600 cláusulas. Em quanto ao percentagem de instâncias satisfazíveis é 100% para instâncias com 400 cláusulas ou menos. Portanto, a mais cláusulas, menos percentagem de instâncias satisfazíveis.

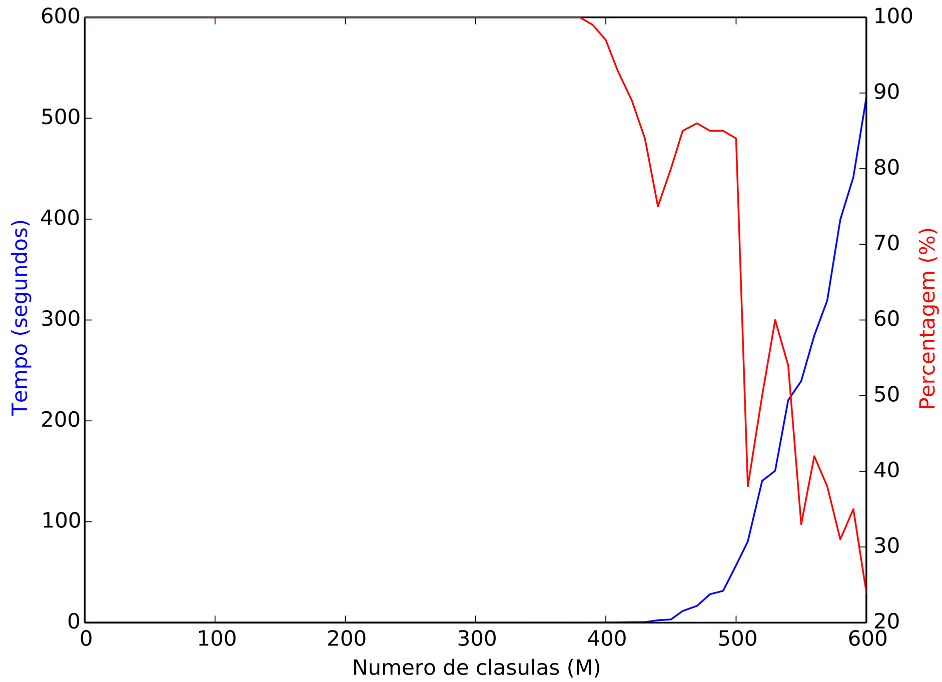

Figura 1: Curva de resposta de tempo (azul), curva de percentagem (vermelho)

3.2 Experimento 2

3.2.1 Descrição

Para $K = 2$ (ou seja, 2-SAT) e $N = 100$, levantar a curva de resposta de tempo, e apresentá-la sobreposta à curva de percentagem de problemas satisfazíveis. Cada ponto deve ser obtido a partir de pelo menos 100 instâncias geradas aleatoriamente. Apresentar e discutir o formato do gráfico.

3.2.2 Resultados

Da mesma forma que o gráfico em 3.1.2, a figura 2 mostra os tempos promedio de resposta e o percentagem de instâncias satisfazíveis, mas esta vez tem as instâncias são K-literais com $K = 2$. Neste caso o percentagem de instâncias satisfazíveis diminui rápidamente ainda para valores M/N muito próximo de 1. Por outro lado, no gráfico mostra que para valores entre 200 e 300 cláusulas o tempo promedio de resposta é consideravelmente mais grande que para outros valores em que seu tempo é muito próximo a 0.

