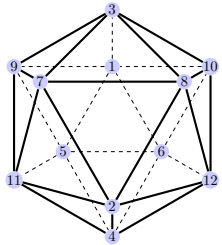
Metaheuristicas - 5 - Alg. de solução única: Conceitos

Alexandre Checoli Choueiri

20/03/2023



1. Introdução
2. Vizinhança
3. Ótimo local
4. Fitness landscape (paisagem de adaptabilidade)
5. Atividades

Introdução

Introdução

A idéia geral

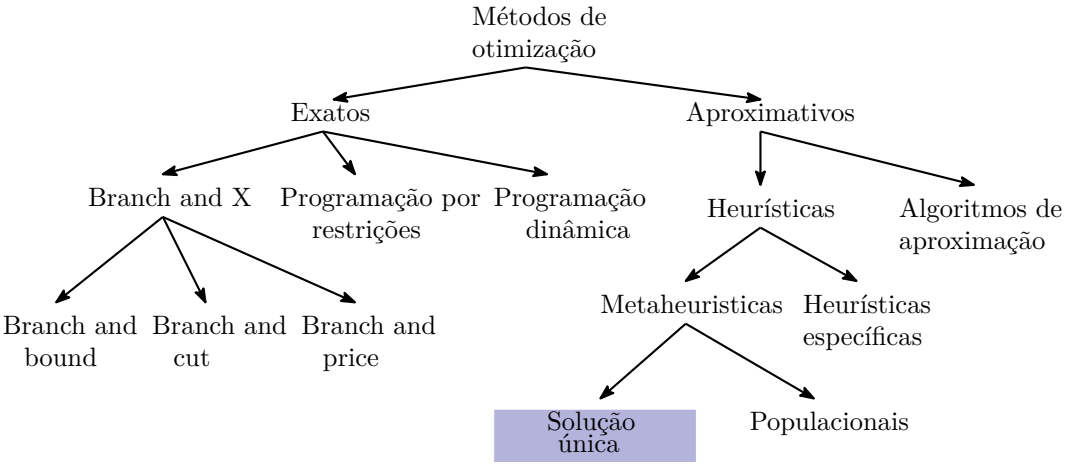
Como vimos, as metaheuristicas precisam de uma (ou mais) solução inicial para começarem o processo de otimização. Aprendemos os conceitos dos algoritmos gulosos para geração de soluções iniciais, de forma que podemos começar a estudar uma grande família de metaheuristicas: **algoritmos de solução única**.

Como vimos, as metaheuristicas precisam de uma (ou mais) solução inicial para começarem o processo de otimização. Aprendemos os conceitos dos algoritmos gulosos para geração de soluções iniciais, de forma que podemos começar a estudar uma grande família de metaheuristicas: **algoritmos de solução única**.

Como o nome sugere, essa família de metaheuristicas sempre trabalha com **uma solução** a cada iteração.

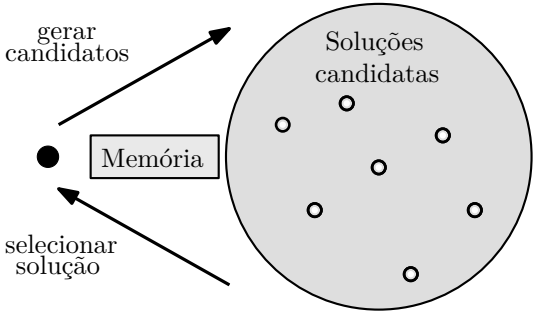
Introdução

Na árvore de métodos



Introdução

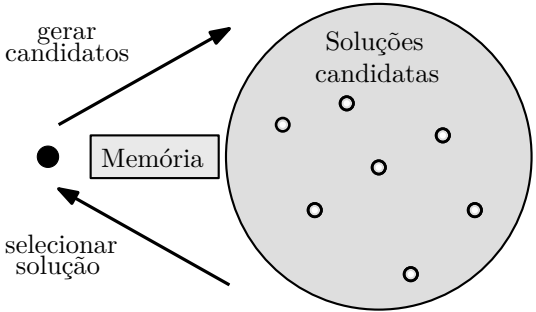
A idéia geral



Alguns conceitos são comuns a todos os algoritmos de solução única. Iterativamente uma heurística gera novas soluções, e a partir delas faz a reposição da solução atual.

Introdução

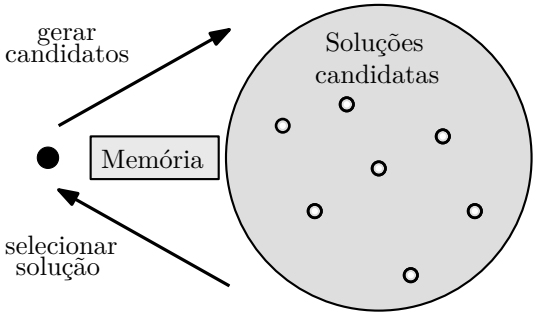
A idéia geral



Na etapa de geração, um conjunto de soluções é gerado a partir da solução atual s. O conjunto $C(s)$ é geralmente obtido a partir de transformações locais na solução s.

Introdução

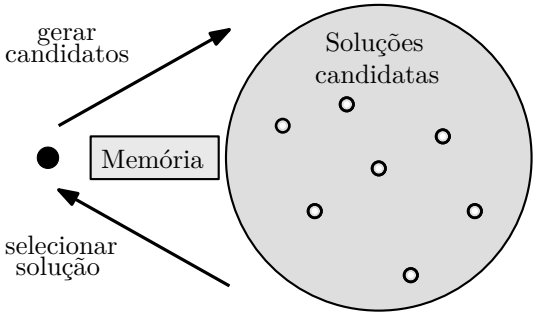
A idéia geral



Na etapa de reposição, uma seleção é feita dentre os elementos do conjunto de candidatos $C(s)$ para substituir a solução atual s, ou seja, uma solução $s' \in C(S)$ é selecionada para ser a nova solução. O processo é repetido até que um critério de parada seja atingido.

Introdução

A idéia geral



A geração e reposição podem ser processos sem memória, ou seja, os dois procedimentos são baseados exclusivamente na solução atual. Do contrário, um histórico de buscas passadas é armazenado e utilizado nas fases de geração/reposição.

Introdução

Algoritmo genérico solução única

O algoritmo abaixo ilustra um *template* de alto nível para todas as metaheurísticas de solução única.

Algorithm 1 Template de alto nível: algoritmos de solução única

Input: solução inicial s_0
$t = 0$
while Critério de parada = Falso **do**
  $C =$ GeraConjunto(s_t)
  $s_{t+1} =$ SelecionaElemento(C)
  $t = t + 1$
end while
return s_{t+1}

Introdução

Conceitos comuns

Dois conceitos são comuns a todas as metaheuristicas de solução única:

1. A geração da solução inicial (já vimos uma família de algoritmos).
2. A estrutura de vizinhança.

Vizinhança

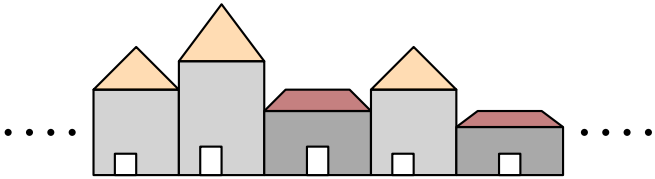
Vizinhança

O que é um vizinho?

A primeira pergunta que temos que responder é muito simples...

Vizinhança

O que é um vizinho?

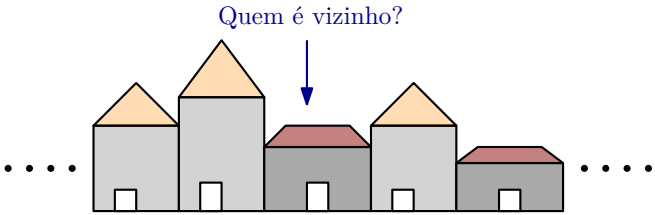


A primeira pergunta que temos que responder é muito simples...

O que é um vizinho?

Vizinhança

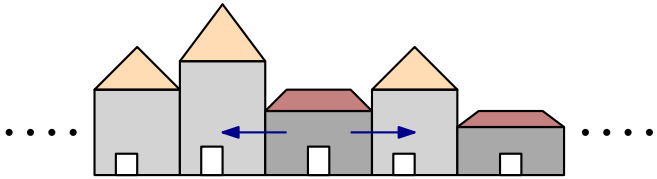
O que é um vizinho?



Considerando a casa acima, quem são seus vizinhos?

Vizinhança

O que é um vizinho?



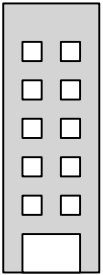
Em um primeiro momento podemos responder de pronto:

1. Vizinho é aquele que está ao lado

De forma que **a casa acima possui 2 vizinhos**.

Vizinhança

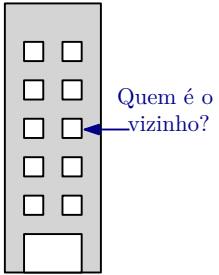
O que é um vizinho?



E agora? Podemos usar a mesma definição de vizinho?

Vizinhança

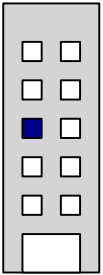
O que é um vizinho?



Quais são os vizinhos do apartamento acima?

