Heurísticas e Metaheuristicas

Conceitos Básicos (Parte 2)

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

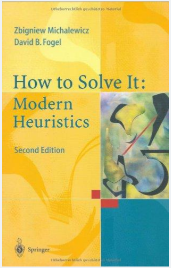
1. Conceitos Básicos
   - Representação
   - Objetivo
   - Função de Avaliação

2. Problema de Otimização
   - Definição
   - Vizinhança e Ótimo Local

Referência

Livro

- Esse conteúdo está baseado no livro texto:
- MICHLEWICZ, Zbigniew; FOGEL, David B. How to solve it: modern heuristics. 2nd. ed. Berlin: Springer c2010 554 p. ISBN 9783642061349.
- Capítulo 2, seções 2.1 - 2.5;



Introdução

Descrição

- Quando estivermos falando de uma abordagem algorítmica para resolver algum problema, três conceitos básicos são comuns.
- Independente de qualquer técnica que usarmos, os conceitos estarão presentes:
  - A Representação
  - O Objetivo
  - A Função de Avaliação

Representação

Visão geral

- De forma generalizada, a representação codifica soluções candidatas alternativas para manipulação.
- Por exemplo:

- SAT, onde temos n variáveis que são bits lógicos (0 ou 1).
  - Uma forma óbvia de representar uma solução candidata seria uma *string* binária de tamanho n: `001001001001...11` em que cada posição dessa *string* representaria uma variável do problema.
  - O tamanho do espaço de busca com essa representação é 2^n, e cada ponto desse espaço é uma solução viável para o SAT.

Representação

Visão geral

- De forma generalizada, a representação codifica soluções candidatas alternativas para manipulação.
- Por exemplo:

- TSP, onde temos n cidades a serem visitadas.
  - Como vimos o problema, o espaço de busca de um TSP-simétrico por exemplo $(dist(i,j) = dist(j,i))$ é $\frac{(n-1)!}{2}$, que gera números muito grandes.
  - Uma representação para uma solução candidata nesse caso poderia ser um vetor de inteiros de tamanho n em que a primeira posição é a cidade de início e fim da rota e as outras as cidades consecutivas a partir da primeira: `{1, 2, 3, ..., 20}`.

Representação

Considerações

- Para cada problema, a **representação** de uma solução potencial e a sua correta interpretação implica diretamente no espaço de busca e no seu tamanho.
- Definir o espaço de busca correto é fundamental!
- Usar uma representação equivocada pode adicionar muitas soluções inviáveis ou duplicadas; ou mesmo excluir a solução correta.

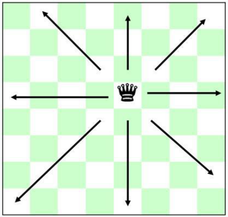
Exemplos

- Codificação Binária:
  - SAT
  - Mochila
- Permutação (vetor de valores):
  - TSP
  - Problemas de Alocação (Scheduling)

Representação - Reduzindo o espaço de buscas

Problema das 8 rainhas

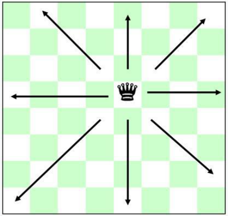
- Por exemplo, seja o problema das 8 rainhas:
- O problema é colocar 8 rainhas num tabuleiro de xadrez 8x8 sem que nenhuma esteja sob ataque de outra.



Representação - Reduzindo o espaço de buscas

Problema das 8 rainhas

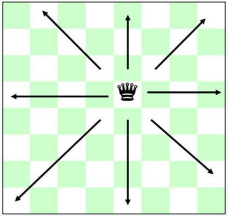
- Uma representação seria um vetor de 8 valores, representando a casa de cada rainha (de 1 a 64):
  - $64^8 = 281.474.976.710.656$
- Porém, rainhas não podem estar na mesma linha! (deve haver uma por linha)
- Guardar apenas a coluna (1 a 8) de cada uma:
  - $8^8 = 16.777.216$



Representação - Reduzindo o espaço de buscas

Problema das 8 rainhas

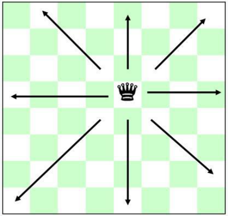
- Uma representação seria um vetor de 8 valores, representando a casa de cada rainha (de 1 a 64):
  - $64^8 = 281.474.976.710.656$
- Porém, rainhas não podem estar na mesma linha! (deve haver uma por linha)
- Guardar apenas a coluna (1 a 8) de cada uma:
  - $8^8 = 16.777.216$



