



**CEAR**  
CENTRO DE ENERGIAS  
ALTERNATIVAS E RENOVÁVEIS  
UNIVERSIDADE FEDERAL DA PARAÍBA



# Busca Tabu para Otimização

Introdução ao método otimização combinatória  
Resolução do problema da mochila

**Orientador:** Prof. Dr. Juan Moises Mauricio Villanueva

**Aluno:** Miguel Marques Ferreira

06 de maio de 2024.

1. Contextualização;
2. Lista Tabu de Soluções;
3. Critério de Aspiração;
4. Fluxo do Algoritmo Busca Tabu;
5. Apresentação do Problema da Mochila;
6. Implementação da Busca Tabu em Python;
7. Aplicações;
8. Referências.



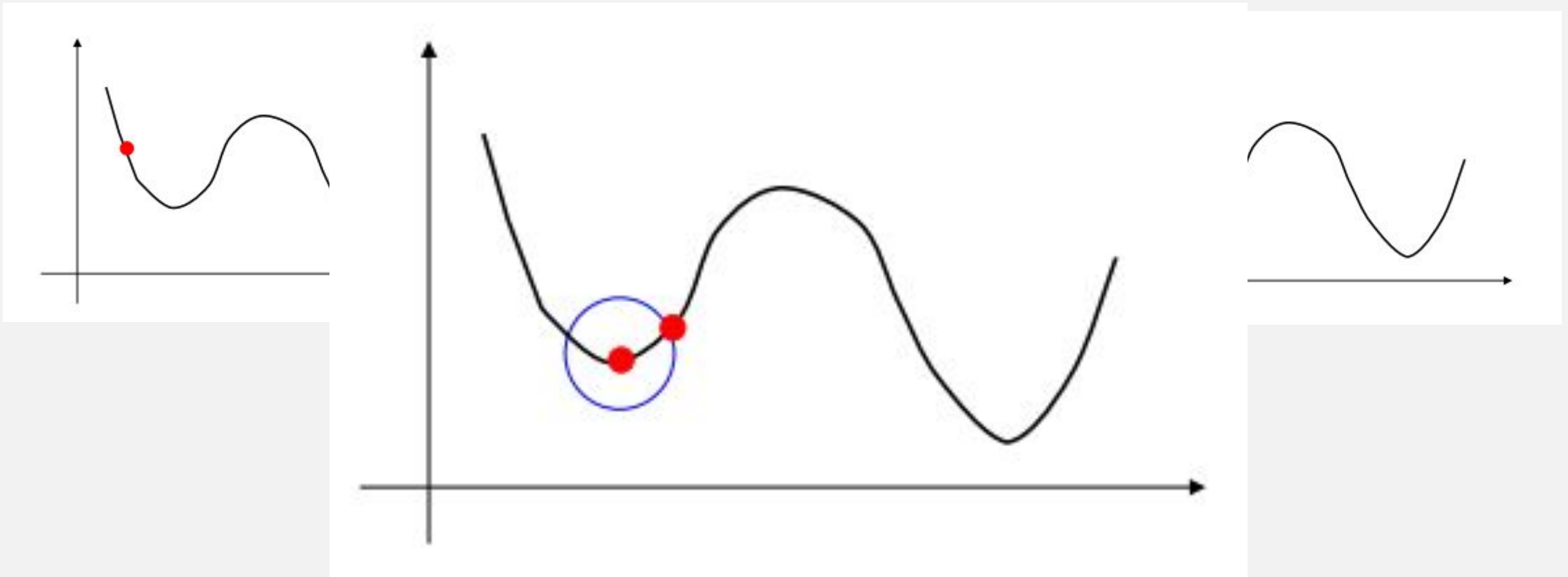
# 1. Contextualização

- O método de Busca Tabu foi proposto por Fred Glover em 1986.
  - **"Future Paths for Integer Programming and Links to Artificial Intelligence,"**  
Computers and Operations Research, Vol. 13, No. 5, 533-549, 1986.
- Método de busca local
  - Explorar o espaço de soluções movendo-se de uma solução para outra que seja seu melhor vizinho;
  - Estrutura de memória para armazenar as soluções geradas (ou características dessas);
  - Essas características possibilitam Busca Tabu escapar de ótimos locais.
- Afinal, por que busca tabu? O que seria um tabu?



