# MAC5727 – Algoritmos de Aproximação

Departamento de Ciência da Computação – IME/USP

*Primeiro Semestre de 2000*

*"If the optimal solution is unattainable then it is reasonable to sacrifice optimality and settle for a "good" feasible solution that can be computed efficiently. Of course, we would like to sacrifice as little optimality as possible, while gaining as much as possible in efficiency. Trading-off optimality in favor of tractability is the paradigma of approximation algorithms."* [2]

## 1. Introdução

Considere o *Problema do Caixeiro Viajante*: dado um grafo $G = (V, E)$ e custos $c_e \geq 0$ para cada aresta $e$ em $E$, encontrar um circuito Hamiltoniano em $G$ de custo mínimo.

Um algoritmo óbvio para este problema é o seguinte: teste todos os circuitos em $G$ e escolha um de menor custo. O problema deste algoritmo é que ele é muito lento: se $n$ é o número de vértices de $G$, pode existir um número exponencial em $n$ de circuitos, e com isso esse algoritmo poderia levar anos para executar para uma instância onde $n = 100$, por exemplo.

A noção do que é um algoritmo eficiente foi sugerida por Edmonds e Cobham: um algoritmo é eficiente se executa em tempo polinomial no *tamanho* da entrada. Infelizmente não se sabe se existe ou não um algoritmo eficiente para resolver o TSP. O que se sabe é que se $\mathbf{P} \neq \mathbf{NP}$—uma conjectura que muitos cientistas acreditam ser verdadeira—então não existe um algoritmo eficiente para o TSP.

O que podemos fazer para contornar essa dificuldade? Temos duas alternativas: ou desistimos de procurar um algoritmo polinomial e procuramos algorimos que executem rapidamente pelo menos para as instâncias que aparecem na prática, ou desistimos de buscar soluções ótimas e nos satisfazemos com algo próximo de ótimo. Algoritmos de aproximação se enquadram na segunda alternativa: são algoritmos eficientes que produzem soluções aproximadamente ótimas.

O estudo de algoritmos de aproximação, além de ser um modo matematicamente formal de se analisar a qualidade de uma heurística, traz também mais informação sobre a dificuldade dos problemas. A busca de algoritmos exatos levou a uma primeira categorização dos problemas quanto à sua dificuldade. O estudo de algoritmos de aproximação para problemas conhecidamente difíceis tem refinado essa categorização.

Resumindo, o estudo de algoritmos de aproximação é uma possível forma de se atacar um problema conhecidamente difícil. É também uma maneira de se entender melhor a natureza da dificuldade do problema.

### 1.1. Objetivos.

Familiarizar os alunos com as técnicas de desenvolvimento e análise de algoritmos de aproximação para problemas combinatórios, e com os resultados de teoria de complexidade relacionados a aproximações.

1.2. **Método.** Neste curso, estudaremos algoritmos de aproximação de acordo com as metodologias usadas. Estaremos particularmente interessados em métodos derivados de programação linear.

Em paralelo ao estudo de como os vários métodos de desenvolvimento de algoritmos se aplicam à busca de aproximações, mostraremos também alguns resultados de teoria de complexidade relacionados com aproximações.

## 2. Critério de avaliação

Haverão duas ou três provas e várias listas de exercícios.

**Data provável das provas**:

- P1: 7 de abril (ou mais tarde).
- P2: 12 de maio (talvez essa desapareça).
- P3: 9 de junho.

A média das provas será a média arimética das notas de prova. A média das listas será a média arimética das notas das listas.

A média final levará em conta a sua média de prova, de listas e participação no curso em geral.

## 3. Outras informações

Várias informações sobre a disciplina estarão disponíveis na página `http://www.ime.usp.br/~cris/mac5727/`.

Teremos uma *mailing-list* para facilitar e estimular a comunicação entre os alunos e o professor. Assumirei que os alunos estão checando esta *mailing-list* periodicamente, de forma que avisos enviados na *mailing-list* sejam oficiais.

Vocês podem me encontrar na *sala 150-B* do IME-USP.

## 4. Bibliografia

A bibliografia principal é


1. M. Goemans, *Approximation algorithms*, 1994, notas de aula disponíveis em `http://www-math.mit.edu/goemans/`.
2. D. Hochbaum (ed.), *Approximation Algorithms for NP-hard problems*, PWS Publishing Company, 1997.
3. E.W. Mayr, H.J. Prömel, and A. Steger (eds.), *Lectures on Proof Verification and Approximation Algorithms*, Springer, 1998.
4. V. Vazirani, *Approximation Algorithms*, preprint, 1999, disponíveis em `http://www.cc.gatech.edu/fac/Vijay.Vazirani/`.
5. D. Williamson, *Lecture Notes on Approximation Algorithms*, Fall 1998, disponíveis em `http://www.research.ibm.com/people/w/williamson/`.


Eventualmente você pode querer consultar alguns dos textos abaixo para uma revisão em alguns tópicos relacionados.


6 V. Chvátal, *Linear programming*, W.H. Freeman and Company, New York, 1983.

7 T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, MIT Press & McGraw-Hill, 1992.

8 M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, Freman, New York, 1979.