Vizinhança

O que é um vizinho?



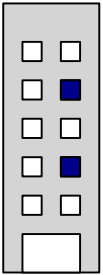
Usando a nossa primeira definição:

1. Vizinho é aquele que está ao lado

O apartamento acima possui somente um vizinho.

Vizinhança

O que é um vizinho?



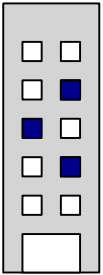
Porém, podemos alterar a nossa definição de vizinho:

1. Vizinho é aquele que está acima ou abaixo

O que se desdobra em 2 outros vizinhos.

Vizinhança

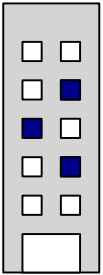
O que é um vizinho?



Qual poderia ser a definição de vizinho, para que os apartamentos destacados sejam considerados como vizinhos?

Vizinhança

O que é um vizinho?



1. Vizinho é aquele que está a "1 apartamento de distância"em qualquer direção.

Essa definição engloba os 3 apartamentos acima destacados.

Vizinhança

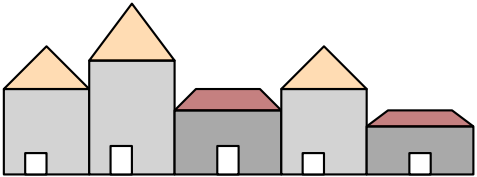
O que é um vizinho?

Conclusões

1. Vizinho ou vizinhança são conceitos que suportam mais de uma definição.
2. Para cada definição de vizinhança, conjuntos diferentes de elementos podem ser considerados como vizinhos.

Vizinhança

O jogo da carta e do vizinho

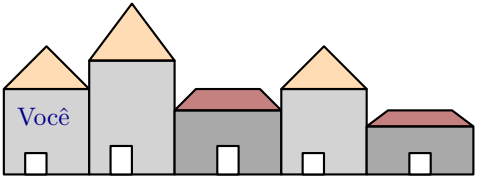


Com esse novo conhecimento, vamos determinar um novo jogo, chamado:

O JOGO DA CARTA E DO VIZINHO

Vizinhança

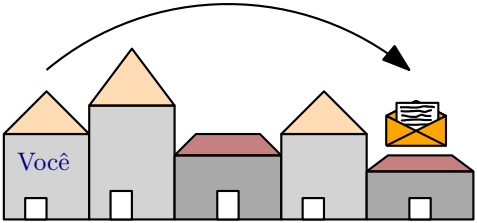
O jogo da carta e do vizinho



Neste jogo houve um evento cataclismático na terra, e não existe mais internet. Você mora na casa indicada acima.

Vizinhança

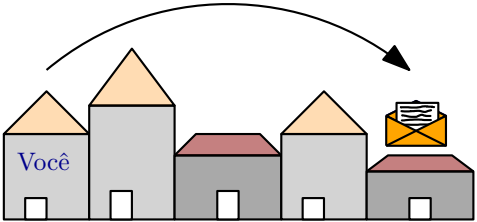
O jogo da carta e do vizinho



A única forma de comunicação é por cartas que são entregues de vizinho para vizinho, até chegar no destino.

Vizinhança

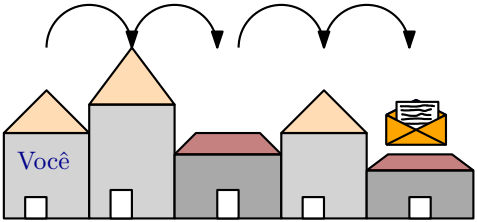
O jogo da carta e do vizinho



Considerando você e onde a carta deve chegar, o jogo consiste em determinar a definição de vizinhança adequada para que a carta chegue ao destino. Para o exemplo acima, qual poderia ser uma definição que ganhe o jogo?

Vizinhança

O jogo da carta e do vizinho

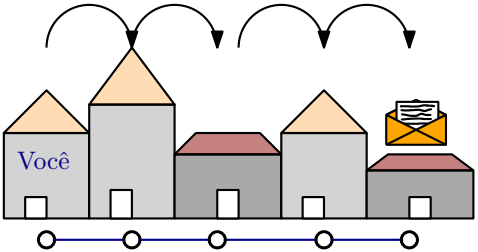


A primeira que encontramos:

1. Vizinho é aquele que está ao lado

Vizinhança

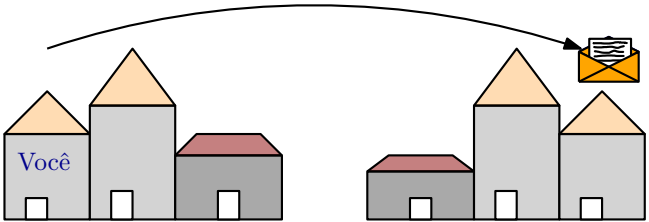
O jogo da carta e do vizinho



Com essa definição a sua carta chega ao destino final.

Vizinhança

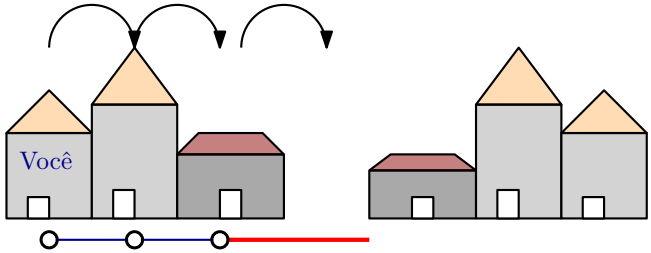
O jogo da carta e do vizinho



Considerando a mesma vizinhança definida anteriormente, você consegue levar a carta ao destino nesse caso?

Vizinhança

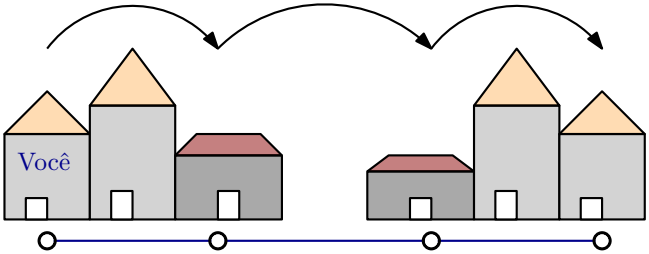
O jogo da carta e do vizinho



Não é possível! **Como podemos definir uma estrutura de vizinhança neste caso para vencermos o jogo**?

Vizinhança

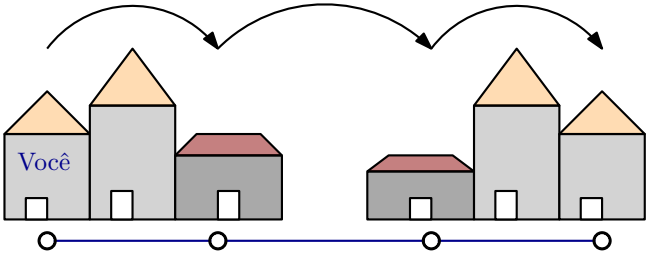
O jogo da carta e do vizinho



1. Vizinho é aquele que está a "2 casas de distância"em todos os lados.

Vizinhança

O jogo da carta e do vizinho



Com essa definição conseguimos alcançar o nosso objetivo.

Vizinhança

O jogo da carta e do vizinho

Agora chegamos no nível **Dark Souls** do jogo da vizinhança.

Vizinhança

O jogo da carta e do vizinho

Considere a sua localização e o local onde a carta deve ser entregue.

Vizinhança

O jogo da carta e do vizinho

Qual a vizinhança que ganha o jogo?

Vizinhança

O jogo da carta e do vizinho

Ora, mas essa é muito fácil, me dê uma mais difícil!

1. Vizinho é aquele que está a "no máximo 6.378 km (raio da terra) de distância"em todos os lados.

Vizinhança

O jogo da carta e do vizinho

Intuitivamente não podemos aceitar essa vizinhança... **a vizinhança deve ser algo local/próximo do ponto considerado**.

Conclusões

1. Vizinho ou vizinhança são conceitos que suportam mais de uma definição.
2. Para cada definição de vizinhança, conjuntos diferentes de elementos podem ser considerados como vizinhos.
3. A **distância** é uma parte importante na definição de uma vizinhança.
4. De comum acordo (sem uma formalização), intuímos que uma vizinhança deve ser algo **próximo**.

Vizinhança

Por quê tudo isso?

Mas por quê estamos discutindo conceitos abstratos de vizinhança?

Vizinhança

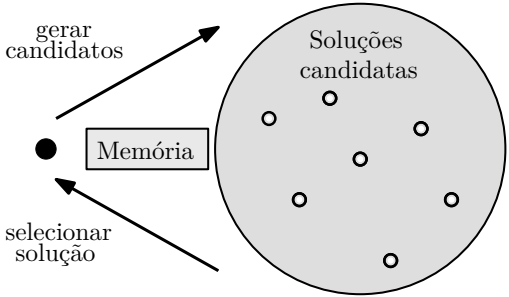
Por quê tudo isso?

Mas por quê estamos discutindo conceitos abstratos de vizinhança?

Analisemos novamente a ideia geral de uma metaheurística de solução única. **Qual a relação de vizinhos com o esquema**?

