

Busca Tabu

Prof. Igor da Penha Natal
Departamento de Informática
Universidade Estadual de Maringá

Baseado no material do professor Ademir Constantino

BUSCA TABU

- M. Laguna, Tabu Search Tutorial, II Escuela de Verano Latino-Americana de Investigación Operativa, 1995.
- A. Diaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato e F. T. Tseng, Optimización Heurística Y Redes Neurolales, Editorial Paraninfo, Espanha, 1996.
- <http://www.tabusearch.net/>

Princípios gerais

- Proposto por Fred Glover, 1986.
- Seu principal componente é o uso de memória adaptativa para criar uma busca mais flexível.
 - * analogia → *escalar montanha* → precisa lembrar (memória) de pontos chaves para ser capaz de tomar decisões ao longo do caminho.
- Aurélio:
 - * **Tabu**: Que é objeto forte de censura; que é interdito, proibido.
- Língua inglesa: *Tabu* ou *Taboo*

Princípios gerais

- Contrasta com métodos que:
 - * não utilizam memória (*simulated annealing*)
 - * utilizam estruturas rígidas de memória (*branch-and-bound*).
- A memória de soluções (ou atributos) visitadas no passado é usada para proibir movimentos que levariam a soluções já visitadas.
- Memória \Rightarrow necessidade de estrutura de dados eficiente de memória.

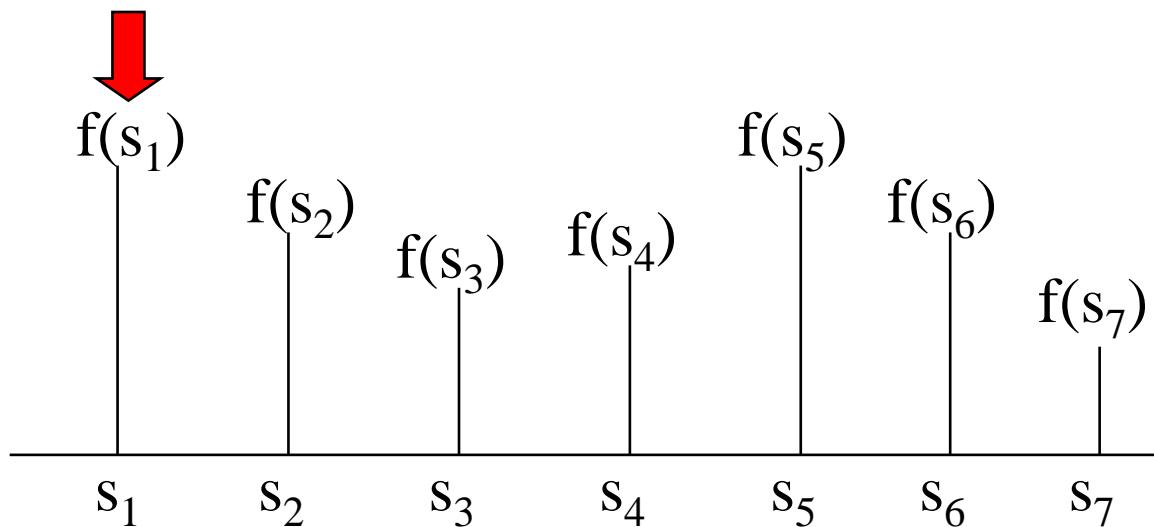
Busca Tabu

- Utiliza uma memória para evitar que o algoritmo pesquise regiões já exploradas no espaço de pesquisa
- Começa com uma solução inicial e, a cada iteração, move para a melhor solução na vizinhança (“corrente”):
 - * Movimentos que levam a soluções na **lista tabu** são proibidos (“em princípio”)
- **Lista tabu** : estrutura de memória básica, formada pelas soluções (ou movimentos) proibidas (**tabu**).
- **Prazo tabu** ou **tabu-tenure**: é o número de iterações (tempo) que soluções (ou movimentos) permanecem na lista tabu.

Busca Tabu - Algoritmo Básico

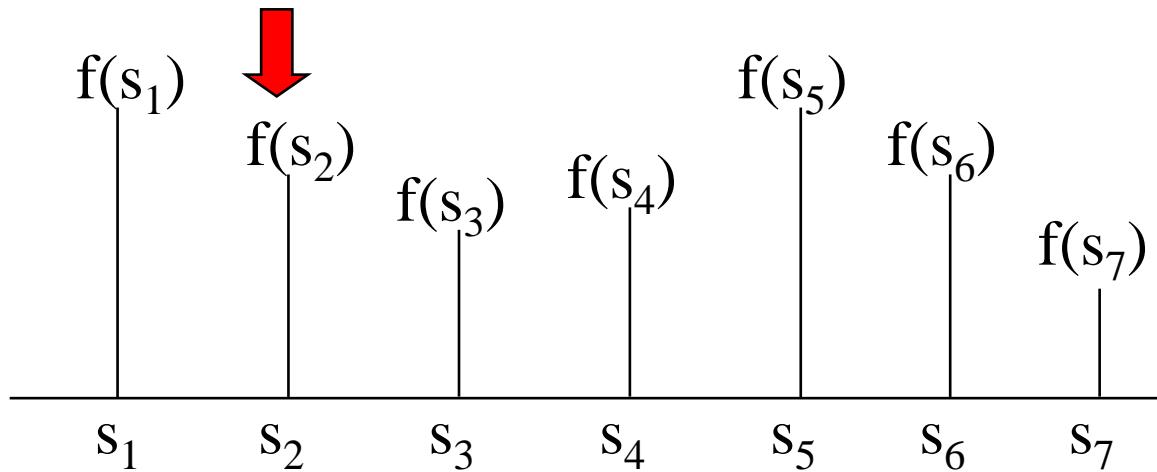
```
s ← s0; s* ← s0; lista-tabu ← ∅  
while critério-de-parada do  
    Encontrar a melhor solução s' pertencente a N(s) \ lista-tabu  
    if f(s') < f(s*)  
        then s* ← s'  
    end-if  
    lista-tabu ← lista-tabu ∪ {s}  
    s ← s'  
    Atualizar lista-tabu (prazos)  
end-while
```

Busca Tabu-Ilustração



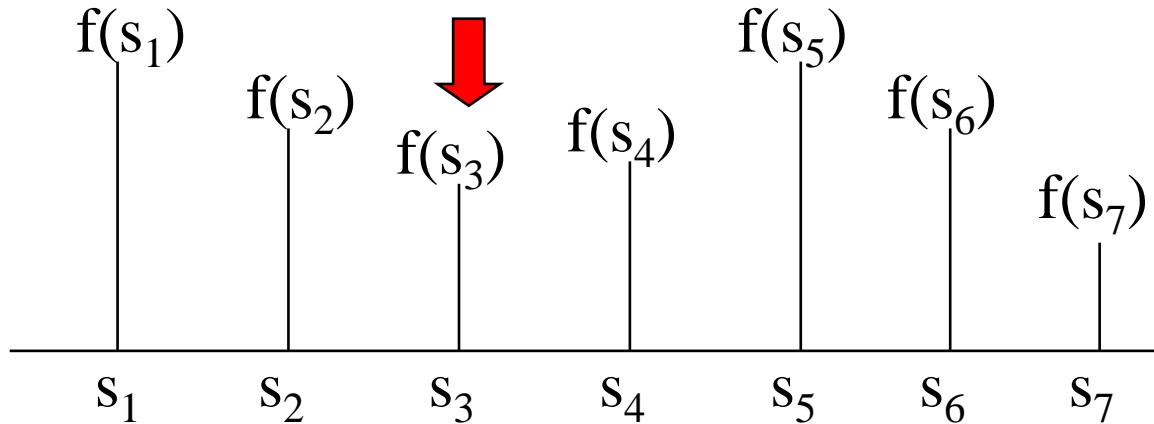
Lista Tabu: (s_1)

Busca Tabu-Ilustração



Lista Tabu: (s_1, s_2)

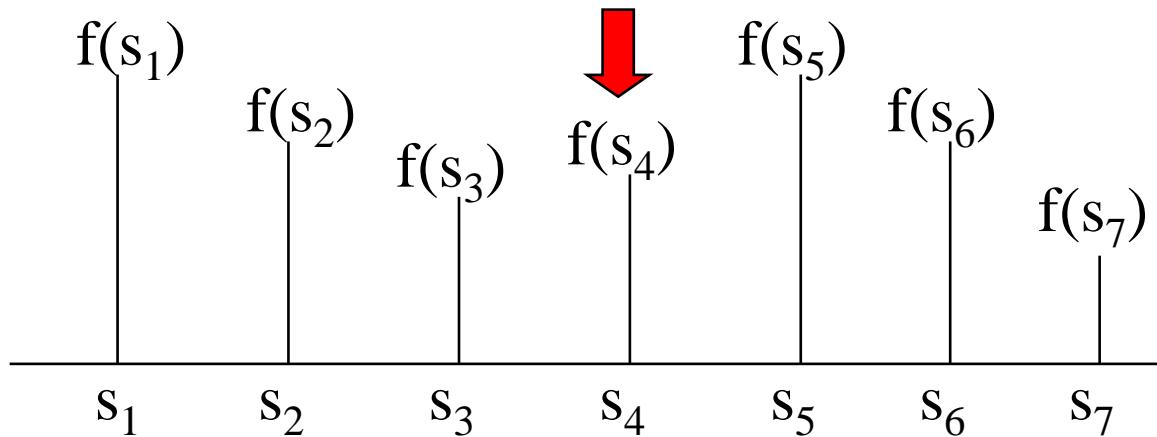
Busca Tabu-Ilustração



Lista Tabu: (s_1, s_2, s_3)

