Solução Exata do Problema da Clique Máxima

Alexandre Prusch Züge[1], **Renato Carmo**[1] **(orientador)**

[1] Departamento de Informática da Universidade Federal do Paraná
Centro Politécnico da UFPR, 81531-990, Curitiba, PR

alexandrezuge@gmail.com, renato@inf.ufpr.br

Abstract. *Several algorithms for the Maximum Clique Problem are available, many of them based on the branch and bound technique. In this work we describe a general branch and bound algorithm for the exact solution of the Maximum Clique Problem, we review eight algorithms from the literature and describe each one of them by modifying the general algoritm. We implemented these algorithms and ran several experiments. Experimental results are presented for comparison.*

Resumo. *Vários algoritmos para o Problema da Clique Máxima são encontrados na literatura, grande parte deles empregando a técnica de Branch & Bound. Neste trabalho é descrito um algoritmo genérico de Branch & Bound para solução exata do Problema da Clique Máxima, são revisados oito algoritmos disponíveis na literatura e cada um dos algoritmos é descrito como uma modificação do algoritmo genérico. Implementamos estes algoritmos e executamos experimentos cujos resultados são apresentados para comparação.*

1. Introdução

O Problema da Clique Máxima (CM) é o problema de encontrar uma clique de tamanho máximo em um grafo dado. Tratando-se de um problema $\mathcal{NP}$-difícil, não se conhece algoritmo de tempo polinomial para sua solução exata. Por outro lado, a literatura apresenta diversas propostas de solução exata do problema, relatando resultados experimentais surpreendentes se confrontados com os limites analíticos conhecidos.

A principal motivação do trabalho apresentado em [Züge 2011], do qual o presente texto constitui um resumo esquemático, é a de contribuir para um melhor entendimento deste intrigante fenômeno que vez por outra assoma na fronteira entre teoria e prática da Ciência da Computação: um problema computacional que apresenta os mais desalentadores indícios de intratabilidade e para o qual, ao mesmo tempo, conhece-se algoritmos capazes de obter a solução exata de instâncias de interesse prático e tamanho considerável.

Dentre os algoritmos para solução exata do CM encontrados na literatura, os mais bem-sucedidos parecem ser aqueles baseados na técnica de *Branch & Bound*. É a esta classe de algoritmos que se dedica o estudo de [Züge 2011].

A principal contribuição original do trabalho apresentado em [Züge 2011] é a formulação de um esquema genérico de *Branch & Bound* para o CM que permite que os diversos algoritmos propostos na literatura sejam descritos e implementados como instâncias de um mesmo “meta-algoritmo”. Tal formulação representa um modesto primeiro passo em direção ao objetivo de longo prazo mencionado acima: uma melhor compreensão das razões para o descompasso entre os resultados analíticos e experimentais presentes na literatura a respeito da solução exata do CM. Ao mesmo tempo, por descrever diversos algoritmos a partir de um ponto de vista unificado, nossa formulação permite estabelecer termos de comparação menos voláteis para seu desempenho do que, por exemplo, o puro e simples tempo de execução em um determinado ambiente computacional, usado por muitos autores.

Contribuição secundária de [Züge 2011] é a efetiva implementação de tal "meta-algoritmo", bem como a implementação de oito dos mais bem-sucedidos algoritmos para o CM propostos na literatura como suas particulares instâncias. É notável o fato de que a implementação feita corresponde de maneira muito próxima à sua descrição em "alto nível", o que de certa forma confere à implementação o caráter de "prova de conceito". As implementações estão publicamente disponíveis e foram feitas com o cuidado de permitir a futura incorporação de novos algoritmos que se enquadrem na mesma formulação geral.

Terceira contribuição, decorrente desta última, reside no fato de que pela primeira vez na literatura são apresentados resultados experimentais destes oito algoritmos passíveis de comparação direta, posto que implementados e executados no mesmo ambiente computacional.

Finalmente, cabe mencionar como contribuição o fato de nossa implementação ter sido utilizada com sucesso como ferramenta na obtenção dos resultados experimentais em um trabalho independente, promovido pelo grupo de pesquisa em Sistemas Distribuídos da mesma instituição.