Vizinhança

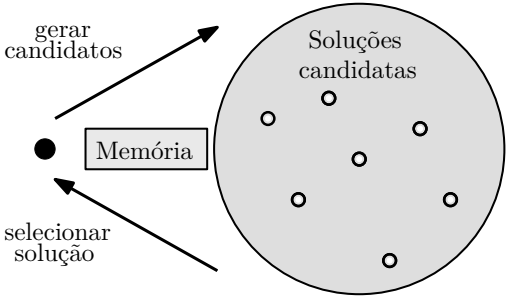
Por quê tudo isso?



Na etapa de geração, um conjunto de soluções é gerado a partir da solução atual s. O conjunto $C(s)$ é geralmente obtido a partir de transformações locais na solução s.

Vizinhança

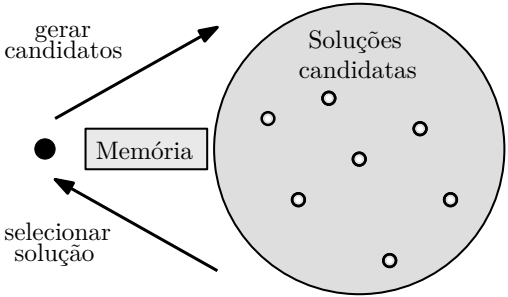
Por quê tudo isso?



Na etapa de geração, um conjunto de soluções é gerado a partir da solução atual s. O conjunto $C(s)$ é geralmente obtido a partir de **transformações locais na solução** s.

Vizinhança

Por quê tudo isso?

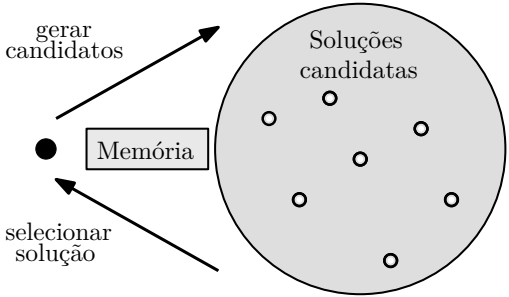


Essas transformações locais em s podem ser pensadas como soluções próximas a s, ou ainda,

soluções vizinhas de s

Vizinhança

Definições formais



Ou seja, o conceito de vizinhança também é usado para soluções de problemas de otimização. Vejamos as definições formais.

Vizinhança

Definições formais

Definição

Vizinhança (Neighborhood): Uma função vizinhança N é um mapeamento $N : S \rightarrow 2^S$ que atribui a cada solução $s \in S$ um conjunto de soluções $N(s) \subset S$

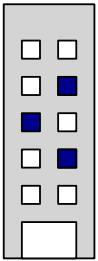
Vizinhança

Definições formais

Definição

Vizinhança (Neighborhood): Uma função vizinhança N é um mapeamento $N : S \rightarrow 2^S$ que atribui a cada solução $s \in S$ um conjunto de soluções $N(s) \subset S$

Você consegue estabelecer uma relação entre essa definição e o exemplo inicial dos vizinhos em um prédio?



A definição de vizinhança depende do tipo de problema de otimização. Primeiramente ela foi definida para os problemas contínuos.

Definição

Vizinhança em problemas contínuos: A vizinhança $N(s)$ em um espaço contínuo é a bola com centro s e raio igual a $\epsilon > 0$.

Note que existe uma dimensão de **distância** inerente ao problema: o raio é definido em função de uma determinada distância (dentre muitas possíveis).

Vizinhança

Definições formais

Se definirmos a distância pela **Norma euclidiana**, temos que a distância entre duas soluções s e s' é dada por:

Vizinhança

Definições formais

Se definirmos a distância pela **Norma euclidiana**, temos que a distância entre duas soluções s e s' é dada por:

$$||s' - s|| = \sqrt{(s'_1 - s_1)^2 + (s'_2 - s_2)^2 + ...(s'_n - s_n)^2}$$

Seja $s = [2, 3]$ e $s' = [3, 4]$, qual a distância entre as soluções?

Vizinhança

Definições formais

Se definirmos a distância pela **Norma euclidiana**, temos que a distância entre duas soluções s e s' é dada por:

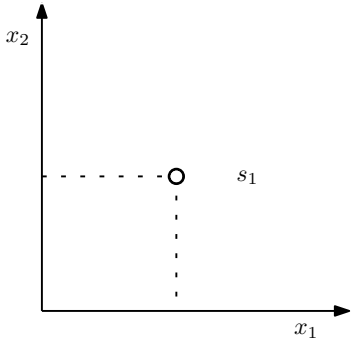
$$||s' - s|| = \sqrt{(s'_1 - s_1)^2 + (s'_2 - s_2)^2 + ...(s'_n - s_n)^2}$$

Seja $s = [2, 3]$ e $s' = [3, 4]$, qual a distância entre as soluções?

$$||s' - s|| = \sqrt{(3-2)^2 + (4-3)^2} = \sqrt{2}$$

Vizinhança

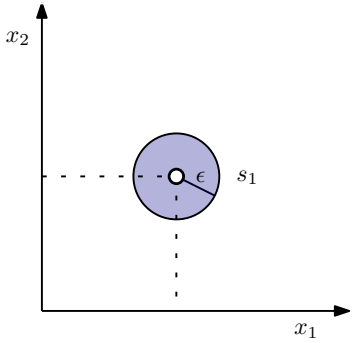
Definições formais



Em duas dimensões, com a norma euclidiana, podemos representar a vizinhança de uma solução s simplesmente fixando uma distância ϵ, e verificando todas as soluções que não ultrapassam essa distância.

Vizinhança

Definições formais



O que gera de fato um raio de vizinhos de s (todas as soluções dentro do círculo na imagem acima).

Vizinhança

Definições formais

Também podemos definir vizinhança para problemas de otimização discretos (os que vamos tratar na disciplina), como:

Definição

Vizinhança em problemas discretos: Em um problema de otimização discreta, a vizinhança $N(s)$ de uma solução s é representada pelo conjunto $\{s' : d(s', s) \leq \epsilon\}$, em que d é o **tipo de distância** relacionada com o **operador de movimento.**

Vizinhança

Definições formais

Também podemos definir vizinhança para problemas de otimização discretos (os que vamos tratar na disciplina), como:

Definição

Vizinhança em problemas discretos: Em um problema de otimização discreta, a vizinhança $N(s)$ de uma solução s é representada pelo conjunto $\{s' : d(s', s) \leq \epsilon\}$, em que d é o **tipo de distância** relacionada com o **operador de movimento.**

A definição de vizinhança em problemas discretos **depende fortemente das estruturas usadas para a representação das soluções**. Uma definição de distância pode **servir para alguns problemas e não servir para outros**.

Vizinhança

Definições formais

Também podemos definir vizinhança para problemas de otimização discretos (os que vamos tratar na disciplina), como:

Definição

Vizinhança em problemas discretos: Em um problema de otimização discreta, a vizinhança $N(s)$ de uma solução s é representada pelo conjunto $\{s' : d(s', s) \leq \epsilon\}$, em que d é o **tipo de distância** relacionada com o **operador de movimento.**

O operador de movimento é definido como a operação que realizamos em uma solução s_1 para transformá-la em uma nova solução s_2.

Vizinhança

Definições formais

EXEMPLO: Considerando o problema da mochila com 4 itens e a representação binária de soluções. Podemos ter a solução s_1 abaixo:

$$s_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

EXEMPLO: Considerando o problema da mochila com 4 itens e a representação binária de soluções. Podemos ter a solução s_1 abaixo:

$$s_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right]$$

Uma distancia muito utilizada para problemas com representações binárias é a **distância de Hamming**. A distância de Hamming consiste na soma do número de posições em que duas soluções diferem entre si. Considerando a solução s_1 e a solução s_2 dada por:

$$s_2 = \left[\begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \right]$$

Qual a distância entre as duas soluções?

Vizinhança

Definições formais

Verificamos quais elementos das duas soluções são diferentes:

$$s_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right]$$

$$s_2 = \left[\begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \right]$$

Só existe 1 bit de diferença entre as duas soluções, de forma que $d(s_1, s_2) = 1$. Lembre-se que a vizinhança de uma solução s são todas as soluções que estão a uma distância $\leq \epsilon$. Se considerarmos a distância de **Hamming** e $\epsilon = 1$, qual poderia ser um **operador de movimento** que ao ser aplicado em uma solução s gera uma solução vizinha s'?

Vizinhança

Exemplo mochila

Um operador de movimento clássico para problemas com codificação binária é o **flip**. O operador *flip* inverte o elemento de uma solução: por exemplo, se ele era 0 vira 1, se era 1 vira 0.

Vizinhança

Exemplo mochila

Um operador de movimento clássico para problemas com codificação binária é o **flip**. O operador *flip* inverte o elemento de uma solução: por exemplo, se ele era 0 vira 1, se era 1 vira 0.

$$s_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right]$$

Vizinhança

Exemplo mochila

Um operador de movimento clássico para problemas com codificação binária é o **flip**. O operador *flip* inverte o elemento de uma solução: por exemplo, se ele era 0 vira 1, se era 1 vira 0.

$$s_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right] \rightarrow \text{flip}(s_1,3)$$

Vizinhança

Exemplo mochila

Um operador de movimento clássico para problemas com codificação binária é o **flip**. O operador *flip* inverte o elemento de uma solução: por exemplo, se ele era 0 vira 1, se era 1 vira 0.

$$s_1 = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right] \rightarrow \text{flip}(s_1,3) \rightarrow \left[\begin{array}{cccc} 1 & 0 & 0 & 1 \end{array} \right]$$

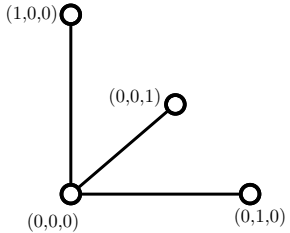
Vizinhança

Exemplo mochila

EXERCÍCIO: Considerando a distância Hamming com $\epsilon = 1$, o operador *flip* e a solução inicial binária s mostrada acima, quais são seus vizinhos?

