

Heurísticas e Metaheurísticas

Métodos Gulosos

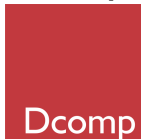
Prof. Guilherme de Castro Pena

guilherme.pena@ufsj.edu.br

Sala: DCOMP 3.11

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

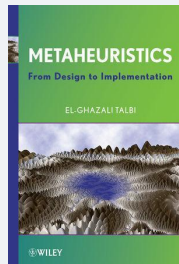
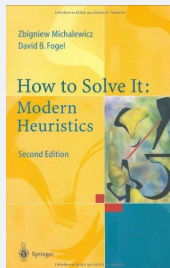
Material adaptado do Prof. André (UFV)



Referência

Livro

- ▶ Esse conteúdo está baseado nos livros:
- ▶ MICHLEWICZ, Zbigniew; FOGEL, David B. How to solve it: modern heuristics. 2nd. ed. Berlin: Springer c2010 554 p. ISBN 9783642061349 (Capítulo 4, seções 4.1 - 4.6).
- ▶ Talbi, El-Ghazali; Metaheuristics: From Design to Implementation, Wiley Publishing, 2009 (Capítulo 1, seção 1.3.4).



Métodos Tradicionais

Visão Geral:

- ▶ Vimos métodos que trabalham com **soluções completas**:
 - ▶ Incluem todas as variáveis de decisão.
 - ▶ Pode-se facilmente comparar duas soluções.
 - ▶ Vantagem: já tem alguma solução em qualquer momento que parar a busca.

Exemplos:

- ▶ Métodos com solução completa:
 - ▶ Busca Exaustiva
 - ▶ Busca Local
 - ▶ Metaheurísticas em geral (Que veremos)

Métodos Tradicionais

Visão Geral:

- ▶ No entanto, existem métodos que constroem **soluções completas** por uma sequência de passos:
 - ▶ Eles trabalham com *soluções parciais* que vão sendo incrementadas.
 - ▶ No geral, esse incremento do passo a passo é feito de forma guiada.
 - ▶ Vantagem: Alguns deles podem ser mais simples.

Exemplos:

- ▶ Métodos que constroem uma solução completa:
 - ▶ **Métodos Gulosos**
 - ▶ Divisão e Conquista
 - ▶ Programação Dinâmica
 - ▶ Branch and Bound
 - ▶ Algoritmo A*

Agenda

1 Métodos Gulosos

- Introdução
- Conceitos
- Exemplos

Introdução

Visão geral:

- ▶ Em algoritmos gulosos (ou métodos gulosos), começamos do zero (solução vazia) e construímos uma solução atribuindo valores a uma variável de decisão de cada vez, até que uma solução completa seja gerada.
- ▶ A razão pelo uso dos métodos gulosos é a simplicidade comparado a outros.
- ▶ Eles assumem a existência de uma heurística que escolhe a melhor alternativa, a mais **lucrativa**, daí o nome **guloso**.
- ▶ Mas como principal desvantagem, o método guloso não consegue enxergar muito adiante, ele é preso à sua decisão local.
- ▶ Isso quer dizer que a otimalidade local não necessariamente levará a otimalidade global.

Introdução

Observação

- ▶ **Um método guloso não é Busca Local!!!**
- ▶ A Busca Local trabalha com soluções completas.
- ▶ Mas podem ser (e geralmente são) combinados:
 - ▶ Método guloso constrói uma solução inicial.
 - ▶ Busca Local melhora a solução.

Conceitos

Considerações:

- ▶ Para entendermos melhor a aplicação de métodos gulosos em problemas de otimização, primeiramente devemos ver:
 - ▶ uma solução como um conjunto finito de elementos, $E = \{e_1, e_2, \dots, e_n\}$.
 - ▶ a função objetivo que mapeia que mapeia uma combinação desses elementos em um valor, $f : 2^E \rightarrow \mathbb{R}$.
 - ▶ e o espaço de busca definido como $F \subset 2^E$.
- ▶ Assim, um conjunto vazio é definido como solução inicial.
- ▶ A cada passo, a heurística local seleciona um novo elemento $e_i \in E$ para entrar no conjunto.
- ▶ Esse elemento não pode ser trocado por outro e não existe *backtracking* sobre as decisões já tomadas.
- ▶ Tipicamente, métodos gulosos são algoritmos determinísticos.

