

# Heurísticas e Metaheurísticas

## Heurísticas e Metaheurísticas

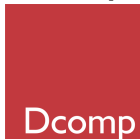
Prof. Guilherme de Castro Pena

guilherme.pena@ufsj.edu.br

Sala: DCOMP 3.11

Departamento de Ciência da Computação  
Universidade Federal de São João del-Rei

*Material adaptado do Prof. André (UFV)*



# Agenda

## 1 Heurísticas e Metaheurísticas - Introdução

- Introdução
- Objetivos
- Otimização

## 2 Heurísticas X Metaheurísticas

- Heurísticas
- Metaheurísticas

## 3 Classes de Problemas

- Classes de Problemas

# Introdução

## Visão geral:

- ▶ **Heurísticas e Metaheurísticas** são métodos que visam encontrar **soluções viáveis** para problemas, em sua maioria chamados de **problemas de otimização**, e de uma forma que a busca por essas soluções seja realizada de forma **eficiente** a nível dos recursos computacionais utilizados.

# Objetivos

## Objetivos gerais da disciplina:

- ▶ Apresentar técnicas básicas e avançadas para construção de heurísticas e metaheurísticas eficientes para problemas de otimização.
- ▶ Propor, implementar e analisar a eficiência desses métodos aplicados a problemas de otimização específicos.

# Otimização

## Visão geral:

- ▶ Campo de conhecimentos cujas técnicas visam determinar os **extremos** (máximos ou mínimos) de **funções**, em **domínios** (espaços de solução) determinados.
  - ▶ **Função:** representa algum fator de mérito relacionado com o problema que se deseja analisar.
  - ▶ Se for um problema de rotas e transporte de objetos que envolve o uso de caminhões por exemplo, a função pode representar o custo de cada rota, o lucro alcançado com tais objetos transportados, etc.
  - ▶ **Extremo (máximo ou mínimo):** representa a melhor maneira possível de se planejar tais rotas. Maximizar o lucro, minimizar o custo (uso de combustível por exemplo).
  - ▶ **Domínio:** corresponde ao conjunto das alternativas diferentes disponíveis para planejar o problema em questão.

# Agenda

## 1 Heurísticas e Metaheurísticas - Introdução

- Introdução
- Objetivos
- Otimização

## 2 Heurísticas X Metaheurísticas

- Heurísticas
- Metaheurísticas

## 3 Classes de Problemas

- Classes de Problemas

# Heurísticas X Metaheurísticas

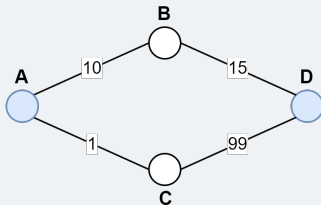
## Diferenças nas definições (Heurísticas):

- ▶ **Heurísticas** são algoritmos com uma sequência finita de instruções não ambíguas que podem ser executadas por um computador para resolver um problema.
- ▶ Elas nem sempre retornam a (melhor) solução de um problema e nem quão perto se pode chegar dela.
- ▶ A ideia é garantir uma solução “boa”, melhor que a maioria, num tempo razoável.

# Heurísticas X Metaheurísticas

## Diferenças nas definições (Heurísticas):

- ▶ Um detalhe é que **heurísticas são específicas para o problema abordado**.
- ▶ Por exemplo, **qual o menor caminho para ir do ponto A ou D?**

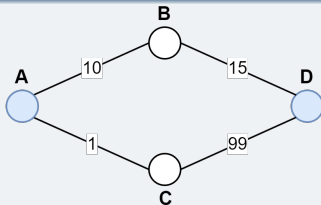


- ▶ Uma heurística chamada **gulosa**, diz que, partindo-se do início (A), a cada nó, vá para o mais próximo (menor distância).
- ▶ Dessa forma, conseguimos uma solução: (A - C - D) e podemos dizer que essa solução tem custo 100 (soma das distâncias).