Vizinhança

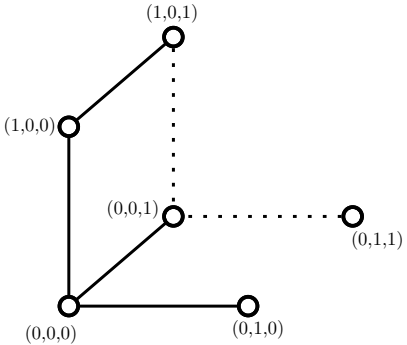
Exemplo mochila



Geramos 3 vizinhos da solução inicial. Agora quais são os vizinhos da solução $(0, 0, 1)$?

Vizinhança

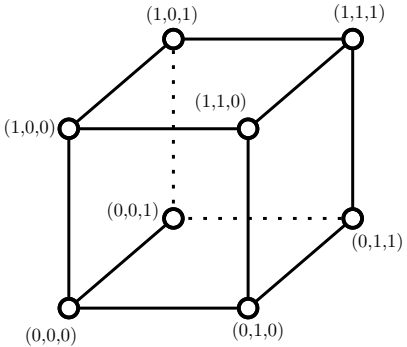
Exemplo mochila



E assim podemos gerar todas as soluções da vizinhança a partir do operador **flip**, da distancia Hamming e $\epsilon = 1$.

Vizinhança

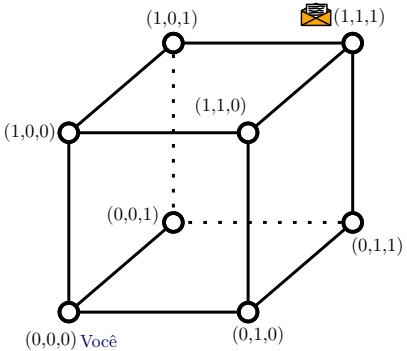
Exemplo mochila



E assim podemos gerar todas as soluções da vizinhança a partir do operador **flip**, da distancia Hamming e $\epsilon = 1$.

Vizinhança

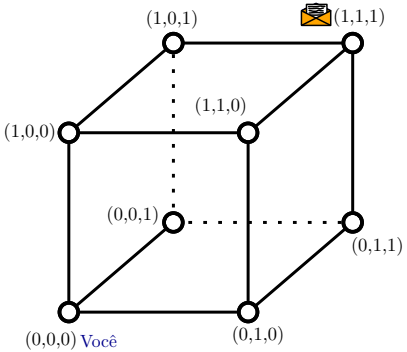
Exemplo mochila



Achou que toda aquela história do jogo do vizinho e da carta era a toa?

Vizinhança

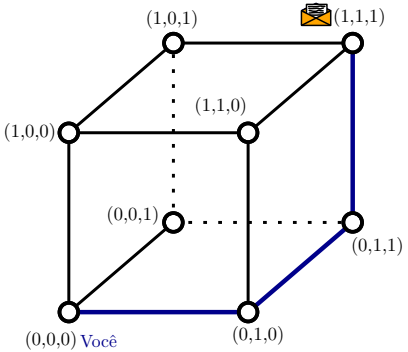
Exemplo mochila



Com a nossa definição de vizinhança (distancia, operador e ϵ), é possível entregar a carta?

Vizinhança

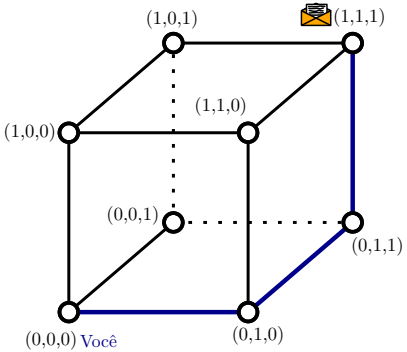
Exemplo mochila



Com a nossa definição de vizinhança (distancia, operador e ϵ), é possível entregar a carta?
SIM!

Vizinhança

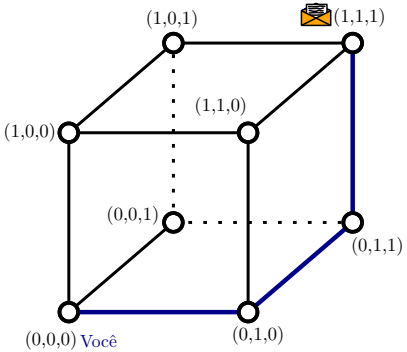
Exemplo mochila



Usamos o operador *flip* nesta vizinhança. Quantos movimentos foram necessários para sairmos de $(0, 0, 0)$ e chegarmos a $(1, 1, 1)$?

Vizinhança

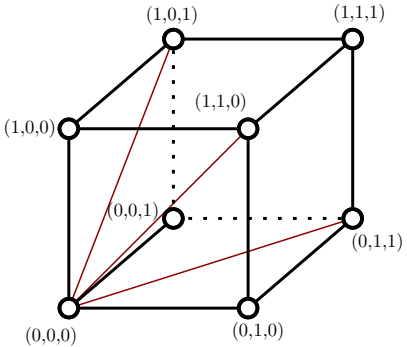
Exemplo mochila



Usamos o operador *flip* nesta vizinhança. Quantos movimentos foram necessários para sairmos de $(0, 0, 0)$ e chegarmos a $(1, 1, 1)$? É possível determinar uma vizinhança em que esse caminho seja feito com 2 movimentos?

Vizinhança

Exemplo mochila



Se alterarmos $\epsilon = 2$ geramos novos vizinhos, o que encurta o caminho até a solução.

Vizinhança

Exemplo TSP

EXEMPLO: Considere o problema do caixeiro viajante com 3 cidades. Pense em uma vizinhança para o problema (distância, ϵ, operador de movimento).

Vizinhança

Exemplo TSP

EXEMPLO: Considere o problema do caixeiro viajante com 3 cidades. Pense em uma vizinhança para o problema (distância, ϵ, operador de movimento).
Uma representação para a solução do TSP com n cidade é uma permutação de n elementos:

$$\pi = \{\pi_1, \pi_2, ..., \pi_n\} \tag{1}$$

E pode ser modelada como um vetor de inteiros, um para cada ponto. Um operador comum em problemas permutacionais é o **swap**.

Vizinhança

Exemplo TSP

EXEMPLO: Considere o problema do caixeiro viajante com 3 cidades. Pense em uma vizinhança para o problema (distância, ϵ, operador de movimento).
Uma representação para a solução do TSP com n cidade é uma permutação de n elementos:

$$\pi = \{\pi_1, \pi_2, ..., \pi_n\} \tag{1}$$

E pode ser modelada como um vetor de inteiros, um para cada ponto. Um operador comum em problemas permutacionais é o **swap**.

Dada uma permutação π, um movimento *swap(i,j)* consiste em trocar os elementos π_i e π_j de posição.

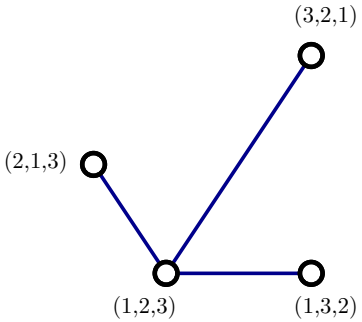
Vizinhança

Exemplo TSP

Considerando o operador swap, $\epsilon = 2$ e a distância Hamming, **construa a vizinhança para o TSP com 3 cidades** (construa toda a vizinhança, vizinhos, vizinhos de vizinhos, etc...).

Vizinhança

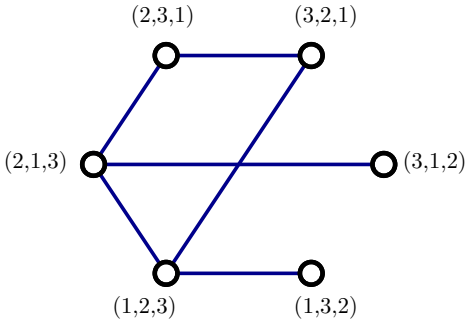
Exemplo TSP



Considerando o operador swap, $\epsilon = 2$ e a distância Hamming, **construa a vizinhança para o TSP com 3 cidades** (construa toda a vizinhança, vizinhos, vizinhos de vizinhos, etc...).

Vizinhança

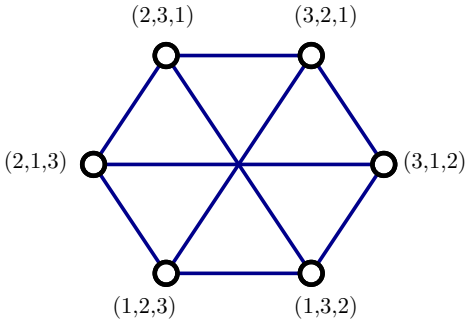
Exemplo TSP



Considerando o operador swap, $\epsilon = 2$ e a distância Hamming, **construa a vizinhança para o TSP com 3 cidades** (construa toda a vizinhança, vizinhos, vizinhos de vizinhos, etc...).