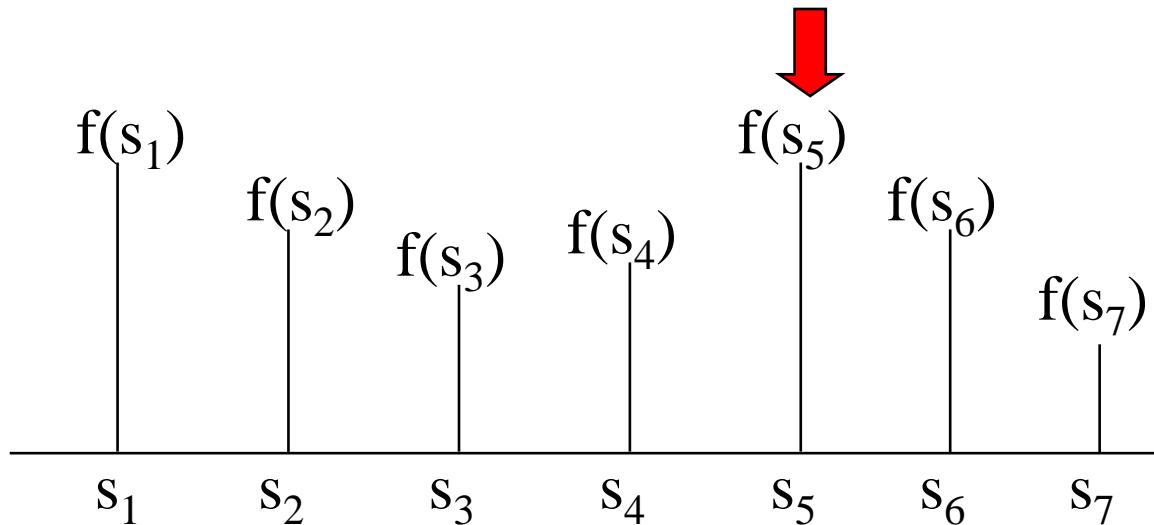
não há movimento aprimorante:
 s_4 é a melhor solução vizinha

Busca Tabu-Ilustração



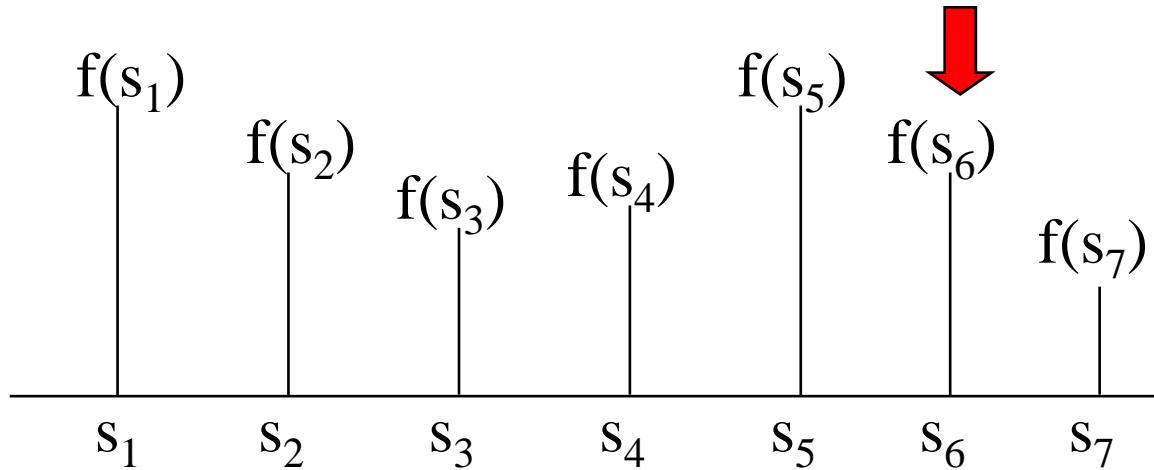
Lista Tabu: (s_1, s_2, s_3, s_4)

Busca Tabu-Ilustração



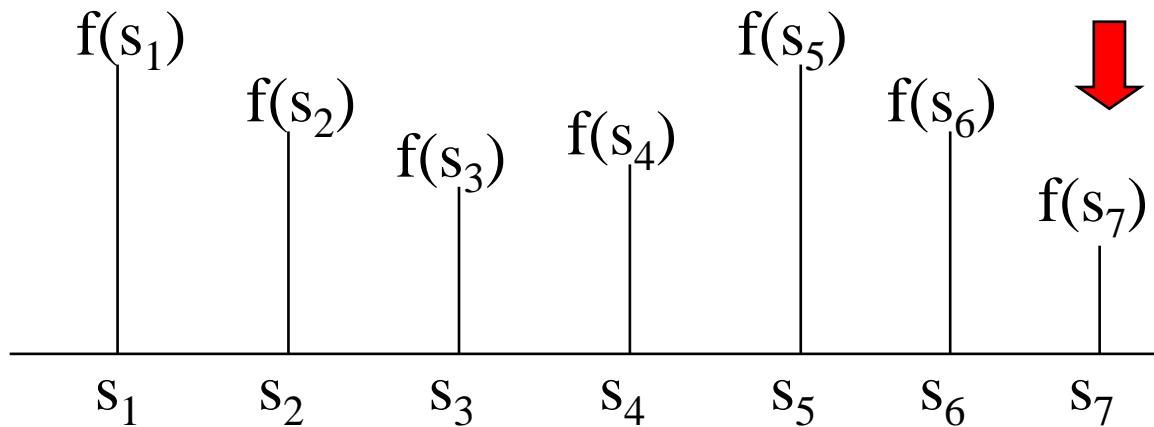
Lista Tabu: $(s_1, s_2, s_3, s_4, s_5)$

Busca Tabu-Ilustração



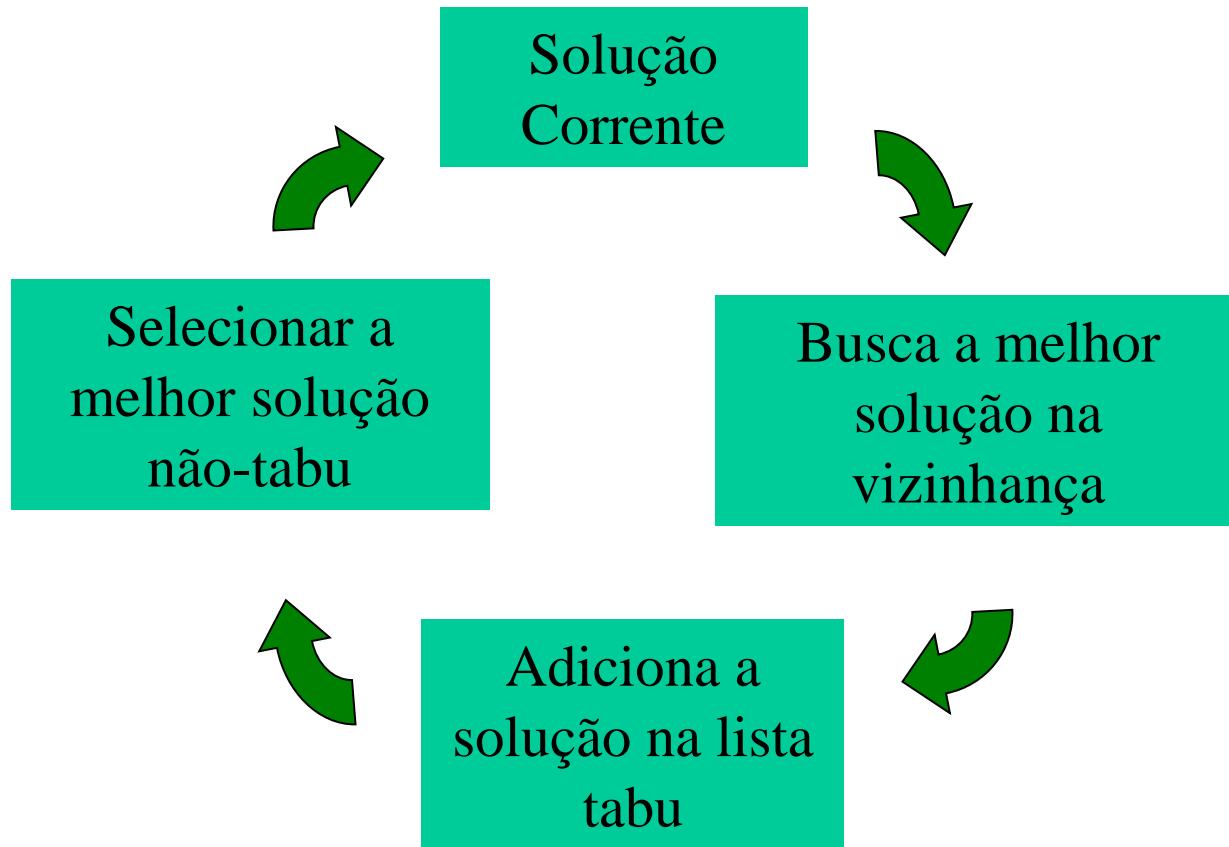
Lista Tabu: $(s_1, s_2, s_3, s_4, s_5, s_6)$