Conceitos

Considerações:

- ▶ Para entendermos melhor a aplicação de métodos gulosos em problemas de otimização, primeiramente devemos ver:
 - ▶ uma solução como um conjunto finito de elementos, $E = \{e_1, e_2, \dots, e_n\}$.
 - ▶ a função objetivo que mapeia que mapeia uma combinação desses elementos em um valor, $f : 2^E \rightarrow \mathbb{R}$.
 - ▶ e o espaço de busca definido como $F \subset 2^E$.
- ▶ Assim, um conjunto vazio é definido como solução inicial.
- ▶ A cada passo, a heurística local seleciona um novo elemento $e_i \in E$ para entrar no conjunto.
- ▶ Esse elemento não pode ser trocado por outro e não existe *backtracking* sobre as decisões já tomadas.
- ▶ Tipicamente, métodos gulosos são algoritmos determinísticos.

Conceitos

Considerações:

- ▶ Vejamos um *template* de um método guloso, onde:
 - ▶ s é uma solução inicial vazia.
 - ▶ s será uma solução parcial quando tiver um subconjunto $\{e_1, e_2, \dots, e_k\}$ de elementos $e_i \in E$.
 - ▶ A heurística local é o critério de decisão para selecionar o próximo elemento.
 - ▶ O conjunto F representa soluções viáveis.

Algorithm 1: Template do método guloso

```
1  $s \leftarrow \emptyset$ 
2 while Solução Completa não encontrada do
3    $e_i \leftarrow \text{HeuristicaLocal}(E \setminus \{e/e \in s\})$ 
4   #Próximo elemento de E menos os já selecionados
5   if  $s \cup e_i \in F$  then
6      $s \leftarrow s \cup e_i$ 
7   end
8 end
9 return  $s$  (solução completa)
```

Exemplos

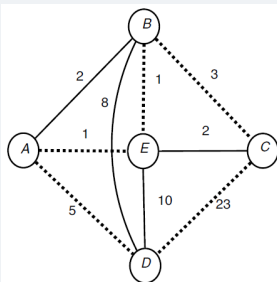
Problemas Comuns

- ▶ Como exemplos mais comuns de problemas com diferentes heurísticas gulosas podemos citar:
 - ▶ TSP
 - ▶ Problema da Mochila 0/1
 - ▶ Árvore Geradora Mínima

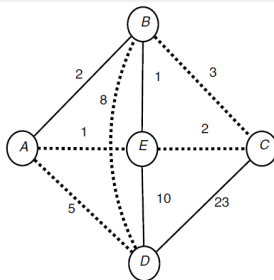
Exemplos

TSP

- ▶ No problema do TSP, o conjunto E é definido pelo conjunto de arestas.
- ▶ O conjunto F de soluções viáveis é definido pelos subconjuntos de 2^E que formam ciclos Hamiltonianos.
- ▶ O método guloso mais conhecido é selecionar a cada passo, **o vizinho mais próximo** de acordo com as distâncias nas arestas.



Greedy final solution : A – E – B – C – D – A
with cost = 33



Better solution : A – E – C – B – D – A
with cost = 19

Exemplos

TSP

- ▶ Quais seriam outras heurísticas gulosas para o TSP?

Exemplos

TSP

- ▶ Quais seriam outras heurísticas gulosas para o TSP?
 - ▶ Inserir o vizinho mais distante.
 - ▶ Inserir uma cidade no meio de uma rota, que geraria o menor aumento da rota (Note que este difere do vizinho mais próximo porque considera todas cidades da rota).
 - ▶ Entre outras.

Exemplos

TSP

- ▶ Quais seriam outras heurísticas gulosas para o TSP?
 - ▶ Inserir o vizinho mais distante.
 - ▶ Inserir uma cidade no meio de uma rota, que geraria o menor aumento da rota (Note que este difere do vizinho mais próximo porque considera todas cidades da rota).
 - ▶ Entre outras.

Exemplos

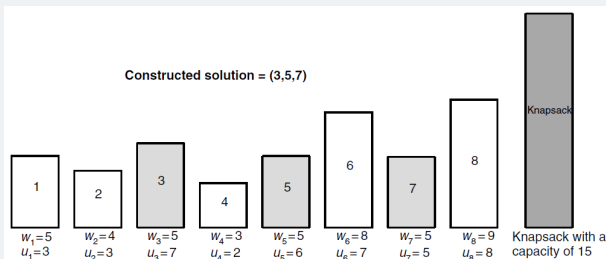
TSP

- ▶ Quais seriam outras heurísticas gulosas para o TSP?
 - ▶ Inserir o vizinho mais distante.
 - ▶ Inserir uma cidade no meio de uma rota, que geraria o menor aumento da rota (Note que este difere do vizinho mais próximo porque considera todas cidades da rota).
 - ▶ Entre outras.

