

Busca Tabu

Notação e Termos:

k = Número das últimas iterações na **memória da Busca Tabu**

$T(k)$ = Lista de movimentos proibidos, **Lista Tabu**, considerando últimas k iterações.

$\mathcal{N}(S)$ = Conjunto de soluções vizinhas de S

- ▶ O **Critério de Aspiração** permite aceitar uma solução baseado em sua qualidade, mesmo que tal solução faça movimentos da lista tabu.

BUSCA TABU SIMPLIFICADA - MINIMIZAÇÃO

1. Seja S uma solução inicial
2. $S^+ \leftarrow S$ (mantém atualizado a melhor solução encontrada)
3. Enquanto não atingir condição de parada faça
4. Escolha soluções candidatas $V \subseteq \mathcal{N}(S)$
5. Enquanto $V \neq \emptyset$ faça
6. Pegue $S' \in V$ de peso mínimo e faça $V \leftarrow V \setminus \{S'\}$
7. Se $c(S') \leq c(S^+)$ então // (critério de aspiração)
8. $S^+ \leftarrow S'$, $S \leftarrow S'$ e $V \leftarrow \emptyset$
9. senão
10. se não é Tabu($S' \leftarrow S$) então
11. $S \leftarrow S'$ e $V \leftarrow \emptyset$
12. Devolva S^+

Possíveis implementações das subrotinas:

CONDIÇÕES DE PARADA

- ▶ Quando atingir limite de tempo de CPU
- ▶ Quando melhor solução não for atualizada por certo tempo

ESCOLHA DE SOLUÇÕES CANDIDATAS

- ▶ Escolha todas soluções vizinhas (se vizinhança for pequena)
- ▶ Escolha aleatoriamente um subconjunto dos vizinhos

Possíveis implementações das subrotinas:

CRITÉRIO-ASPIRAÇÃO

- ▶ No passo 7, note que mesmo uma solução com movimento Tabu, pode ser escolhida. Este é o critério de aspiração mais comum.

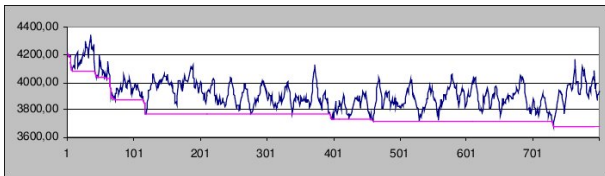
TABU($S' \leftarrow S$)

- ▶ Considere cada solução formada por elementos E
- ▶ Cada iteração do algoritmo de Busca Tabu representa um tempo
- ▶ Cada elemento $e \in E$ possui um marcador de tempo t_e (inicialmente $-\infty$)
- ▶ t_e é o último momento que e entrou ou saiu de uma solução
- ▶ Se estamos no momento t e for inserido/removido elemento e tempo t_e tal que $t - t_e \leq k$, então movimento é tabu.

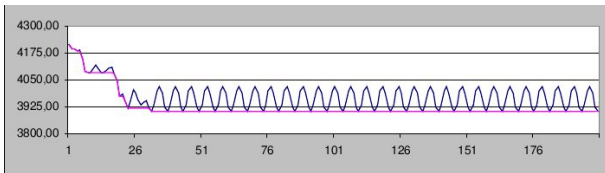
Valores adequados para k

Depende do problema, mas $k = 7$ é um bom ponto de partida

Exemplo de comportamento sem ciclo



Exemplo de comportamento com ciclo ($k = 3$)

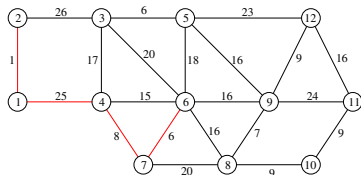


Exemplo: k -Árvore

k -Árvore: Dado um grafo $G = (V, E)$ com custo c_e em cada aresta $e \in E$, encontrar uma árvore com k vértices de peso mínimo.

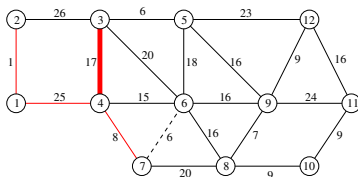
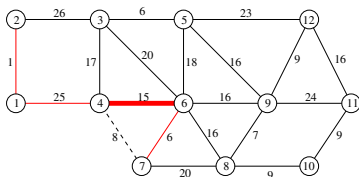
Teorema: O problema da k -Árvore é um problema NP-difícil.