Vizinhança

Exemplo TSP



Considerando o operador swap, $\epsilon = 2$ e a distância Hamming, **construa a vizinhança para o TSP com 3 cidades** (construa toda a vizinhança, vizinhos, vizinhos de vizinhos, etc...).

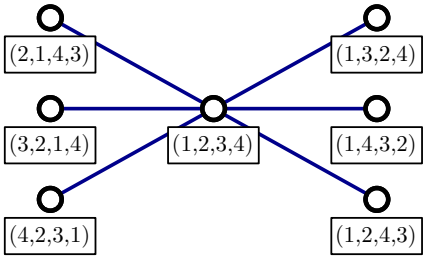
Vizinhança

Exemplo TSP

Considerando o operador swap, $\epsilon = 2$ e a distância Hamming, **construa a vizinhança da solução acima** (somente da solução).

Vizinhança

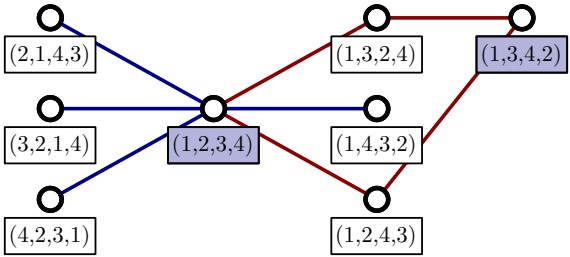
Exemplo TSP



Determine pelo menos 2 **caminhos** usando a vizinhança, partindo de $(1, 2, 3, 4)$ até $(1, 3, 4, 2)$. Quantos movimentos foram necessários?

Vizinhança

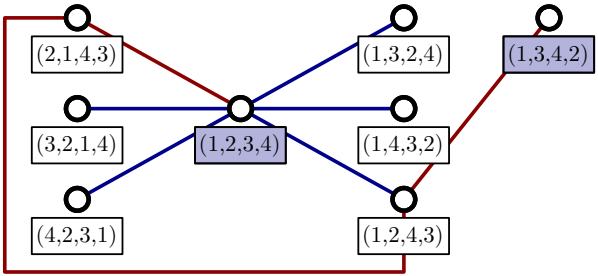
Exemplo TSP



Desta forma 2 movimentos são necessários nos 2 caminhos. Encontre um caminho em que seja necessário pelo menos 3 movimentos.

Vizinhança

Exemplo TSP



Neste caminho usamos 3 movimentos swap.

Conclusão

1. Ao determinar uma vizinhança, implicitamente estamos criando caminhos entre soluções vizinhas. Esses caminhos são percorridos ao executarmos o operador de movimento nas soluções. **Isso implica que existem diversos caminhos possíveis entre duas soluções s e s'.**

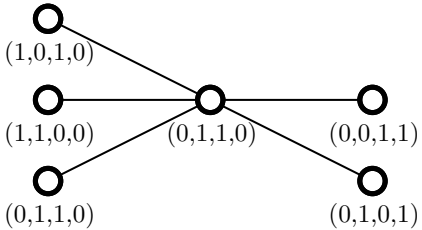
Vizinhança

Exemplo mochila

EXEMPLO: Considere o problema da mochila, o operador **swap**, a distância Hamming e $\epsilon = 2$. Crie a vizinhança da solução acima.

Vizinhança

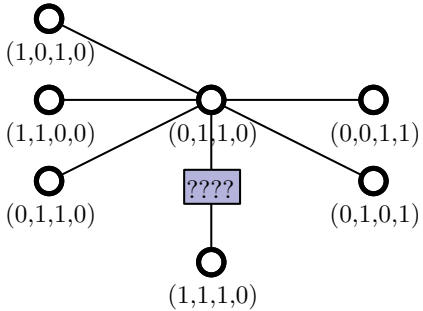
Exemplo mochila



A vizinhança da solução fica da seguinte forma.

Vizinhança

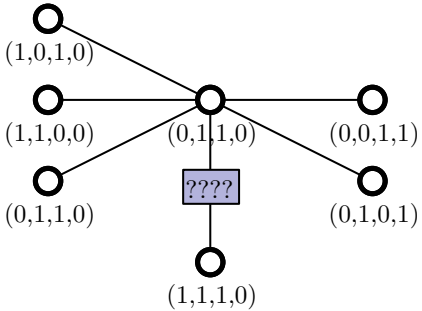
Exemplo mochila



Com essa definição de vizinhança, é possível, partindo de $s = (0, 1, 1, 0)$ chegar na solução $s' = (1, 1, 1, 0)$?

Vizinhança

Exemplo mochila



NÃO. Por quê isso ocorre?

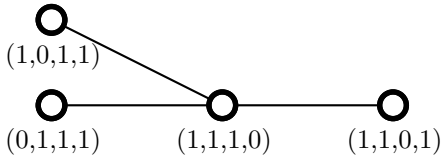
Vizinhança

Exemplo mochila

Inicialmente vamos criar a vizinhança da solução $(1, 1, 1, 0)$.

Vizinhança

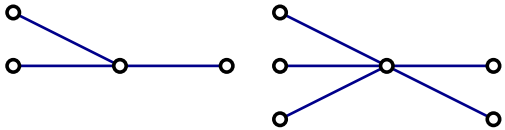
Exemplo mochila



Inicialmente vamos criar a vizinhança da solução $(1, 1, 1, 0)$.

Vizinhança

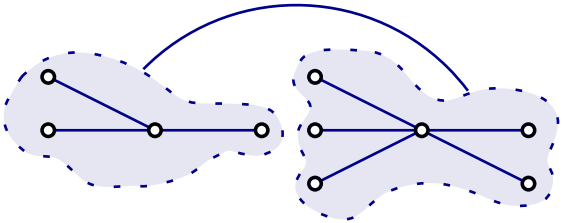
Exemplo mochila



Se colocarmos os dois grafos de vizinhança lado a lado...

Vizinhança

Exemplo mochila



Se colocarmos os dois grafos de vizinhança lado a lado... Percebemos que não existem conexões entre os elementos dos mesmos, ou seja, não existe caminho entre as soluções s e s'.

Conclusão

1. Ao determinar uma vizinhança, implicitamente estamos criando caminhos entre soluções vizinhas. Esses caminhos são percorridos ao executarmos o operador de movimento nas soluções. **Isso implica que existem diversos caminhos possíveis entre duas soluções s e s'.**
2. Dependendo da escolha de vizinhança, **pode não ser possível encontrar um caminho que conecte todas as soluções**.

Conclusão

1. Ao determinar uma vizinhança, implicitamente estamos criando caminhos entre soluções vizinhas. Esses caminhos são percorridos ao executarmos o operador de movimento nas soluções. **Isso implica que existem diversos caminhos possíveis entre duas soluções s e s'.**
2. Dependendo da escolha de vizinhança, **pode não ser possível encontrar um caminho que conecte todas as soluções**.

Com esse conhecimento, finalmente conseguimos entender as três características que uma boa representação de solução deve possuir (visto na apresentação "Representação de soluções").

Vizinhança

Conclusões

1. **Completude**: todas as soluções de um problema (pelo menos as factíveis) devem ser passíveis de representação.

Vizinhança

Conclusões

1. **Completude**: todas as soluções de um problema (pelo menos as factíveis) devem ser passíveis de representação.
2. **Conectividade**: é necessário que exista um **caminho** entre quaisquer duas soluções.

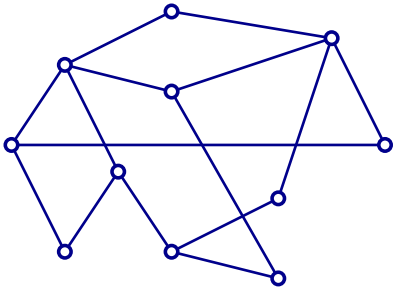
Vizinhança

Conclusões

1. **Completude**: todas as soluções de um problema (pelo menos as factíveis) devem ser passíveis de representação.
2. **Conectividade**: é necessário que exista um **caminho** entre quaisquer duas soluções.
3. **Eficiência**: a representação precisa ser de fácil manipulação pelos **operadores de movimento**. Também deve facilitar o cálculo da função objetivo e verificação de violação de restrições.

Vizinhança

Casos - vizinhanças diferentes

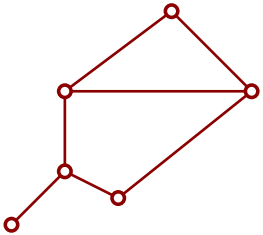


N_1

Considere que determinamos uma vizinhança N_1 para um determinado problema, gerando o grafo acima.

Vizinhança

Casos - vizinhanças diferentes

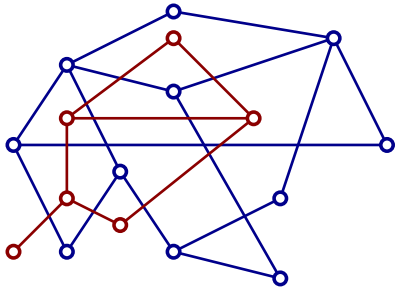


N_2

Ao determinarmos uma nova vizinhança N_2 geramos um novo grafo de soluções.