Exemplos

Problema da Mochila 0/1

- ▶ No problema da Mochila 0/1, o conjunto E é definido pelo conjunto de itens com peso e valor cada.
- ▶ O conjunto F de soluções viáveis é definido pelos subconjuntos de 2^E que cabem (em peso) dentro da mochila.
- ▶ O método guloso mais conhecido é selecionar a cada passo o objeto de **maior** razão (valor/peso) ou de forma equivalente, o **menor** custo/benefício que seria (peso/valor).



Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?

Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?
 - ▶ Inserir por ordem de itens mais leves.
 - ▶ Inserir por ordem de itens mais pesados.
 - ▶ Inserir por ordem de itens de menor valor.
 - ▶ Inserir por ordem de itens de maior valor.
 - ▶ Entre outras.

Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?
 - ▶ Inserir por ordem de itens mais leves.
 - ▶ Inserir por ordem de itens mais pesados.
 - ▶ Inserir por ordem de itens de menor valor.
 - ▶ Inserir por ordem de itens de maior valor.
 - ▶ Entre outras.

Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?
 - ▶ Inserir por ordem de itens mais leves.
 - ▶ Inserir por ordem de itens mais pesados.
 - ▶ Inserir por ordem de itens de menor valor.
 - ▶ Inserir por ordem de itens de maior valor.
 - ▶ Entre outras.

Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?
 - ▶ Inserir por ordem de itens mais leves.
 - ▶ Inserir por ordem de itens mais pesados.
 - ▶ Inserir por ordem de itens de menor valor.
 - ▶ Inserir por ordem de itens de maior valor.
 - ▶ Entre outras.

Exemplos

Problema da Mochila 0/1

- ▶ Quais seriam outras heurísticas gulosas para o problema da Mochila 0/1?
 - ▶ Inserir por ordem de itens mais leves.
 - ▶ Inserir por ordem de itens mais pesados.
 - ▶ Inserir por ordem de itens de menor valor.
 - ▶ Inserir por ordem de itens de maior valor.
 - ▶ Entre outras.

Exemplos

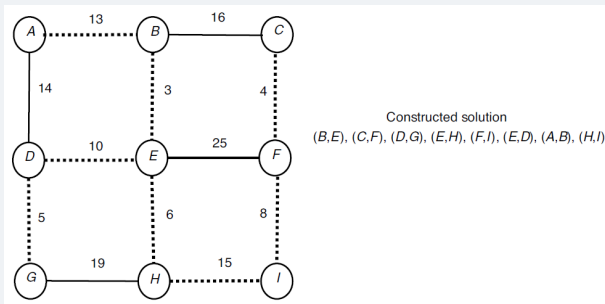
Árvore Geradora Mínima

- ▶ No problema da Árvore Geradora Mínima, existe um algoritmo de Kruskal que é guloso mas gera soluções ótimas.
- ▶ Nesse problema, o conjunto E é definido pelo conjunto de arestas.
- ▶ O conjunto F de soluções viáveis é definido pelos subconjuntos de E que são árvores.

Exemplos

Árvore Geradora Mínima

- ▶ O algoritmo de Kruskal pega as arestas de menor valor tomando cuidado pra não fazer ciclos (pois assim não formaremos uma árvore). Em seguida, vai pegando as próximas arestas de menor peso até que todos os vértices estejam ligados.



Exercícios

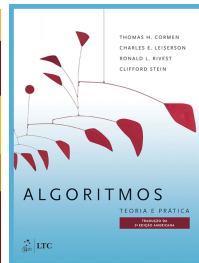
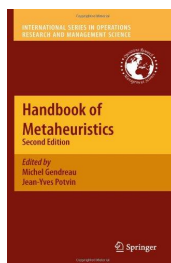
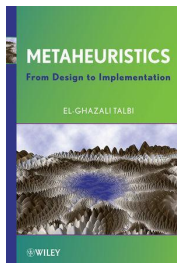
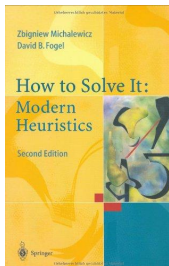
- ➊ Vamos fazer dois métodos gulosos para o TSP: (a) Usando o vizinho mais próximo. (b) Usando o vizinho mais distante.
- ➋ Agora dois métodos gulosos para o problema da Mochila 0/1: (a) Usando o melhor custo benefício (peso/valor). (b) Usando os itens mais leves.
- ➌ Em seguida, vamos aplicar cada um deles com suas respectivas buscas locais realizadas no último exercício e ver se obtemos melhoras ou não.

Obs: Podemos usar a mesmas instâncias utilizadas na busca local para ambos.

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.