



Uma Introdução à Busca Tabu

André Gomes

*Departamento de Ciência da Computação,
Instituto de Matemática e Estatística,
Universidade de São Paulo, SP, Brasil*

Novembro de 2009

INTRODUÇÃO

- Método de Busca Local
 - Explora o Espaço de Soluções movendo-se de uma solução para outra que seja seu melhor vizinho
 - Estrutura de memória adaptativa para armazenar as soluções geradas (ou características dessas)
 - Possibilitam escapar de ótimos locais
- Metaheurística

METAHEURÍSTICA

- Usada em casos que podem ser modelados como problemas de maximizar (ou minimizar) uma função cujas variáveis tem certas restrições.
- Estratégia para resolver problemas NP Difíceis.
- Utilizam procedimentos de busca em vizinhanças dentro do espaço de pesquisa
- Tem melhores soluções:
 - evitam paradas prematuras em ótimos locais.
- Combinam escolhas aleatórias e conhecimento histórico para se guiarem pelo espaço de pesquisa

ESPAÇO DE PESQUISA

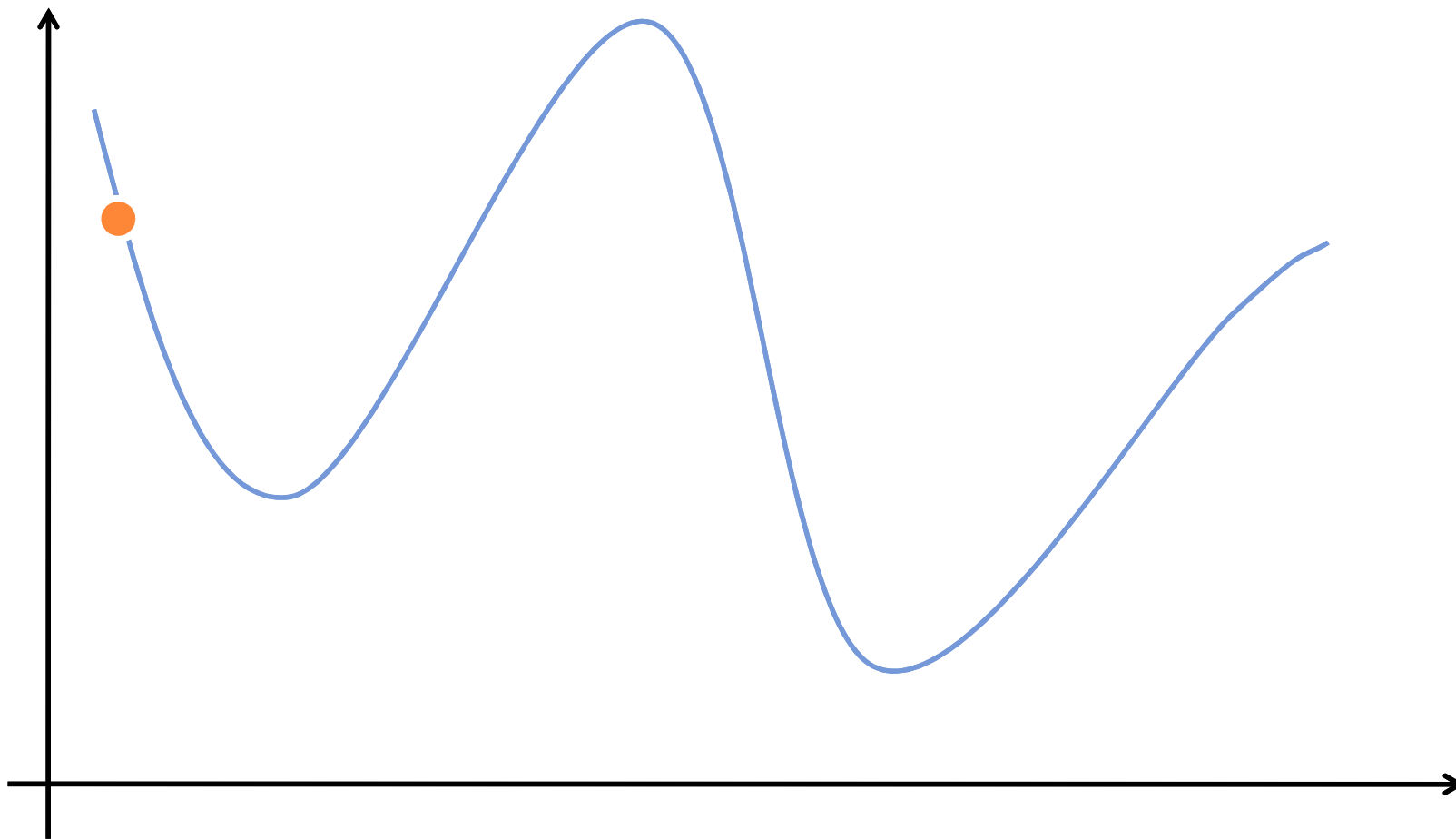
- O espaço de todas as soluções possíveis é chamado de **espaço de pesquisa**.
- A procura por uma solução é equivalente a procurar por algum ponto extremo (mínimo ou máximo) neste espaço.

BT: BREVE HISTÓRICO

- Origem no fim dos anos 60 e início dos anos 70.
- Proposto na sua forma atual por Fred Glover em 1986 [1]
- Aspectos teóricos pesquisados somente após alguns anos depois da definição inicial [2], [3]
 - Por muito tempo sabia-se que funcionava muito bem, mas ninguém sabia explicar o por que.