Vizinhança

Casos - vizinhanças diferentes



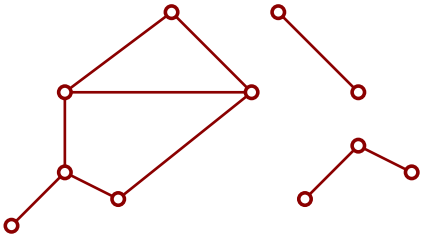
N_1

N_2

Sendo que pode não haver conexões entre os grafos gerados por duas vizinhanças diferentes.

Vizinhança

Casos - não conectividade

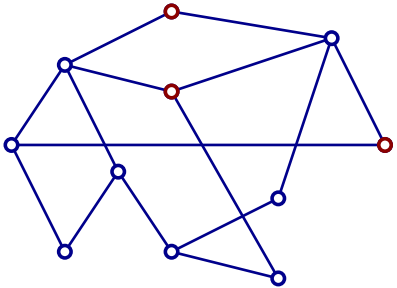


N_2

Ainda, pode acontecer de uma mesma vizinhança ser desconectada, como vimos no exemplo do problema da mochila.

Vizinhança

Casos - infactibilidade

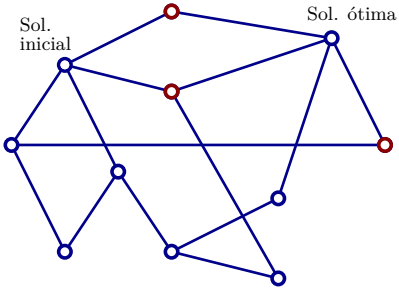


N_1

Considere a vizinhança gerada para um determinado problema, em que os nós (soluções) em vermelho são infactíveis.

Vizinhança

Casos - infactibilidade

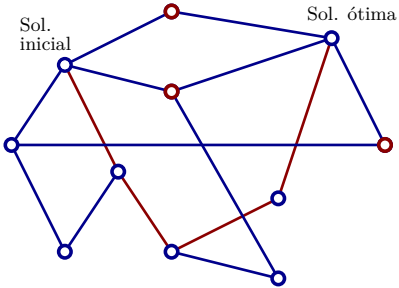


N_1

Ainda, partindo da solução atual mostrada, quantos movimentos seriam necessários para chegarmos a solução ótima?

Vizinhança

Casos - infactibilidade

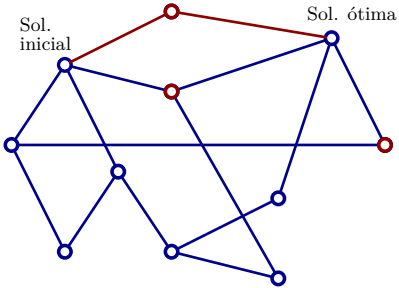


N_1

O número mínimo de movimentos para chegarmos ao ótimo é 4. Existe algo que poderíamos fazer para encurtar essa distância?

Vizinhança

Casos - infactibilidade

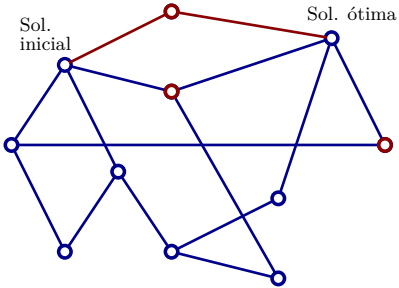


N_1

Se visitássemos uma solução infactível, **conseguimos chegar ao ótimo em apenas 2 movimentos**!

Vizinhança

Casos - infactibilidade

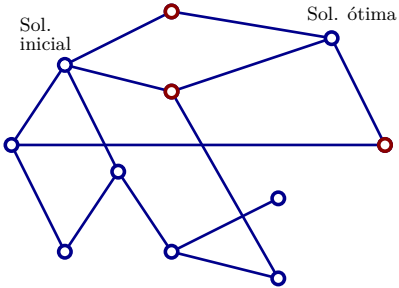


N_1

Ainda, em alguns casos podemos resolver o problema da conectividade entre grafos de soluções justamente aceitando soluções infactíveis.

Vizinhança

Casos - infactibilidade



N_1

No grafo desconectado por soluções factíveis acima, **conseguiríamos chegar na solução ótima usando um nó infactível intermediário**.

Conclusão

1. Ao determinar uma vizinhança, implicitamente estamos criando caminhos entre soluções vizinhas. Esses caminhos são percorridos ao executarmos o operador de movimento nas soluções. **Isso implica que existem diversos caminhos possíveis entre duas soluções s e s'.**
2. Dependendo da escolha de vizinhança, **pode não ser possível encontrar um caminho que conecte todas as soluções**.
3. Em alguns casos pode fazer sentido **aceitar soluções infactíveis** para chegarmos em soluções melhores.

Ótimo local

Ótimo local

Relembrando

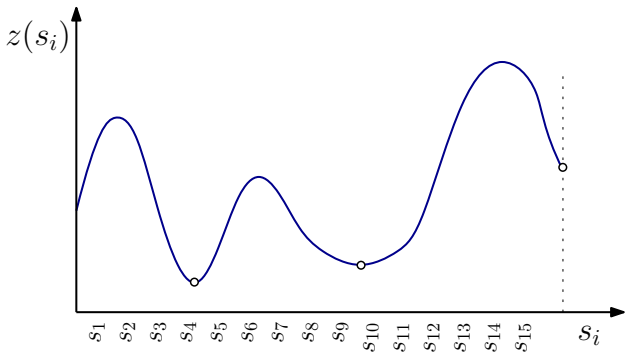
Definição

Ótimo local: Relativamente a uma função de vizinhança N, uma solução $s \in S$ é um ótimo local se ela tem uma qualidade de solução melhor do que todos os seus vizinhos, ou seja, para um problema de minimização:

$$f(s) \leq f(s'), \forall s' \in N(s)$$

Ótimo local

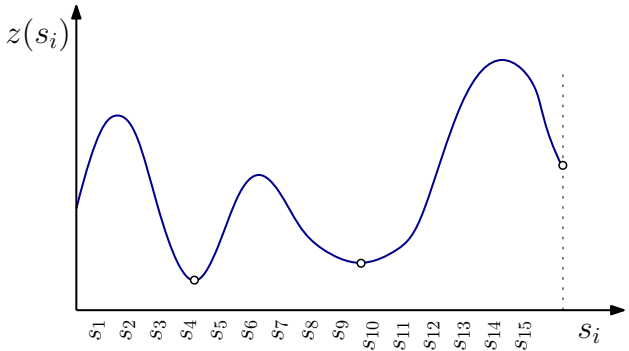
Relembrando



Fica mais fácil entender o conceito de ótimo local gráficamente. Lembra-se da representação que fizemos das soluções do problema do TSP vs os seus custos?

Ótimo local

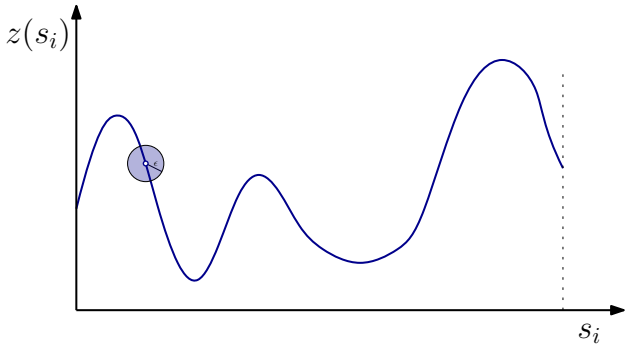
Relembrando



OBS: Já sabemos que a disposição das soluções no eixo S_i é muito mais complexa do que isso, bem como a função objetivo.

Ótimo local

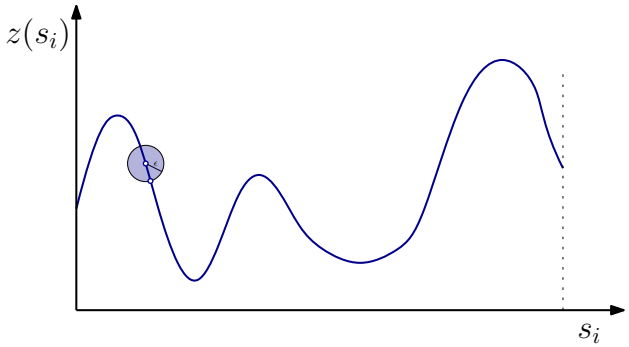
Relembrando



Podemos representar a vizinhança de uma solução (que está no eixo s_i) diretamente na função objetivo.

Ótimo local

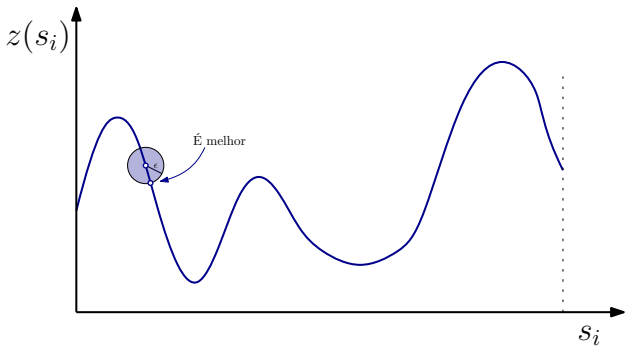
Exemplo



Considerando a vizinhança da solução mostrada na imagem, podemos dizer que ela é um ótimo local? Ou seja, $f(s) \leq f(s'), \forall s' \in N(s)$?

