Instituto de Computação – UNICAMP
Algoritmos de Aproximação
http://www.ic.unicamp.br/~lehilton/mo418a/

Instruções para seminário

Conforme o programa, cada alun@ deverá estudar um conteúdo, preparar resumo e apresentar um seminário, conforme o seguinte:

→ Serão formados grupos de tamanho um, dois ou três. No máximo um grupo com três. O tempo de seminário será de aproximadamente 1h para apresentação individual e de 2h para apresentação em dupla. Tod@s do grupo deverão apresentar.

→ Cada grupo irá escolher um tópico em acordo com o professor. O conteúdo e o nível de detalhes apresentados serão ajustados de acordo com o tempo de apresentação.

→ Deverá ser confeccionado um resumo que é um material de apoio à apresentação e será distribuído para a turma com antecedência. Esse resumo deve ser feito utilizando-se Latex e desenvolvido em um repositório compartilhado via Gitlab.

→ A apresentação poderá usar quadro e/ou projetor. Em qualquer caso, o resumo pode ser uma versão estendida do esquema de aula utilizado (das notas de aula ou do conjunto de slides). O resumo deve ser autocontido, i.e., deve conter uma breve motivação do problema estudado, as definições, enunciados, demonstrações e referências principais.

1 Tópicos

Cada grupo deve sugerir ao professor dois ou mais tópicos de interesse, que correspondem a partes de capítulo de livro ou de artigo selecionado. Os tópicos devem ser escolhidos da lista abaixo (pelo menos um), ou de outros capítulos e fontes de interesse não listados. Se desejarem, poderão incluir uma justificativa de porque gostariam de apresentar algum tópico. A escolha dos tópicos e a ordem de apresentação será definida pelo professor.

De acordo com a quantidade de conteúdo cobertos, alguns dos tópicos abaixo estão discriminados para trabalho individual ou grupo. Os conteúdos dos tópicos não discriminados podem ser ajustados de acordo com o tamanho do grupo.

→ **APTAS para bin-packing:** O capítulo 3.3 do livro apresenta uma técnica importante para problemas de empacotamento chamada *linear grouping*. Nesse exemplo, é considerado apenas um empacotamento unidimensional, mas a técnica pode ser generalizada para diversas outras versões (geométricas ou não) e é utilizada para obter diversos esquemas de aproximação assintóticos. Qualquer pessoa interessada em empacotamento deve conhecer linear grouping. (sugestão para trabalho individual)

→ **Shortest Superstring:** No curso não apresentarei nenhum problema relacionado a strings. Nesse problema, recebemos um conjunto de strings e queremos criar uma (super) string que contém todas as outras como substring. O livro de Vazirani, capítulo 7.1, apresenta uma 4-aproximação por meio de um belo algoritmo que relaciona uma entrada a um grafo de prefixo. Uma curiosidade é que muitos acreditam que um algoritmo guloso simples é uma 2-aproximação, mas ninguém sabe provar.

→ **Explorando propriedades da solução extremal:** Vimos que conhecer uma solução extremal do PL pode ser útil para resolver o vertex-cover. Um outro exemplo para o qual conhecer a estrutura do PL é útil é o escalonamento de tarefas em máquinas não relacionadas. O exercício 11.1 do livro mostra um algoritmo elegante que encontra uma 2-aproximação para esse problema estudando com cuidado o PL. Veja também o Capítulo 17 do livro de Vazirani.

→ **Dual fitting para UFLP:** Entre as estratégias baseadas em primal-dual está o dual fitting. Vimos no início do curso uma aplicação dessa técnica para o set cover: encontrar uma solução dual (inviável) que "paga" o valor de uma solução integral viável; depois, encontrar um número que divide a solução para torná-la viável. Essa técnica é utilizada com bastante sucesso para o UFLP e dá um fator 1.861 [6], que melhor do que a 3-aproximação vista em sala. Uma curiosidade é que ao invés de calcular algebricamente o fator de aproximação, para esse problema podemos simplesmente utilizar um programa linear que revela o fator de aproximação [3].

→ **MAXCUT em grafos densos:** Embora vimos uma aplicação dos limitantes de Chernoff apenas para encontrar uma solução aproximada, eles são utilizados extensivamente para projetar algoritmos probabilísticos (para problemas de otimização ou não). Por exemplo, enquanto saber se um grafo tem 3-coloração é NP-difícil, se ele tiver, então podemos, com alta probabilidade, colori-lo com 3 cores em tempo polinomial . Do mesmo modo, aproximar MAXCUT por um fato melhor que 0.941 é NP-difícil, mas se o grafo for muito denso, então, com alta probabilidade, podemos encontrar um corte de tamanho tão próximo do ótimo quanto se queira em tempo polinomial.

→ **Cuts e tree-metrics:** Por falta de tempo, pulamos o Capítulo 8. Nesse capítulo há problemas importantes (como o multiway cut problem) e técnicas fundamentais. Em particular, é importante conhecer as chamadas *tree metrics*. Como vimos, pressupor que uma instância é métrica, i.e., que o peso das arestas é uma função de distância torna o problema muito mais fácil em vários exemplos (k-center, TSP, etc.), porque nesse caso podemos utilizar a desigualdade triangular. Se restringirmos a função de distância ainda mais, é natural pensar que o problema pode ficar ainda mais fácil. Esse é o caso de vários problemas restritos a tree metrics, i.e., supomos que o grafo subjacente é uma árvore e que a função de distância corresponde a um caminho na árvore. De fato, vários problemas que em geral são NP-difíceis, são polinomiais para tree metrics. (sugestão para trabalho grupo)

