

# Busca Tabu

The background is dark with a vertical gradient. It features several decorative elements: a large curved grey shape on the left side; faint circular arrows and dashed lines in the upper and lower right areas, suggesting a search or iterative process.

Profa  
Mariana

# Introdução

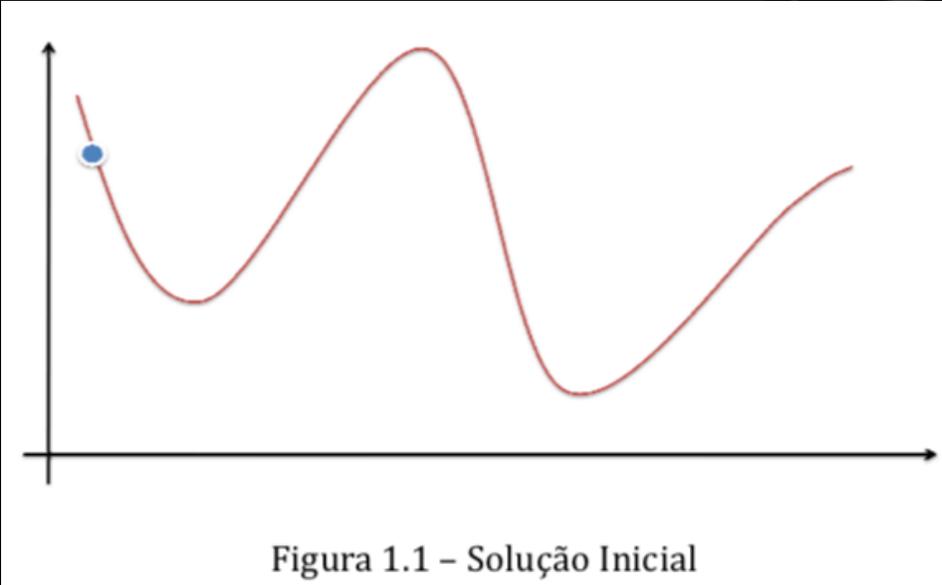
Busca Tabu (*Tabu Search* –TB) é uma técnica computacional de busca, conhecida por superar (em geral) o problema da convergência local em problemas de otimização.

O método evita retornar a um ótimo local visitado previamente, de forma a superar a otimalidade local e atingir um resultado ótimo ou próximo ao ótimo global.