# Heurísticas X Metaheurísticas

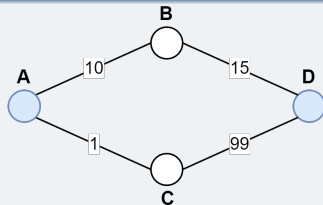
## Diferenças nas definições (Heurísticas):



- ▶ (A - C - D) é **uma solução viável**, resolve o problema, mas **é a solução ótima?**
- ▶ Essa heurística, não permite mais alterar a solução, deixando-a assim “presa”.
- ▶ Da para notar que não é a melhor solução, mas então como encontrar tal solução?
- ▶ Nesse caso podemos usar alguma **Metaheurística!**

# Heurísticas X Metaheurísticas

## Diferenças nas definições (Heurísticas):



- ▶ (A - C - D) é **uma solução viável**, resolve o problema, mas **é a solução ótima?**
- ▶ Essa heurística, não permite mais alterar a solução, deixando-a assim “presa”.
- ▶ Da para notar que não é a melhor solução, mas então como encontrar tal solução?
- ▶ Nesse caso podemos usar alguma **Metaheurística!**

# Heurísticas X Metaheurísticas

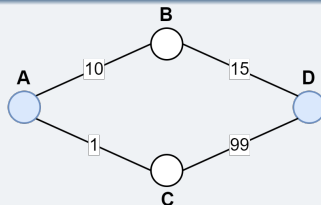
## Diferenças nas definições (Metaheurísticas):

- ▶ **Metaheurísticas** também são algoritmos com uma sequência finita de instruções não ambíguas que podem ser executadas por um computador para resolver um problema.
- ▶ Elas também nem sempre retornam a (melhor) solução de um problema.
- ▶ No entanto, uma grande diferença é de que elas tem a capacidade de fugir de **ótimos locais**, conseguindo alterar a solução atual e buscar por soluções melhores no espaço.



# Heurísticas X Metaheurísticas

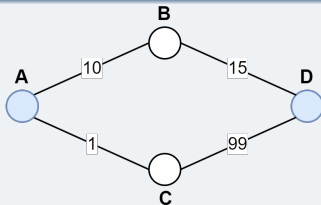
## Diferenças nas definições (Metaheurísticas):



- ▶ Em geral, as metaheurísticas partem de uma solução viável, e a partir desta, faz modificações na busca por melhores soluções.
- ▶ Podemos usar a heurística gulosa anterior para dar a nossa solução viável de partida.
- ▶ Assim, temos a solução viável (A - C - D) com custo 100.

# Heurísticas X Metaheurísticas

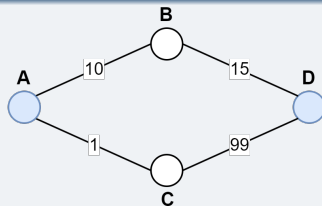
## Diferenças nas definições (Metaheurísticas):



- ▶ Nesse momento, para alterarmos a solução em questão, a metaheurística poderia fazer uma troca entre um ponto do grafo não usado, com algum que está presente na solução.
- ▶ Nesse caso, o ponto B ainda não foi usado, logo, podemos trocá-lo com os presentes na solução (A - C - D), verificar a viabilidade e posteriormente o custo.
- ▶ Aplicando essa ideia, geramos 3 novas soluções:  
(B - C - D), (A - B - D) e (A - C - B).

# Heurísticas X Metaheurísticas

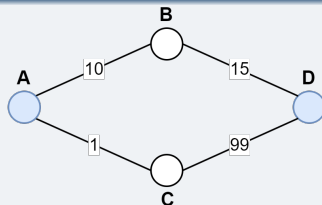
## Diferenças nas definições (Metaheurísticas):



- ▶ Ao verificar a viabilidade,  $(B - C - D)$  é uma **solução inviável**, pois não parte do nó A, e nem existe caminho de B para C.
- ▶ A solução  $(A - B - D)$  é viável, satisfaz as restrições de começar em A e terminar em D, e ainda tem custo menor que a atual, cujo valor é 25 (**encontramos uma solução melhor!**).
- ▶ A outra possibilidade,  $(A - C - B)$ , também é uma **solução inviável**.