Ótimo local

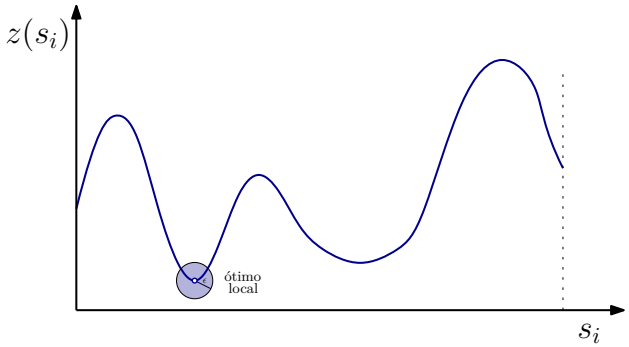
Exemplo



Não, existe uma solução melhor (na verdade existem várias, todas abaixo da solução no centro da bola).

Ótimo local

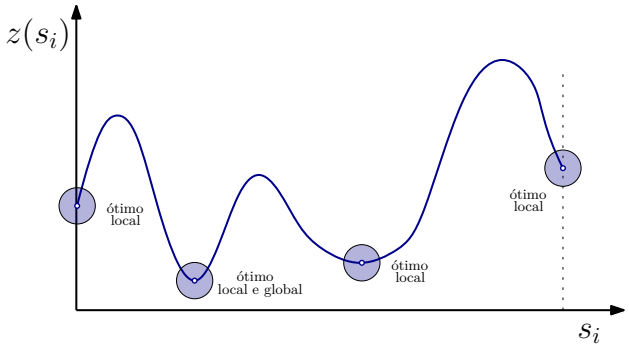
Exemplo



Já a solução mostrada pode ser considerada um **ótimo local**: nenhuma solução na vizinhança tem valor melhor do que ela. Existem mais ótimos locais no exemplo acima?

Ótimo local

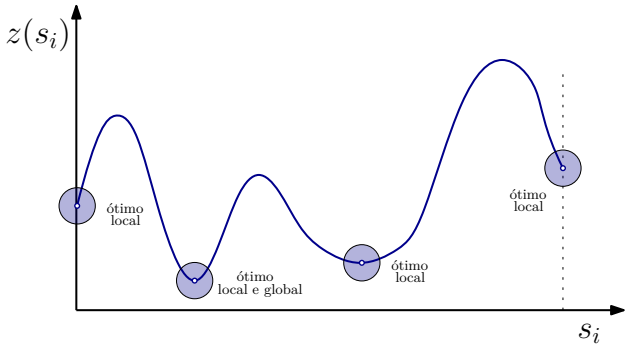
Exemplo



Existem **4 ótimos locais** no exemplo.

Ótimo local

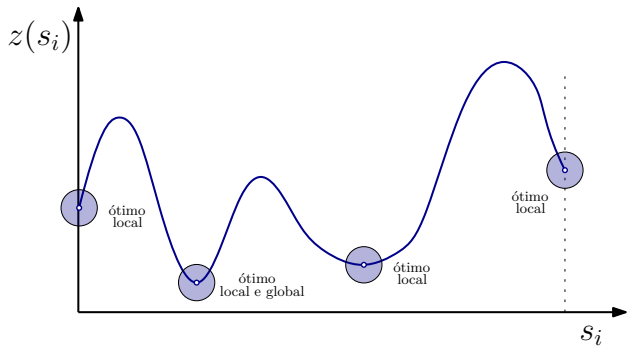
Exemplo



Ainda, percebe-se que o **ótimo global é um ótimo local**.

Ótimo local

Exemplo



O objetivo dos problemas de otimização é **encontrar o ótimo global**.

Fitness landscape (paisagem de adaptabilidade)

Fitness landscape

Espaço de busca

Finalmente, podemos definir o conceito de **espaço de busca** e **fitness landscape**.

Fitness landscape

Espaço de busca

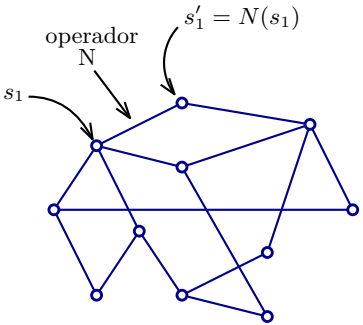
Finalmente, podemos definir o conceito de **espaço de busca** e **fitness landscape**.

Definição

Espaço de busca: O espaço de busca é definido como um grafo $G = (S, E)$, onde o conjunto de vértices S corresponde ao conjunto de soluções que é definida pela representação computacional, e o conjunto de arcos E corresponde ao operador de movimento usado para gerar novas soluções (vizinhança).

Fitness landscape

Espaço de busca



A definição nada mais é do que o que estamos estudando até agora.

Fitness landscape

Fitness landscape

Definição

Fitness landscape: O fitness landscape (*paisagem de adaptabilidade*) pode ser definida por uma tupla (G, f), em que o grafo G representa o espaço de busca e f representa a função objetivo que guia a busca por soluções.

Fitness landscape

Fitness landscape

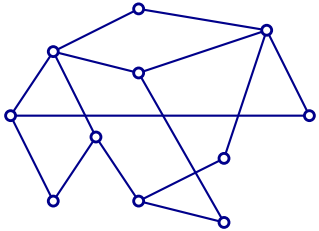
Definição

Fitness landscape: O fitness landscape (*paisagem de adaptabilidade*) pode ser definida por uma tupla (G, f), em que o grafo G representa o espaço de busca e f representa a função objetivo que guia a busca por soluções.

Uma forma conveniente de descrever o *fitness landscape* é usando termos geográficos. Assim, considerando o espaço de busca como o piso, elevamos cada solução a uma altitude equivalente a sua qualidade (função objetivo). Obtemos então uma paisagem feita de vales, colinas, penhascos, planaltos, etc...

Fitness landscape

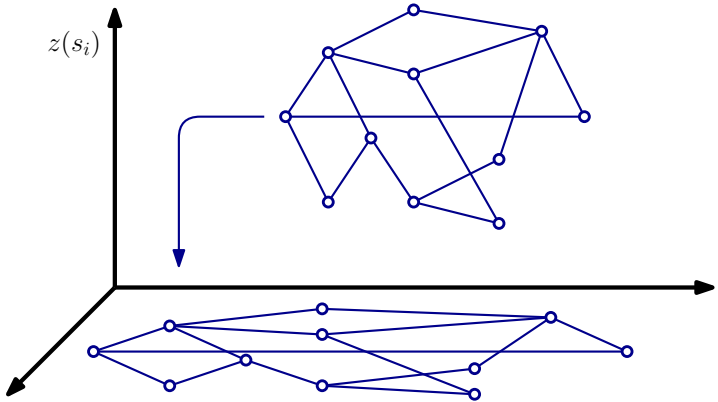
Fitness landscape



Com o grafo G do espaço de busca.

Fitness landscape

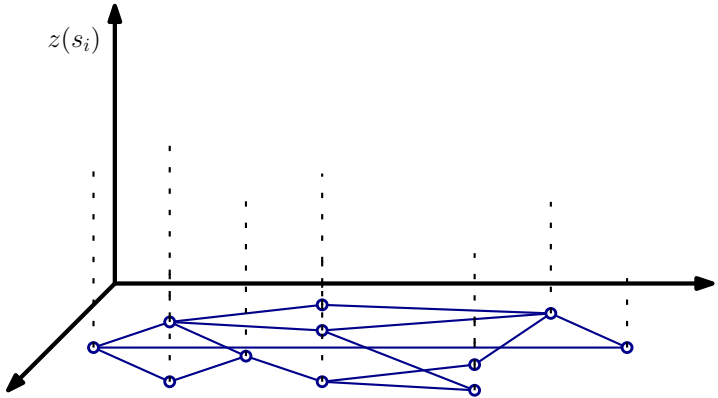
Fitness landscape



"Achatado"no piso.

Fitness landscape

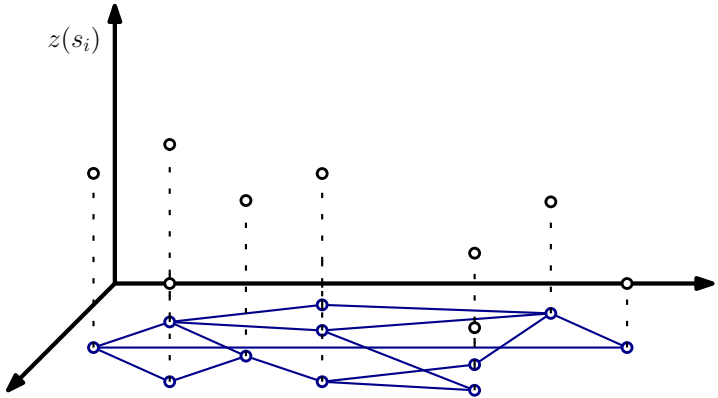
Fitness landscape



Calculamos o custo de cada solução.

