Parâmetros de Complexidade Para O Problema da Satisfazibilidade Probabilística

Wendel Luiz Sviatovski Aragão (PIBIC/Fundação Araucária/UEPG), José Carlos Ferreira da Rocha (Orientador), aragaowendel@hotmail.com:jrocha@uepg.br.

Universidade Estadual de Ponta Grossa/Departamento de Informática.

Ciências Exatas e da Terra/Ciência da Computação -1.03.00.00-7.

PALAVRAS-CHAVE: Satisfazibilidade probabilística, Programação linear, Clarkson..

RESUMO

O problema da satisfazibilidade probabilística (SATP) diz respeito a verificação da existência e uma distribuição conjunta de probabilidades que atenda à restrições probabilísticas impostas por um conjunto atribuições de probabilidades sobre cláusulas proposicionais. Este problema pode ser solucionado com técnicas de programação linear. Como o problema é NP-completo mesmo pequenas instâncias podem consumir muito tempo de computação. Considerando isso, este trabalho avalia se o emprego do algoritmo de Clarkson reduz o tempo gasto por algoritmos de programação linear na solução de SATPs. A implementação proposta deste algoritmo utiliza uma heurística que explora a associação entre as cláusulas durante a solução dos SATPs. O desempenho do algoritmo proposto foi comparado ao do algoritmo simplex revisado. Os resultados indicam que o algoritmo de Clarkson/Chazelle é significativamente mais eficiente.

INTRODUÇÃO

O problema da satisfazibilidade probabilística (SATP) recebe como entrada um conjunto de cláusulas proposicionais, definidas sobre um conjunto **X** de variáveis e associadas à medidas de probabilidade (Hooker, 1992) e retorna uma resposta do tipo Sim/Não informando se existe uma distribuição de probabilidades conjuntas sobre as variáveis em **X** que satisfaça as probabilidades atribuídas às cláusulas.

Uma maneira de abordar a solução de SATPs é formulá-los como programas lineares e então resolvê-los com um algoritmo de otimização adequado. Contudo, dependendo do número de variáveis e do número de cláusulas do SATP esta abordagem pode ser ineficiente (Cozman, Campos e Rocha, 2008). Considerando isto, este trabalho avaliou o desempenho de um procedimento que estende o algoritmo de Clarkson para solução de programas de otimização. O procedimento proposto foi testado em um conjunto insatisfazíveis. Cada SATP foi transformado em um programa linear e resolvido pelo método simplex revisado e com o uso do algoritmo proposto. Os resultados obtidos foram avaliados em termos de tempo de computação. Os resultados mostram que o procedimento proposto foi mais eficiente do que o simplex revisado.

MATERIAL E MÉTODOS

Seja **X**={x1, x2 ..., xn} um conjunto de variáveis proposicionais e **C** = {C_1, C_2 ..., C_m} um conjunto de cláusulas definidas sobre as variáveis de **X**. Em lógica probabilística, cada cláusula C_j em C de é associada a uma medida de probabilidade π_j. Assim, uma base de conhecimento em lógica probabilística é uma tripla **L**=(**X**,**C**,Π), com Π =

$\{\pi_1 \ldots,\pi_n\}$. Neste contexto, a satisfazibilidade probabilística (SATP) diz respeito a verificar se existe uma distribuição de probabilidades conjunta **p**(**X**) que satisfaça o modelo especificado em **L**. Se todas as cláusulas tem o mesmo número de variáveis diz-se que o problema é um k-SATP.

Hooker (1991) descreve uma forma de como converter SATPs em problemas de programação linear. A idéia básica é associar cada uma das 2^n instanciações conjuntas em **X** a t = 2.