# Heurísticas X Metaheurísticas

## Diferenças nas definições (Metaheurísticas):



- ▶ Então nossa solução atual agora é (A - B - D) com custo 25.
- ▶ A metaheurística poderia tentar continuar, mas chegaríamos a soluções piores ou inviáveis.
- ▶ Para evitar isso, geralmente é estipulado um **critério de parada** para finalizar a busca por novas soluções.



# Agenda

## 1 Heurísticas e Metaheurísticas - Introdução

- Introdução
- Objetivos
- Otimização

## 2 Heurísticas X Metaheurísticas

- Heurísticas
- Metaheurísticas

## 3 Classes de Problemas

- Classes de Problemas

# Classes de Problemas

## Visão Geral:

- ▶ Conforme visto em PAA, os problemas da computação são incluídos em algumas classes de problemas:
  - ▶ Problemas Tratáveis
  - ▶ Problemas Intratáveis ou Difíceis
  - ▶ Problemas de Decisão
  - ▶ Problemas de Otimização

# Classes de Problemas

## Problemas Tratáveis:

- ▶ Problemas que podem ser resolvidos em tempo polinomial:

$$T(n) = O(n^k)$$

- ▶ onde  $n$  é o tamanho da entrada e  $k$  uma constante.
- ▶ Obs: como trabalhamos com a complexidade assintótica, então algoritmos com complexidade logarítmica podem ser considerados polinomiais.

## Exemplos:

- ▶ Ordenação
- ▶ Caminho mínimo
- ▶ Árvore geradora mínima
- ▶ Fluxo máximo

# Classes de Problemas

## Problemas Intratáveis ou Difíceis:

- ▶ problemas para os quais não se conhece um algoritmo de tempo polinomial que o resolva. O melhor algoritmo conhecido é de **tempo exponencial**.

## Exemplos:

- ▶ Torres de hanoi
- ▶ Caixeiro viajante
- ▶ Cobertura mínima de vértices
- ▶ Clique máxima, etc.

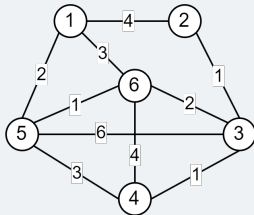
# Classes de Problemas

## Problemas de Decisão (sim ou não):

- ▶ São problemas que questionam a existência de uma solução. A resposta (saída) destes problemas é SIM ou NÃO.

## Exemplo:

- ▶ Dado um **grafo** valorado e um número  $k$ . Deseja-se saber se existe ou não um **Ciclo Hamiltoniano** com custo  $\leq k$ .
- ▶ Um **Ciclo Hamiltoniano** em um grafo  $G = (V, A)$  é um ciclo simples que passa por todos os vértices do grafo uma única vez.



$k = 16$ : SIM

$k = 10$ : NÃO

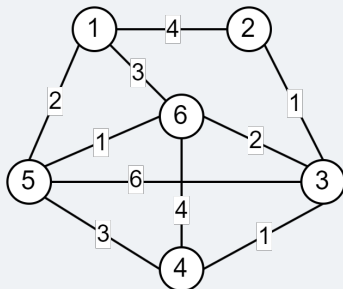
# Classes de Problemas

## Problemas de Otimização:

- ▶ Estes problemas consistem em encontrar uma solução que **minimize** ou **maximize** uma **função objetivo**.

## Exemplo:

- ▶ Determinar o *Ciclo Hamiltoniano* de **menor custo** de um grafo valorado (problema do **Caixeiro Viajante**).