Representação - Reduzindo o espaço de buscas

Problema das 8 rainhas

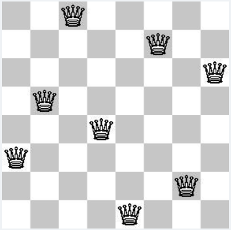
- Mas as rainhas também não podem estar na mesma coluna! (deve haver uma por coluna)
- Permutação dos valores 1, 2, 3, 4, 5, 6, 7, 8:
  - $8! = 40.320$



Representação - Reduzindo o espaço de buscas

Problema das 8 rainhas

- Mas as rainhas não podem estar na mesma coluna! (deve haver uma por coluna)
- Permutação dos valores 1, 2, 3, 4, 5, 6, 7, 8:
  - $8! = 40.320$
  - **Uma solução:** $[3, 6, 8, 2, 4, 1, 7, 5]$



Objetivo

Visão geral

- De forma generalizada, o **objetivo** descreve o propósito que deve ser satisfeito.
- Uma vez que se conhece o espaço de busca, deve-se decidir o que estamos procurando, qual é o objetivo do nosso problema?
- Na modelagem, o objetivo no geral pode ser traduzido como uma expressão matemática.
- Por exemplo:
  - No TSP o objetivo é tipicamente minimizar a distância total viajada, o que em termos matemáticos seria:

$$min \sum dist(x, y)$$

  - Já no SAT o objetivo é encontrar o vetor de bits que satisfaça a expressão booleana (TRUE). Ele não é traduzido em uma expressão matemática.

Objetivo

Visão geral

- De forma generalizada, o **objetivo** descreve o propósito que deve ser satisfeito.
- Uma vez que se conhece o espaço de busca, deve-se decidir o que estamos procurando, qual é o objetivo do nosso problema?
- Na modelagem, o objetivo no geral pode ser traduzido como uma expressão matemática.
- Por exemplo:
  - No TSP o objetivo é tipicamente minimizar a distância total viajada, o que em termos matemáticos seria:

$$min \sum dist(x, y)$$

  - Já no SAT o objetivo é encontrar o vetor de bits que satisfaça a expressão booleana (TRUE). Ele não é traduzido em uma expressão matemática.

Função de Avaliação

Visão geral

- De forma generalizada, a **função de avaliação** retorna um valor específico que indica a qualidade de qualquer solução particular em sua representação (ou minimamente, permite a comparação da qualidade de duas soluções candidatas).

- A função de avaliação não é a mesma coisa que o objetivo.
- É um mapeamento do espaço de busca em um conjunto de valores:
  - A cada elemento do espaço de busca é associado um valor que indicará sua qualidade.
  - Permite comparar a importância das potenciais soluções.
- Pode ser do tipo:
  - Ordinal: estabelece um ranking das soluções.
  - Numérica: além do ranking, estabelece o *grau* de qualidade.

Função de Avaliação

Visão geral

- De forma generalizada, a **função de avaliação** retorna um valor específico que indica a qualidade de qualquer solução particular em sua representação (ou minimamente, permite a comparação da qualidade de duas soluções candidatas).

- A função de avaliação não é a mesma coisa que o objetivo.
- É um mapeamento do espaço de busca em um conjunto de valores:
  - A cada elemento do espaço de busca é associado um valor que indicará sua qualidade.
  - Permite comparar a importância das potenciais soluções.
- Pode ser do tipo:
  - Ordinal: estabelece um ranking das soluções.
  - Numérica: além do ranking, estabelece o *grau* de qualidade.

Função de Avaliação

TSP

- O objetivo é minimizar a distância total percorrida.
- Uma função de avaliação pode mapear uma solução na sua distância total.
- A comparação de duas soluções **não apenas informa qual é melhor, mas quanto é melhor**.

- Em alguns problemas pode ser (computacionalmente) difícil avaliar exatamente a qualidade de uma solução.
  - Às vezes basta uma aproximação.
  - Ou basta poder comparar duas soluções para saber qual é melhor.