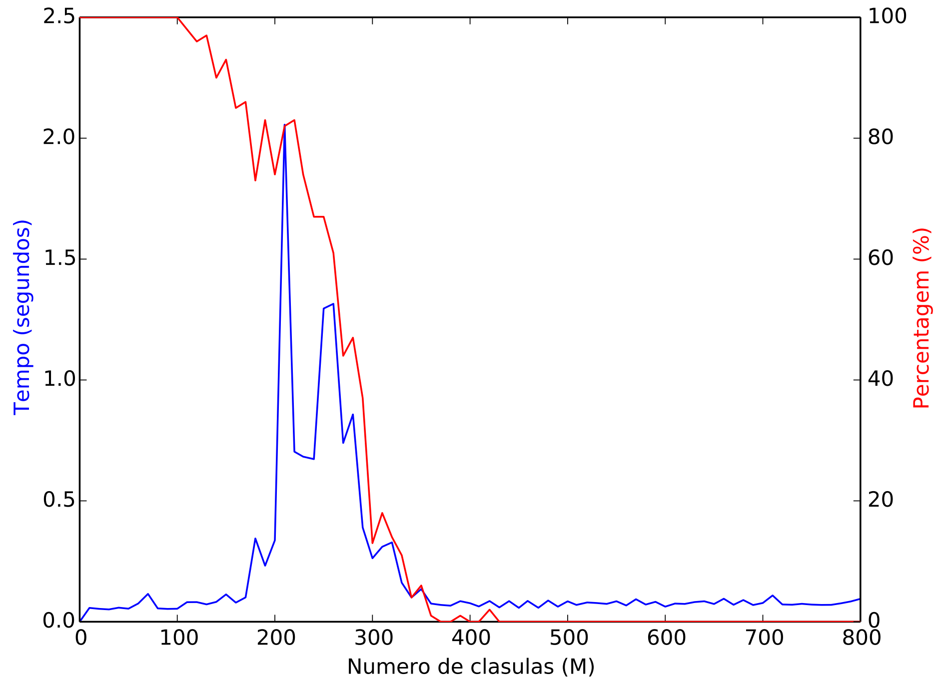

Figura 2: Curva de resposta de tempo (azul), curva de percentagem (vermelho)

3.3 Experimento 3

3.3.1 Descrição

Apresentar 5 gráficos mostrando o tempo de execução em função de N para $K = 3$. Em cada gráfico, o valor de M/N deve ser fixo. Os cinco gráficos devem ser feitos para N variando de 100 a 1000, em intervalos de 100. Os valores de M/N de cada um dos 5 gráficos são 1; 3; 4,3; 6; e 8. Discutir a natureza da curva obtida em cada caso, se polinomial ou exponencial.

3.3.2 Resultados

Na tabela 1 mostra os tempos promedio em segundos de cada configuração de parâmetros. Para as relações $M/N = 1.0$ e $M/N = 3.0$ os tempos são menores a 1 segundo, mas para $M/N = 4.3$ os tempos são consideravelmente mais grandes levando pouco mais de uma hora para $N = 1000$. Além, para $M/N = 6.0$ os tempos são muito grandes levando mais de um dia para seu execução com $N = 600$ e sem calcular para valores de N maiores. Porém, para $M/N = 8.0$ os tempos não aumentam muito apesar de ter muitas cláusulas, e como se mostrou em 3.1.2 não tem muitas instâncias satisfazíveis, portanto pode conclui-se que esta relação não é muito dependente de N porque na maioria das vezes as instâncias são insatisfazíveis.

N	$M/N = 1.0$	$M/N = 3.0$	$M/N = 4.3$	$M/N = 6.0$	$M/N = 8.0$
100	0.042	0.050	10.278	6327.574	100.724
200	0.092	0.137	58.735	7732.361	224.791
300	0.071	0.108	116.031	15253.935	384.112
400	0.131	0.322	230.58	40823.234	681.084
500	0.178	0.434	512.05	91834.454	800.268
600	0.196	0.422	741.912	131082.805	811.481
700	0.168	0.668	998.048	-	811.101
800	0.188	0.763	1806.502	-	842.849
900	0.296	0.765	2501.011	-	895.298
1000	0.199	0.717	4700.17	-	874.026

Tabela 1: Comparação de tempos (segundos) de Max 3-SAT

A figura 3 mostra os tempos para $M/N = 1.0$. Esta curva tem um comportamento polinomial, quase lineal, porque os tempos aumentan proporcionalmente a N ainda para $N = 1000$.

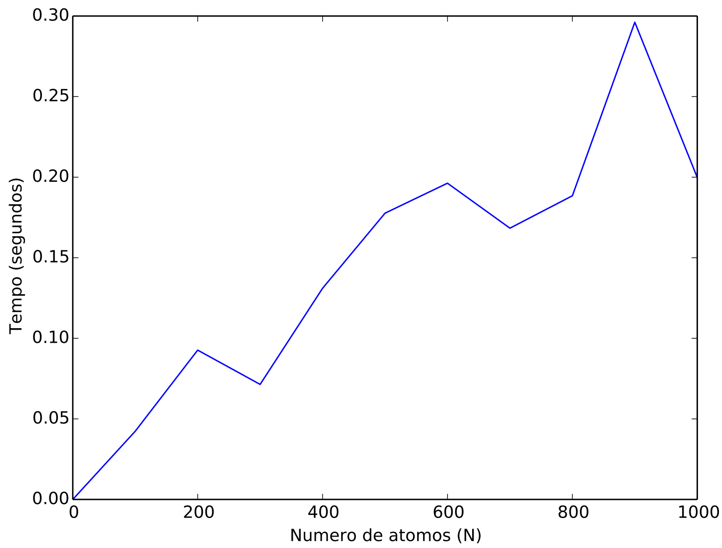

Figura 3: Curva de resposta de tempo para $M/N = 1.0$

Da mesma forma que a figura anterior, a figura 4 mostra uma curva com comportamento polinomial, mas com tempos maiores.