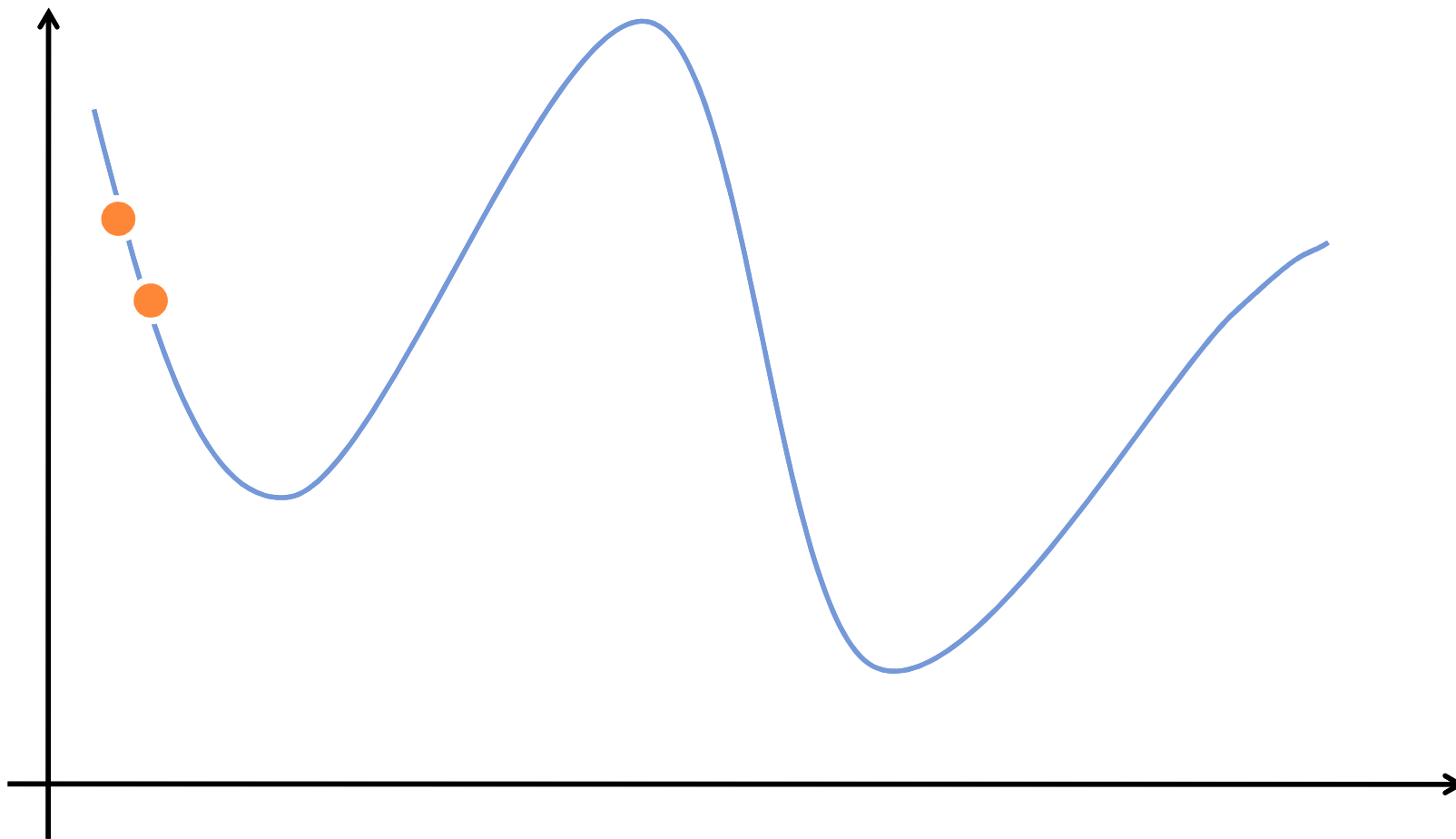
BUSCA TABU

1ª Idéia: Usar heurística de descida



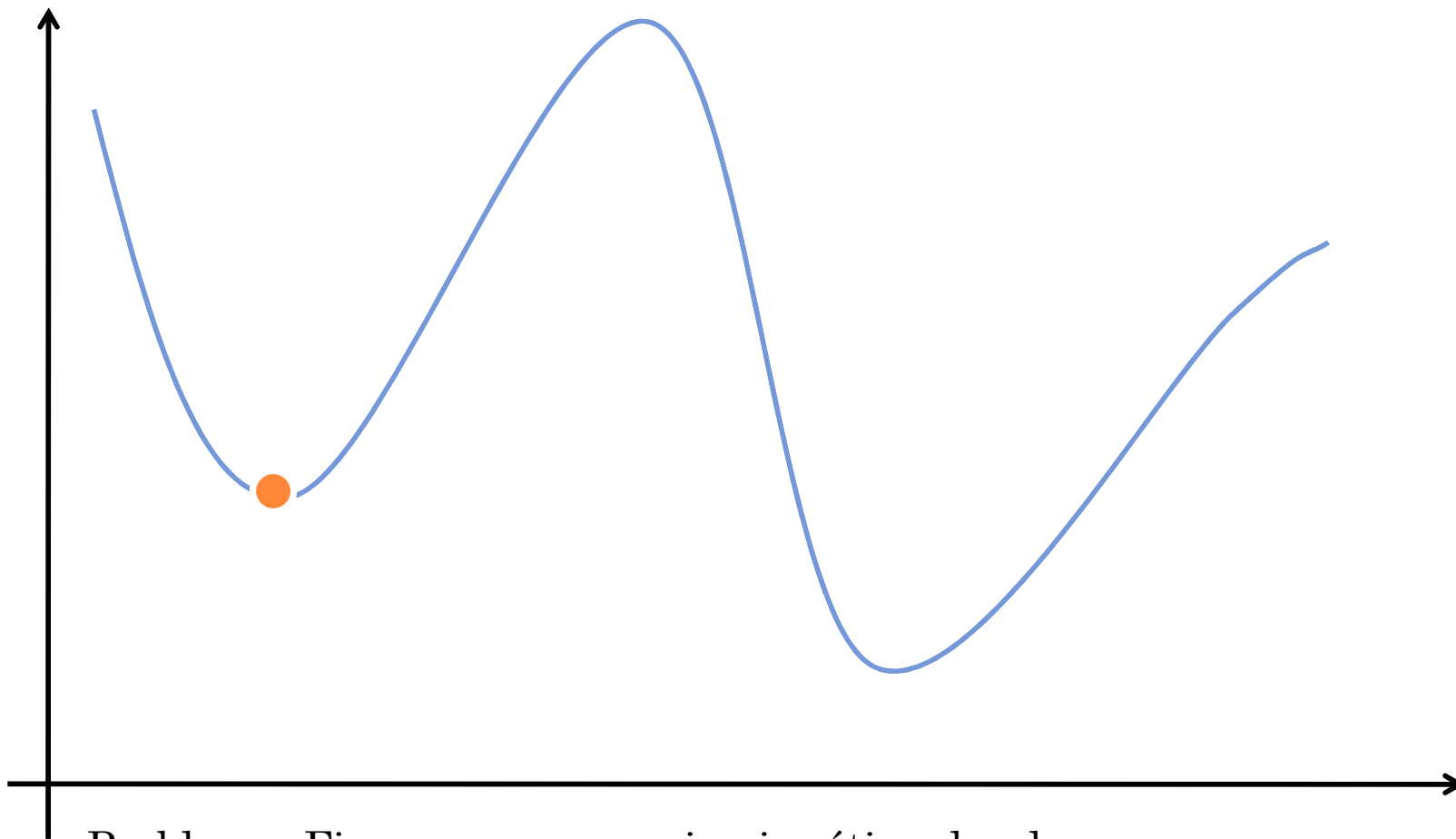
BUSCA TABU

1ª Idéia: Usar heurística de descida



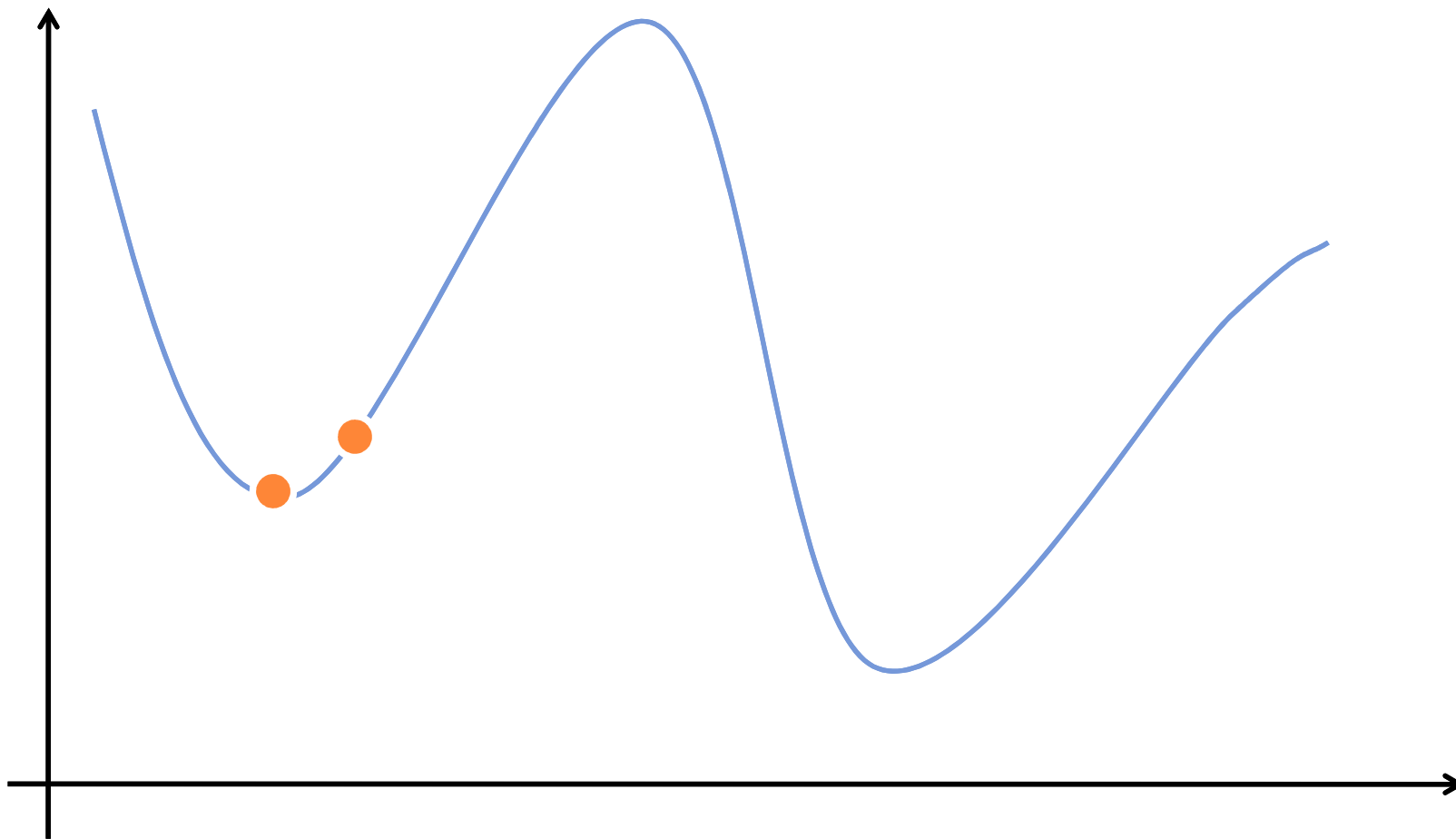
BUSCA TABU

1ª Idéia: Usar heurística de descida



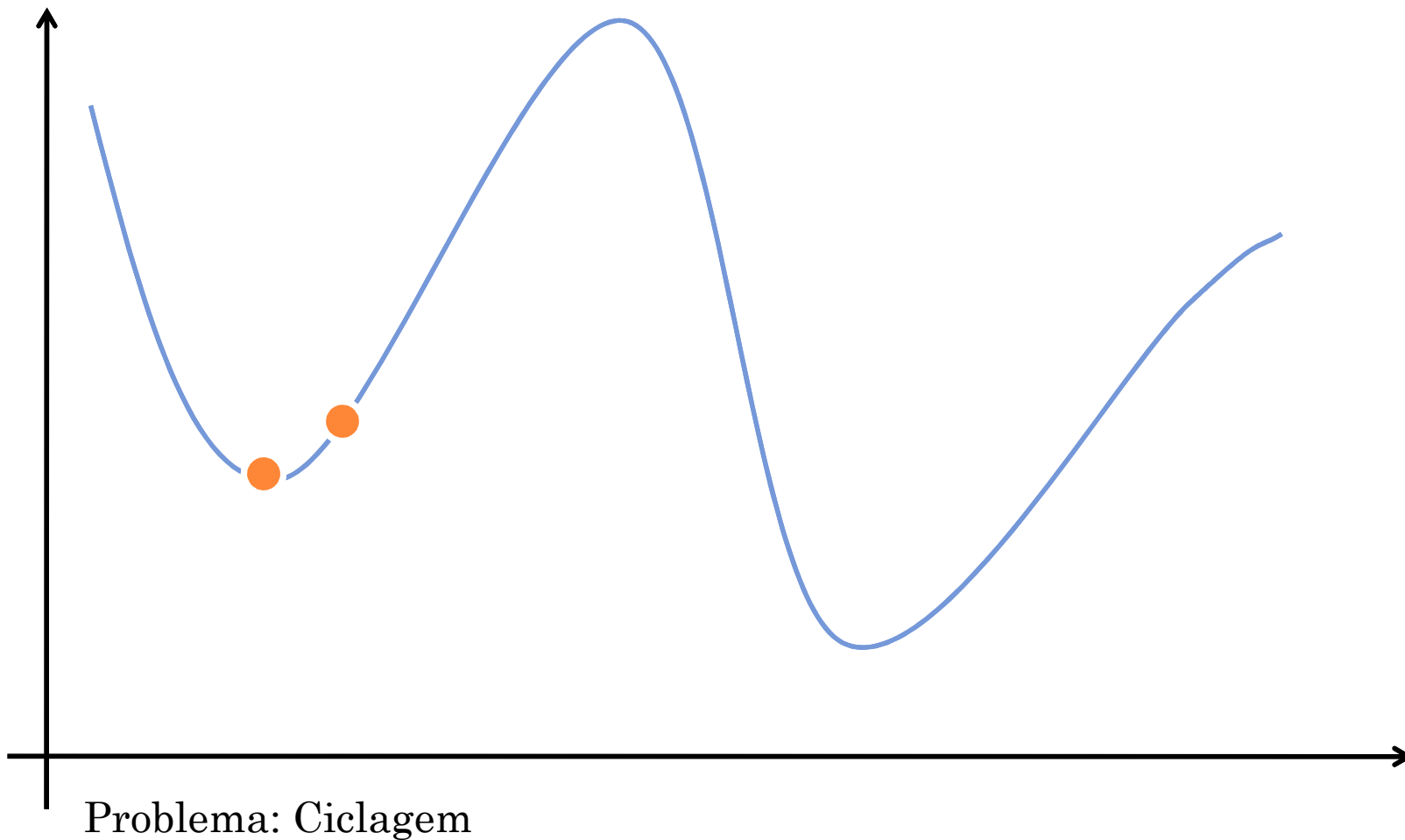
BUSCA TABU

2ª Idéia: Mover-se para o melhor vizinho, que pode ser de piora



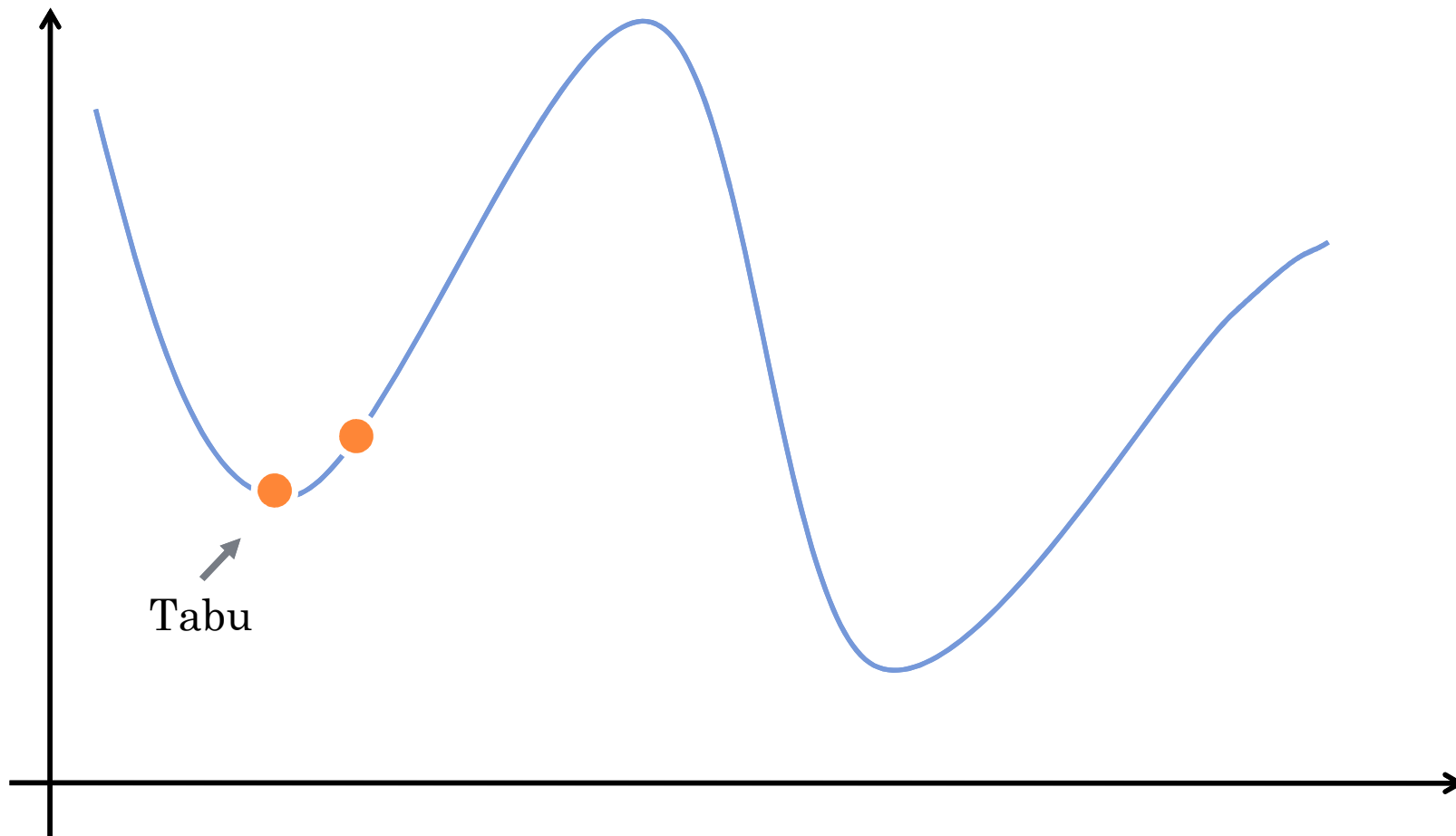