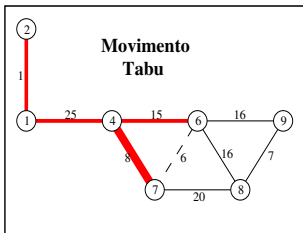
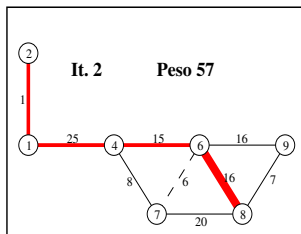
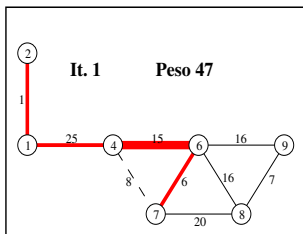
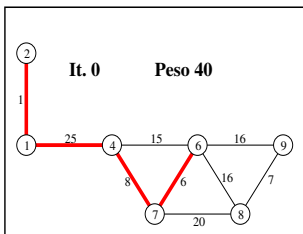
Movimentos:



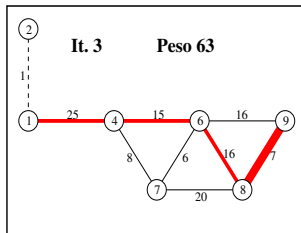
*Melhor movimento estatico
(mantem conjunto de vertices)*

Melhor movimento dinamico



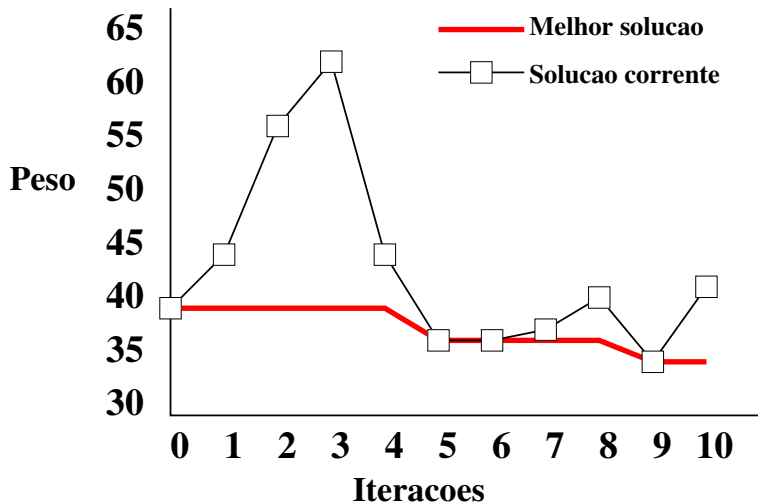


TABU



Arestas ficam na Lista Tabu por 2 iterações

Comportamento do algoritmo para 10 iterações com melhor movimento possível



Busca Tabu

Exercícios faça algoritmos Busca Tabu para os seguintes problemas:

1. Caixeiro Viajante: TSP
2. Corte de Peso Máximo: MaxCut
3. Satisfatibilidade de Peso Máximo: MaxSat

Algoritmos Genéticos

- ▶ Baseado em idéias da Evolução Natural
- ▶ Mantém uma população de indivíduos
- ▶ Há cruzamento entre indivíduos gerando novos indivíduos
- ▶ Pode haver mutação nos indivíduos
- ▶ Há seleção dos melhores indivíduos
- ▶ No decorrer do tempo, há indivíduos que se destacam

ALGORITMO GENÉTICO SIMPLIFICADO

1. Seja P uma população inicial de indivíduos
2. Repita
3. Obtenha novos indivíduos N a partir de P por
4. Cruzamentos:
5. Selecione pares de indivíduos em P
6. Produza filhos de cada par
7. Mutação:
8. Selecione indivíduos de P
9. Produza filhos de maneira assexuada
10. Atualize P selecionando indivíduos de $P \cup N$
11. Até atingir critério de parada
12. Devolva indivíduo $e \in P$ mais adaptado