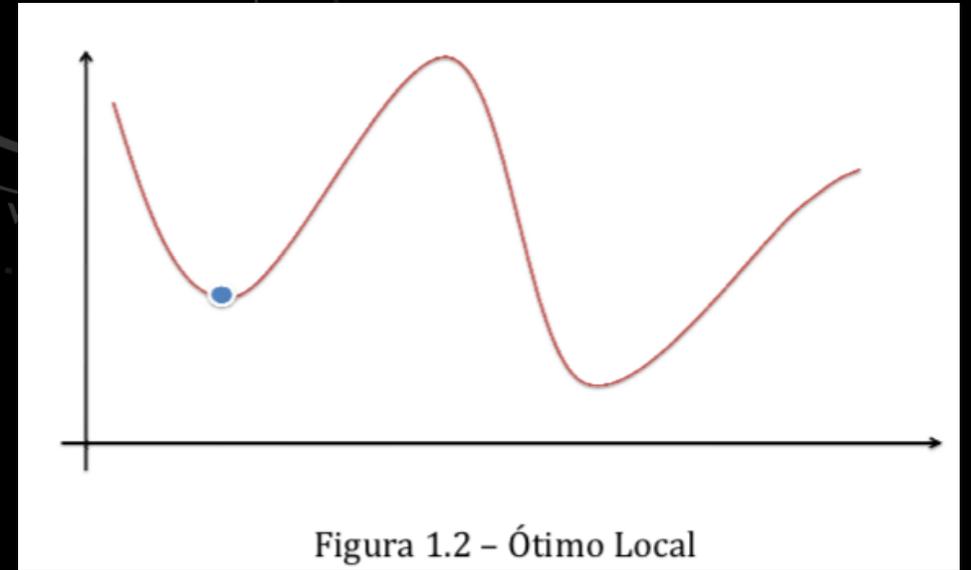
TABU: proibição



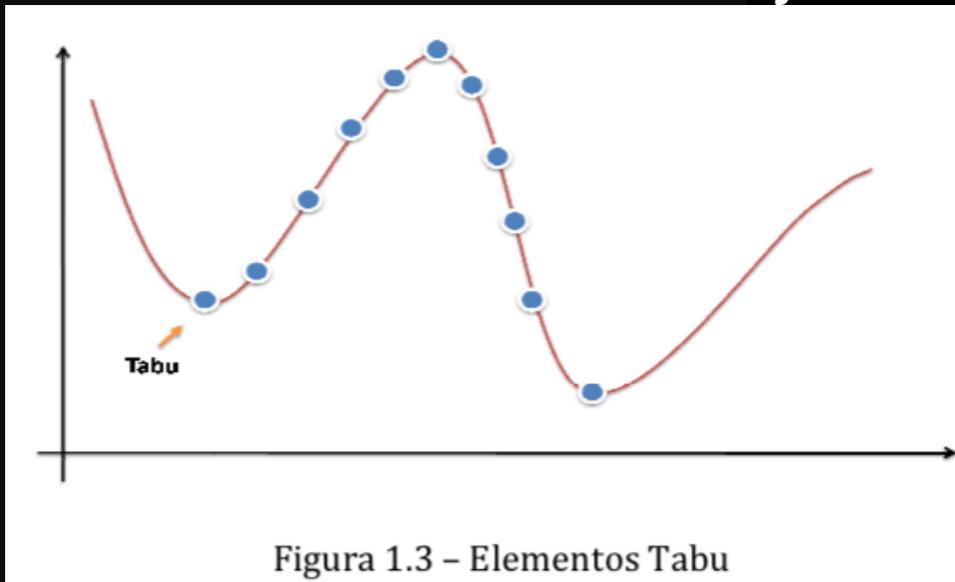
Partindo de uma solução inicial



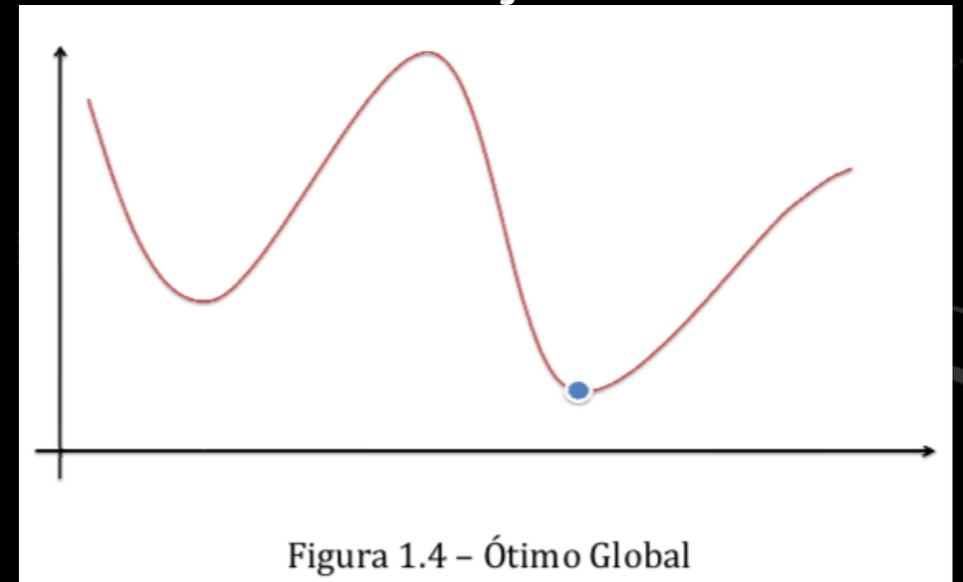
Busca-se na vizinhança a melhor sol.



Geralmente não se aceita sol. já visitadas



Para se alcançar o ótimo



# Conceito

Partindo de uma solução inicial, a busca move-se, a cada iteração, para a melhor solução na vizinhança, não aceitando movimentos que levem a soluções já visitadas por permanecerem armazenadas em uma lista tabu.

A lista permanece na memória guardando soluções já visitadas (tabu) durante um determinado espaço de tempo ou certo número de iterações (prazo tabu ou tempo tabu).

Como resultado final é esperado que se encontre um ótimo global ou a solução mais próxima deste.

