Reconhecendo Grafos com até 1 Cruzamento

André Carvalho Silva[1], **Orlando Lee**[1] *

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas, São Paulo - Brasil

andre.silva,lee@ic.unicamp.br

Abstract. *The crossing number $cr(G)$ of a graph G is the least number of crossings over all possible drawings of G. This paper describes a minor improvement over a naive algorithm for deciding if $cr(G) \leq 1$.*

Resumo. *O número de cruzamentos de um grafo G é o menor número de cruzamentos dentre todos os desenhos de G. Este artigo descreve uma pequena melhora para um algoritmo ingênuo para decidir se $cr(G) \leq 1$.*

1. Introdução

Um *desenho* D de um grafo $G = (V(G), E(G))$ é a união das imagens de:

- uma função injetiva $\phi : V(G) \to \mathbb{R}^2$, e
- uma função ϕ_e para cada aresta $e = uv$, tal que ϕ_e é um mapeamento contínuo entre o intervalo real $[0, 1]$ e um subconjunto do $\mathbb{R}^2$, $\phi_e(0) = \phi(u)$ e $\phi_e(1) = \phi(v)$, ou vice-versa, e $\phi(V) \cap \phi_e([0, 1] \backslash \{0, 1\}) = \emptyset$.

No texto, não faremos distinção entre as arestas e vértices do grafo e o conjunto de pontos representando elas em D. Seja H um subgrafo de G, usamos a notação $D[H]$ para representar o desenho de H em D.

Seja D um desenho de um grafo G, um *cruzamento* é uma interseção entre o interior de duas arestas. O *número de cruzamentos* $cr(D)$ de D é o número de interseções entre arestas em D. O *número de cruzamentos* $cr(G)$ de um grafo G é o menor $cr(D)$ dentre todos os desenhos D de G.

O número de cruzamentos de um grafo tem aplicações práticas nas áreas de VLSI (*Very Large Scale Integration*) e de desenhos de grafos (*graph drawing*). Leighton [Leighton 1983] mostrou que a área necessária para representar um circuito elétrico está intrinsecamente relacionada ao número de cruzamentos do grafo que representa o circuito. Purchase [Purchase 1997] fez uma pesquisa mostrando que o desenhos de grafos que minimizam o número de cruzamentos são os mais fáceis de entender visualmente.

Como apontado por Székely [Székely 1997], resultados sobre o número de cruzamentos de grafos podem ser utilizados para provar, de forma simples, alguns problemas difíceis de geometria discreta.

Dado um grafo G e um inteiro k, determinar se $cr(G) \leq k$ é um problema NP-Completo [Garey and Johnson 1983]. Sabe-se que o problema é *fixed-parameter tractable* para k fixo [Grohe 2001]. Informalmente, isto significa que para cada k fixo existe um algoritmo polinomial que decide se $cr(G) \leq k$.

*Pesquisa financiada pela FAPESP Proc. 2014/14375-9, 2015/04385-0 e 2015/11937-9, CNPq Proc. 311373/2015-1

Conhece-se na literatura um algoritmo ingênuo que dados G e k, decide se $cr(G) \leq k$. Este artigo apresenta um algoritmo que, apesar de não ser asintoticamente melhor, pode melhorar a complexidade em até $|E(G)|$ para grafos mais densos. Este artigo apresenta uma melhoria para algoritmo simples que determina se um grafo tem um ou menos cruzamentos. Na Seção 2 descreveremos o algoritmo ingênuo. Na Seção 3 descrevemos nosso algoritmo.

2. Algoritmo Ingênuo para Decidir se $cr(G) \leq k$

Nesta seção descreveremos um algoritmo simples para decidir se $cr(G) \leq k$.

Usamos indução em k. Se $k = 0$ podemos usar um algoritmo de planaridade com complexidade linear no tamanho dos vértices [Hopcroft and Tarjan 1974].

Por indução em k, sabemos se $cr(G) < k$. Se for, então o problema está resolvido. Suponha o contrário. Precisamos verificar se $cr(G) = k$. Seja e, f um par de arestas distintas de G. Denotamos por $G_{e,f}$ o grafo obtido subdividindo e e f e identificando as subdivisões. Denominamos este novo vértice de v. Note que se D é um desenho de $G_{e,f}$ com $k-1$ cruzamentos D também é um desenho de G com $n+1$ cruzamentos, então, por hipótese, $G_{e,f}$ não pode ser desenhado com menos que $k-1$ cruzamentos.