Busca Tabu-Ilustração

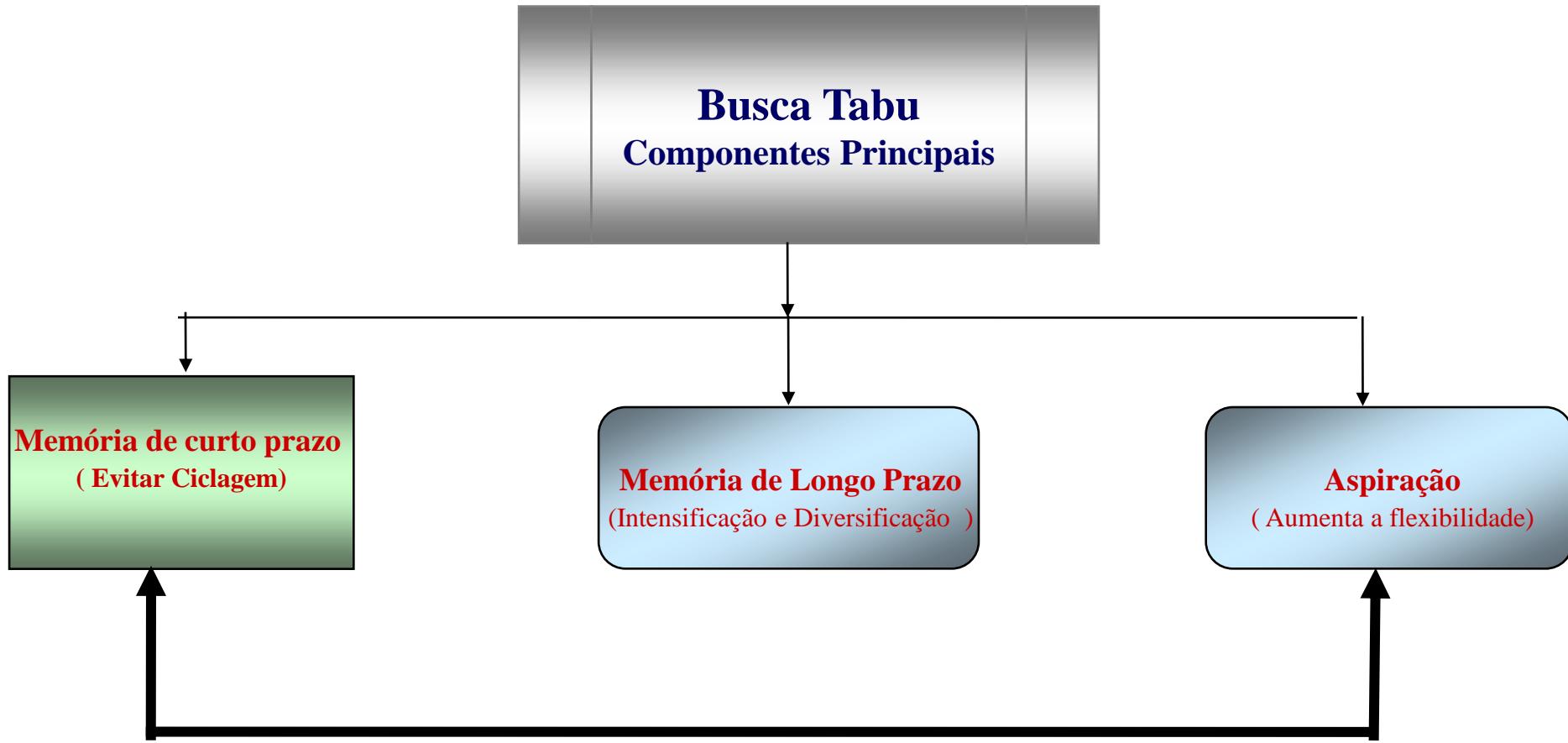


Melhor solução: s_7

Idéia Básica de Busca Tabu



Componentes da Busca Tabu



Estrutura de Memória



Uso da memória

- As estruturas de memória tem quatro princípios:
 - * recentidade, frequência → utilizada em estratégias mais “sofisticadas” de tabu.
 - * qualidade → habilidade de diferenciar o mérito de uma solução durante a busca. Podendo identificar elementos que são comuns entre as melhores soluções e elementos que fazem parte das ruins.
 - * influência → considera o impacto dessas escolhas durante a busca.

Uso da memória

- O uso de memória pode ser explícito ou através de atributos.
 - * explícito armazena soluções completas, por exemplo, armazenando soluções visitadas durante a busca.
 - * atributos de soluções que se modificam depois de alguns movimentos
 - grafo → adição e/ou retirada de uma aresta.
 - *scheduling* → índice das tarefas.

Classificação da memória

- As estratégias com o uso da memória estão divididas em:
 - * memória de curto prazo
 - recentidade
 - * memória de longo prazo
 - frequência
 - intensificação e diversificação

Classificação da memória

- Efeito das memórias: modificar a vizinhança de $N(x)$ para $N^*(x)$ como resultado da história seletiva de estados encontrados durante a busca.
- Memória de curto prazo :
 - * $N^*(x) = N(x) - \{\text{elementos tabu}\}$
- Memória de longo prazo:
 - * $N^*(x)$ é uma expansão de $N(x)$ com solução que $\notin N(x)$

Busca Local - Revisão

(Versão para minimização)

inicie com uma solução S

faça

melhora \leftarrow False

$S' \leftarrow \underline{\text{seleciona}}$

$\Delta\text{custo} \leftarrow \text{custo}(S') - \text{custo}(S)$

se $\Delta\text{custo} < 0$ então

$S \leftarrow S'$

melhora $\leftarrow 1$

enquanto (melhora)

Como é o seleciona ?

Busca Local - Revisão

- seleciona pode ser:
 - * H1 - Analise todas as soluções vizinhas e escolha a que leve ao menor Δ custo, pode ser Δ custo ≥ 0 .
 - analisar todas pode ser computacionalmente muito caro
 - * H2 - Analise todas as soluções vizinhas e escolha a que leve ao menor Δ custo, mas com Δ custo < 0 .
 - * H3 - Analise as soluções vizinhas e escolha a primeira que leve ao Δ custo < 0 .

Busca Local - exemplo - sequenciamento

- Sequenciar 4 tarefas em uma máquina com o objetivo de minimizar o atraso
 - * Tempos de processamento = $(6, 4, 8, 2)$
 - * Datas de entrega = $(9, 12, 15, 8)$
- Suponha a solução $S = (1, 2, 3, 4)$, construída aleatoriamente (poderíamos pensar em uma heurística de construção mais elaborada)
 - * **término** = $(6, 10, 18, 20)$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 - * **Atraso** = $0 + 0 + 3 + 12 = 15$

Busca Local - exemplo - sequenciamento

- $S = (1 \ 2 \ 3 \ 4)$, atraso = 15
- Vizinhança sugerida: trocar a tarefa i com a tarefa j

Busca Local - exemplo - sequenciamento

- Qual movimento realizar?
 - * Se a cada iteração forem analisados todos os movimentos possíveis, então será o $i=2$ e $j=4$.
 - * Se for realizado o primeiro movimento que melhora a função objetivo então será $i=1$ e $j=4$.
- Vamos supor que todos os movimentos estão sendo analisados. Então
 - * $S = (1 \text{ } 4 \text{ } 3 \text{ } 2)$, atraso = 9

Busca Local - exemplo - sequenciamento

- $S = (1 \ 4 \ 3 \ 2)$, atraso = 9

<i>i</i>	<i>j</i>	valor movimento	atraso	
1	2	+2	11	
1	3	+8	17	
1	4	0	9	
2	3	-4	5	←
2	4	+6	15	
3	4	+7	16	

- $S = (1 \ 4 \ 2 \ 3)$, atraso = 5

Busca Local - exemplo - sequenciamento

- $S = (1 \ 4 \ 2 \ 3)$, atraso = 5

<i>i</i>	<i>j</i>	valor movimento	atraso
1	2	+3	8
1	3	+10	15
1	4	0	5
2	3	-4	9
2	4	-4	9
3	4	+13	18

- $S = (4 \ 1 \ 2 \ 3)$, atraso = 5

Busca Local - exemplo - sequenciamento

- $S = (4 \ 1 \ 2 \ 3)$, atraso = 5

<i>i</i>	<i>j</i>	valor movimento	atraso
1	2	+3	8
1	3	+8	13
1	4	0	5
2	3	+4	9
2	4	+5	10
3	4	+18	23

- Entrou em ciclo.
- Como evitar isso? Resposta: Lista Tabu.

Busca Tabu - memória de curto prazo

- Memória de curto prazo: guarda atributos de soluções recentes (recentidade).
- A estratégia é de **classificar como proibido** certos movimentos ou soluções, através de atributos, armazenando-os na lista **tabu** durante um certo **tempo**.
- Atributos selecionados são denominados de **tabu-ativos**.
- Soluções que contém elementos **tabu-ativos** tornam-se **tabu**.
- Para tornar a busca mais flexível, também temos como desclassificar um movimento de tabu como não tabu por algum **critério de aspiração**.

Busca Tabu - memória de curto prazo

- Elementos
 - * O que proibir e como?
 - Atributo ou solução
 - Lista Tabu
 - * Por quanto tempo?
 - Tempo Tabu
 - * Critério de Aspiração
 - É uma maneira de se retirar o estado tabu-ativo de um movimento se ele se apresenta com características muito boas e não ocasiona ciclo.