BUSCA TABU

2ª Idéia: Mover-se para o melhor vizinho, que pode ser de piora



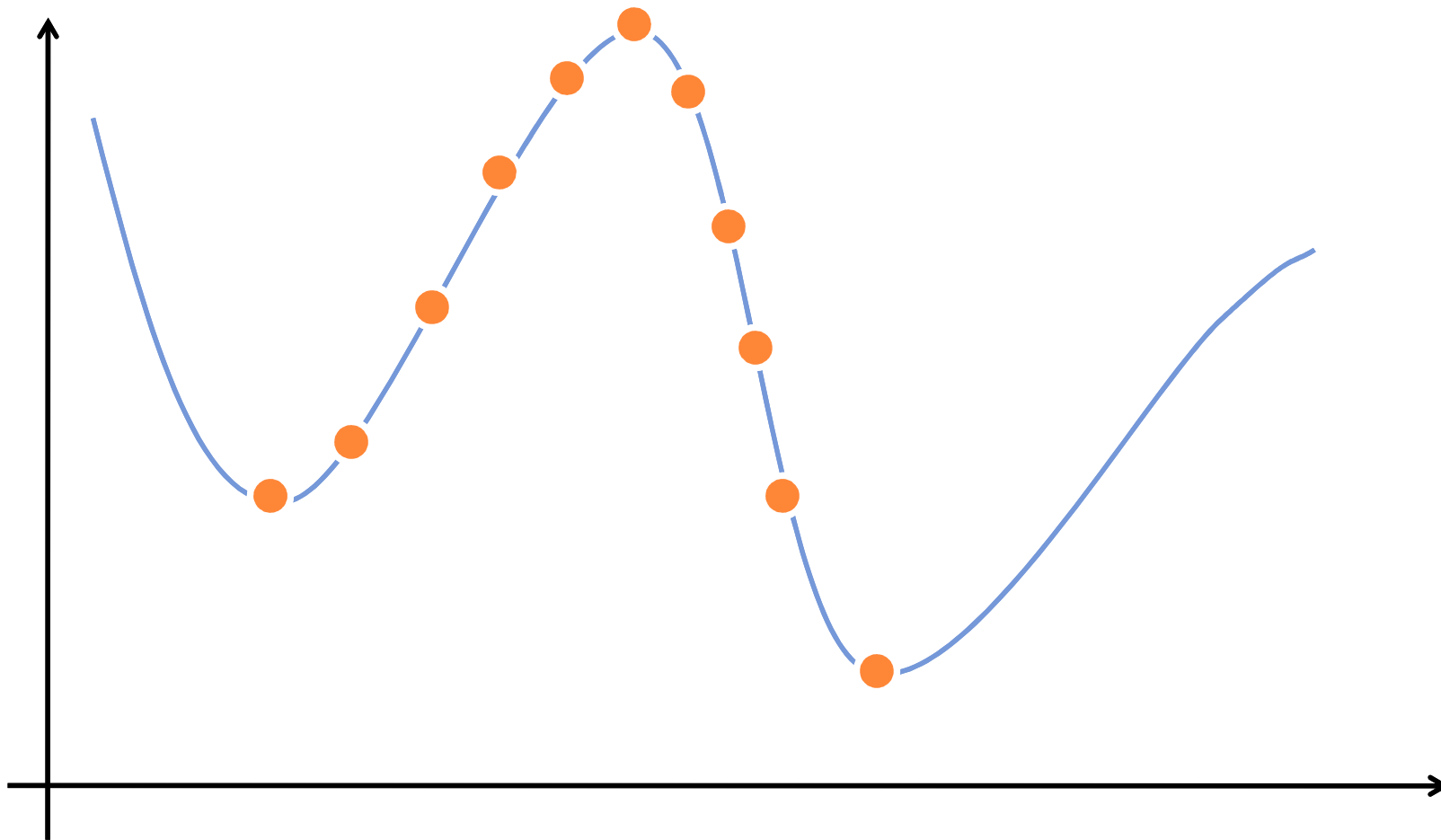
BUSCA TABU

3ª Idéia: Criar uma lista tabu



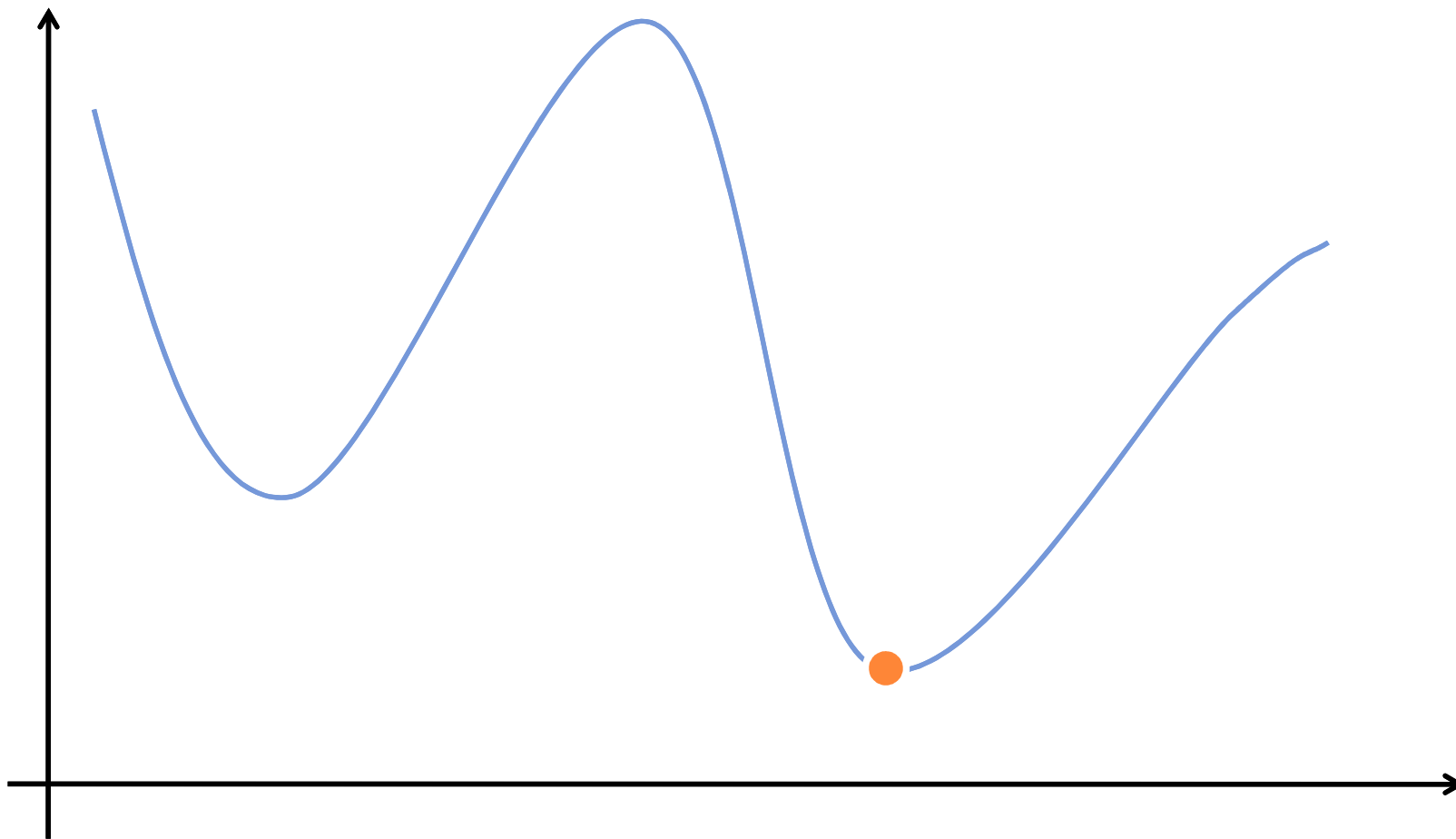
BUSCA TABU

3ª Idéia: Criar uma lista tabu



BUSCA TABU

3ª Idéia: Criar uma lista tabu



PROBLEMAS

- Inviável armazenar todas soluções geradas
 - Idéia: Armazenar apenas as últimas $|T|$ soluções geradas
 - Evita ciclos de até $|T|$ iterações
 - Problema: Pode ser inviável armazenar $|T|$ soluções e testar se uma solução está ou não em T
 - Idéia: Criar uma Lista Tabu de movimentos reversos
- A Lista Tabu de movimentos pode ser restritiva
 - Impede a ida à soluções geradas ou não geradas
- O algoritmo pode entrar em looping
 - T pode evitar que isto ocorra
 - garantindo o não retorno, por $|T|$ iterações, a uma solução já visitada anteriormente;
 - Mas, também pode proibir movimentos para soluções que ainda não foram visitadas.

4ª IDÉIA: CRITÉRIO DE ASPIRAÇÃO

- Retirar o status tabu de um movimento em certas circunstâncias:
 - Aspiração por objetivo: aceitar um movimento tabu se ele melhorar o valor da função objetivo global
 - Aspiração por default: Realizar o movimento tabu mais antigo se todos os movimentos possíveis forem tabu.

