UNIVERSIDADE DE SÃO PAULO

INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

TRABALHO DE CONCLUSÃO DE CURSO

Aluna: Karina Suemi Awoki

Orientadora: Cristina Gomes Fernandes

1 Introdução

O assunto a ser abordado no Trabalho de Conclusão de Curso se enquadra na área de Otimização Combinatória e diz respeito ao *Problema da Bissecção Mínima*. Para definir o problema precisamente, seguem primeiro algumas definições.

Seja $G = (V, E)$ um grafo e S um conjunto de vértices de G tal que $\emptyset \neq S \subset V$. Denota-se por $\delta(S)$ o conjunto das arestas de G com exatamente um extremo em S. Um *corte* em G é um conjunto C de arestas tal que $C = \delta(S)$ para algum conjunto S de vértices de G com $\emptyset \neq S \subset V$. Denota-se por $\bar{S}$ o conjunto $V \setminus S$. Note que $\delta(\bar{S}) = C$ também. O par de conjuntos S e $\bar{S}$ é chamado de *par de margens* de C. Note que se G não é conexo, um corte pode ter mais de um par de margens. O corte C é uma *bissecção* se possui um par de margens S e $\bar{S}$ tal que $||S| - |\bar{S}|| \in \{0, 1\}$, ou seja, S e $\bar{S}$ particionam o grafo basicamente ao meio. O número $|C|$ é chamado de *largura* do corte C.

O *Problema da Bissecção Mínima* consiste em, dado um grafo, encontrar uma bissecção no grafo de largura mínima.

Sabe-se que o problema da bissecção é NP-difícil [3] e a melhor aproximação conhecida para o caso geral do problema tem razão $O(\lg n)$ [5], onde n é o número de vértices do grafo. Por outro lado, sabe-se que, para árvores e grafos que de uma certa maneira se assemelham a árvores, há um algoritmo polinomial de programação dinâmica para encontrar uma bissecção mínima, proposto por Jansen, Karpinski, Lingas e Seidel [4].

Fernandes, Schmidt e Taraz têm interesse em entender a estrutura dos grafos cuja bissecção mínima tem largura grande, de modo a desenvolver bons algoritmos de aproximação para o problema ou identificar melhor as classes de grafos onde o problema torna-se especialmente difícil. Para tanto, eles têm feito estudos de certas questões em árvores, em grafos que têm uma estrutura semelhante às árvores, e também em grafos planares [1, 2], para os quais a complexidade do problema encontra-se em aberto.

Vários outros resultados são conhecidos para o Problema da Bissecção Mínima, porém este trabalho de conclusão de curso irá se concentrar no estudo do problema em árvores, e possivelmente em grafos semelhantes a árvores.

2 Objetivos

O principal objetivo deste trabalho é o estudo, implementação e análise de um algoritmo recente, proposto por Fernandes, Schmidt e Taraz [1], para encontrar uma bissecção aproximadamente mínima em árvores de grau limitado. A vantagem deste algoritmo sobre o algoritmo de Jansen et al. [4], que encontra uma bissecção de largura mínima, é que este tem um consumo de tempo linear, enquanto que o segundo tem um consumo de tempo cúbico no número de vértices da árvore. Assim mesmo, o plano é também implementar o algoritmo de Jansen et al., para fins de comparação da largura das bissecções produzidas.

Como já mencionado, o algoritmo de Jansen et al. baseia-se em programação dinâmica. Já o algoritmo de Fernandes et al., para atingir um consumo linear, utiliza técnicas bem conhecidas de percursos de árvore, como busca em profundidade, bem como um rastreamento de informações mais cuidadoso que permite que o processamento todo seja executado em tempo linear. É um algoritmo mais refinado, dividido na implementação de uma série de etapas menores.

Uma vez implementados os dois algoritmos, será feita uma análise da sua performance em árvores binárias geradas aleatoriamente, e eventualmente em uma classe maior de árvores de grau limitado.

Se houver tempo, serão abordadas também as generalizações dos algoritmos implementados para uma classe mais ampla de grafos, com grau máximo e *largura arbórea limitada*, que são os tais grafos que se assemelham a árvores, mencionados acima. Os dois algoritmos discutidos acima possuem uma versão mais geral que funciona para essa classe de grafos [2, 4]. Assim, com um pouco mais de esforço, deve ser possível adaptar as implementações para árvores para que funcionem para esta classe mais ampla de grafos. Se houver tempo então, o estudo será estendido para abranger também esta classe mais ampla de grafos.