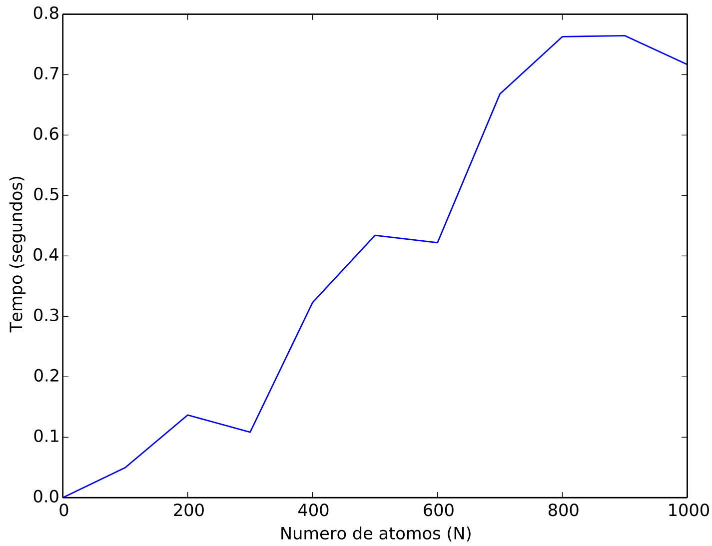

Figura 4: Curva de resposta de tempo para $M/N = 3.0$

Para $M/N = 4.3$ a curva de resposta de tempo tem um comportamento exponencial como mostra a figura 5 com tempos muito grandes de mais de uma hora (para $N = 1000$). Os aumentos de tempo são maiores enquanto o valor N continúa aumentando.

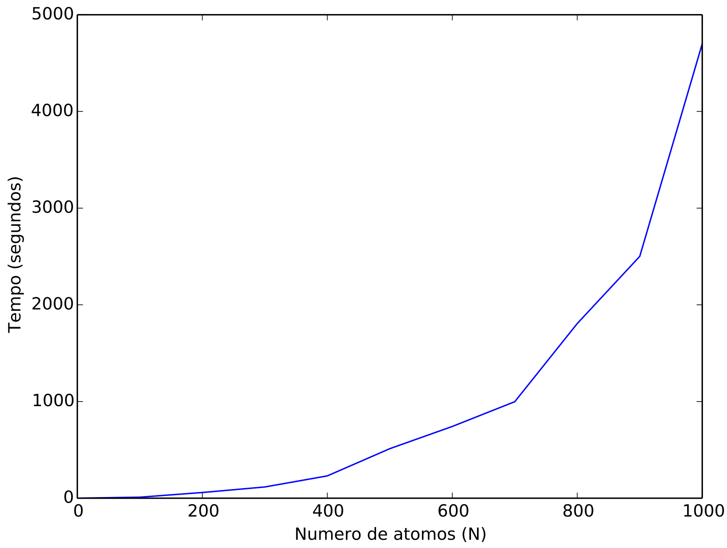

Figura 5: Curva de resposta de tempo para $M/N = 4.3$

Tão como mostrava a figura 5, na curva mostrada na figura 6 tem um comportamento exponencial com tempos de mais de um dia para $N = 600$.