Seja e, f um par de arestas distintas de G. Por indução verificamos se $cr(G_{e,f}) = k-1$. Suponha que sim e seja D um desenho de $G_{e,f}$ com $k-1$ cruzamentos. Logo, existe um desenho de G com k cruzamentos, e portanto $cr(G) = k$. Se $n < k-1$, similarmente, temos que $cr(G) \leq k-1$, contrariando nossa hipótese. Caso $cr(G_{e,f}) > k-1$ temos que G não possui nenhum desenho com no máximo k cruzamentos tal que e e f se cruzam.

Logo se não existe nenhum par de arestas distintas e, f de G tal que $cr(G_{e,f}) \leq k-1$, concluímos que não existe nenhum desenho D' de G com $cr(D) \leq k$.

Note que a cada passo indutivo no algoritmo, adicionamos duas arestas e um vértice a mais no grafo. Portanto, são gerados no máximo $(|E(G)| + 2k)^2 + 1$ subproblemas e no caso base executamos um algoritmo de complexidade $O(|V(G)| + k)$. Logo, analisando a árvore da recursão, concluímos que o algoritmo tem complexidade $O((|E(G)| + 2k)^{2k}(|V(G)| + k))$.

3. Reconhecendo Grafos com até um cruzamento

Quando $k = 1$ o algoritmo ingênuo descrito na seção anterior tem complexidade $O(|E(G)|^2|V(G)|)$. Nesta seção, descrevemos como podemos melhorar esse algoritmo neste caso.

Um par de arestas $e, f \in E(G)$ é um *par cruzante* se existe um desenho de G com um cruzamento no qual $D[e]$ e $D[f]$ se cruzam. Um subgrafo H de G é um *1-subgrafo* de G se $cr(H) = 1$. Seja H um subgrafo de um grafo G. Denotamos por $D[H]$ o desenho de H contido em D.

Lema 1. *Seja G um grafo não-planar. Se e, f é um par cruzante de um grafo G então e, f também é um par cruzante de todo 1-subgrafo H de G.*

Demonstração. Seja H um 1-subgrafo de H e seja D um desenho com um cruzamento de G no qual e e f se cruzam. Já que $D[H]$ tem que ter um cruzamento, o mesmo deve conter e e f pois são as únicas arestas que se cruzam em D. Logo, $D[H]$ tem um cruzamento. $\square$

O lema anterior mostra que não é necessário verificar se $cr(G_{e,f}) = 1$ para todo par de arestas de G. Basta apenas verificar para todo par de arestas cruzante de um 1-subgrafo H de G.

Assim, podemos modificar o algoritmo da seguinte maneira. Verificamos de G é planar. Se sim então o problema está resolvido. Senão, existe um subgrafo H de G que é uma subdivisão de um $K_{3,3}$ ou K_5. O subgrafo H pode ser obtido através do algoritmo de planaridade em tempo linear no número de vértices [Hopcroft and Tarjan 1974]. Verificamos se $cr(G_{e,f}) = 0$ para cada par de arestas e, f distintas de H. Caso sim para algum par, então $cr(G) \leq 1$. Caso contrário, $cr(G) > 1$.

O algoritmo tem complexidade $O(|E(H)|^2|V(G)|)$. No pior caso, no qual $G = H$, a complexidade é a mesma do algoritmo descrito na seção anterior. Dado que o número de arestas de H é linear com relação ao número de vértices de G, em grafos densos, temos uma melhora de até $|E(G)|$ no tempo de execução do algoritmo com relação ao ingênuo.

Podemos usar este algoritmo como caso base no algoritmo da seção anterior, melhorando o tempo de execução do mesmo. No caso em que é conhecido um 1-subgrafo H de G específico, podemos utilizar a informação sobre os pares cruzantes de H para melhorar o tempo de execução.

Referências

[Garey and Johnson 1983] Garey, M. R. and Johnson, D. S. (1983). Crossing number is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316.

[Grohe 2001] Grohe, M. (2001). Computing crossing numbers in quadratic time. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 231–236, New York, NY, USA. ACM.

[Hopcroft and Tarjan 1974] Hopcroft, J. and Tarjan, R. (1974). Efficient planarity testing. *J. ACM*, 21(4):549–568.

[Leighton 1983] Leighton, F. T. (1983). *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-exchange Graph and Other Networks*. MIT Press, Cambridge, MA, USA.

[Purchase 1997] Purchase, H. (1997). Which aesthetic has the greatest effect on human understanding? In DiBattista, G., editor, *Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer Berlin Heidelberg.

[Székely 1997] Székely, L. A. (1997). Crossing numbers and hard Erdős problems in discrete geometry. *Combinatorics, Probability and Computing*, 6:353–358.