PROCEDIMENTO BUSCA TABU

procedimento *BT*

1. Seja s_0 solução inicial;
 2. $s^* \leftarrow s$; {Melhor solução obtida até então}
 3. $\text{Iter} \leftarrow 0$; {Contador do número de iterações}
 4. $\text{MelhorIter} \leftarrow 0$; {Iteração mais recente que forneceu s^* }
 5. Seja BTmax o número máximo de iterações sem melhora em s^* ;
 6. $T \leftarrow \emptyset$; {Lista Tabu}
 7. Inicialize a função de aspiração A ;
 8. enquanto $(\text{Iter} - \text{MelhorIter} \leq \text{BTmax})$ faça
 9. $\text{Iter} \leftarrow \text{Iter} + 1$;
 10. Seja $s' \leftarrow s \oplus m$ o melhor elemento de $V \subseteq N(s)$ tal que o movimento m não seja tabu
($m \notin T$)
 ou s' atenda a condição de aspiração ($f(s') < A(f(s))$);
 11. Atualize a Lista Tabu T ;
 12. $s \leftarrow s'$;
 13. se $f(s) < f(s^*)$ então
 14. $s^* \leftarrow s$;
 15. $\text{MelhorIter} \leftarrow \text{Iter}$;
 16. fim-se;
 17. Atualize a função de aspiração A ;
 18. fim-enquanto;
 19. Retorne s^* ;
- fim** *BT*;

DEFINIÇÃO FORMAL: BUSCA TABU

- Busca começa partindo de uma solução inicial s_0 qualquer
- O algoritmo Busca Tabu explora, a cada iteração, um subconjunto V da vizinhança $N(s)$ da solução corrente s .
- O membro s' de V com melhor valor nessa região segundo a função $f(\cdot)$ torna-se a nova solução corrente mesmo que s' seja pior que s
 - $f(s') > f(s)$ para um problema de minimização
- Critério de escolha do melhor vizinho é utilizado para escapar de um ótimo local.
- Esta estratégia pode fazer com que o algoritmo cicle.

DEFINIÇÃO FORMAL: BUSCA TABU (CONT.)

- T clássica:
 - Contém os movimentos reversos dos últimos $|T|$ movimentos realizados
 - $|T|$ é um parâmetro do método
 - Fila de tamanho fixo
 - Quando um novo movimento é adicionado, o mais antigo sai
- Ao explorar V em $N(s)$ da solução corrente,
 - ficam excluídos da busca os vizinhos s' que são obtidos de s por movimentos m que constam na lista tabu.

DEFINIÇÃO FORMAL: BUSCA TABU (CONT.)

- Para cada possível valor v da função objetivo existe um nível de aspiração $A(v)$: uma solução s' em V pode ser gerada se $f(s') < A(f(s))$, mesmo que o movimento m esteja na lista tabu.
- A função de aspiração A é tal que, para cada valor v da função objetivo, retorna outro valor $A(v)$, que representa o valor que o algoritmo aspira ao chegar de v

- Exemplo simples de aplicação de aspiração:
 - considere $A(f(s)) = f(s^*)$ tal que s^* é a melhor solução encontrada até então.
 - Neste caso, aceita-se um movimento tabu somente se ele conduzir a um vizinho melhor que s^* (aspiração por objetivo).
- Soluções melhores a solução s^* corrente, mesmo que geradas por movimentos tabu, não foram visitadas anteriormente,
 - Evidencia de que T pode impedir a ida à soluções geradas ou não geradas.

CRITÉRIOS DE PARADA

- Principais critérios de parada:
 - Pára-se ao atingir um certo número máximo de iterações sem melhora no valor da melhor solução.
 - Pára-se quando o valor da melhor solução chega a um limite inferior conhecido (ou próximo dele).
 - evita a execuções desnecessárias ao encontrar uma solução ótima suficientemente “boa”.

PARÂMETROS DE CONTROLE

- Principais parâmetros de controle :
 - Cardinalidade $|T|$ da lista tabu
 - Função de aspiração A
 - Cardinalidade do conjunto V de soluções vizinhas testadas em cada iteração
 - Btmax: número máximo de iterações sem melhora no valor da melhor solução.

PRESCRIÇÕES ESPECIAIS

- Lista tabu dinâmica:
 - Tamanho variável no intervalo $[t_{\min}, t_{\max}]$
 - Tamanho deve ser mudado periodicamente (p.ex., a cada $2t_{\max}$ iterações)
 - Objetivo: diminuir a probabilidade de ciclagem
- Passagem por regiões planas
 - Aumentar o tamanho da lista enquanto estiver na região plana
 - Retornar ao tamanho original quando houver mudança no valor da função de avaliação

INTENSIFICAÇÃO EM BT

- É comum em métodos de Busca Tabu incluir estratégias de intensificação
 - Objetivo: concentrar a pesquisa em regiões consideradas promissoras.
 - Ex: retornar à uma solução já visitada para explorar sua vizinhança de forma mais efetiva.
 - Ex2: incorporar atributos das melhores soluções já encontradas durante o progresso da pesquisa e estimular componentes dessas soluções a tornarem-se parte da solução corrente.
- Componentes não associadas às boas soluções não recebem técnicas de intensificação
- Ex de procedimento: Path-Relinking (Reconexão por Caminhos)

PATH RELINKING

- Estratégia de intensificação para explorar trajetórias que conectavam soluções elite (Glover, 1996)
- Consiste em gerar e explorar caminhos no espaço de soluções partindo de uma ou mais soluções elite e levando a outras soluções elite.
 - são selecionados movimentos que introduzem atributos das soluções guia na solução corrente.
- Objetivo: incorporar atributos de soluções de boa qualidade, favorecendo a seleção de movimentos que as contenham.

PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



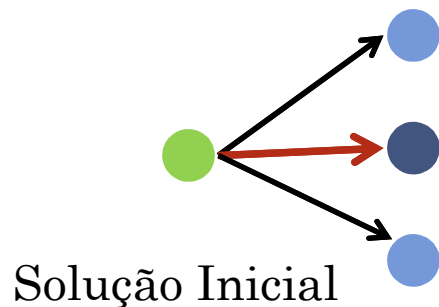
Solução Inicial



Solução Final

PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



Solução Final

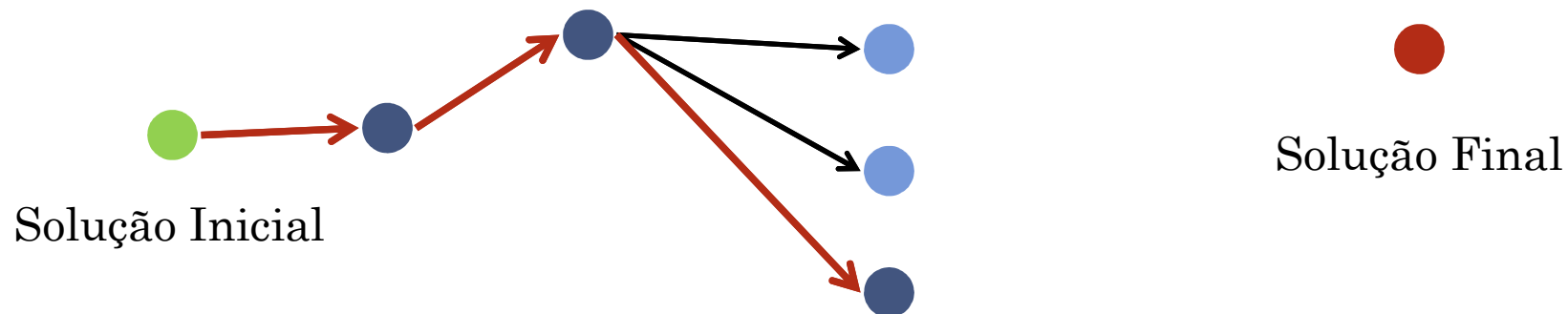
PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