Pequenas instâncias do SATP podem ser resolvidas com o método simplex e suas extensões (simplex revisado, método de geração de colunas). Contudo, tal abordagem pode se tornar ineficiente à medida que cresce o número de variáveis (n) e cláusulas (m). Clarkson (1988) desenvolveu um procedimento baseado em busca aleatória para otimização linear que tem se mostrado eficiente na solução de problemas com muitas restrições. O algoritmo seleciona aleatoriamente um conjunto de restrições, resolve o subproblema derivado e verifica se a solução encontrada atende todas as restrições. Caso isto não ocorra, o processo é repetido.

Chazelle (1993), Agarwal e Sen (2000) e Fukuda, Gartnër e Szedlák (2018) estenderam o procedimento de Clarkson a fim evitar o emprego de uma estratégia aleatória na escolha das restrições que devem ser consideradas a cada iteração do algoritmo. Este trabalho propõe um procedimento que explora a mesma abordagem na solução de SATPs. No procedimento proposto, o SATP dá origem a um programa linear **P** e a um grafo não dirigido **G**. Os nós de **G** simbolizam as cláusulas em **C** e os arcos indicam que as cláusulas conectadas compartilham no mínimo duas variáveis. Na sequência componentes biconexos de **G** são determinados (Goodrich e Tamassia, 2010) e ordenados em ordem decrescente do número de nós.

Seja B_1 …, B_r a lista ordenada dos componentes biconexos de **G**. O procedimento proposto remove B_1 lista e resolve o problema linear derivado a partir da consideração das restrições em **P** que são associadas às cláusulas de B_1. Se o problema é insatisfazível então o procedimento termina e informa o resultado. Caso contrário, o procedimento seleciona o próximo componente B_k cuja conjunto de variáveis compartilhadas com B_1 é máximo; se nenhum componente atende esta condição, B_k é o maior componente da lista restante. Os componentes B_1 e B_k são unidos e o procedimento resolve o SATP relativo às cláusulas contidas em $B_1 \cup B_k$. Este processo é repetido enquanto o resultado obtido indicar que o problema é satisfazível ou não houver mais componentes a serem processados.

Como pode ser visto, o procedimento aqui descrito explora uma heurística gulosa que procura, a cada passo, resolver SATPs menores que o original e cujas cláusulas definem restrições sobre variáveis em comum. O procedimento foi testado em um conjunto de 3-SATPs gerados aleatoriamente. A amostra de SATPs continha problemas com n=15 e m entre 100 e 200. Foram gerados 50 SATPs para cada valor de m. Os SATPs foram gerados utilizando um procedimento que usava uma distribuição uniforme para selecionar quais variáveis comporiam cada cláusula. Os valores de Π também foram produzidos aleatoriamente de acordo com uma distribuição uniforme.

Os SATPs foram resolvidos com o algoritmo simplex revisado, como implementado pelo software *lpsolve* (Eikland e Notebeard, 1995) e com o procedimento proposto. O tempo de computação gasto na solução dos problemas foi registrado ao longo dos testes. A média e o desvio padrão do tempo de processamento foi calculado para cada valor de m. Os resultados obtidos foram submetidos ao teste T de Student a fim de verificar diferenças significativas no desempenho experimental de cada procedimento.

RESULTADOS E DISCUSSÃO

A Tabela 1 mostra os resultados obtidos nos experimentos. Como pode ser observado, o procedimento combinado do algoritmo de Clarkson, com a aplicação de heurística gulosa e programação linear reduziu o tempo médio gasto na solução dos SATPs. O teste T pareado, aplicado sobre as 50 instâncias de cada teste rejeitaram a hipótese nula, com α =0.05. Isto indica que, as diferenças observadas foram efetivamente significativas.