Memória de curto prazo: lista tabu e atributo

- O que proibir e como?
- Essa decisão envolve
 - * escolha de estrutura de dados.
 - * o tipo de vizinhança adotada na heurística de busca.
 - * a representação da solução.
 - * como será feita a verificação se certo movimento é tabu

Memória de curto prazo: lista tabu e atributo

Nosso exemplo sequenciamento

- * movimento = sai da solução S1 para solução S2 trocando i com j .
- * Depois de feito esse movimento, o que proibir?
 - ir a solução S1 ?
 - trocar i com j , ou vice-versa?
 - trocar i com alguém (ou j com alguém)?
- * Cada uma dessas regras tem a sua influência na eficiência do algoritmo.

Memória de curto prazo: lista tabu e atributo

- ir a solução S1.
 - * atributo → uma solução inteira, um vetor.
 - * lista tabu → lista de soluções
- trocar i com j , ou vice-versa.
 - * atributo → (i, j)
 - * lista tabu → matriz $n \times n$
- trocar i com alguém (ou j com alguém).
 - * atributo → i
 - * lista tabu → vetor de tamanho n
- *Qual desses critérios utilizar para construir a lista tabu?*

Memória de curto prazo: lista tabu e atributo

- Exemplo do Problema do Caixeiro Viajante

- * solução → Sequência de cidades (vetor)
- * vizinhança
 - trocar a cidade i com a cidade j
- Exemplos de atributos:
 - * solução (vetor) → ir a uma solução S.
 - * (i, j) → proibir a troca de i com j .

Memória de curto prazo: lista tabu e atributo

- Exemplo do Problema da Mochila
- Problema da mochila 0-1
 - * solução $\rightarrow x_i = 1$ ou 0 (o item i está ou não na mochila)
 - * vizinhança
 - tirar o item i da mochila ($x_i=1 \rightarrow x_i=0$)
 - colocar o item i na mochila ($x_i=0 \rightarrow x_i=1$)
- Exemplos de atributos:
 - * solução (vetor) \rightarrow ir a uma solução S.
 - * (i) \rightarrow proibir i de sair (ou entrar) na mochila.

Memória de curto prazo: lista tabu e atributo

- Exemplo de Planejamento da Produção
- Planejamento da produção de N itens em T períodos
 - * solução → x_{it} = quantidade
 - * vizinhança : mover uma quantidade q da produção do item i do período t para o período t' .
- Exemplos de atributos:
 - * (q,i,t,t') → proibir q de i de sair de t' e ir para t
 - * (i,t,t') → proibir i de sair de t' e para ir a t .
 - * (i,t) → proibir i de sair (ou ir para) t .

Memória de curto prazo: tempo tabu

- *Tempo tabu* (Quanto tempo o atributo permanece proibido?)
 - * número fixo
 - descoberto experimentalmente.
 - relacionado com o tamanho do problema
 - relacionado com a vizinhança adotada
 - todas as anteriores
 - * número sorteado aleatoriamente entre [Tmin,Tmax]
 - Tmin e Tmax escolhidos como o número fixo
 - * o número fixo ou o Tmin e Tmax podem variar durante a busca

Memória de curto prazo: critério de aspiração

- *Critério de aspiração:*
 - * Esse critério é utilizado para liberar um movimento de seu estado tabu-ativo antes do seu tempo tabu terminar.
- Por exemplo,
 - * Se um certo movimento tabu levar a uma solução melhor que a incumbente (melhor encontrada até o momento) libere-o.
 - * Se todos os movimentos possíveis de serem realizados estão tabu, libere um deles segundo algum critério.

Memória de curto prazo

- Exemplo: Problema do Caixeiro Viajante
- Como armazenar a lista tabu e tempo tabu?
- Proposta: uma matriz ($n \times n$)

Memória de curto prazo

- Lista Tabu & Tempo Tabu & Critério de Aspiração

Lista Tabu Simples

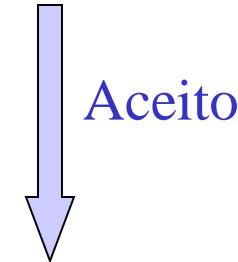
	C2	C3	C4	C5
C1				
C2				
C3			3	
C4				
C5				

Tempo
tabu = 3

Problema PCV

C1	C2	C3	C4	C5
----	----	----	----	----

Initial solution
 $f = 100$



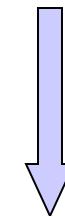
	C2	C3	C4	C5
C1	3			
C2				
C3			2	
C4				
C5				

Tempo
tabu
atualizado

C1	C2	C3	C4	C5
----	----	----	----	----

$f = 90$

Proibido?



$f = 80$ Aceito

Exemplo - sequenciamento

- Solução inicial: $S = (1 \ 4 \ 3 \ 2)$, atraso = 9
- Busca local
 - * movimento: trocar i com j
 - * analise todas as soluções vizinhas e escolha a que leve a menor Δ custo
- Busca Tabu
 - * atributo (i, j) , ou seja, proibido de trocar i com j .
(estrutura de dados)
 - * tempo tabu: não definido
 - * Critério de aspiração: quando for atingida uma solução melhor que a incumbente.

Exemplo - sequenciamento

- $S = (1 \ 4 \ 3 \ 2)$, atraso = 9

i	j	valor movimento	atraso
1	2	+2	11
1	3	+8	17
1	4	0	9
2	3	-4	5 
2	4	+6	15
3	4	+7	16

- $S = (1 \ 4 \ 2 \ 3)$, atraso = 5
- TABU = {(2,3)}, atraso* = 5

Exemplo - sequenciamento

- $S = (1 \ 4 \ 2 \ 3)$, atraso = 5

i	j	valor movimento	atraso
1	2	+3	8
1	3	+10	15
1	4	0	5 
2	3	+4	9
2	4	+4	9
3	4	+13	18

- $S = (4 \ 1 \ 2 \ 3)$, atraso = 5
- TABU = $\{(2,3), (1,4)\}$, atraso* = 5

Exemplo - sequenciamento

- $S = (4 \ 1 \ 2 \ 3)$, atraso = 5

i	j	valor movimento	atraso
1	2	+3	8 ←
1	3	+8	13
1	4	0	5 ← Tabu
2	3	+4	9
2	4	+5	10
3	4	+18	23

- $S = (4 \ 2 \ 1 \ 3)$, atraso = 8
- $TABU = \{(2,3), (1,4), (2,1)\}$, atraso* = 5

Exemplo - sequenciamento

- $S = (4 \ 2 \ 1 \ 3)$, atraso = 8

i	j	valor movimento	atraso
1	2	-3	5 ← Tabu
1	3	+3	11
1	4	+1	9
2	3	+7	15
2	4	0	8 ←
3	4	+13	21

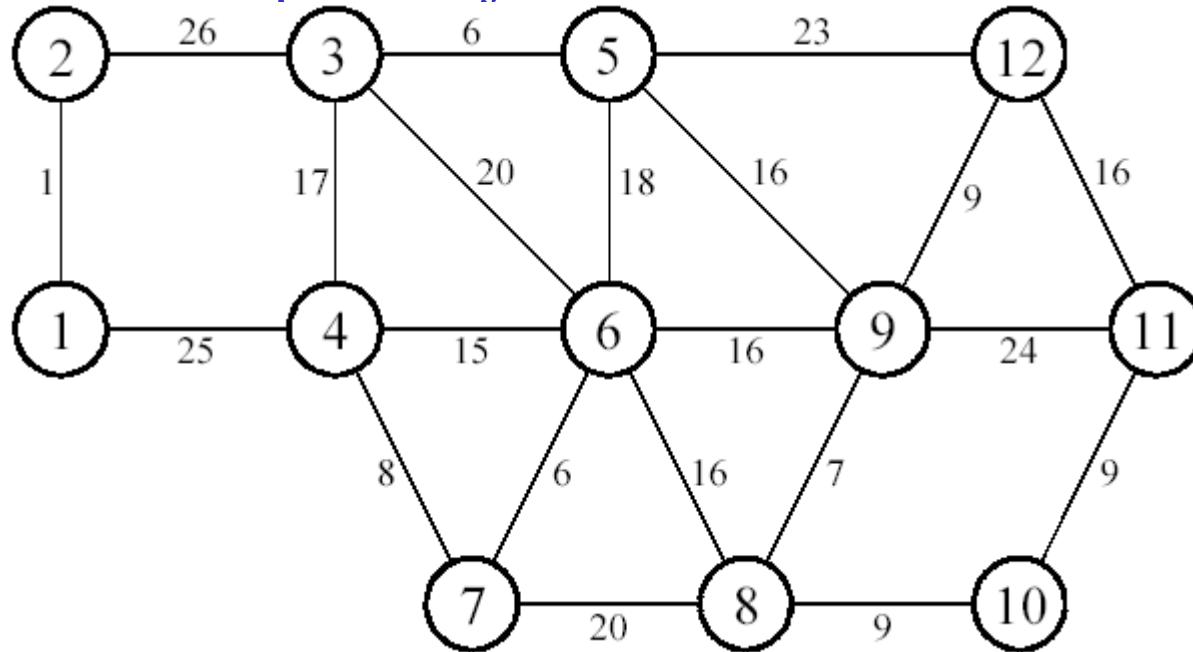
- $S = (2 \ 4 \ 1 \ 3)$, atraso = 8
- $TABU = \{(2,3), (1,4), (2,1), (2,4)\}$, atraso* = 5

Exemplo - sequenciamento

- E assim continua a busca...
- É preciso atualizar a lista, dependendo do tipo de estrutura de dados utilizada. Nesse caso poderíamos usar uma matriz.
- Cada movimento que descartávamos, pois era tabu, poderia ter sido feito se levasse a uma solução com atraso menor que 5 (critério de aspiração)

Exemplo - K-árvore Mínima

- Dado um grafo, encontrar uma árvore com k arestas de forma que a soma dos pesos seja mínima.