Algoritmos Genéticos

Indivíduo	↔	Solução para um problema
População	↔	Conjunto de soluções
Cromossomo	↔	Codificação de uma solução
Aptidão/Fitness	↔	Qualidade da solução
Alelo	↔	cada elemento que forma a solução

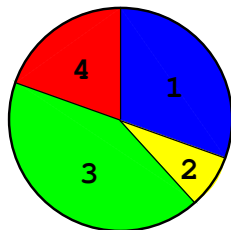
Manutenção da População

- ▶ Mantenha a população P com n elementos.
- ▶ Mantenha de uma população para outra os n_e melhores indivíduos (elite).
- ▶ Use probabilidade para escolher alguns elementos (e.g. método da roleta).
- ▶ Remova da população os indivíduos menos adaptados.

Método da Roleta (Roulette Wheel)

Suponha que temos uma população de $n = 4$ indivíduos:

Item	Aptidão/Fitness	Chance de Escolha
1	150	30%
2	50	10%
3	200	40%
4	100	20%
Total	500	100%



A chance de escolher um indivíduo é proporcional à aptidão.

- ▶ Para gerar uma nova população com m indivíduos, sorteie (rode a roleta) m vezes (pode dar repetições).

Exemplos de codificação

Vetor de bits Cada bit pode representar um elemento que forma uma solução.

Cromossomo 1:

11110000011100000111

Cromossomo 2:

01010101010101010101

Permutação Usado para problemas com ordem, como o TSP.

Cromossomo 1:

1 4 2 8 5 7 6 9 3

Cromossomo 2:

5 2 6 9 4 3 1 7 8

Função de Aptidão/Fitness

- ▶ Função que quantifica o quão bom é uma solução (indivíduo) em relação aos outros.
- ▶ Em geral, a aptidão é o valor da solução.
- ▶ Quanto mais próximo da solução ótima, mais apta é a solução.

Seleção para Cruzamento/Crossover ou Mutação

Exemplos:

- ▶ Selecione de maneira aleatória.
- ▶ Seleção baseada na aptidão/fitness.
- ▶ Um mesmo indivíduo pode estar em vários cruzamentos ou várias mutações.

Cruzamento/Crossover

Cruzamento em 1 ponto:

Pai 1:

11111 111111111

Pai 2:

00000 000000000

Produzem os filhos:

Filho 1:

11111 000000000

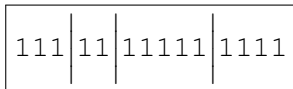
Filho 2:

00000 111111111

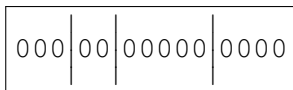
Cruzamento/Crossover

Cruzamento em k pontos:

Pai 1:

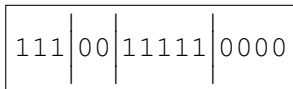


Pai 2:

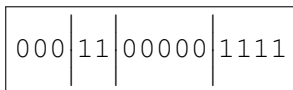


Produzem os filhos:

Filho 1:



Filho 2:



Os k pontos podem ser gerados aleatoriamente. Um filho pode ser gerado de vários pais.

Mutaç o

- ▶ Gera variabilidade na populaç o, principalmente com aleatoriedade
- ▶ Em geral uma mutaç o   obtida por pequenas mudanç as no indiv duo original

Por exemplo: Sortear posiç es no vetor de bits e trocar seus valores

Indiv duo:

1	1	1	1	0	1	0	0	1	0	1	1	1	1	0
---	----------	---	---	---	---	----------	---	---	---	---	----------	---	---	---

Mutaç o:

1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	0
---	----------	---	---	---	---	----------	---	---	---	---	----------	----------	---	---	---

Exemplo: $\max x^2$ para $x \in \{0, 1, \dots, 31\}$

Codificação com vetor de 5 bits. Indivíduo = 01101 ($x = 13$)