Tabela 1 – Resultados experimentais (SR simplex revisado) (Cke Clarkson estendido)

m	SR		Cke		m	SR		Cke	
	Média	DP	Média	DP		Média	DP	Média	DP
100	5211,0	1854,6	555,2	962,4	155	22835,7	11298,8	4002,1	5734,3
105	5986,0	1556,0	521,6	832,7	160	24505,6	14474,3	5492,4	7089,1
110	6862,8	1998,1	649,8	1191,1	165	26058,7	14248,1	5413,4	7372,7
115	7296,3	3015,9	515,9	1067,7	170	28067,3	19335,8	7560,9	9921,3
120	8959,4	3746,6	1129,4	1525,9	175	37833,8	28283,4	10919,5	15323,6
125	9855,9	4702,0	1563,9	1955,7	180	37157,7	25528,9	13796,3	20234,7
130	11162,9	4794,7	2228,6	2787,3	185	35088,2	24962,8	10950,3	15130,8
135	12794,5	6030,4	1185,5	2470,0	190	4609,0	32709,5	12351,5	21308,1
140	13773,0	6959,5	1863,4	3239,5	195	51865,8	50874,9	17273,5	21938,5
145	15442,8	8732,6	2399,0	3732,2	200	80915,5	60683,4	17932,9	29917,9
150	15188,4	10652,8	1775,4	3500,6					

O Gráfico da Figura 1a mostra o desempenho médio de ambos os algoritmos para cada valor de m. Como observado experimentalmente por Finger e De Bonna (2010) , para n fixo, o tempo gasto na solução de SATPs usando programação linear cresce exponencialmente o valor de m. Como, pode ser visto no gráfico, o procedimento proposto também apresenta uma relação exponencial entre o número de cláusulas e o tempo de processamento. Esta conclusão é apoiada pelo fato da correlação entre m e o logaritmo do tempo de processamento foi superior a 0,95 para ambos os algoritmos. Contudo, deve ser observado que houve uma redução de 42% no tempo médio de processamento. Adicionalmente deve ser notado que quanto maior o número de componentes em que o grafo foi decomposto, menor o tempo médio de processamento (Figura 1b).



Figura 1 – Gráfico A) Número de componentes versus tempo. B) Número de cláusulas versus tempo.

CONCLUSÃO

O procedimento proposto neste trabalho possibilitou a redução do tempo de processamento gasto na detecção da instatisfazibilidade de SATPs usando programação linear. O procecimeto é uma implementação estendida do algoritmo de Clarkson. No futuro, pretende-se ampliar os testes considerando computação paralela e combiná-lo com algoritmos que exploram a transformação do SATP em SAT.

AGRADECIMENTOS

Ao CNPq, à Fundação Araucária e à UEPG pelo suporte à pesquisa.

REFERÊNCIAS

CAMPOS, C.P.; COZMAN, F.G.; ROCHA, J. F. da; Probabilistic Logic with Independence**. International Journal of Approximate Reasoning**, p. 3-7, Setembro 2008.

CHAZELLE, Bernard. Cutting hyperplanes for divide-and-conquer. **Discrete & Computational Geometry**, p. 145–158, fev. 1993.

FINGER, Marcelo.; DE BONA, Glauber. Probabilistic Satisfiability: Logic-Based Algorithms and Phase Transition. **International Joint Conference on Artificial Inteligence**, p.528-533, 2011.

FUKUDA, K., GÄRTNER, B. & SZEDLÁK, May. Combinatorial redundancy detection. **M. Ann Oper Res** , p. 47–65, 2018

GOODRICH, Michael.; TAMASSIA, Roberto.; **Data Structures and Algorithms in Java**. 5th Edition. Wiley, 2010. 720 p.

HOOKER, J.N. Mathematical Programming Methods for Reasoning under Uncertainty. **Operations Research Proceedings 1991**, Berlin, p. 23-34,1992.

LPSOLVE: Mixed Integer Linear Programming (MILP) solver. Version 4.1. [S.l.]: K. Eikland P. Notebeard, 1995. Disponível em: <https://sourceforge.net/p/lpsolve/wiki/Home/>. Acesso em: 30 ago. 2018.

AGARWAL, Pankaj. K.; SEN, Sandeep.; Randomized algorithms for geometric optimization problems. **Handbook of Randomized Computation**, p. 151-201, 2001.