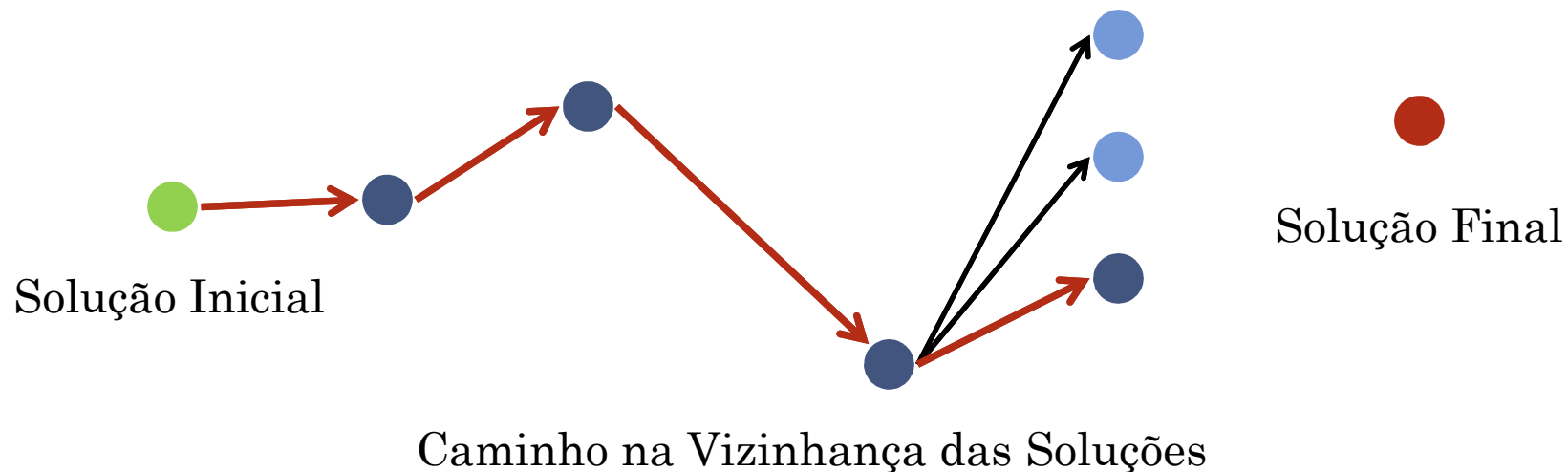
PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



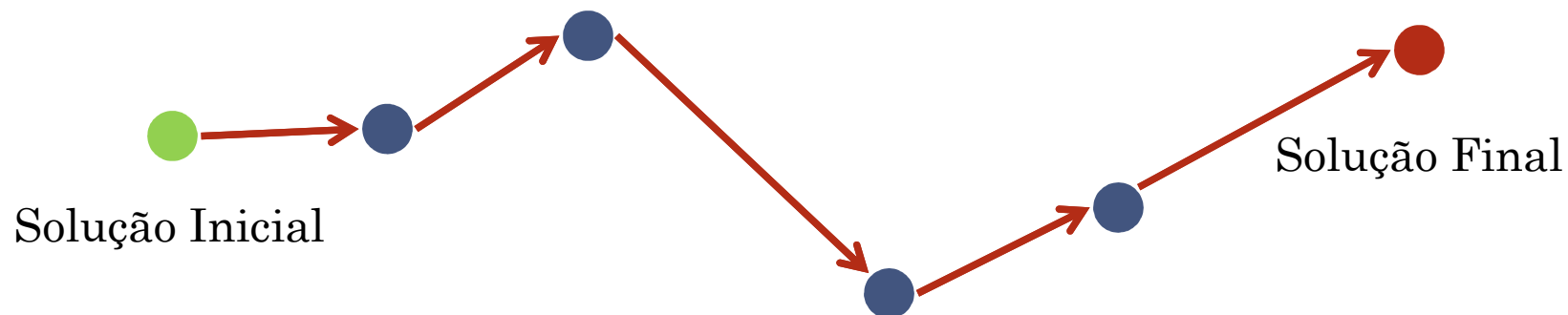
PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



PATH RELINKING: EXEMPLO

- Caminho é gerado selecionando movimentos que introduzam na solução inicial atributos da solução guia.
- A cada passo, todos os movimentos que incorporam atributos da solução guia são avaliados e o melhor movimento é selecionado:



Caminho na Vizinhança das Soluções

DIVERSIFICAÇÃO EM BT

- Técnicas que tipicamente utilizam memória de longo prazo.
- Objetivo: redirecionar a pesquisa para regiões ainda não suficientemente exploradas do espaço de soluções.
- Ao contrário das estratégias de intensificação, buscam gerar soluções com atributos significativamente diferentes daqueles encontrados nas melhores soluções obtidas.
- Utilizada somente em determinadas situações,
 - Ex: dada uma solução s , não existem movimentos m de melhora para ela (indicando que o algoritmo já exauriu a análise naquela região)

APLICAÇÃO BT: PROBLEMA DA MOCHILA

- Seja um conjunto de objetos, uma unidade de cada, com peso e benefício dado abaixo e uma mochila de capacidade $b = 23$

Objeto (j)	1	2	3	4	5
Peso (w_j)	4	5	7	9	6
Benefício (p_j)	2	2	3	4	4

- Representação de uma solução: $s = (s_1, s_2, \dots, s_5)$, onde $s_j \in \{0, 1\}$
- Movimento $m =$ troca no valor de um bit
- Lista tabu = $\{ \langle \text{posição do bit alterado} \rangle \}$
- $|T| = 1;$
- $BT_{\max} = 1;$
- Aspiração por objetivo.

FUNÇÃO DE AVALIAÇÃO

$$f(s) = \sum_{j=1}^n p_j s_j - \alpha \times \max\{0, \sum_{j=1}^n w_j s_j - b\}$$

sendo α uma penalidade, por exemplo, $\alpha = \sum_{j=1}^n p_j = 15$.