A colaboração com o grupo de pesquisa em Sistemas Distribuídos resultou na publicação de [Duarte Jr. et al. 2010], onde a implementação de [Züge 2011] é descrita e é utilizada para determinar cliques em centenas de grafos com centenas de vértices.

Uma versão preliminar do trabalho de [Züge 2011] foi apresentada em comunicação verbal na edição de 2010 do Latin American Workshop on Cliques (LAW-Cliques).

As contribuições originais resultantes do trabalho estão publicadas em [Carmo e Züge 2011].

O restante deste texto está organizado como segue: a seguir são detalhadas algumas definições e notação. Na seção 2 são apresentados resultados sobre a complexidade de CM e algumas soluções exatas para o problema. Um esquema de *Branch & Bound* para CM e diversos algoritmos são apresentados na seção 3. Finalmente, na seção 4 são apresentados a implementação e os resultados experimentais.

1.1. Definições e notação

Utilizamos a nomenclatura e notação usuais da teoria dos grafos (vide [Züge 2011, Seção 1.1]). Em particular, dado um grafo G, a vizinhança de um vértice v em G é denotada $\Gamma_G(v)$. Uma clique em G é um subconjunto de $V(G)$ que induz um subgrafo completo de G. Dado $Q \subseteq V(G)$, definimos $\Gamma_G^{\cap}(Q) := \bigcap_{v \in Q} \Gamma_G(v)$, se $Q \neq \emptyset$ ou $\Gamma_G^{\cap}(Q) := V(G)$ quando $Q = \emptyset$.

2. O Problema da Clique Máxima: Teoria e Prática

A versão de CM em problema de decisão é o problema de decidir, dados um grafo G e um inteiro k, se G tem uma clique de tamanho k. Este problema não só é $\mathcal{NP}$-completo

[Garey e Johnson 1979] como não é tratável por parâmetro fixo (na parametrização em que k é o parâmetro) [Downey e Fellows 1999].

O Problema da Clique Máxima é o correspondente problema de otimização que consiste em encontrar uma clique de tamanho máximo em um grafo G dado. O problema não só é $\mathcal{NP}$-difícil em decorrência do dito acima, mas não admite sequer uma $|V(G)|^{1/3}$ aproximação [Bellare et al. 1995].

Um grafo com n vértices pode ter até $3^{n/3}$ cliques maximais de tamanho $n/3$ [Moon e Moser 1965], de maneira que determinar uma clique máxima à base de enumerar explicitamente as cliques maximais do grafo tem complexidade de tempo de pior caso $\Omega(3^{n/3})$. É recente a proposta de um algoritmo que efetivamente enumera todas as cliques maximais de um grafo em tempo $O(3^{n/3})$ [Tomita et al. 2006].

Que é possível determinar uma clique máxima em tempo (de pior caso) assintoticamente menor que o necessário para enumerar todas as cliques maximais de um grafo de n vértices é demonstrado por [Tarjan e Trojanowski 1976] que apresenta um algoritmo que determina uma clique máxima do grafo em tempo $O(2^{n/3})$. Esta cota foi posteriormente reduzida para $O(2^{0.304n})$ [Jian 1986] e em seguida para $O(2^{0.276n})$ [Robson 1986] [Züge 2011, Seção 2.1].

Por outro lado, a implementação e experimentação com algoritmos para a solução exata do CM disponíveis na literatura sugerem que os resultados negativos (de aproximabilidade, tratabilidade por parâmetro fixo) e as estimativas de pior caso mencionados acima apresentam um retrato incompleto da realidade. Efetivamente, diversos dos autores que propõem algoritmos para a solução exata do CM (por exemplo [Östergård 2002, Tomita e Kameda 2007, Tomita et al. 2010]) reportam tempos de execução surpreendentes se confrontados com o que os resultados acima sugerem.

3. Algoritmos para a Solução Exata do Problema da Clique Máxima

Dentre os algoritmos mais bem-sucedidos do ponto de vista experimental constam-se aqueles que implementam a abordagem conhecida como *Branch & Bound*. O leitor encontra em [Züge 2011, Capítulo 4] uma resenha dos principais dentre estes. Referências para outras abordagens são encontradas em [Züge 2011, Seção 2.3]. O diagrama na Figura 1 mostra a "linha do tempo" da publicação de alguns destes algoritmos. Uma seta no diagrama significa que o trabalho apontado se baseia no trabalho anterior. Nem todos estes algoritmos são discutidos neste resumo mas todos são discutidos em [Züge 2011].