Fitness landscape

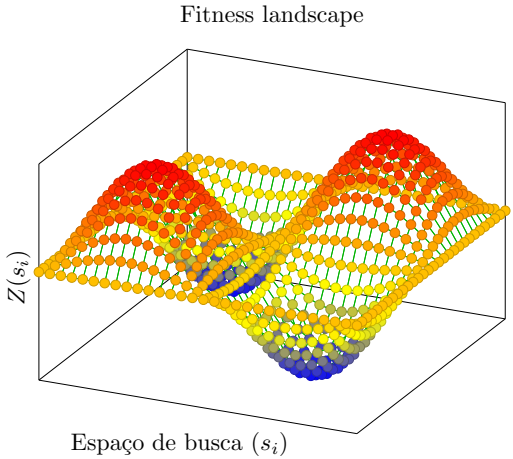
Fitness landscape



Elevando-a a uma "altitude".

Fitness landscape

Fitness landscape



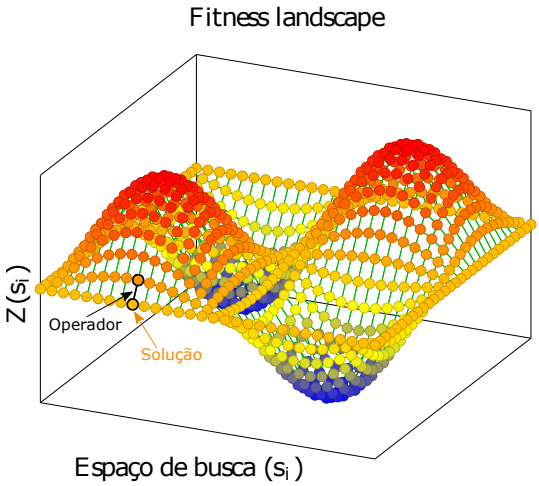
Fitness landscape

Fitness landscape

Uma metaheurística de solução única pode ser vista como uma trajetória (ou uma caminhada) na paisagem para encontrar o vale mais baixo (em um problema de minimização). Duas soluções sucessivas são sempre vizinhas, e podem ser acessadas por meio do **operador de movimento**.

Fitness landscape

Fitness landscape



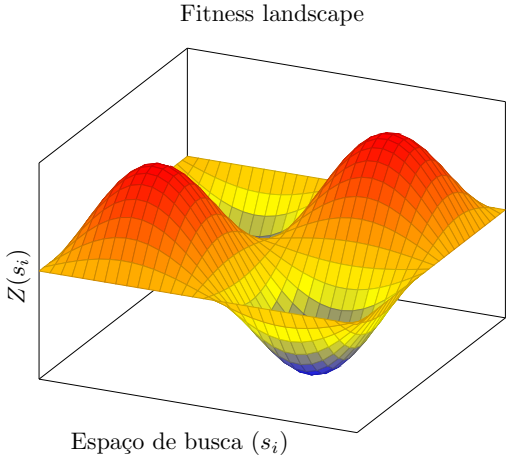
Fitness landscape

Fitness landscape

Visualmente, uma representação do *fitness landscape* sem os nós (círculos representando as soluções) fica mais "agradável".

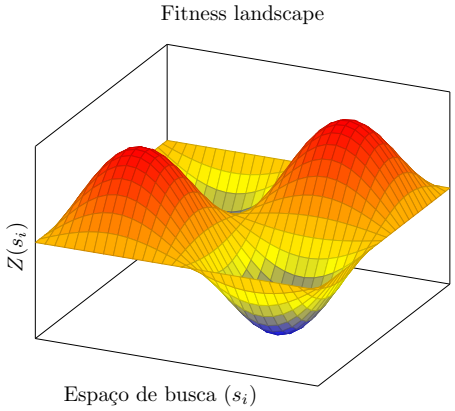
Fitness landscape

Fitness landscape



Fitness landscape

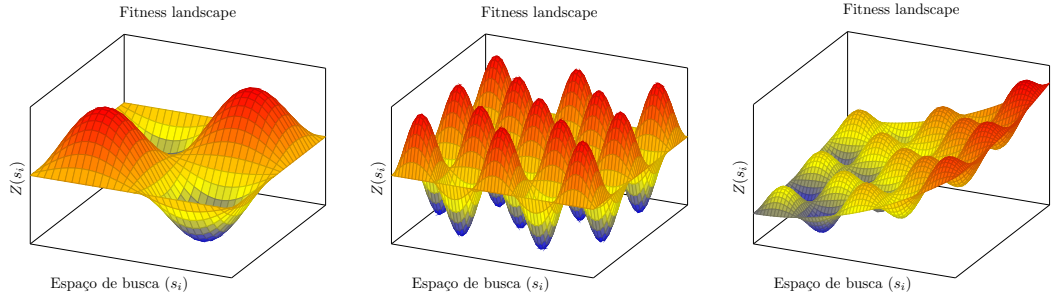
Fitness landscape



ATENÇÃO: Lembre dos grafos do espaço de busca, nada garante que todas as soluções são conectadas! Isso é uma simplificação que fazemos para ajudar na visualização do *landscape*.

Fitness landscape

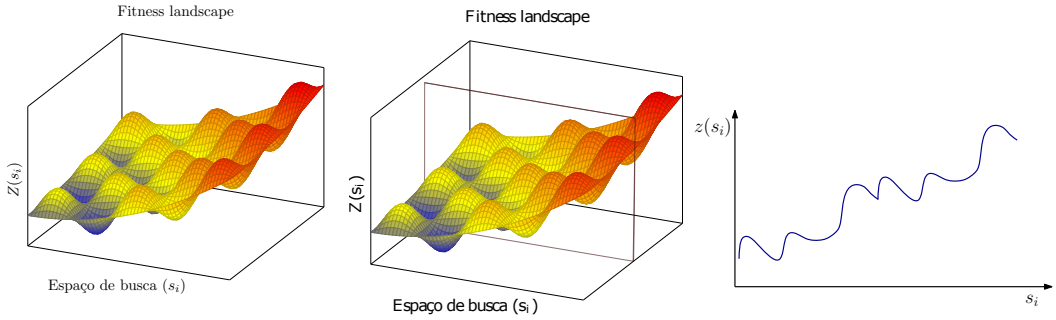
Fitness landscape



Cada problema de otimização gera um *fitness landscape* diferente. As metaheurísticas tentam explorar o landscape de forma eficiente.

Fitness landscape

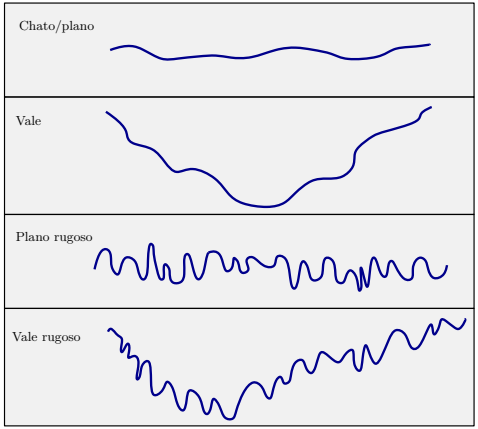
Fitness landscape



Vemos que a nossa simplificação do *fitness landscape* é um "corte" no modelo original.

Fitness landscape

Classificações



Os diferentes padrões de landscape vão influenciar no design das metaheurísticas.

Conclusão

1. O *fitness landscape* é a topologia de um problema de otimização.
2. Essa topologia é o que usamos para realizar o design de metaheurísticas.

Atividades

Atividade 1

1. Explique as principais componentes de um algoritmo de solução única.
2. Qual é a relação entre os algoritmos de solução única e o conceito de vizinhança?
3. O que é vizinhança em um problema de otimização discreto? E operador de movimento?
4. Considere as duas soluções abaixo:

$$s_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \end{bmatrix} s_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

   Calcule a distância entre as soluções considerando a distância Euclidiana e a Hamming.
5. Como funcionam os operadores de movimento *flip* e *swap*?
6. Considerando a solução inicial $s_1 = [0, 0, 1, 0, 1]$ e o operador *flip*, gere os vizinhos de s_1.
7. Seja a solução inicial $s_1 = [0, 0, 1, 0, 1]$ e a solução final $s_2 = [1, 0, 1, 1, 1]$ com o operador *flip*. Mostre um caminho possível entre s_1 e s_2 gerado pela vizinhança do *flip*.

Atividade 1

8 Considerando uma vizinhança N, sempre vai existir um caminho entre duas soluções s e s'? Quais casos podem ocorrer?

9 Você leu um artigo de um algoritmo que usa soluções infactíveis em algumas iterações. Qual é a lógica por trás disso? Quais problemas esse tipo de abordagem espera sanar?

10 O que é um ótimo local? Você estuda um algoritmo que garante chegar em um ótimo local, o que pode ser dito sobre a solução global do problema?

11 O que é espaço de busca? Como ele se relaciona com o *fitness landscape*?

12 Busque 2 operadores de movimento (vizinhanças) para os seguintes problemas, e forneça um exemplo numérico da execução no movimento para cada um deles:

- TSP
- Mochila
- VRP

Atividade 2

1. Considerando o problema escolhido para ser desenvolvido na disciplina, pense em um operador de movimento e uma vizinhança para o mesmo. Em seguida busque na literatura (livros, artigos) se já existe algum operador definido para o problema (para algumas classes de problemas é muito difícil definir uma vizinhança, pela complexidade na própria representação computacional das soluções).