# 1. Contextualização

1ª Ideia: Utilizar Heurística de Descida para encontrar o melhor mínimo ou máximo da função objetivo.

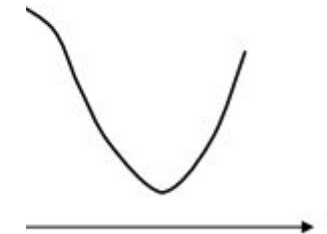
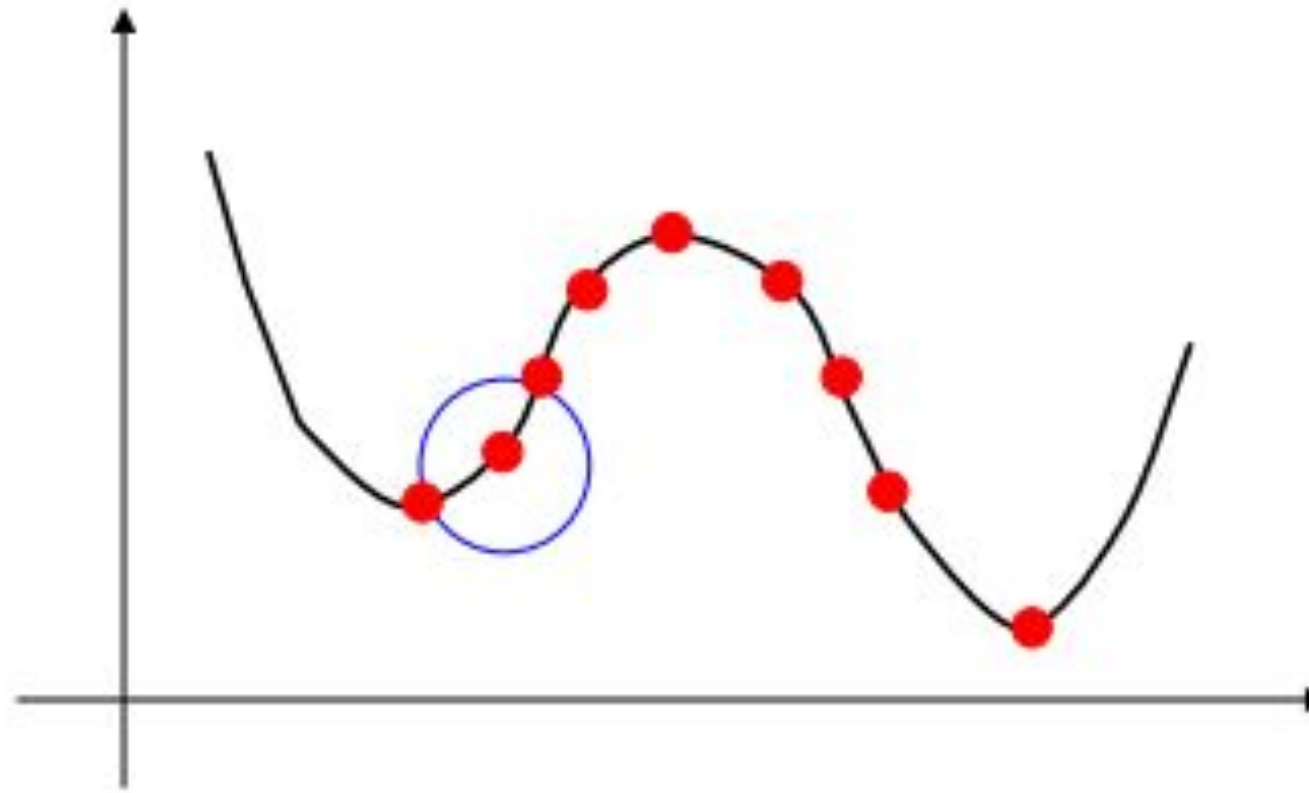
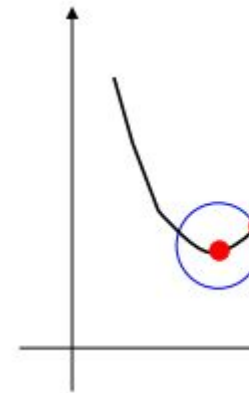