População de tamanho $n = 4$

Cruzamento com 1 ponto

Seleção por roleta

Inicialização aleatória

Exemplo: $\max x^2$ para $x \in \{0, 1, \dots, 31\}$

Seleção de indivíduos

No. do Indivíduo	População Inicial	Valor de x	Aptidão $f(x)$	Probab. na roleta	No. Vezes selecionado
1	01101	13	169	0,14	1
2	11000	24	576	0,49	2
3	01000	8	64	0,06	0
4	10011	19	361	0,31	1
Soma			1170	1	4
Média			293	0,25	1
Máximo			576	0,49	2

Exemplo: $\max x^2$ para $x \in \{0, 1, \dots, 31\}$

Cruzamento

No. do Indivíduo	Pares p/ Cruzamento	Ponto de Cruzamento	Filhos	x	Aptidão $f(x) = x^2$
1	0110 1	4	01100	12	144
2	1100 0	4	11001	25	625
2	11 000	2	11011	27	729
4	10 011	2	10000	16	256
Soma				1	1754
Média					439
Máximo					729

Exemplo: $\max x^2$ para $x \in \{0, 1, \dots, 31\}$

Mutação

No. do Indivíduo	Antes Mutação	Após Mutação	x	Aptidão $f(x) = x^2$
1	<u>0</u> 1101	<u>1</u> 1100	28	784
2	11000	11001	25	625
2	11000	11011	27	729
4	10 <u>0</u> 11	10 <u>1</u> 11	23	529
Soma				2667
Média				666,75
Máximo				784

Exemplo: Caixeiro Viajante

Codificação

- ▶ Codificação usando ordem/seqüência
- ▶ Conjunto de cidades: $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- ▶ Cada rota é uma permutação

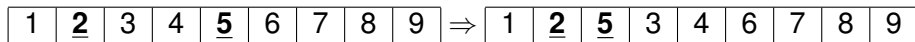
Exemplo de rota:

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Exemplo: Caixeiro Viajante

Exemplo de Mutação: Migração

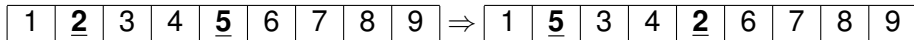
- ▶ Pegue dois alelos e coloque o segundo na seqüência do primeiro
- ▶ Esta mutação é pontual e preserva a ordem da maioria dos alelos



Exemplo: Caixeiro Viajante

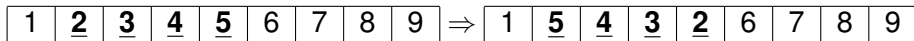
Exemplo de Mutação: Troca

- ▶ Pegue dois alelos e troque a posição deles



Exemplo de Mutação: Inversão

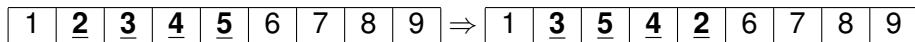
- ▶ Pegue um trecho e inverta a seqüência do trecho



Exemplo: Caixeiro Viajante

Exemplo de Mutação: Rearranjo aleatório de trecho

- ▶ Pegue um trecho e faça um rearranjo aleatório de trecho



Exemplo: Caixeiro Viajante

Exemplo de Cruzamento 1:

- ▶ Copie um trecho de um pai para o filho.

1	2	3	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	8	9
---	---	---	----------	----------	----------	----------	---	---

 \Rightarrow

			4	5	6	7		
--	--	--	---	---	---	---	--	--

9	3	7	<u>8</u>	<u>2</u>	<u>6</u>	<u>5</u>	1	4
---	---	---	----------	----------	----------	----------	---	---

- ▶ Complete os demais elementos a partir do trecho, com elementos do segundo pai na ordem em que aparecem e sem considerar os já inseridos

1	2	3	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	8	9
---	---	---	----------	----------	----------	----------	---	---

 \Rightarrow

3	8	2	4	5	6	7	1	9
---	---	---	---	---	---	---	---	---

9	3	7	<u>8</u>	<u>2</u>	<u>6</u>	<u>5</u>	1	4
---	---	---	----------	----------	----------	----------	---	---

Algoritmos Genéticos

Exercícios faça algoritmos genéticos para os seguintes problemas:

1. Caixeiro Viajante: TSP (faça mais rotinas para mutação, cruzamentos, etc)
2. Corte de Peso Máximo: MaxCut
3. Satisfatibilidade de Peso Máximo: MaxSat