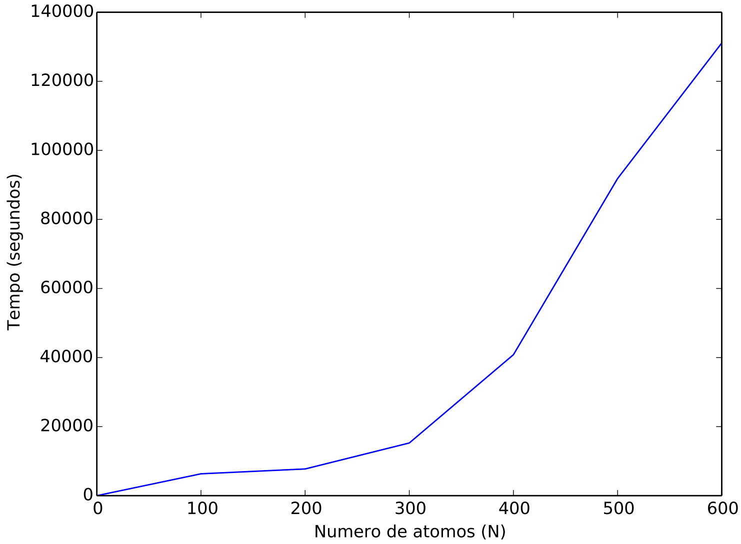

Figura 6: Curva de resposta de tempo para $M/N = 6.0$

Por último, a figura 7 mostra novamente uma curva de comportamento polinomial que já foi explicado ao inicio desta seção.

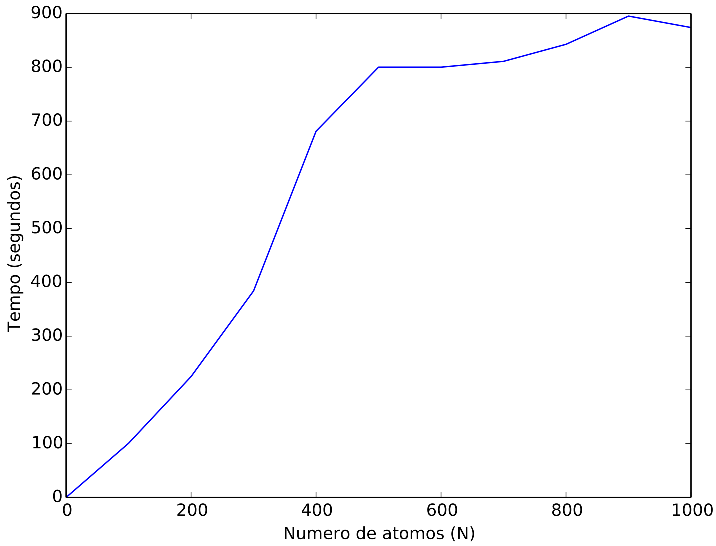

Figura 7: Curva de resposta de tempo para $M/N = 8.0$

3.4 Experimento 4

3.4.1 Descrição

Apresentar 5 gráficos mostrando o tempo de execução em função de N para $K = 2$. Em cada gráfico, o valor de M/N deve ser fixo. Os cinco gráficos devem ser feitos para N variando de 100 a 1000, em intervalos de 100. Os valores de M/N de cada um dos 5 gráficos são 1; 3; 4,3; 6; e 8. Discutir a natureza da curva obtida em cada caso, se polinomial ou exponencial.

3.4.2 Resultados

Na tabela 2 tem a mesma informação que a tabela 1 mas para instâncias com $K = 2$. Em geral, enquanto a relação M/N aumenta os tempos vão aumentando também. Em $M/N = 3.0$ os tempos são menores a 1 segundo, excepto para $N = 200$ que pode deve-se a instâncias muito complicadas para o programa, esto é possível porque a geração de dados é totalmente aleatória. Além, pode-se notar que as diferencias entre os tempos para as diferentes relações não são muito diferente numa de outra, sendo apenas 2 segundos para $M/N = 8.0$ e $N = 1000$.

N	$M/N = 1.0$	$M/N = 3.0$	$M/N = 4.3$	$M/N = 6.0$	$M/N = 8.0$
100	0.040	0.259	0.057	0.063	0.075
200	0.048	18.241	0.078	0.099	0.136
300	0.052	0.076	0.111	0.152	0.218
400	0.062	0.128	0.159	0.221	0.354
500	0.060	0.115	0.193	0.337	0.615
600	0.076	0.188	0.310	0.446	0.700
700	0.073	0.191	0.382	0.659	0.982
800	0.073	0.217	0.398	0.890	1.184
900	0.0790	0.272	0.551	1.128	2.731
1000	0.286	0.359	0.582	1.535	2.275

Tabela 2: Comparação de tempos (segundos) de Max 2-SAT

A figura 8 mostra os tempos para $M/N = 1.0$. Esta curva tem um comportamento polinomial e com aumentos de tempo proporcionais a N.