# 1. Contextualização

2ª Ideia: Mover para o Melhor Vizinho

O melhor



ma Lista Tabu



## 2. Lista Tabu de Soluções

- Assim como é computacionalmente inviável testar todas as soluções disponíveis no espaço de busca, quando este é consideravelmente grande, armazenar todas as soluções geradas também se torna inviável!
- O que pode ser feito?
  - Armazenar apenas as últimas  $|T|$  (lista tabu) soluções geradas;
  - **Observação:** Uma lista com as  $|T|$  últimas soluções evita ciclos de até  $|T|$  iterações;
  - **Problema:** Pode ser inviável armazenar  $|T|$  soluções e testar se uma solução está ou não na Lista Tabu;
  - **Ideia:** Criar uma Lista Tabu de movimentos reversos.
- **Problema:** Uma Lista Tabu de movimentos pode ser muito restritiva (impede o retorno a uma solução já gerada anteriormente e também a outras soluções ainda não geradas)



### 3. Critério de Aspiração

- O critério de aspiração surgem como uma 4ª ideia;
- Quanto se retira a situação tabu de um movimento sob determinadas circunstâncias:
  - **Exemplo:** aceitar um movimento, mesmo que tabu, se ele melhorar o valor da função objetivo global (Critério de aspiração por objetivo);
- Aspiração por padrão
  - Realiza-se o movimento tabu mais antigo se todos os possíveis movimentos forem tabus;



## 4. Fluxo do Algoritmo Busca Tabu

- A Busca Tabu começa a partir de uma solução inicial qualquer ( $s_0$ );
- A cada iteração, é explorado um subconjunto  $V$  da vizinhança  $N(s)$  da solução corrente  $s$ .
- O membro  $s'$  de  $V$  com melhor valor nessa região segundo a função objetiva torna-se a nova solução corrente mesmo que  $s'$  seja pior que  $s$ , isto é, que  $f(s') > f(s)$  para um problema de minimização.
- O critério de escolha do melhor vizinho é utilizado para escapar de um ótimo local. Porém, esta estratégia pode fazer com que o algoritmo cicle, isto é, que retorne a uma solução já gerada anteriormente.
- Para evitar que isto ocorra, existe uma lista tabu  $T$ , a qual é uma lista de movimentos proibidos.
- A lista tabu clássica contém os movimentos reversos aos últimos  $|T|$  movimentos realizados (onde  $|T|$  é um parâmetro do método) e funciona como uma fila de tamanho fixo, isto é, quando um novo movimento é adicionado à lista, o mais antigo sai.





## 4. Fluxo do Algoritmo Busca Tabu

- Assim, na exploração do subconjunto  $V$  da vizinhança  $N(s)$  da solução corrente  $s$ , ficam excluídos da busca os vizinhos  $s'$  que são obtidos de  $s$  por movimentos  $m$  que constam na lista tabu.
- A lista tabu reduz o risco de ciclagem, uma vez que:
  - garante o não retorno, por  $|T|$  iterações, a uma solução já visitada anteriormente;
  - proibir movimentos para soluções que ainda não foram visitadas.
    - função de aspiração, que é um mecanismo que retira, sob certas circunstâncias, o status tabu de um movimento.
- 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  $v$ .