3 Método

O estudo do algoritmo de Fernandes et al. já foi iniciado, e também foi feito um roteiro de como a implementação deve ser feita. A linguagem escolhida para a implementação é `C` e os algoritmos implementados estão sendo colocados na página do gitHub. O roteiro, que já está sendo seguido, consiste em uma divisão da implementação do algoritmo em várias etapas, que permite um melhor acompanhamento do trabalho que está sendo desenvolvido, e uma organização da forma de estudo. O algoritmo encontra-se descrito em um do-

cumento longo [6] juntamente com a sua análise teórica. Este documento tem servido como base para o entendimento de cada etapa, e de como a implementação de cada etapa deve ser feita para que a implementação obtida seja de fato linear.

Reuniões frequentes têm sido feitas junto à supervisora para apresentar as etapas já implementadas, bem como para tirar dúvidas sobre as próximas etapas ou detalhes da implementação. Em maio e junho, Tina Schmidt, uma das autoras do algoritmo que está sendo implementado, estará visitando o IME, e a parte da implementação que já estiver pronta será apresentada a ela, e algumas discussões estão planejadas.

Em paralelo ao estudo e desenvolvimento da implementação, serão lidos dois artigos da literatura. São eles o artigo de Garey, Johnson e Stockmeyer [3], que contém a prova de que o problema é NP-difícil, e o artigo de Jansen et al., que apresenta o algoritmo de programação dinâmica para o problema em árvores. Após a implementação do algoritmo de Fernandes et al., será feita a implementação do algoritmo de Jansen et al. e a comparação dos dois.

4 Cronograma

As etapas do trabalho são as seguintes:

1. Implementar um algoritmo que encontra um caminho mais longo em uma árvore.

2. Implementar os algoritmos derivados dos chamados lemas de cortes aproximados (*approximate cutting lemmas*) apresentados em [6].

3. Implementar um algoritmo que numera os vértices da árvore de uma maneira particular. Essa numeração será usada em um dos algoritmos seguintes.

4. Implementar o algoritmo referente ao teorema da dobra do diâmetro (*doubling diameter theorem*) apresentado em [6].

5. Implementar o algoritmo da bissecção, descrito no principal teorema de [6], que usa todos os anteriores.

6. Estudar a prova que o problema da bissecção é NP-difícil [3].

7. Estudar e implementar o algoritmo de programação dinâmica de Jansen et al. [4] para árvores.

8. Implementar um algoritmo para gerar árvores binárias aleatórias e, usando árvores geradas por este algoritmo, realizar experimentos com os dois algoritmos de bissecção.

9. Se houver tempo, estudar a generalização dos algoritmos para grafos com largura árborea limitada.

10. Preparar a apresentação e o pôster do TCC.

11. Redigir o texto do TCC.

As etapas de 1 a 3 já foram concluídas. O quadro abaixo apresenta o planejamento das etapas para os próximos meses:

Etapas	abr	mai	jun	jul	ago	set	out	nov
4	x							
5	x	x						
6		x	x					
7			x	x				
8				x	x			
9					x	x	x	
10						x	x	
11		x	x	x	x	x	x	x

Referências

[1] Cristina G. Fernandes, Tina J. Schmidt, and Anusch Taraz. On the structure of graphs with large minimum bisection. In J. Nešetřil and M. Pellegrini, editors, *The Seventh European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB)*, volume 16 of *CRM Series*, pages 291–296. Scuola Normale Superiore, 2013.

[2] Cristina G. Fernandes, Tina J. Schmidt, and Anusch Taraz. On minimum bisection and related partition problems in graphs with bounded tree width. Submitted, 2015.

[3] Michael R Garey, David S. Johnson, and Larry Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, pages 237–267, 1974.

[4] Klaus Jansen, Marek Karpinsk, Andrzej Lingas, and Eike Seide. Polynomial time approximation schemes for MAX-BISECTION on planar and geometric graphs. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2010 of *Lecture Notes in Computer Science*, pages 365–375. Springer, 2001.

[5] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 255–264. ACM, 2008.

[6] Tina J. Schmidt. The minimum bisection problem. PhD Thesis in preparation.