Função de Avaliação

Considerações Gerais

- Em muitos problemas a função não é dada.
  - Ela deve então ser definida.
- Algumas regras:
  - Uma solução que atende o objetivo deve ter o melhor valor possível.
  - Uma que não atende não pode ter valor melhor.
- A função de avaliação deve ser apropriada:
  - para a representação usada
  - para o objetivo proposto
  - para os operadores que movem de uma solução para outra no espaço

Função de Avaliação

Considerações Gerais

- Em muitos problemas a função não é dada.
  - Ela deve então ser definida.
- Algumas regras:
  - Uma solução que atende o objetivo deve ter o melhor valor possível.
  - Uma que não atende não pode ter valor melhor.
- A função de avaliação deve ser apropriada:
  - para a representação usada
  - para o objetivo proposto
  - para os operadores que movem de uma solução para outra no espaço

Função de Avaliação

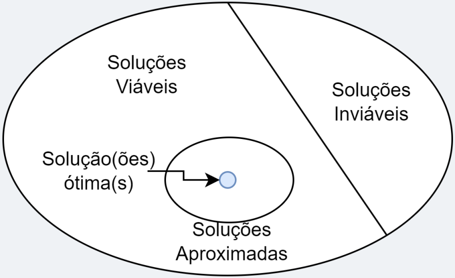
Considerações Gerais

- Em muitos problemas a função não é dada.
  - Ela deve então ser definida.
- Algumas regras:
  - Uma solução que atende o objetivo deve ter o melhor valor possível.
  - Uma que não atende não pode ter valor melhor.
- A função de avaliação deve ser apropriada:
  - para a representação usada
  - para o objetivo proposto
  - para os operadores que movem de uma solução para outra no espaço

Função de Avaliação

Considerações Gerais

- E se houver restrições?
  - Podem existir soluções inviáveis.
  - Existe o espaço de busca $\mathcal{S}$, e um subconjunto de soluções viáveis $\mathcal{F} \subseteq \mathcal{S}$.



- Soluções inviáveis, podem ser representadas com penalidades que deixam o valor numérico de uma função de avaliação muito alto (ou muito baixo) dependendo do tipo de problema.

Agenda

1. Conceitos Básicos
   - Representação
   - Objetivo
   - Função de Avaliação

2. Problema de Otimização
   - Definição
   - Vizinhança e Ótimo Local

Problema de Otimização (= de busca)

Definição

- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, encontrar uma solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- Obs: Esse é o caso para um problema de minimização.
- Para maximização a mudança seria a seguinte:

$$\text{avaliação}(x) \geq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- Os termos **problema de otimização** e **problema de busca** são considerados sinônimos.
- A busca por melhores soluções viáveis é o problema de otimização.

Problema de Otimização (= de busca)

Definição

- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, encontrar uma solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- Note que o objetivo não aparece! Aliás, o problema não aparece!
- Pode ser um SAT, TSP, etc..
- Tudo o que um algoritmo de otimização precisa saber é: como a solução é **representada**; a **função de avaliação**; e como **gerar soluções candidatas**.
- Mas se a função de avaliação não corresponder ao objetivo, estará buscando a resposta certa para o problema errado..

Problema de Otimização (= de busca)

Definição

- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, encontrar uma solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- Note que o objetivo não aparece! Aliás, o problema não aparece!
- Pode ser um SAT, TSP, etc..
- Tudo o que um algoritmo de otimização precisa saber é: como a solução é **representada**; a **função de avaliação**; e como **gerar soluções candidatas**.
- Mas se a função de avaliação não corresponder ao objetivo, estará buscando a resposta certa para o problema errado..

Problema de Otimização (= de busca)

Definição

- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, encontrar uma solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- O ponto x que satisfaz a condição acima é chamado o **ótimo global**, mas nem sempre é fácil encontrá-lo.
- Aliás, é difícil pro SAT, TSP, ..
- Entretanto, encontrar a *melhor solução* é mais fácil quando se considera apenas um pequeno subconjunto (viável ou não) do espaço de busca (**ótimo local**).
- Muitas técnicas de busca/otimização usam esse fato.