## 4. Fluxo do Algoritmo Busca Tabu

- Duas regras são normalmente utilizadas para interromper o procedimento:
  - Quando é atingido um certo número máximo de iterações sem melhora no valor da melhor solução, para-se a execução do algoritmo.
  - Quando o valor da melhor solução chega a um limite inferior conhecido (ou próximo dele). Esse segundo critério evita a execução desnecessária do algoritmo quando uma solução ótima é encontrada ou quando uma solução é julgada suficientemente boa (erro aceitável atingido).
- Em resumo, os parâmetros principais de controle do método de Busca Tabu são:
  - a cardinalidade  $|T|$  da lista tabu;
  - a função de aspiração  $A$ ;
  - a cardinalidade do conjunto  $V$  de soluções vizinhas testadas em cada iteração;
  - o número máximo de iterações sem melhora no valor da melhor solução.



# 5. Apresentação do Problema da Mochila

- É um problema de otimização combinatória (NP-Completo) estudado por mais de um século (desde ~1897).
- Resolvido por variados algoritmos.
- A Mochila Valiosa ou o problema da mochila consiste na seguinte situação:
  - Um ladrão entra em uma loja carregando mochila (bolsa) que **pode transportar 35 kg de peso**. A loja possui 10 itens, cada um com um peso e preço específicos.
  - O dilema do ladrão é fazer uma seleção de itens que **maximize** o valor (ou seja, preço total) sem exceder o peso da mochila. Temos que ajudar o ladrão a fazer a seleção.



# 5. Apresentação do Problema da Mochila

- Itens da loja com seus respectivos valores e pesos.

Item	1	2	3	4	5	6	7	8	9	10
Valor	1200	200	300	1000	1500	800	2000	40	500	3000
Peso	10	8	2	4	15	5	3	1	12	9



# 5. Apresentação do Problema da Mochila

- Definição de indivíduo (cromossomo, semelhante à implementação do AG).
- Definição dos itens como os genes dos cromossomos.
- Código
  - 0 = não leva item, 1 = leva item.

Item	1	2	3	4	5	6	7	8	9	10
Cromossomo	1	0	1	0	0	0	1	0	1	0





# 5. Apresentação do Problema da Mochila


- Função-objetivo (Fitness) – Maximizar o valor total dos itens.
  - Restrição: O peso total dos itens  $\leq 35$  kg.
- A equação matemática que expressa esta função é:

$$fitness = \sum_{i=1}^n c_i v_i; \text{ if } \sum_{i=1}^n c_i w_i \leq kw$$

- Onde:
  - $n$  = tamanho do cromossomo
  - $c_i$  =  $i$  ésimo gene
  - $v_i$  =  $i$  ésimo valor
  - $w_i$  =  $i$  ésimo peso
  - $kw$  = peso da mochila



# 6. Implementação da Busca Tabu em Python



Name	Last commit message	Last commit da...
..		
tabu_search_trials	fim do algoritmo busca tabu com o problema da mochila	last week
core.py	fim do algoritmo busca tabu com o problema da mochila	last week
fitness.py	fim do algoritmo busca tabu com o problema da mochila	last week
mochila.csv	busca tabu implementacao inicial	last week
plots.py	fim do algoritmo busca tabu com o problema da mochila	last week
problema_da_mochila.py	busca tabu implementacao inicial	last week

[https://github.com/Miguel-mmf/smart-automation/tree/main/ava03/pt2/busca\\_tabu](https://github.com/Miguel-mmf/smart-automation/tree/main/ava03/pt2/busca_tabu)



# 7. Aplicações

- Aplicações:
  - Gravação de arquivos desperdiçando o mínimo espaço em cada mídia;
  - Corte e empacotamento;
  - Carregamento de veículos;
  - Alocação de recursos em geral;
  - Naves espaciais.



## 8. Referências

- GOLDBARG, Elizabeth. **Otimização Combinatória e Meta-heurísticas - Algoritmos e Aplicações**. [Digite o Local da Editora]: Grupo GEN, 2015. E-book. ISBN 9788595154667. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595154667/>. Acesso em: 04 mai. 2024.
- Material baseado nas notas de aula do Prof. Dr. Marcone Jamilson Freitas Souza (UFOP);
- <http://www.decom.ufop.br/prof/marcone/>

