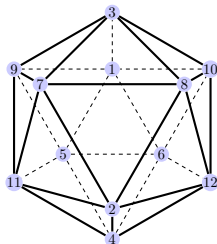


Metaheurísticas - Local Search 2OPT

Alexandre Checoli Choueiri

30/03/2023



① Exemplo

② Atividade

Busca Local

Exemplo

EXEMPLO: Implemente um algoritmo de busca local para o problema do caixeiro viajante usando a estrutura de vizinhança swap e 2-OPT.

Antes de partirmos para a etapa de implementação, os principais problemas que poderiam ocorrer devem estar bem resolvidos. Fazemos isso pensando no design da nossa implementação antes de implementarmos. **OBS: Não se anime! mesmo com esse cuidado cometeremos erros.**

Busca Local

Exemplo

A vizinhança 2-OPT consiste em remover os arcos $(i, i + 1)$ e $(j, j + 1)$ e criar os arcos (i, j) e $(i + 1, j + 1)$

REMOVER

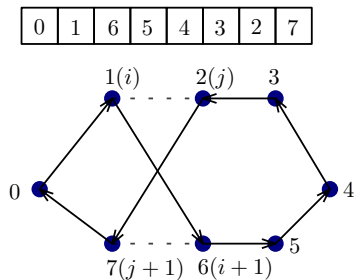
$(i, i + 1)$ e $(j, j + 1)$

ADICIONAR

(i, j) e $(i + 1, j + 1)$

Busca Local

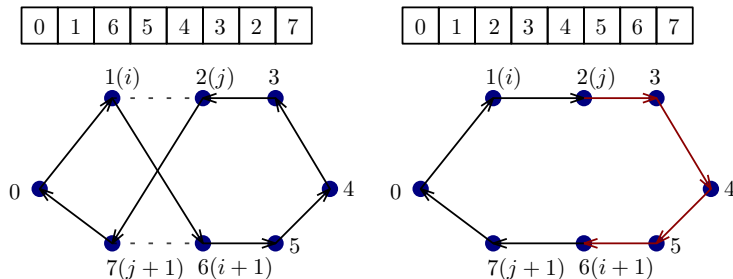
Exemplo



Podemos verificar o que ocorre com o operador 2-OPT junto a estrutura de solução.

Busca Local

Exemplo

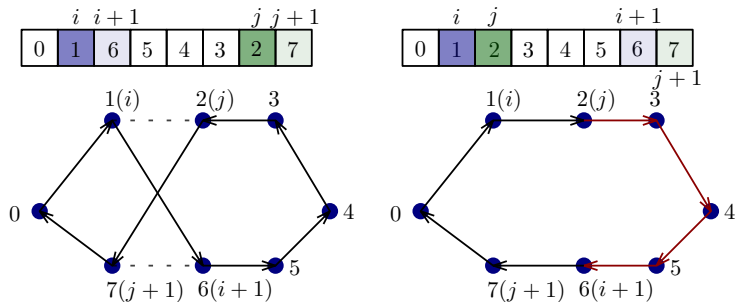


A primeira pergunta que devemos responder é:

Dada uma solução (vetor) s e os elementos $i, i + 1, j, j + 1$, como alterar s para operar o 2-OPT?

Busca Local

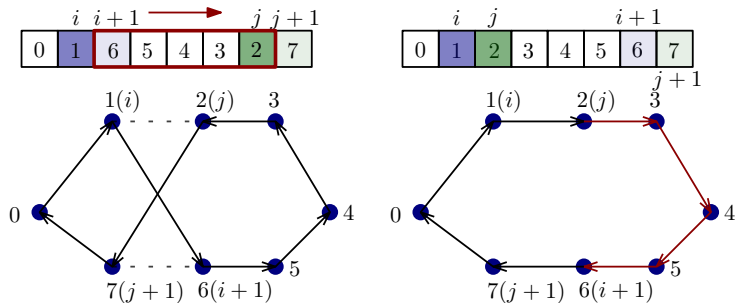
Exemplo



A figura acima mostra as alterações realizadas no vetor de solução. Como podemos generalizar essas transformações para quaisquer $i, i + 1, j, j + 1$?

Busca Local

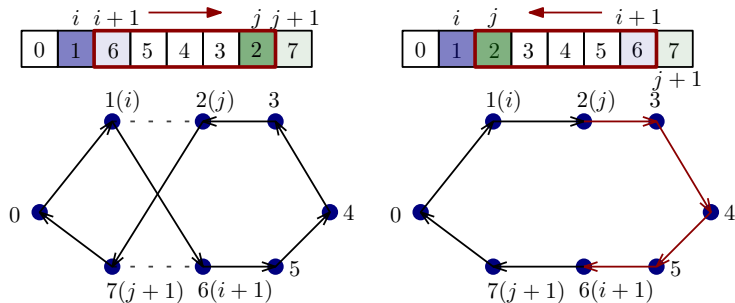
Exemplo



Note que os elementos i e $j + 1$ **não** foram alterados.