Problema de Otimização (= de busca)

Definição

- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, encontrar uma solução $x \in \mathcal{F}$ tal que

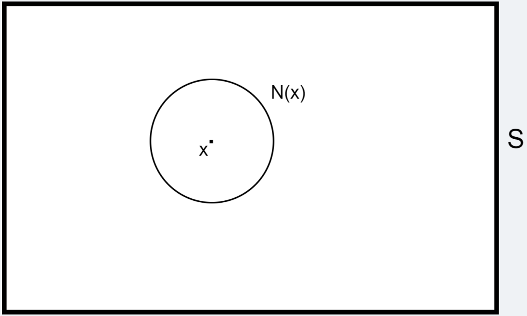
$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

- O ponto x que satisfaz a condição acima é chamado o **ótimo global**, mas nem sempre é fácil encontrá-lo.
- Aliás, é difícil pro SAT, TSP, ..
- Entretanto, encontrar a *melhor solução* é mais fácil quando se considera apenas um pequeno subconjunto (viável ou não) do espaço de busca (**ótimo local**).
- Muitas técnicas de busca/otimização usam esse fato.

Vizinhança (*Neighborhood*)

Vizinhança

- Se concentrarmos em uma região do espaço de busca **próxima** de alguma solução, é buscar na **vizinhança dessa solução**.



- **A vizinhança N(x) de x é o conjunto de pontos do espaço de busca S próximos de x por alguma medida.**

Vizinhança (*Neighborhood*)

Vizinhança

- A proximidade entre soluções pode ser definida de diferentes maneiras.
- Uma delas por exemplo, pode-se definir uma função de distância entre soluções:

$$\text{dist: } \mathcal{S} \times \mathcal{S} \to \mathbb{R}$$

  e então definir a vizinhança, para algum $\epsilon \geq 0$, como

$$\mathrm{N(x)} = \{y \in \mathcal{S} : \mathrm{dist(x,\ y)} \leq \epsilon\}$$

- Um y que satisfaz a condição é dito estar na vizinhança ϵ de x.

Vizinhança (*Neighborhood*)

Vizinhança

- A proximidade entre soluções pode ser definida de diferentes maneiras.
- Pode-se definir um mapeamento no espaço de busca:

$$\text{m: } \mathcal{S} \rightarrow 2^{\mathcal{S}}$$

e esse mapeamento define uma vizinhança para algum ponto $x \in S$.

TSP

- Por exemplo, define-se no TSP um mapeamento pelo operador *2-swap* que gera um novo conjunto de soluções potenciais a partir de uma solução corrente.
- Cada solução que pode ser gerada trocando-se duas cidades de lugar em uma rota (solução corrente), é dita estar na vizinhança dessa solução pelo operador *2-swap*.

Vizinhança (*Neighborhood*)

Vizinhança

- A proximidade entre soluções pode ser definida de diferentes maneiras.
- Pode-se definir um mapeamento no espaço de busca:

$$\text{m: } \mathcal{S} \rightarrow 2^{\mathcal{S}}$$

e esse mapeamento define uma vizinhança para algum ponto $x \in S$.

TSP

- Por exemplo, define-se no TSP um mapeamento pelo operador *2-swap* que gera um novo conjunto de soluções potenciais a partir de uma solução corrente.
- Cada solução que pode ser gerada trocando-se duas cidades de lugar em uma rota (solução corrente), é dita estar na vizinhança dessa solução pelo operador *2-swap*.

Vizinhança (*Neighborhood*)

TSP

- Em particular, uma solução x (uma permutação de $n = 20$ cidades):

  $15 - 3 - 11 - 19 - 17 - 2 - \cdots - 6$

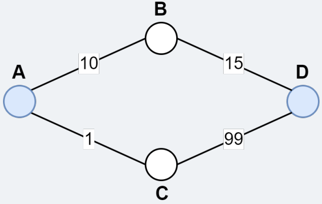
  possui $\frac{n(n-1)}{2}$ vizinhos, que incluem:

  - $15 - 17 - 11 - 19 - 3 - 2 - \cdots - 6$ (trocando cidades das posições 2 e 5).
  - $2 - 3 - 11 - 19 - 17 - 15 - \cdots - 6$ (trocando cidades das posições 1 e 6).
  - $15 - 3 - 6 - 19 - 17 - 2 - \cdots - 11$ (trocando cidades das posições 3 e 20).
  - etc.

Vizinhança (*Neighborhood*)

Nosso exemplo

- Se recordarem, o nosso exemplo com as cidades A - B - C - D:



  também definia uma mapeamento de vizinhança para encontrar uma solução melhor, que era, trocar uma cidade dentro da solução, com uma cidade que não está na solução.