# Elementos da técnica

**TABU:** proibição



**RESTRICÇÕES TABU:**  
inibe certos  
movimentos



**CRITÉRIOS DE ASPIRAÇÃO:**  
decidem quando os  
movimentos  
classificados  
como tabu  
podem ser  
executados



# Elementos da técnica

**TABU:** um movimento que já foi realizado não é mais permitido.

**RESTRICÇÕES TABU:** são controladas por uma lista que memoriza os últimos movimentos executados.

**CRITÉRIOS DE ASPIRAÇÃO:** são introduzidos para determinar quando uma restrição tabu pode ser quebrada, ou seja, a restrição é ignorada e o movimento, mesmo classificado como proibido, é executado.

Um critério de aspiração bastante utilizado é o seguinte: ignorar a restrição tabu se a solução formada por um determinado movimento proibido for melhor que a melhor solução encontrada até o momento.

# Elementos da técnica

**ESTRATÉGIAS DE OSCILAÇÃO:** admite oscilação entre soluções factíveis e infactíveis durante o processo de busca, pois considerar soluções infactíveis é um meio de tentar escapar de ótimos locais.

O fato de permitir temporariamente soluções infactíveis tende a direcionar a busca mais rapidamente para soluções factíveis cada vez melhores.

# Elementos da técnica

**LISTA TABU:** a lista de memória (lista tabu) armazena cada movimento executado. O movimento deverá permanecer nesta lista durante um tempo tabu (3 iterações, por exemplo).

Enquanto um determinado movimento permanecer na lista tabu, será considerado proibido e só poderá ser executado, se este resultar num valor de isolamento melhor que o melhor valor obtido em todas as iterações anteriores (critério de aspiração). No caso de maximizar uma função procura-se sempre o maior valor de troca possível.

# Algoritmo

1. Seja  $s_0$  solução inicial;
  2.  $s^* \leftarrow s_0$ ;                    {Melhor solução obtida até então}
  3. Iter  $\leftarrow 0$ ;                    {Contador do número de iterações}
  4. 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** (Iter – MelhorIter  $\leq$  BTmax) **faça**
  9.     Iter  $\leftarrow$  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.     MelhorIter  $\leftarrow$  Iter ;
  16.    **fim-se**;
  17.    Atualize a função de aspiração A;
  18. **fim-enquanto**;
  19. **Retorne**  $s^*$ ;
- fim** BT;



## *Busca Tabu aplicada ao Problema da Mochila*

# Exemplo

[http://www.lac.inpe.br/~lorena/cap/Aula\\_C02.pdf](http://www.lac.inpe.br/~lorena/cap/Aula_C02.pdf)

Seja 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 =  $\{ \langle \text{posição do bit alterado} \rangle \}$

$|T| = 1;$

BTmax = 1;

Aspiração por objetivo.



## Busca Tabu aplicada ao Problema da Mochila

- Função de Avaliação:

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

- **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  $f(s^*) = 6$

*Iter* = 0; *MelhorIter* = 0;



## *Passo 1: Devemos, agora, analisar todos os vizinhos de $s$ e calcular a função de avaliação deles*

Vizinhos de $s$	Peso dos 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

Melhor vizinho:  $s' = (01011)$ , com  $f(s') = 10$

Como  $s'$  é o melhor vizinho de  $s$ , então  $s \leftarrow s'$ , isto é, a nova solução corrente passa a ser:  $s = (01011)$

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  $f(s') > f(s^*)$ )

$Iter = 1$ ;  $MelhorIter = 1$ ;

Como  $(Iter - MelhorIter) = (1 - 1) = 0 \leq BTmax = 1$ , então o procedimento de exploração do espaço de soluções deve continuar.



## *Passo 2: Determinemos, agora, o melhor vizinho de $s = (01011)$ :*

Vizinhos de $s$	Peso dos vizinhos de $s$	Benefício 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

Melhor vizinho:  $s' = (00011)$ , com  $f(s') = 8$

Como  $s'$  é o melhor vizinho de  $s$ , então  $s \leftarrow s'$  (mesmo sendo  $f(s')$  pior que  $f(s)$ ), isto é, a nova solução corrente passa a ser:  $s = (00011)$

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  $f(s^*) = 10$

$Iter = 2$ ;  $MelhorIter = 1$ ;

Como  $(Iter - MelhorIter) = (2 - 1) = 1 \leq BTmax = 1$ , então o BT continua.