→ **Shifting geométrico:** Uma técnica muito bem sucedida é a dos chamados algoritmos de deslocamento (*shifting algorithms*) [1, 5]. A ideia é encontrar um conjunto de partições da entrada, cada uma composta por partes adjacentes (em um determinado sentido) de forma que: (i) resolver o problema restrito a cada parte seja mais fácil; e (ii) para alguma partição, a união das soluções parciais forneça uma boa solução geral. Algoritmos desse tipo são encontrados em vários domínios, particularmente em problemas geométricos. Um exemplo clássico é o PTAS para TSP quando restrito a instâncias euclideanas, apresentado no capítulo 10.1. Esse algoritmo utiliza uma programação dinâmica sofisticada e é analisado utilizando-se a noção de deslocamento aleatorizado. (sugestão para trabalho grupo)

→ **Shifting em grafos:** Uma aplicação de shifting em grafos é o algoritmo de Brenda Baker para o problema do conjunto independente máximo, apresentado no capítulo 10.2. Relembre que esse problema é NP-difícil e sequer tem aproximação subpolinomial a não ser que P $=$ NP. Se restringimos o problema para grafos planares, no entanto, a situação muda completamente e podemos encontrar um PTAS para o problema. O motivo disso é que em grafos planares podemos utilizar a técnica de deslocamento, que nesse caso, é chamada de técnica de Baker. (sugestão para trabalho grupo)

→ **Inaproximabilidade do UFLP:** Um das ferramentas normalmente utilizadas para atestar que um problema não tem aproximação constante a não ser que P $=$ NP é encontrar uma L-redução a partir do set cover, que não tem $c \ln n$-aproximação com $c < 1$ a não ser P $=$ NP. A prova dessa inaproximabilidade para set cover, no entanto, foi um longo caminho, que passa por resultados longos e alguns mais fracos (descritos no capítulo 16.4) e só foi fechado completamente em 2013 por Dinur e Steurer [2]. Mas também é possível utilizar uma redução (de oráculo) a partir do set cover para encontrar um limiar de inaproximabilidade constante, que é o resultado de inaproximabilidade do UFLP [4]. As mesmas técnicas podem ser utilizadas para encontrar limiar de inaproximabilidade para outros problemas semelhantes, como o problema das k-medianas. (sugestão para trabalho individual)

→ **Unique Games Conjecture:** Unique Games é um problema que não vimos e é apresentado no livro no capítulo 13.3. A conjectura é que um determinado problema de promessa relacionado (i.e., a versão do Unique Games em que só temos instâncias cujo o valor ótimo é muito grande, ou cujo o valor ótimo é muito pequeno) é NP-difícil e está descrita em mais detalhes no capítulo 16.5. O motivo pelo qual essa conjectura é importante é que uma boa aproximação para vários problemas importantes implica em resolver o problema da promessa em tempo polinomial. Como muitos acreditam que a conjectura é verdadeira, isso implicaria em um resultado de inaproximabilidade para o problema em questão. A vantagem é que muitas vezes o liminar de inaproximabiliade obtido é melhor do que se supuséssemos apenas P = NP.

2 Exercícios

Juntamente com o resumo, deve haver a sugestão da resolução de um exercício sobre o tópico. Se o exercício não for original, então transcreva e cite a fonte. Deverá ser possível resolver o exercício apenas tendo assistido ao seminário e possivelmente consultado o material de apoio.

Cada alun@ deverá responder (individualmente) todos os exercícios propostos pelos grupos de que não fizerem parte. A união desses formará uma lista de exercícios que deverá ser entregue utilizando o mesmo formato e modo de submissão das outras listas e será corrigida normalmente.

3 Repositório

Os resumos e/ou slides deverão ser confeccionados utilizando Latex e disponibilizados no seguinte repositório: `https://gitlab.com/zurzir/seminarios-mo418-2019`

Vocês deverão, juntamente com a escolha dos grupos e dos tópicos, enviar-me o endereço de e-mail associado a conta do `https://gitlab.com/` que vocês utilizarem. Se não tiverem conta, vocês podem criar uma conta, ou logar usando contas existentes de outras plataformas.

Eu irei disponibilizar alguns exemplos de template que vocês podem seguir. Após a escolha dos grupos e tópicos, irei criar uma pasta para cada grupo. Observem que o repositório será compartilhado, então vocês podem acessar o material de outros grupos. Por favor, alterem apenas o conteúdo na sua pasta.

Se você nunca utilizou git, dê uma olhada rápida na documentação `https://git-scm.com/doc` e avise-me em caso de dúvidas.

A razão para utilizar o repositório é para que eu possa revisar e sugerir alterações no conteúdo. Assim, uma versão preliminar do material deve está disponível com o máximo de antecedência; no pior caso, com **pelo menos uma semana antes do seminário**.

4 Cronograma

→ 10/10 - Enviar e-mail com sugestões de tópicos e formação de grupos.

→ 12/10 - Definição dos tópicos e ordem de apresentação.

→ 07/11 a 03/12 - Dias reservados para seminários (tentativo).

Referências

[1] B. S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, Jan. 1994.

[2] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 624–633, New York, NY, USA, 2014. ACM.

[3] C. G. Fernandes, L. A. A. Meira, F. K. Miyazawa, and L. L. C. Pedrosa. A systematic approach to bound factor-revealing LPs and its application to the metric and squared metric facility location problems. *Mathematical Programming*, 153(2):655–685, 2015.

[4] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999.

[5] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *J. ACM*, 32(1):130–136, Jan. 1985.

[6] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *Journal of the ACM*, 50(6):795–824, 2003.