- Solução atual: {A - C - D}
- Soluções vizinhas: {B - C - D}, {A - B - D} e {A - C - B}.
- **Obs:** Esse exemplo difere do TSP porque só queremos sair de A e chegar em D com a menor distância, e não visitar todas as cidades.

Vizinhança (*Neighborhood*)

SAT

- Para o SAT, poderíamos definir uma vizinhança (um mapeamento) do tipo *1-flip* que visa inverter 1 bit da solução corrente, para gerar novas soluções potenciais.
- Por exemplo, a seguinte solução corrente x (uma string binária de $n = 20$ valores):

  01101010001000011111

  possui n vizinhos, que incluem:

  - 11101010001000011111 (invertendo o bit 1).
  - 00101010001000011111 (invertendo o bit 2).
  - 01001010001000011111 (invertendo o bit 3).
  - etc.

Ótimo Local

Definição

- Com o conceito de vizinhança definido, agora podemos discutir o conceito de **ótimo local (*Local Optimum*)**.
- Matematicamente a definição é a seguinte:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- Em palavras, é a solução que tem o melhor (maior ou menor, dependendo do problema) valor entre todas as suas soluções vizinhas.
- Geralmente é fácil verificar se uma solução é ou não um ótimo local, quando a vizinhança é pequena.

Ótimo Local

Definição

- Com o conceito de vizinhança definido, agora podemos discutir o conceito de **ótimo local** (***Local Optimum***).
- Matematicamente a definição é a seguinte:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- Em palavras, é a solução que tem o melhor (maior ou menor, dependendo do problema) valor entre todas as suas soluções vizinhas.
- Geralmente é fácil verificar se uma solução é ou não um ótimo local, quando a vizinhança é pequena.

Ótimo Local

Definição

- Com o conceito de vizinhança definido, agora podemos discutir o conceito de **ótimo local (*Local Optimum*)**.
- Matematicamente a definição é a seguinte:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- Em palavras, é a solução que tem o melhor (maior ou menor, dependendo do problema) valor entre todas as suas soluções vizinhas.
- Geralmente é fácil verificar se uma solução é ou não um ótimo local, quando a vizinhança é pequena.

Ótimo Local

Definição

- Comparando as definições de **ótimo local** e **ótimo global**:
- **Ótimo Local**:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- **Ótimo Global**:
- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, o **ótimo global** é a solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

Ótimo Local

Definição

- Comparando as definições de **ótimo local** e **ótimo global**:
- **Ótimo Local**:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- **Ótimo Global**:
- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, o **ótimo global** é a solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

Ótimo Local

Definição

- Comparando as definições de **ótimo local** e **ótimo global**:
- **Ótimo Local**:
- Uma solução x é um **ótimo local** em uma vizinhança N se e somente se

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in N(x)$$

- **Ótimo Global**:
- Dado um espaço de busca $\mathcal{S}$ e sua parte viável $\mathcal{F} \subseteq \mathcal{S}$, o **ótimo global** é a solução $x \in \mathcal{F}$ tal que

$$\text{avaliação}(x) \leq \text{avaliação}(y), \forall y \in \mathcal{F}$$

Vizinhança, Ótimo Local e Global

Exemplo Gráfico

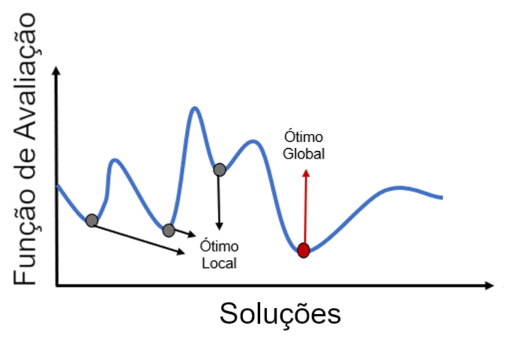
- Na imagem, a curva em azul representa cada uma das soluções possíveis de um conjunto, e o seu valor resultante da função de avaliação.

- Dada uma **solução inicial**, a **vizinhança** está definida como aquelas soluções que são **próximas** desta solução (círculo pontilhado).

Vizinhança, Ótimo Local e Global

Exemplo Gráfico

- Nessa outra imagem, considerando um **problema de minimização**, os **ótimos locais** são representados.