### *Passo 3: Determinemos, agora, o melhor vizinho de $s = (00011)$*

Vizinhos de $s$	Peso dos vizinhos de $s$	Benefício 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
$(00010)^t$	9	4	4

Melhor vizinho:  $s' = (00111)$ , com  $f(s') = 11$

Como  $s'$  é o melhor vizinho de  $s$ , então  $s \leftarrow s'$ , isto é, a nova solução corrente passa a ser:  $s = (00111)$

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  $f(s') > f(s^*)$ )

$Iter = 3$ ;  $MelhorIter = 3$ ;

Como  $(Iter - MelhorIter) = (3 - 3) = 0 \leq BTmax = 1$ , então o procedimento de exploração do espaço de soluções continua.



## *Passo 4: Determinemos, agora, o melhor vizinho de $s = (00111)$*

Vizinhos de $s$	Peso dos 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

Observe que o vizinho com o **melhor** valor para a função de avaliação é  $s' = (00011)$ , com  $f(s') = 8$ , mas esta **solução é tabu**, uma vez que o bit da terceira posição está na lista tabu. Como o critério de aspiração desta solução **não é** satisfeito, pois  $f(s') = 8 \leq f(s^*) = 11$ , esta solução **não é aceita**. Desta forma, considera-se o **melhor vizinho não tabu**, a saber:

Melhor vizinho:  $s' = (00101)$ , com  $f(s') = 7$  (Critério de Desempate)

Como  $s'$  é o melhor vizinho de  $s$  (mesmo sendo de piora), então  $s \leftarrow s'$ , isto é, a nova solução corrente passa a ser:  $s = (00101)$

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  $f(s^*) = 11$

$Iter = 4$ ;  $MelhorIter = 3$ ;

Como  $(Iter - MelhorIter) = (4 - 3) = 1 \leq BTmax = 1$ , então prossegue a busca.



## *Passo 5: Determinemos, agora, o melhor vizinho de $s = (00101)$*

Vizinhos de $s$	Peso dos 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

Observe que o vizinho com o melhor valor para a função de avaliação é  $s' = (00111)$ , com  $f(s') = 11$ . Entretanto, esta **solução é tabu**, uma vez que o bit da quarta posição está na lista tabu. Como o critério de aspiração desta solução **não é** satisfeito, pois  $f(s') = 11 \leq f(s^*) = 11$ , esta solução não é aceita. Desta forma, considera-se o **melhor vizinho não tabu**, a saber (já aplicado um critério de desempate):

Melhor vizinho:  $s' = (10101)$ , com  $f(s') = 9$

Desta forma, a nova solução corrente passa a ser:  $s = (10101)$ , 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  $f(s^*) = 11$

$Iter = 5$ ;  $MelhorIter = 3$ ;

Como  $(Iter - MelhorIter) = (5 - 3) = 2 > BTmax = 1$ , então **PARE**. O método de Busca Tabu retorna, então,  $s^* = (00111)$  como solução final, com valor  $f(s^*) = 11$ .

# Referências

SIQUEIRA, P. H. Metaheurísticas e Aplicações. Parte III – Outras Metaheurísticas. PPGMNE/UFPR. Notas de Aula.

GOMES, A. Uma introdução à Busca Tabu. USP. Disponível em:  
<https://www.ime.usp.br/~gold/cursos/2009/mac5758/AndreBuscaTabu.pdf>

Problema da mochila

[http://www.lac.inpe.br/~lorena/cap/Aula\\_C02.pdf](http://www.lac.inpe.br/~lorena/cap/Aula_C02.pdf)

[https://www.ime.usp.br/~gold/cursos/2009/mac5758/2611/Busca\\_Tabu.pdf](https://www.ime.usp.br/~gold/cursos/2009/mac5758/2611/Busca_Tabu.pdf)

Job-shop:

<https://www.ime.unicamp.br/~chico/mt852/tabu.pdf>

Mínima arborescência e AG+BT:

[https://inf.ufes.br/~crangel/busca\\_tabu-michael.doc](https://inf.ufes.br/~crangel/busca_tabu-michael.doc)