- Solução Inicial obtida de forma gulosa para k=4: escolher a aresta de mínimo acréscimo na soma dos pesos.

Exemplo - K-árvore Mínima

- Construção gulosa:

Passo	Candidatos	Seleção	Peso Total
1	(1, 2)	(1, 2)	1
2	(1, 4), (2, 3)	(1, 4)	26
3	(2, 3), (3, 4), (4, 6), (4, 7)	(4, 7)	34
4	(2, 3), (3, 4), (4, 6), (6, 7), (7, 8)	(6, 7)	40

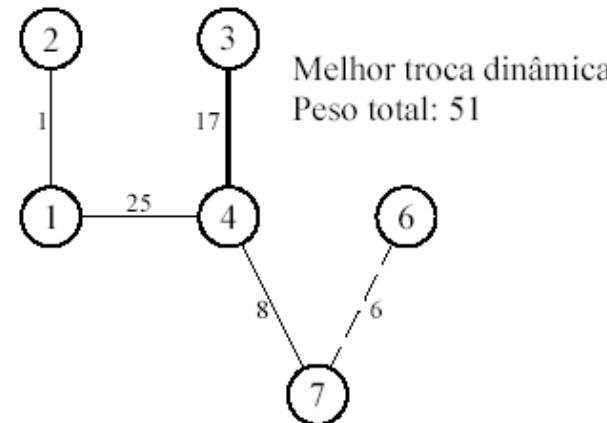
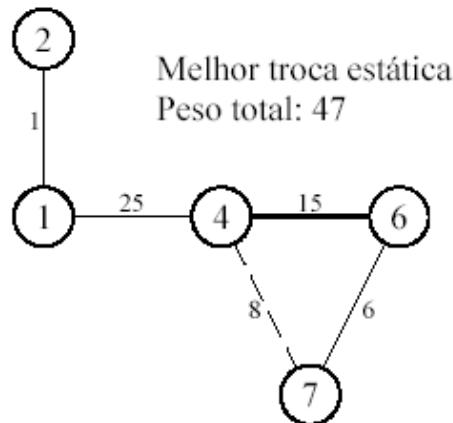
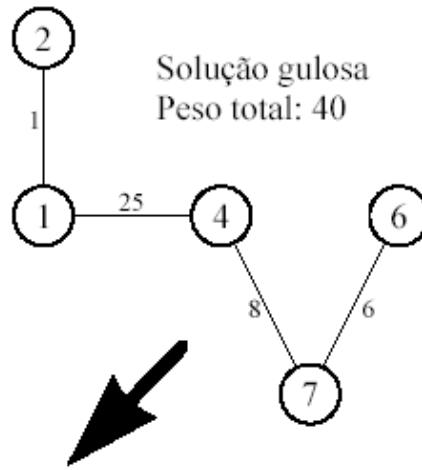
- Atributos de uma solução: arestas na árvore e aresta fora da árvore.
- Movimento: troca de uma aresta na árvore por uma aresta fora da árvore.
- Escolha do movimento: o melhor movimento não tabu em relação a função objetivo.

Exemplo - K-árvore Mínima

- Critério de Aspiração: execute um movimento tabu se a solução resultante tem um valor de função melhor que o encontrado até o momento.
- Critério de Ativação Tabu:
 - * Um movimento é tabu se:
 1. a aresta retirada ou a aresta incluída for tabu-ativa;
 2. a aresta retirada e a aresta incluída são tabu-ativas.
 - * O critério 1 é mais restritivo (proíbe mais movimentos)
- Duração tabu (critério 1):
 - Duração tabu de aresta retirada: 2 iteração;
 - Duração tabu de aresta incluída: 1 iteração.

Exemplo - K-árvore Mínima

- Tipos de movimentos de troca:

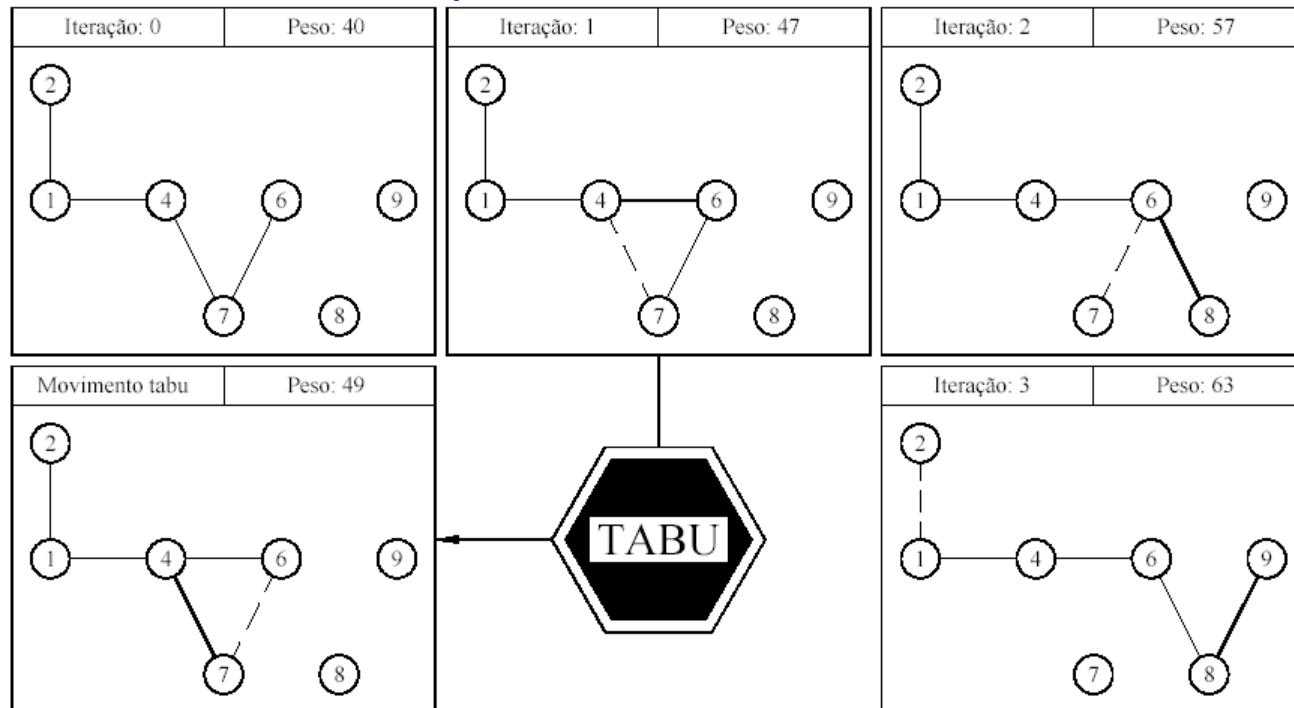


Exemplo - K-árvore Mínima

- Iterações da Busca Tabu

Iteração	Duração Tabu		Incluída	Retirada	Peso
	1	2			
1			(4, 6)	(4, 7)	47
2	(4, 6)	(4, 7)	(6, 8)	(6, 7)	57
3	(6, 8), (4, 7)	(6, 7)	(8, 9)	(1, 2)	63

- Efeitos da memória de curto prazo:



Exemplo - K-árvore Mínima

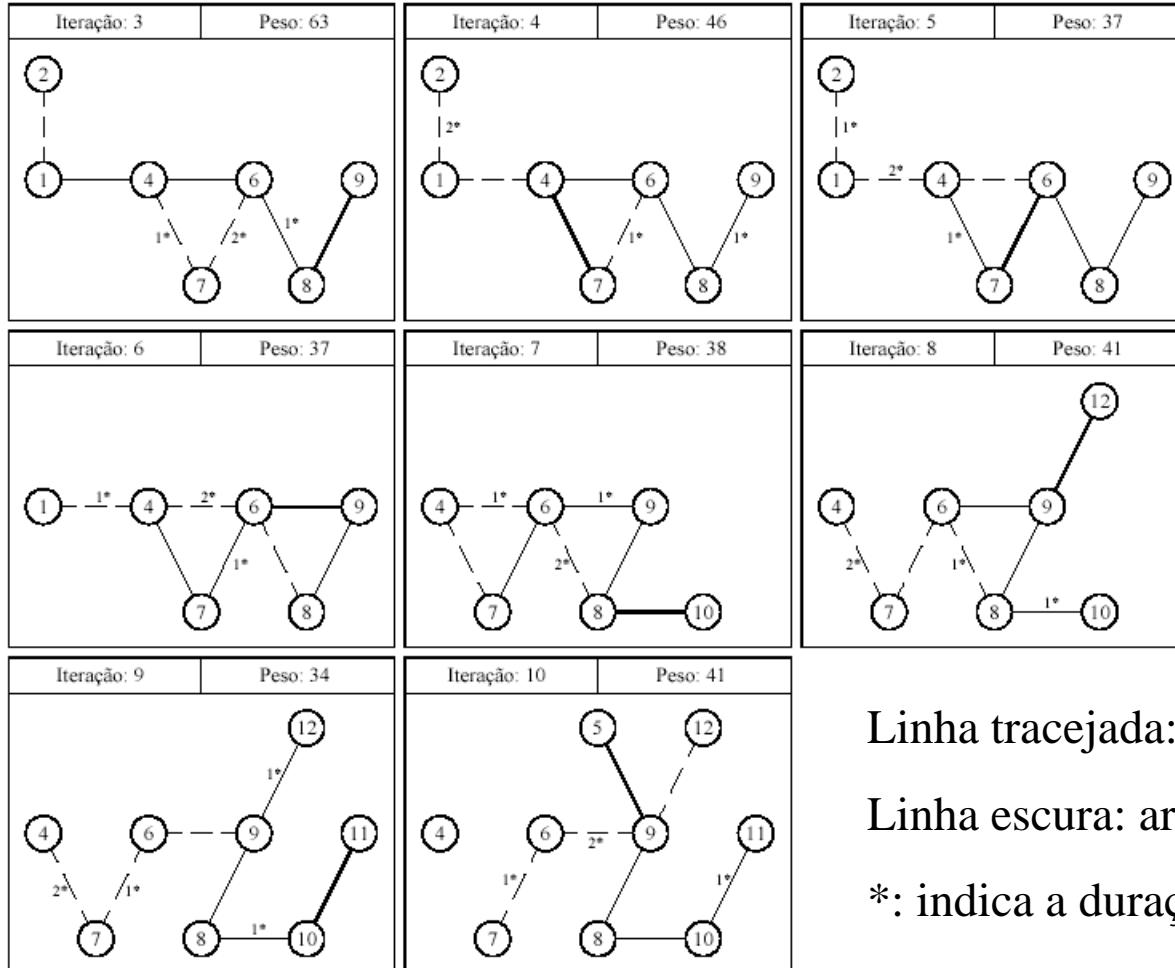
- Iteração da Busca Tabu

Iteração	Duração Tabu		Incluída	Retirada	Valor do Movimento	Peso
	1	2				
3	(6, 8), (4, 7)	(6, 7)	(8, 9)	(1, 2)	8	63
4	(6, 7), (8, 9)	(1, 2)	(4, 7)	(1, 4)	-17	46
5	(1, 2), (4, 7)	(1, 4)	(6, 7)	(4, 6)	-9	37*
6	(1, 4), (6, 7)	(4, 6)	(6, 9)	(6, 8)	0	37
7	(4, 6), (6, 9)	(6, 8)	(8, 10)	(4, 7)	1	38
8	(6, 8), (8, 10)	(4, 7)	(9, 12)	(6, 7)	3	41
9	(4, 7), (9, 12)	(6, 7)	(10, 11)	(6, 9)	-7	34*
10	(6, 7), (10, 11)	(6, 9)	(5, 9)	(9, 12)	7	41

* : representa a melhor solução ao longo das iterações.

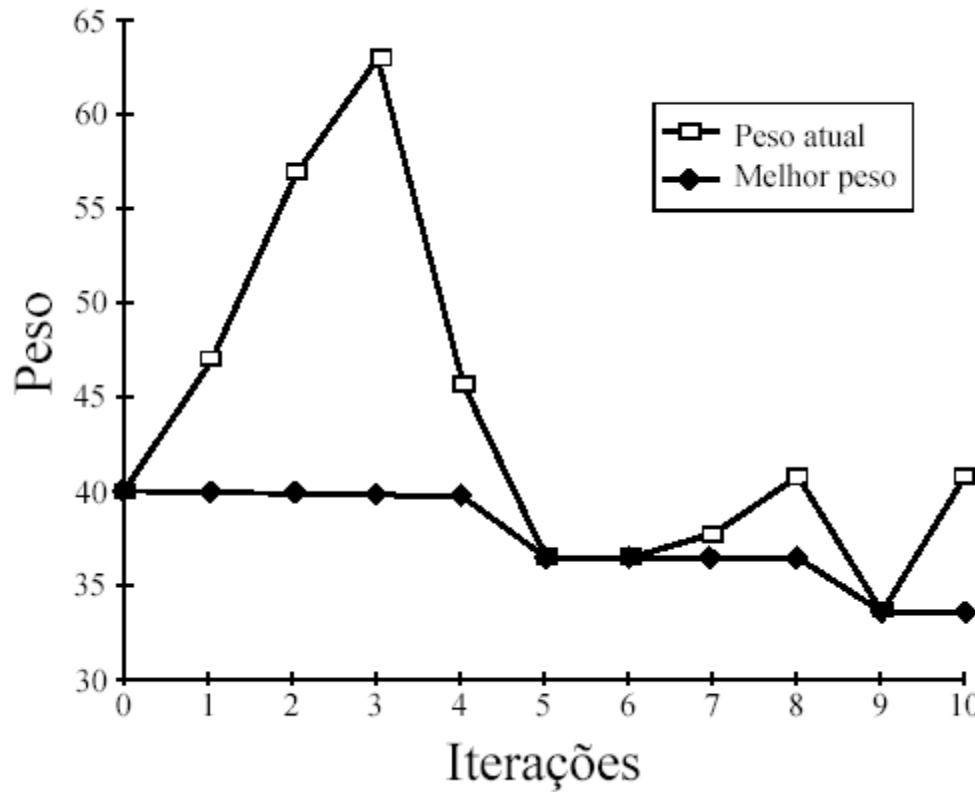
Exemplo - K-árvore Mínima

- Representação gráfica das iterações:



Exemplo - K-árvore Mínima

- Trajetória da Busca Tabu



- O gráfico ilustra os “desvios” que o algoritmo para escapar dos ótimos locais.

Memória de longo prazo

- Modificação da vizinhança para incluir soluções que nunca apareceram
 - * selecionar soluções de elite (ótimos locais de alta qualidade)
- Estratégias mais comum:
 - Medidas de frequência (estratégia mais utilizada)
 - Intensificação e diversificação.

Memória de longo prazo: medida de frequência

- Contabilizar informações durante a busca tabu de curto prazo.
- Dois exemplos de medidas de frequência:
 - * Medida de residência → número de vezes que a tarefa i esteve presente a posição j .
 - análise de um conjunto de soluções
 - * Medida de transição → número de vezes que a tarefa i foi trocada de posição.
 - análise dos passos da heurística

Relembrando o problema de sequenciamento

- Sequenciar 4 tarefas em uma máquina com o objetivo de minimizar o atraso
 - * Tempos de processamento = (6, 4, 8, 2)
 - * Datas de entrega = (9, 12, 15, 8)
- Suponha a solução $S = (1, 2, 3, 4)$, construída aleatoriamente (poderíamos pensar em uma heurística de construção mais elaborada)
 - * **término** = (6, 10, 18, 20)
 ↓ ↓ ↓ ↓
 - * **Atraso** = 0 + 0 + 3 + 12 = 15

Exemplo - Sequenciamento

- Suponha os seguintes movimentos:

(4,2) 1 4 3 2

(2,3) 1 4 2 3

(4,1) 4 1 2 3

(2,1) 4 2 1 3

(2,4) 2 4 1 3

(3,1) 2 4 3 1

(3,2) 3 4 2 1

(4,3) 4 3 2 1

(1,2) 4 3 1 2

(1,3) 4 1 3 2

residência

2 2 3 3

2 1 4 3

1 2 3 4

5 5 0 0

transição

5 5 5 3

apenas os numeradores

Exemplo - Sequenciamento

- Supondo o denominador como sendo o número de soluções = 10 (histórico)

Residência (normalizada)

0.2 0.2 0.3 0.3

0.2 0.1 0.4 0.3

0.1 0.2 0.3 0.4

0.5 0.5 0.0 0.0

Matriz residência
de todas as soluções

Transição (normalizada)

0.5 0.5 0.5 0.3

Memória de longo prazo: medida de frequência

- Construímos a matriz e o vetor utilizando todas as soluções durante a busca.
- Poderíamos, também, utilizar um subgrupo de soluções
 - * soluções de elite;
 - * em heurísticas composta por diferentes procedimentos, poderíamos construir tais matrizes utilizando apenas soluções produzidas por específicos procedimentos.

Matriz residência para as soluções elite

- Suponha os seguintes movimentos:

(4,2)	1 4 3 2	9
(2,3)	1 4 2 3	5
(4,1)	4 1 2 3	5
(2,1)	4 2 1 3	8
(2,4)	2 4 1 3	8
(3,1)	2 4 3 1	11
(3,2)	3 4 2 1	15
(4,3)	4 3 2 1	13
(1,2)	4 3 1 2	15
(1,3)	4 1 3 2	9

Residência

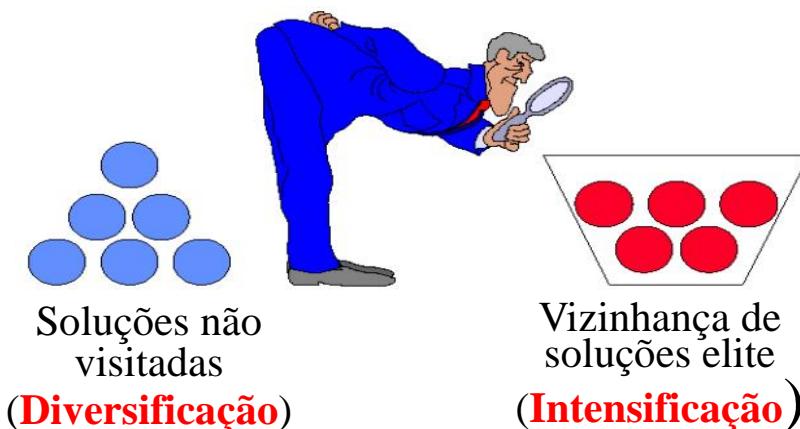
1	1	2	0
1	1	2	0
0	0	0	4
2	2	0	0

Residência
(normalizada)

0.25	0.25	0.5	0.0
0.25	0.25	0.5	0.0
0.0	0.0	0.0	1.0
0.5	0.5	0.0	0.0

Intensificação e Diversificação

- Intensificação:
 - * Uma busca mais minuciosa sobre uma região supostamente promissora.
 - * Concentrar a busca em regiões promissoras do espaço, em torno de boas soluções (soluções elite), modificando as regras de escolha da próxima solução de modo a combinar bons movimentos com a estrutura destas soluções.
- Diversificação:
 - * Guiar a busca para regiões inexploradas do espaço de soluções.
 - * Fazer com que a busca visite regiões ainda inexploradas. Pode ser feito penalizando-se movimentos que voltam a utilizar elementos que estiveram frequentemente presentes em soluções visitadas anteriormente e/ou incentivando a entrada de elementos pouco utilizados no passado.