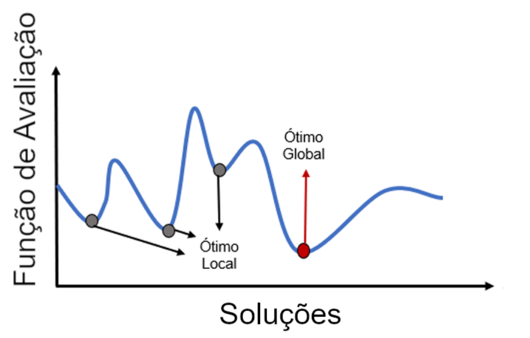


- Pode-se notar que as soluções próximas (vizinhas) a esses pontos tem valores maiores conforme a função de avaliação. Como se os ótimos locais então em **vales** de um relevo.

Vizinhança, Ótimo Local e Global

Exemplo Gráfico

- Ainda considerando um **problema de minimização**, o **ótimo global** também está representado.

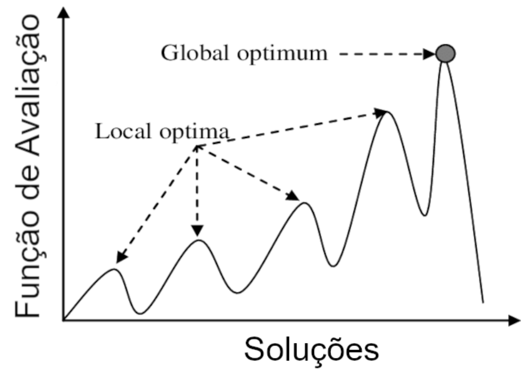


- **Ele possui o menor valor da função de avaliação dentre todo o conjunto.**

Vizinhança, Ótimo Local e Global

Exemplo Gráfico

- Se consideramos um **problema de maximização**, esses ótimos mudam.



- Da mesma forma do relevo, agora eles passam a ocupar os **picos** dessa região.

Vizinhança e Ótimos

Considerações

- Muitos métodos de otimização baseiam-se em vizinhança
  - Geram uma sequência de soluções buscando na vizinhança, ou seja, usam informação local em cada passo da busca.
  - São chamados de métodos ou estratégias de **busca local**.

Exemplo

- Maximizar $F(x) = -x^2$
  - A própria função $F(x)$ pode ser usada para avaliação.
  - Seja $x = 2$ sua melhor solução até o momento; avaliação $F(2) = -4$
  - Pode-se definir a vizinhança ϵ de x como o intervalo 0.1 de cada lado
  - Escolhe-se um ponto x' no intervalo $[1.9; 2.1]$
  - Se $F(x')$ for melhor que $F(2)$, substitui $x = x'$; senão escolhe outro

Vizinhança e Ótimos

Considerações

- Muitos métodos de otimização baseiam-se em vizinhança
  - Geram uma sequência de soluções buscando na vizinhança, ou seja, usam informação local em cada passo da busca.
  - São chamados de métodos ou estratégias de **busca local**.

Exemplo

- Maximizar $F(x) = -x^2$
  - A própria função $F(x)$ pode ser usada para avaliação.
  - Seja $x = 2$ sua melhor solução até o momento; avaliação $F(2) = -4$
  - Pode-se definir a vizinhança ϵ de x como o intervalo 0.1 de cada lado
  - Escolhe-se um ponto x' no intervalo [1.9; 2.1]
  - Se $F(x')$ for melhor que $F(2)$, substitui $x = x'$; senão escolhe outro

Vizinhança e Ótimos

Considerações

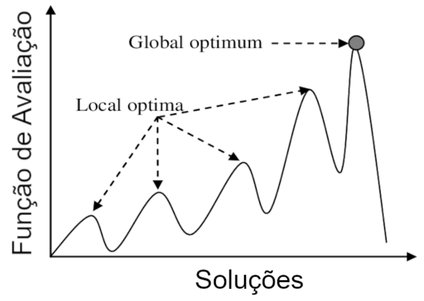
- É claro que funções de avaliação em problemas reais não são tão simples.
- $F(x) = -x^2$ é quadrática com um único máximo ($x = 0$).

- Em problemas reais teríamos uma superfície com vários picos e vales (**como na imagem!**).
- Encontrar o máximo é como chegar ao topo de uma cadeia de montanhas numa neblina forte.