Busca Local

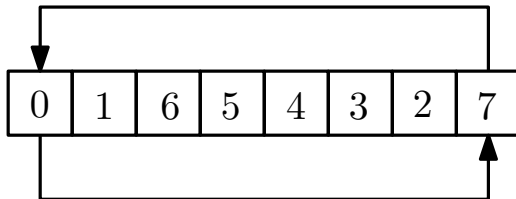
Exemplo



Já a sequência com os pontos começando em $i+1$ até j **foi invertida**. Ou seja, precisamos de uma rotina que faça a inversão dos elementos de um vetor de forma genérica.

Busca Local

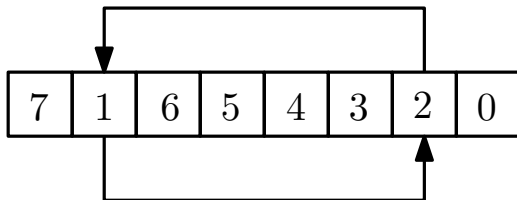
Exemplo



A ideia básica da inversão de elementos em um vetor é que **não precisamos iterar por toda a sequência, mas somente em metade dela**. Considere o exemplo acima.

Busca Local

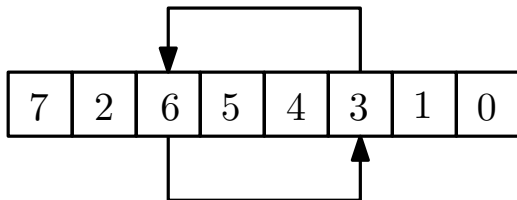
Exemplo



A ideia básica da inversão de elementos em um vetor é que **não precisamos iterar por toda a sequência, mas somente em metade dela**. Considere o exemplo acima.

Busca Local

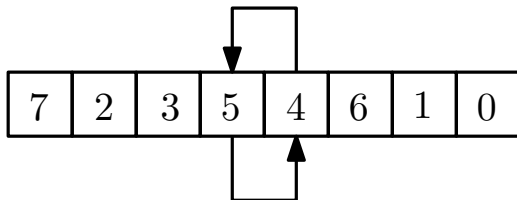
Exemplo



A ideia básica da inversão de elementos em um vetor é que **não precisamos iterar por toda a sequência, mas somente em metade dela**. Considere o exemplo acima.

Busca Local

Exemplo



A ideia básica da inversão de elementos em um vetor é que **não precisamos iterar por toda a sequência, mas somente em metade dela**. Considere o exemplo acima.

Busca Local

Exemplo

7	2	3	4	5	6	1	0
---	---	---	---	---	---	---	---

A ideia básica da inversão de elementos em um vetor é que **não precisamos iterar por toda a sequência, mas somente em metade dela**. Considere o exemplo acima.

Busca Local

Exemplo

O código na sequência inverte os elementos de um vetor começando em *inicio* até *fim*. Inicialmente encontra-se o número de iterações que deverão ser realizadas, como o menor inteiro da divisão do número de elementos do vetor por 2. Em seguida o laço while faz a troca dos elementos começando em *inicio* e *fim*, incrementando *inicio* e decrementando *fim*.

Busca Local

Exemplo

```
Sub inverte_elementos(ByRef v() As Integer, inicio As Integer, fim As Integer)
Dim n_iter As Integer
Dim n_elementos As Integer
n_elementos = fim - inicio + 1 ' numero de elementos na sequencia
n_iter = Int(n_elementos / 2) ' numero de iteracoes
Dim p1 As Integer, p2 As Integer
p1 = inicio
p2 = fim
Dim iter As Integer
iter = 1
Do While iter <= n_iter
Dim aux As Integer
aux = v(p1)
v(p1) = v(p2)
v(p2) = aux
p1 = p1 + 1
p2 = p2 - 1
iter = iter + 1
Loop
End Sub
```


Busca Local

Exemplo

i	$i + 1$						
0	1	6	5	4	3	2	7

A segunda pergunta que temos que responder é:

Como geramos toda a vizinhança 2OPT? Ou seja, quais valores que os índices $i, i + 1, j, j + 1$ podem assumir?

Pela imagem acima percebemos que i pode começar no índice 0.