PASSO 0

- Seja uma solução inicial qualquer, por exemplo:
 - $s = (01010)$
 - $f(s) = 6$
 - Peso corrente da mochila = 14
 - Lista tabu = $T = \emptyset$;
 - Melhor solução até então: $s^* = (01010)$ e $f(s^*) = 6$
 - $\text{Iter} = 0$;
 - $\text{MelhorIter} = 0$;

PASSO 1:

- Analisar todos os vizinhos de s e calcular a função de avaliação deles

Vizinhos de s	Peso vizinhos de s	Benefício dos vizinhos de s	$f(s')$
$(11010)^t$	18	8	8
$(00010)^t$	9	4	4
$(01110)^t$	21	9	9
$(01000)^t$	5	2	2
$(01011)^t$	20	10	10

PASSO 1

- Melhor vizinho: $s' = (01011)$, com $f(s') = 10$
- $s \leftarrow s'$, isto é, $s = (01011)$
- Lista tabu = $T = \{5\}$ (indicando que o bit da quinta posição não pode ser modificado, a não ser que o critério de aspiração seja satisfeito)
- Melhor solução até então: $s^* = (01011)$ e $f(s^*) = 10$ (pois $f(s') > f(s^*)$)
- $\text{Iter} = 1$;
- $\text{MelhorIter} = 1$;
- Como $(\text{Iter} - \text{MelhorIter}) = (1 - 1) = 0 \leq \text{BTmax} = 1$, então o procedimento de exploração do espaço de soluções deve continuar.

PASSO 2

- Determinar o melhor vizinho de $s = (01011)$:

Vizinhos de s	Peso vizinhos de s	Beneficio dos vizinhos de s	$f(s')$
$(11011)^t$	24	12	-3
$(00011)^t$	15	8	8
$(01111)^t$	27	13	-47
$(01001)^t$	11	6	6
$(01010)^t$	14	6	6

PASSO 2

- Melhor vizinho: $s' = (00011)$, com $f(s') = 8$
- $s \leftarrow s'$, isto é, $s = (00011)$
- Lista tabu = $T = \{2\}$ (observa-se que, como a cardinalidade da lista tabu foi fixada em um, então o movimento proibido anterior sai e entra o novo movimento proibido, isto é, o bit da segunda posição não pode ser modificado, a não ser que o critério de aspiração seja satisfeito)
- Melhor solução até então: $s^* = (01011)$ e $f(s^*) = 10$
- $\text{Iter} = 2; \text{MelhorIter} = 1;$
- Como $(\text{Iter} - \text{MelhorIter}) = (2 - 1) = 1 \leq \text{BTmax} = 1$, então BT continua.

PASSO 3

- Determinar o melhor vizinho de $s = (00011)$

Vizinhos de s	Peso vizinhos de s	Beneficio dos vizinhos de s	$f(s')$
$(10011)^t$	19	10	10
$(01011)^t$	20	10	10
$(00111)^t$	22	11	11
$(00001)^t$	6	4	4
$(00010)^t$	6	4	4

PASSO 3

- Melhor vizinho: $s' = (00111)$, com $f(s') = 11$
- $s \leftarrow s'$, isto é, $s = (00111)$
- Lista tabu = $T = \{3\}$ (indicando que o bit da terceira posição não pode ser modificado, a não ser que o critério de aspiração seja satisfeito)
- Melhor solução até então: $s^* = (00111)$ e $f(s^*) = 11$ (pois $f(s') > f(s^*)$)
- $\text{Iter} = 3$; $\text{MelhorIter} = 3$;
- Como $(\text{Iter} - \text{MelhorIter}) = (3 - 3) = 0 \leq \text{Btmax} = 1$, então o procedimento de exploração do espaço de soluções continua.

PASSO 4

- Determinar o melhor vizinho de $s = (00111)$

Vizinhos de s	Peso vizinhos de s	Benefício dos vizinhos de s	$f(s')$
$(10111)^t$	24	13	-2
$(01111)^t$	25	13	-17
$(00011)^t$	15	8	8
$(00101)^t$	13	7	7
$(00110)^t$	16	7	7

- Vizinho com o melhor valor para a função de avaliação: $s' = (00011)$, com $f(s') = 8$,
- Solução é tabu: bit da terceira posição está em $|T|$
- Critério de aspiração não é satisfeito: $f(s') = 8 \leq f(s^*) = 11$,
- Solução não aceita
 - Considera-se o melhor vizinho não tabu, a saber:

PASSO 4

- Melhor vizinho: $s' = (00101)$, com $f(s') = 7$ (Critério de Desempate)
- $s \leftarrow s'$, isto é, $s = (00101)$
- Lista tabu = $T = \{4\}$ (indicando que o bit da quarta posição não pode ser modificado, a não ser que o critério de aspiração seja satisfeito)
- Melhor solução até então: $s^* = (00111)$ e $f(s^*) = 11$
- $\text{Iter} = 4$; $\text{MelhorIter} = 3$;
- Como $(\text{Iter} - \text{MelhorIter}) = (4 - 3) = 1 \leq \text{BTmax} = 1$, então prossegue a busca.

PASSO 5

- Determinar o melhor vizinho de $s = (00101)$

Vizinhos de s	Peso vizinhos de s	Benefício dos vizinhos de s	$f(s')$
$(10101)^t$	17	9	9
$(01101)^t$	18	9	9
$(00001)^t$	6	4	4
$(00111)^t$	23	11	11
$(00100)^t$	7	3	3

- vizinho com o melhor valor para a função de avaliação: $s' = (00111)$, com $f(s') = 11$.
- Solução é tabu: bit da quarta posição está em $|T|$
- critério de aspiração não é satisfeito: $f(s') = 11 \leq f(s^*) = 11$,
- solução não é aceita.
 - considera-se o melhor vizinho não tabu, a saber (já aplicado um critério de desempate):

PASSO 5

- Melhor vizinho: $s' = (10101)$, com $f(s') = 9$
- $s = (10101)$, com $f(s) = 9$
- Lista tabu = $T = \{1\}$ (indicando que o bit da primeira posição não pode ser modificado, a não ser que o critério de aspiração seja satisfeito)
- Melhor solução até então: $s^* = (00111)$ e $f(s^*) = 11$
- $\text{Iter} = 5$; $\text{MelhorIter} = 3$;
- Como $(\text{Iter} - \text{MelhorIter}) = (5 - 3) = 2 > \text{BTmax} = 1$, então PARE. O método de Busca Tabu retorna, então, $s^* = (00111)$ como solução final, com valor $f(s^*) = 11$.

OUTROS EXEMPLOS

- Coloração de Grafos
- QAP

BUSCA TABU X SIMULATED ANNEALING

REFERÊNCIAS

- [1] “Future Paths for Integer Programming and Linksto Artificial Intelligence," Computers and Operations Research, Vol. 13, No. 5, 533-549, 1986.
- [2] Faigle U., Kern W. -Some Convergence Results for Probabilistic Tabu Search – ORSA Journal on Computing 4, (1992), pp. 32-37.
- [3] Fox B.L. -Integrating and accelerating tabu search, simulated annealing and genetic algorithms - Annals of Operations Research 41, (1993) pp. 47-67.
- [4] Glover F. -Tabu Search, Part I - ORSA Journal on Computing 1, (1989).
- [5] Glover F. -Tabu Search, Part II - ORSA Journal on Computing 2, (1990).



Uma Introdução à Busca Tabu

André Gomes

*Departamento de Ciência da Computação,
Instituto de Matemática e Estatística,
Universidade de São Paulo, SP, Brasil*

Novembro de 2009