Vizinhança e Ótimos

Considerações

- Encontrar o máximo é como chegar ao topo de uma cadeia de montanhas numa neblina forte.
  - Você consegue enxergar apenas a proximidade, ou seja, fazer apenas avaliações locais, para decidir o próximo passo (para onde caminhar).
  - Se caminhar sempre para cima, ao fim chegará num pico.
  - Mas pode não ser o pico mais alto (que era o objetivo).
  - Pode ser necessário descer para subir novamente em outro local.



Vizinhança e Ótimos

Considerações

- Vizinhança pequena:
  - Possibilidade de verificar toca a vizinhança
  - Poucas soluções avaliadas para tomar um passo
  - Mas grande chance de ficar preso em ótimo local
- Vizinhança grande:
  - Possibilidade de tomar decisões mais acertadas
- Vizinhança irrestrita (sem restrições/limites):
  - Garantia de ótimo global
  - **Mas computacionalmente inviável verificar toda a vizinhança!**.
- O tamanho da vizinhança deve ser apropriado para o problema.

Vizinhança e Ótimos

Considerações

- Vizinhança pequena:
  - Possibilidade de verificar toca a vizinhança
  - Poucas soluções avaliadas para tomar um passo
  - Mas grande chance de ficar preso em ótimo local
- Vizinhança grande:
  - Possibilidade de tomar decisões mais acertadas
- Vizinhança irrestrita (sem restrições/limites):
  - Garantia de ótimo global
  - **Mas computacionalmente inviável verificar toda a vizinhança!**.
- O tamanho da vizinhança deve ser apropriado para o problema.

Vizinhança e Ótimos

Considerações

- Vizinhança pequena:
  - Possibilidade de verificar toca a vizinhança
  - Poucas soluções avaliadas para tomar um passo
  - Mas grande chance de ficar preso em ótimo local
- Vizinhança grande:
  - Possibilidade de tomar decisões mais acertadas
- Vizinhança irrestrita (sem restrições/limites):
  - Garantia de ótimo global
  - **Mas computacionalmente inviável verificar toda a vizinhança!**.
- O tamanho da vizinhança deve ser apropriado para o problema.

Vizinhança e Ótimos

Considerações

- Vizinhança pequena:
  - Possibilidade de verificar toca a vizinhança
  - Poucas soluções avaliadas para tomar um passo
  - Mas grande chance de ficar preso em ótimo local
- Vizinhança grande:
  - Possibilidade de tomar decisões mais acertadas
- Vizinhança irrestrita (sem restrições/limites):
  - Garantia de ótimo global
  - **Mas computacionalmente inviável verificar toda a vizinhança!**.
- O tamanho da vizinhança deve ser apropriado para o problema.

Próxima Aula

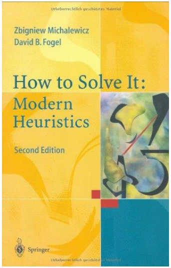
Próxima Aula

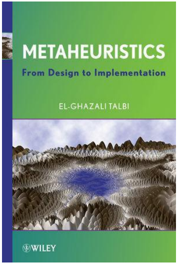
- Na próxima aula falaremos de:
  - Busca Exaustiva
  - Busca Local

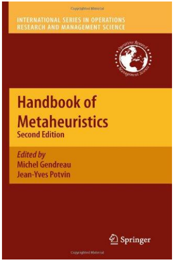
Bibliografias

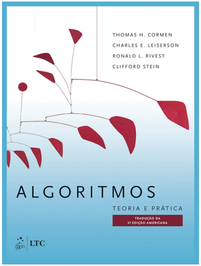
Bibliografia Básica

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









Bibliografias

Bibliografia Complementar

1. GLOVER, Fred; KOCHENBERGER, Gary A. (ed.). Handbook of metaheuristics. Boston: Kluwer, 2003. 556 p. (International series in operations research & management science ; 57).
2. BLUM, Christian Et Al. Hybrid metaheuristics: an emerging approach to optimization. Berlin: Springer 2008 289 p. (Studies in Computational intelligence; 114).
3. DOERNER, Karl F. (ed.) Et Al. Metaheuristics: progress in complex systems optimization. New York: Springer 2007 408 p. (Operations research / computer science interfaces series).
4. GLOVER, Fred; LAGUNA, Manuel. Tabu search. Boston: Kluwer Academic, 1997. 382 p.
5. AARTS, Emile. Local search in combinatorial optimization. Princeton: Princeton University Press, 2003 512 p.
6. 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.