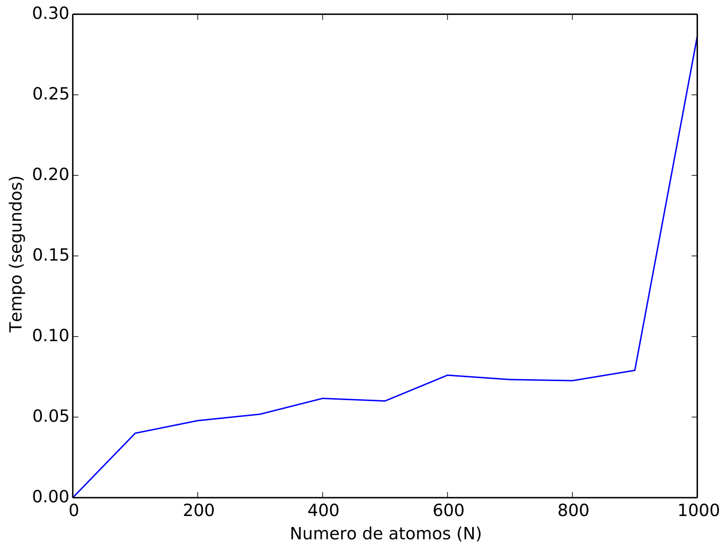

Figura 8: Curva de resposta de tempo para $M/N = 1.0$

Em geral a figura a figura 9 mostra uma curva de comportamento polinomial, mas com uma anormalidade em $N = 200$ que já foi explicado anteriormente.

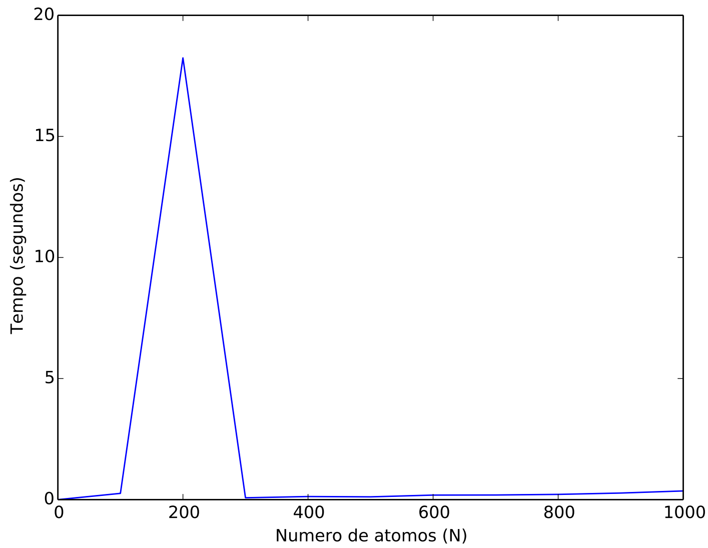

Figura 9: Curva de resposta de tempo para $M/N = 3.0$

Para $M/N = 4.3$ a curva mostrada na figura 10 tem um comportamento polinomial, além já começa a ser quase lineal em N a diferencia das figuras anteriores.

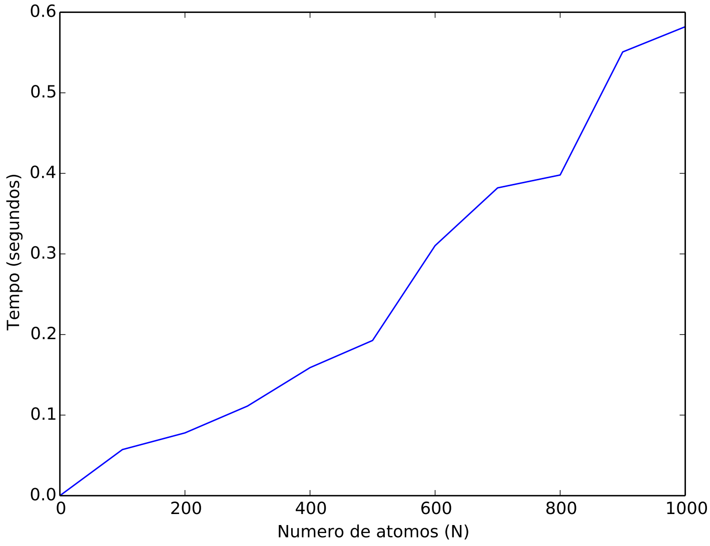

Figura 10: Curva de resposta de tempo para $M/N = 4.3$

A figura 11 já tem uma curva de comportamento exponencial depois de que a figura anterior mostra-se que a curva já estava sendo quase lineal.