# Concluindo

## Podemos concluir que..

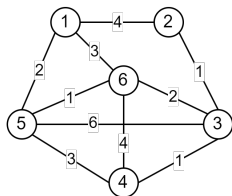
- ▶ Existem alguns problemas especiais para os quais são conhecidos procedimentos exatos e eficiente.
- ▶ Mas para que cada situação seja representada adequadamente adaptações devem ser realizadas.
- ▶ Isso gera um grande conjunto de problemas que não se sabe resolver de maneira exata.

## Para que desenvolver uma heurística?

- ▶ As heurísticas e metaheurísticas são os algoritmos que buscam encontrar soluções viáveis para tais problemas de forma mais eficiente.

# Exercícios

- ❶ **Grafos** são muito úteis para representar problemas de otimização. Cada entrada do problema denominamos **instância**. Por exemplo, o seguinte grafo geraria a seguinte instância:



```
6 9
1 2 4
1 5 2
1 6 3
2 3 1
3 4 1
3 5 6
3 6 2
4 6 4
4 5 3
5 6 1
```

- Onde temos 6 vértices, 9 arestas, e uma lista com a ligação de cada aresta com seus custos.

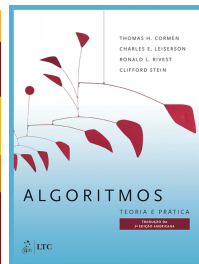
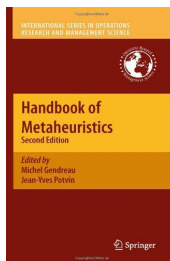
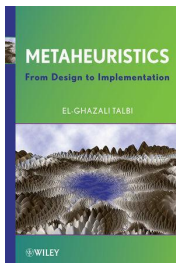
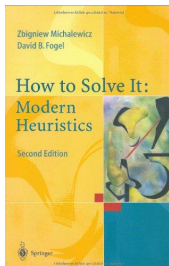
Como exercício, crie em C++ um programa capaz de ler essa instancia a partir de um txt, de forma genérica, independente do número de vértices ou arestas. Guarde a informação em uma matriz e uma lista de adjacência, e por fim imprima para verificar se foi feito de forma correta.



# Bibliografias

## Bibliografia Básica

- ❶ MICHLEWICZ, Zbigniew; FOGEL, David B. How to solve it: modern heuristics. 2nd. ed. Berlin: Springer c2010 554 p. ISBN 9783642061349.
- ❷ Talbi, El-Ghazali; Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.
- ❸ GENDREAU, Michel. Handbook of metaheuristics. 2.ed. New York: Springer 2010 648 p. (International series in operations research & management science ; 146).
- ❹ T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 3rd edition, 2009 (**Pergamum**).



# Bibliografias

## Bibliografia Complementar

- ❶ GLOVER, Fred; KOCHENBERGER, Gary A. (ed.). Handbook of metaheuristics. Boston: Kluwer, 2003. 556 p. (International series in operations research & management science ; 57).
- ❷ BLUM, Christian Et Al. Hybrid metaheuristics: an emerging approach to optimization. Berlin: Springer 2008 289 p. (Studies in Computational intelligence; 114).
- ❸ DOERNER, Karl F. (ed.) Et Al. Metaheuristics: progress in complex systems optimization. New York: Springer 2007 408 p. (Operations research / computer science interfaces series).
- ❹ GLOVER, Fred; LAGUNA, Manuel. Tabu search. Boston: Kluwer Academic, 1997. 382 p.
- ❺ AARTS, Emile. Local search in combinatorial optimization. Princeton: Princeton University Press, 2003 512 p.
- ❻ Gaspar-Cunha, A.; Takahashi, R.; Antunes, C.H.; Manual de Computação Evolutiva e Metaheurística; Belo Horizonte: Editora UFMG; Coimbra: Imprensa da Universidade de Coimbra; 2013.