Em muitos casos, tais algoritmos são apresentados e discutidos por seus autores em um contexto que privilegia o aspecto experimental e o confronto de resultados experimentais com outros algoritmos propostos anteriormente. Esta ênfase muitas vezes acaba por inadvertidamente ocultar o quanto tais algoritmos têm em comum.

Dentre os algoritmos para o CM baseados em *Branch & Bound*, concentramonos sobre aqueles em que o esquema de "branching" é reminiscente do algoritmo de enumeração de cliques maximais de Bron–Kerbosch [Bron e Kerbosch 1973]. Para uma discussão do algoritmo de Bron–Kerbosch e sua relação com esquemas de *Branch & Bound*, para CM, o leitor é referido para [Züge 2011, Seção 3.3].

Como indica a Figura 1, muitos deles "descendem" do algoritmo de enumeração

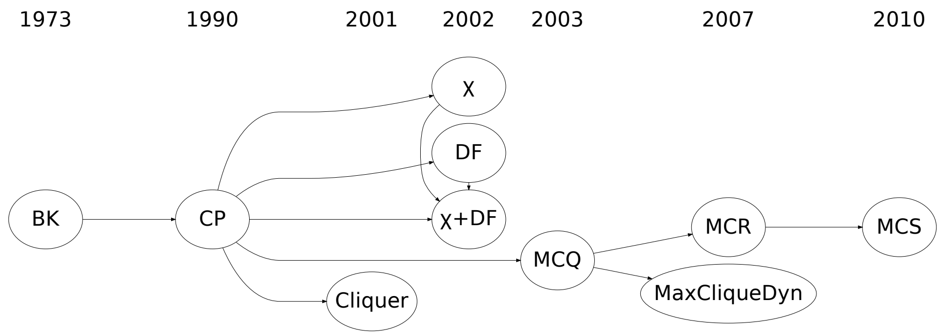

Figura 1. Ordem de publicação dos algoritmos de *Branch & Bound*

de cliques maximais de Bron–Kerbosch [Bron e Kerbosch 1973].

Em [Züge 2011, Capítulo 4] concentramos nossa atenção sobre oito destes algoritmos que, ao tempo de sua respectiva publicação, constavam-se entre os algoritmos de melhor desempenho para o CM descritos na literatura, a saber: CP [Carraghan e Pardalos 1990], DF, χ, χ + DF [Fahle 2002], MCQ [Tomita e Seki 2003], MaxCliqueDyn [Konc e Janezic 2007], MCR [Tomita e Kameda 2007] e MCS [Tomita et al. 2010]. Chamamos esses algoritmos coletivamente de *algoritmos MaxCliqueBB*.

Em [Züge 2011, Capítulo 4], propomos um esquema genérico de *Branch & Bound* para CM. O algoritmo MC apresentado a seguir é uma simplificação do esquema proposto.

MC(G, C, Q, K)

se $|Q| > |C|$ **então**
  $C \leftarrow Q$
enquanto $K \neq \emptyset$ *e* $|Q| + \text{Bound}(G, K) > |C|$ **faça**
  $v \leftarrow \text{Remove}(G, K)$
  $C \leftarrow \text{MC}(G, C, Q \cup \{v\}, K \cap \Gamma(v))$
devolve C

No algoritmo MC a função Bound(G, K) devolve uma cota superior para o tamanho da clique máxima em $G[K]$ e a Remove(G, K) remove um vértice de K e o devolve como resultado.

Os algoritmos MaxCliqueBB detalhados em [Züge 2011] têm as seguintes características, que expomos de maneira muito sumária por limitação de espaço.

CP: aplica uma heurística para escolha de vértices removidos de K baseada em graus, na tentativa de eliminar rapidamente vértices que não fazem parte de uma clique grande. Descrito em [Züge 2011, Seção 4.2].