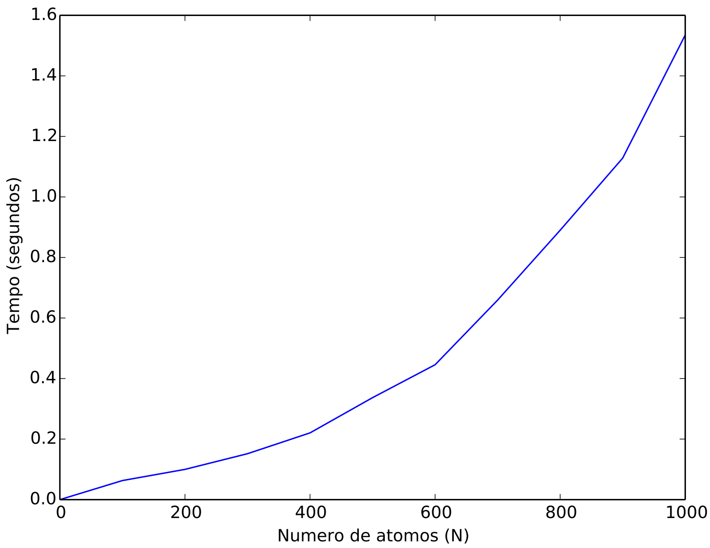

Figura 11: Curva de resposta de tempo para $M/N = 6.0$

Por último, a figura 12 continúa mostrando uma curva exponencial para $M/N =$ 8.0.

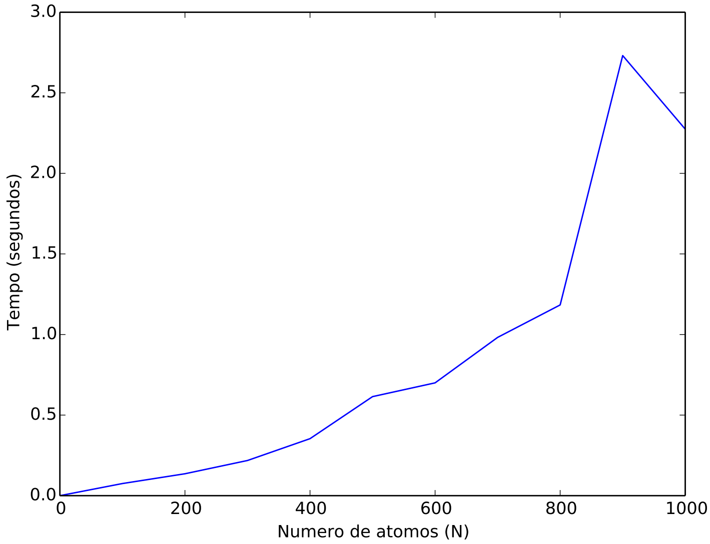

Figura 12: Curva de resposta de tempo para $M/N = 8.0$

4 Conclusões

Pode-se concluir que:

- O valor de K é muito importante para a resolução do problema como pode ser visto nos tempos dos experimentos 1 e 2. Um maior valor de K adiciona complexidade ao problema.
- O intervalo em que Max 3-SAT tem comportamento exponencial é quando $M \geq 430$ e $M \leq 600$ para $N = 100$, ou seja com $M/N \geq 4.3$ e $M/N \leq 6.0$.
- O intervalo em que Max 2-SAT tem comportamento exponencial é quando $M \geq 200$ e $M \leq 300$ para $N = 100$, ou seja com $M/N \geq 2.0$ e $M/N \leq 3.0$.
- Enquanto o valor de M/N é mais grande, é mais provável que o percentagem de instâncias satisfazíveis seja menor.

Referências

[1] 17th International Conference on Theory and Applications of Satisfiability Testing. Ninth Max-SAT Evaluation, 2014.

[2] 17th International Conference on Theory and Applications of Satisfiability Testing. Ninth Max-SAT Evaluation, 2014.

[3] Marcelo Finger. Satisfiability and SAT and MaxSAT Solver, 2015.

[4] Masahiro Sakai. Toysolver, 2014.

[5] Victor W. Marek. *Introduction to Mathematics of Satisfiability*. 2009.