Estratégia de intensificação

- A idéia é intensificar a busca em regiões onde parece ter maiores chances de apresentar soluções de qualidade.
- Exemplos:
 - * modificar regras de escolha de forma a encorajar
 - movimentos e soluções com características “boas” no decorrer da história da busca;
 - combinar movimentos com a estrutura dessas soluções;
 - modificar a função objetivo **favorecendo** os atributos de alta frequência (função objetivo - $k * \Sigma freq(i)$ para minimização)
 - * reiniciar a busca a partir uma solução “boa” (soluções elite).

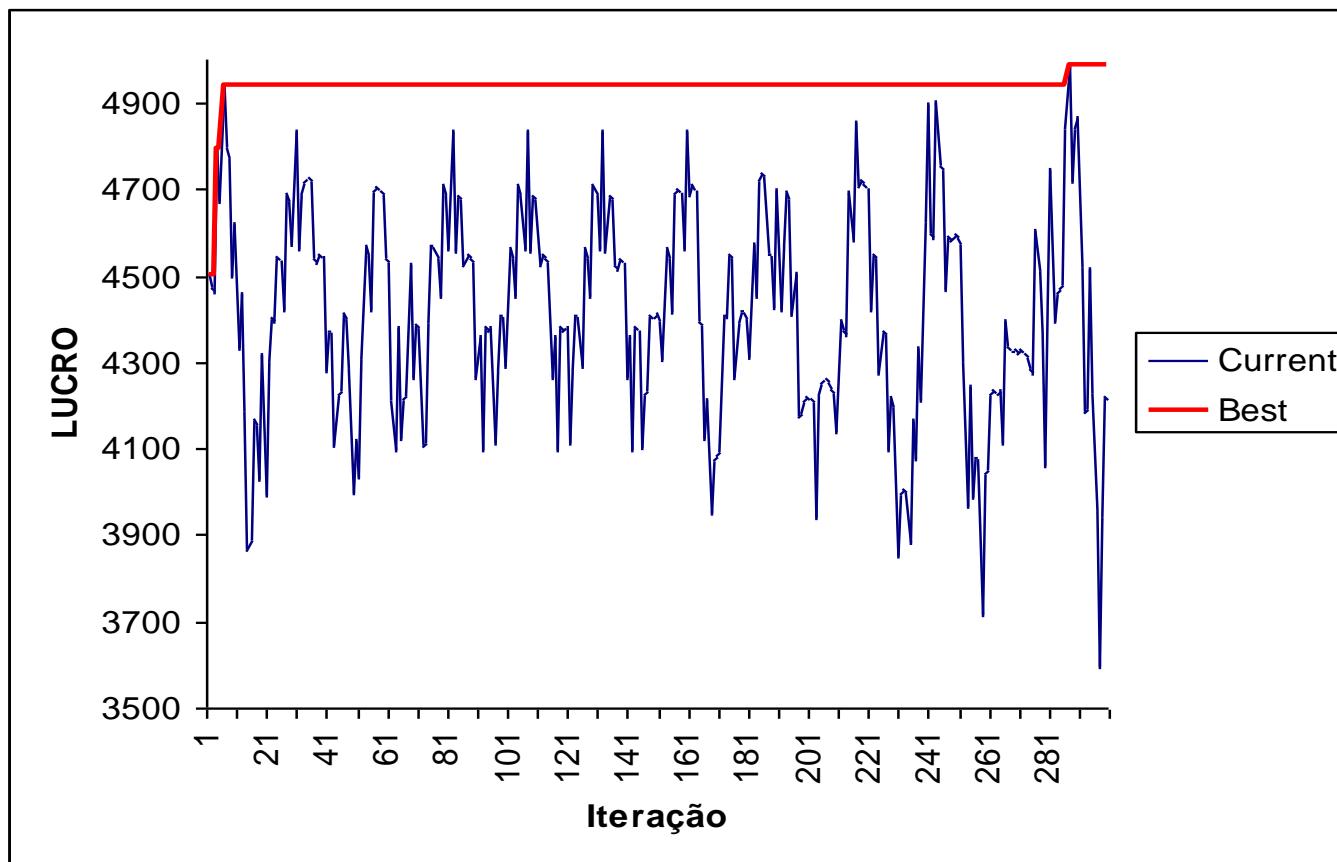
Estratégia de diversificação

- A idéia é tentar explorar regiões ainda inexploradas ou pouco exploradas.
- Exemplos:
 - * modificando as regras de escolha de forma a
 - desencorajar movimentos e soluções muito frequentes no decorrer da história da busca (penalizações)
 - incentivar movimentos e soluções pouco frequentes no decorrer da história da busca (penalizações)
 - modificar a função objetivo **penalizando** os atributos de alta frequência (função objetivo + $k * \Sigma freq(i)$ para minimização)
 - * reiniciar a busca utilizando as informações de frequência

Medida frequência x intensificação e diversificação

- Atributos com:
 - * alta frequência → a partir das soluções de elite → atributos atrativos → intensificação
 - * baixa frequência → a partir das soluções de elite → atributos atrativos → diversificação
 - * baixa frequência → feita a partir de todas as soluções → atributos atrativos → diversificação
 - * etc.

Efeito da Memória de Longo Prazo



Estratégia de Intensificação

- Dois tipos mais comuns de estratégias:
 1. Modificar as regras de escolha para estimular combinações de movimentos e características de soluções historicamente boas;
 2. Retornar à regiões atrativas para pesquisa mais detalhada

Um enfoque simples para o 2o. caso:

Aplicar Busca Tabu com Memória de Curto Prazo

Aplicar uma Estratégia de Seleção de Elite

While Critério_de_Parada **do**

Escolha uma das soluções da elite;

Aplicar Busca Tabu de Curto Prazo a partir da solução escolhida;

Adicionar novas soluções para a lista de elite quando for aplicável;

End-do;

Estratégia de Diversificação

Um Algoritmo Simples:

Aplicar Busca Tabu com Memória de Curto Prazo;

Manter a memória baseada em frequência;

While Critério_de_Parada **do**

 Aplicar Busca Tabu de Curto Prazo a partir da solução escolhida;

 Manter a memória baseada em frequência;

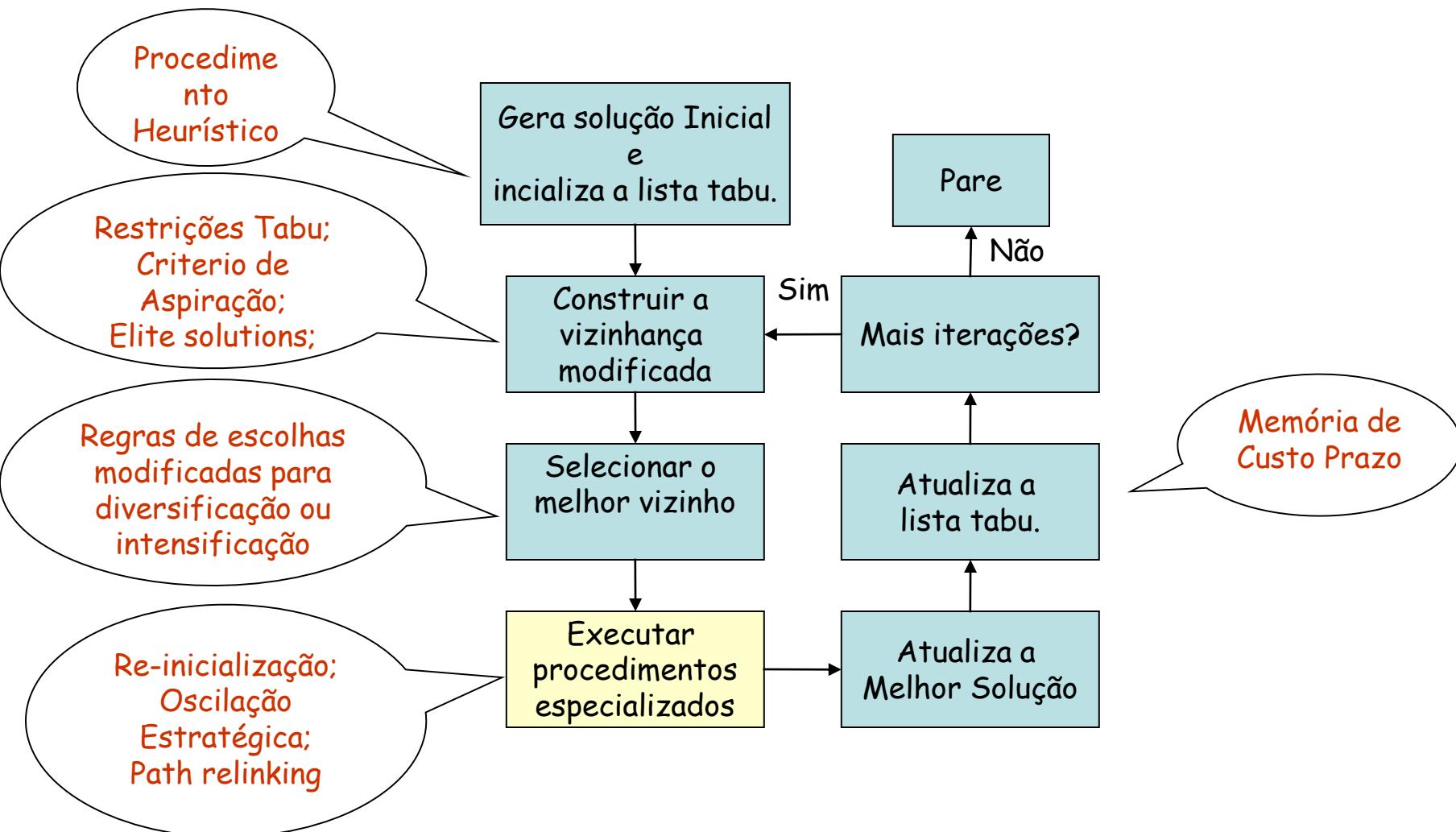
 Criar nova soluções explorando (priorizando) os atributos menos frequentes.