DF: busca remover rapidamente de K vértices que certamente participam ou não podem participar de uma clique grande. Descrito em [Züge 2011, Seção 4.4].

χ**:** utiliza coloração heurística como limitante superior. Descrito em [Züge 2011, Seção 4.5.1].

χ + **DF:** aplica todas as técnicas de DF e χ. Descrito em [Züge 2011, Seção 4.5.2].

MCQ: utiliza uma coloração gulosa como limitante superior e como guia sobre qual vértice remover de K. Descrito em [Züge 2011, Seção 4.5.3].

MaxCliqueDyn: baseado em MCQ, apresenta algumas modificações na heurística de coloração, buscando limitantes mais apertados. Descrito em [Züge 2011, Seção 4.5.4].

MCR: baseado em MCQ, apresenta uma ordenação inicial de vértices que busca fazer com que coloração resulte em limitantes mais apertados. Descrito em [Züge 2011, Seção 4.5.5].

MCS: baseado em MCR, apresenta modificações na coloração, na busca de uma redução maior no número de estados gerados. Descrito em [Züge 2011, Seção 4.5.6].

4. Implementação e Resultados Experimentais

Implementamos os algoritmos MaxCliqueBB usando a linguagem Python e o pacote para manipulação de redes e grafos NetworkX[1]. Este ambiente foi escolhido por sua flexibilidade. Esta implementação é apresentada em [Züge 2011, Capítulo 5].

Para testes comparativos usamos as 66 instâncias do DIMACS Second Implementation Challenge[2]. Os resultados da execução de cada algoritmo em cada uma das instâncias são encontrados em [Züge 2011, Capítulo 5].

Referências

Bellare, M., Goldreich, O., e Sudan, M. (1995). Free bits, pcps and non-approximability-towards tight results. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, p. 422–431. IEEE.

Bron, C. e Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577.

Carmo, R. e Züge, A. (2011). Branch and bound algorithms for the maximum clique problem under a unified framework. *Journal of the Brazilian Computer Society*, p. 1–15.

Carraghan, R. e Pardalos, P. M. (1990). An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6).

Downey, R. e Fellows, M. (1999). *Parameterized complexity*, volume 5. Springer New York.

Duarte Jr., E. P., Garrett, T., Bona, L. C. E., Carmo, R., e Zuge, A. P. (2010). Finding stable cliques of planetlab nodes. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, p. 317–322. IEEE.

Fahle, T. (2002). Simple and fast: Improving a branch-and-bound algorithm for maximum clique. In *Lecture Notes in Computer Science*, p. 47–86. Springer Berlin / Heidelberg.

Garey, M. e Johnson, D. (1979). *Computers and intractability*. Freeman San Francisco.

Jian, T. (1986). An $o(2^{0.304n})$ algorithm for solving maximum independent set problem. *IEEE Trans. Comput.*, 35(9):847–851.

[1] ver `http://networkx.lanl.gov/`

[2] ver `http://dimacs.rutgers.edu/Challenges/`

Konc, J. e Janezic, D. (2007). An improved branch and bound algorithm for the maximum clique problem. *MATCH Communications in Mathematical and in Computer Chemistry*.

Moon, J. e Moser, L. (1965). On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28.

Östergård, P. R. (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1-3):197–207.

Robson, J. (1986). Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440.

Tarjan, R. E. e Trojanowski, A. E. (1976). Finding a maximum independent set. Technical report, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA.

Tomita, E. e Kameda, T. (2007). An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments. *Journal of Global Optimization*, 37(1):95–111.

Tomita, E. e Seki, T. (2003). *An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique*. Springer Berlin/Heidelberg.

Tomita, E., Sutani, Y., Higashi, T., Takahashi, S., e Wakatsuki, M. (2010). A simple and faster branch-and-bound algorithm for finding a maximum clique. In Rahman, M. e Fujita, S., editors, *WALCOM: Algorithms and Computation*, volume 5942, chapter 18, p. 191–203. Springer Berlin Heidelberg, Berlin, Heidelberg.

Tomita, E., Tanaka, A., e Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42.

Züge, A. (2011). Solução exata do problema da clique máxima. Master's thesis, Universidade Federal do Paraná.