Busca Local

Exemplo

i	$i+1$			i	$i+1$		
0	1	6	5	4	3	2	7

No entanto ele **não pode ir até o fim do vetor**, pois devemos considerar os valores $i+1, j$ e $j+1$. Assim, i vai até o fim -3.

Busca Local

Exemplo

		j	$j+1$			j	$j+1$
0	1	6	5	4	3	2	7

Já os valores de j devem começar após os valores de $i+1$, ou seja, $i+2$.

Busca Local

Exemplo

		j	$j+1$			j	$j+1$
0	1	6	5	4	3	2	7

E podem ir até o fim -1, pois devemos considerar $j+1$ como o último elemento.

Busca Local

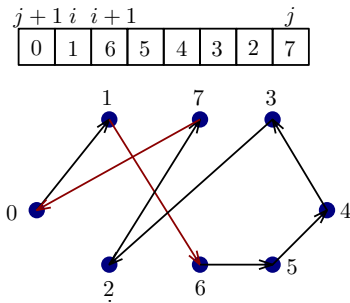
Exemplo

CUIDADO

Com essa seleção de i e j estamos deixando alguns arcos de fora do 2-OPT.

Busca Local

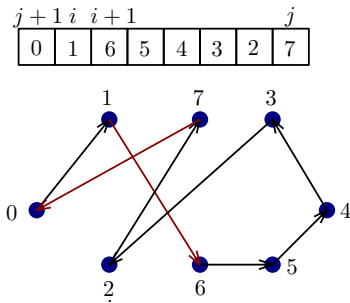
Exemplo



Considere o caso acima, em que queremos remover os arcos em vermelho.

Busca Local

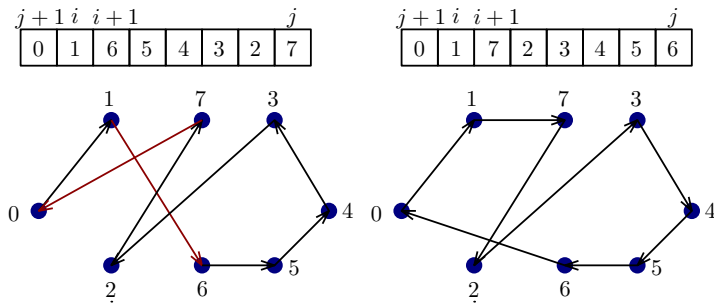
Exemplo



Um dos arcos removidos é composto pelo último e pelo primeiro ponto (7 e 0), de forma que j deveria ser 7 e $j + 1$ 0.

Busca Local

Exemplo



Como o 2OPT inverte as posições de $i + 1$ até j , não há problema em colocarmos o j na última posição do vetor. Com isso, podemos também alterar o limite designado para i , ao invés de fim - 3 deve ser fim - 2.

Busca Local

Exemplo

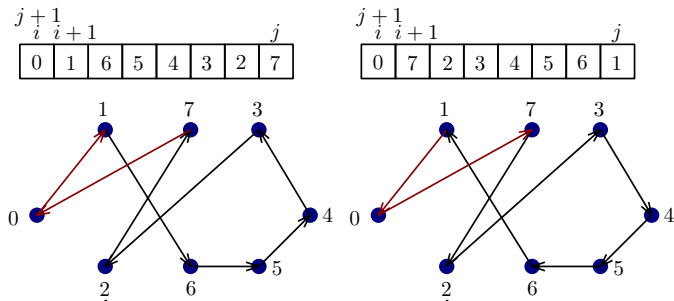
Dessa forma, temos que os limites para i e j devem ser:

Contador	Inicio	Fim
i	0	$\text{fim} - 2$
j	$i + 2$	fim

Se definidos puramente desta forma, a primeira iteração de i gera um movimento 2-OPT redundante.

Busca Local

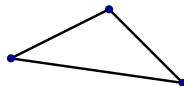
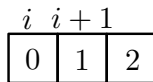
Exemplo



Quando $i = 0$ e $j = \text{fim}$ os dois arcos a serem removidos são **adjacentes**, o que implica simplesmente uma inversão na rota como um todo. Embora não seja um problema, é uma computação a mais.

Busca Local

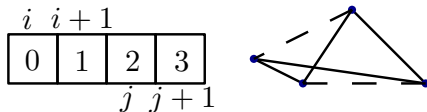
Exemplo



Uma última consideração. Pela definição de varredura dos índices percebemos que não faz sentido executar o movimento 2OPT em rotas com menos de 4 pontos.

Busca Local

Exemplo



Somente a partir de 4 pontos conseguimos remover e reinserir 2 arcos de forma que a rota é alterada.

Atividade 1

1. Considerando o problema de sua escolha, implemente um algoritmo de busca local.