End-do;

Busca tabu

- Características:
 - * uso intensivo de estruturas de memória adaptativas
 - * requer mais engenharia para a sua implementação:
 - muitos parâmetros
 - muitas estruturas e possibilidades de condução da busca
 - * soluções de boa qualidade: as melhores soluções encontradas para muitos problemas
 - * depende pouco da solução inicial, pois possui mecanismos eficientes que permitem escapar de ótimos locais
 - * robustez

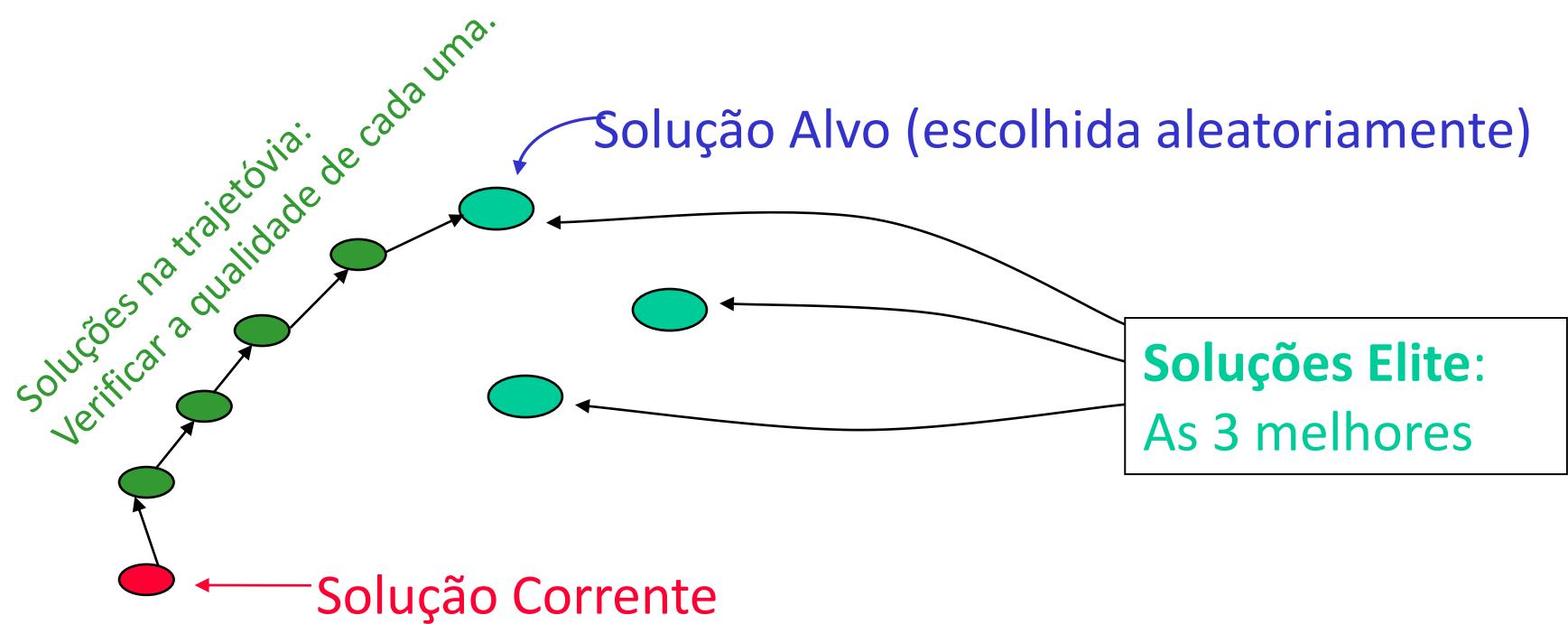
Framework para Busca Tabu



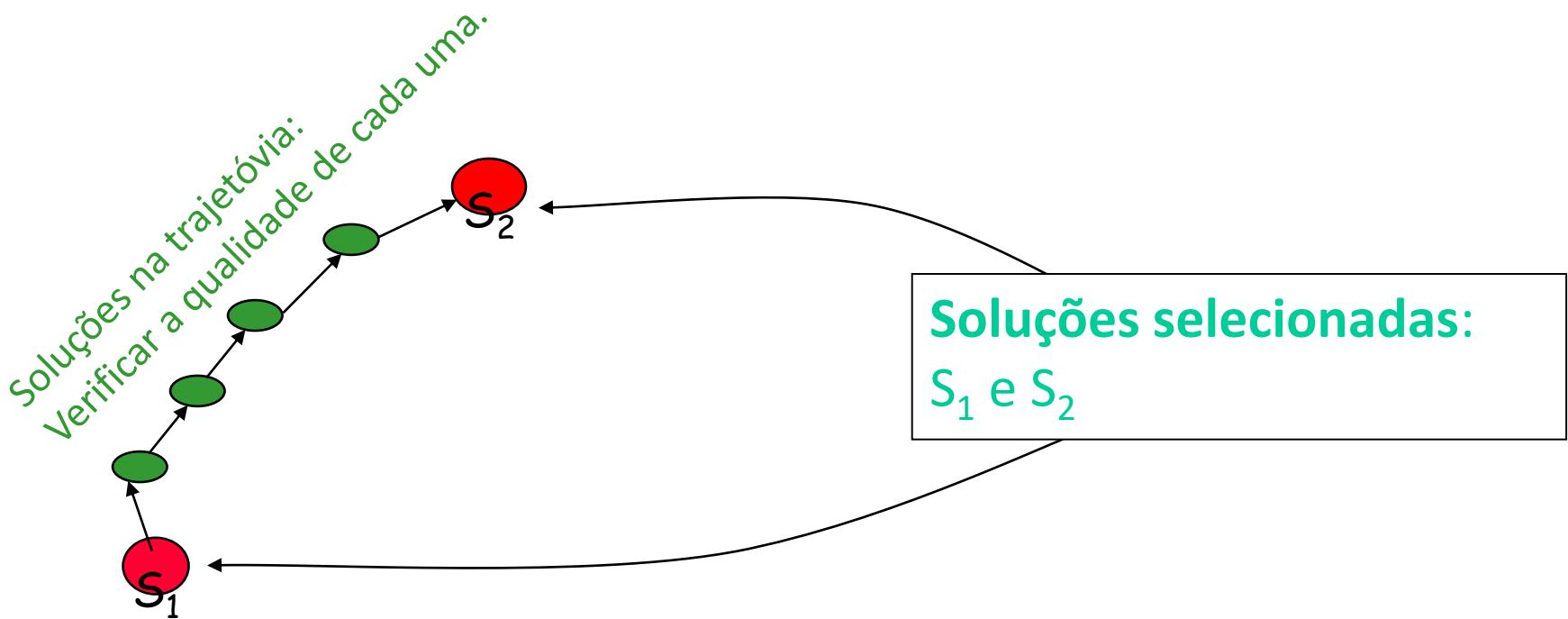
Outras Estratégias

- Oscilação Estatégica
 - * Usado em memória de curto e longo prazo
 - * alternar a busca entre região factível e região infactível
 - * alternar entre diversificação e intensificação
- *Path Relinking*
 - * armazenar um conjunto de soluções de elite
 - * para cada par de soluções de elite:
 - identificar as diferenças entre elas ;
 - verificar a existência de uma trajetória entre as melhores soluções;
 - Selecionar nesta trajetória de soluções melhores do que as extremidades.

Path Relinking



Path Relinking



Questões

- Busta Tabu permite explorar soluções piores que a solução corrente?
- Como implementar uma lista tabu?
- O que é o Tempo Tabu (*Tabu-Tenure*)?
- Porque armazenar somente atributos ao invés da solução completa?
- Para que serve o critério de aspiração?
- Quais os dois tipos de memória usadas em Busca Tabu?
- Como pode ser implementada Memória de Curto Prazo?
- Como pode ser implementada a Memória de Longo Prazo?