

# Tabu Search

Glover, Fred, and Manuel Laguna. "Tabu Search\*." *Handbook of Combinatorial Optimization*. Springer New York, 2013. 3261-3362.



INSTITUTO DE  
INFORMÁTICA  
UFG



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS



“

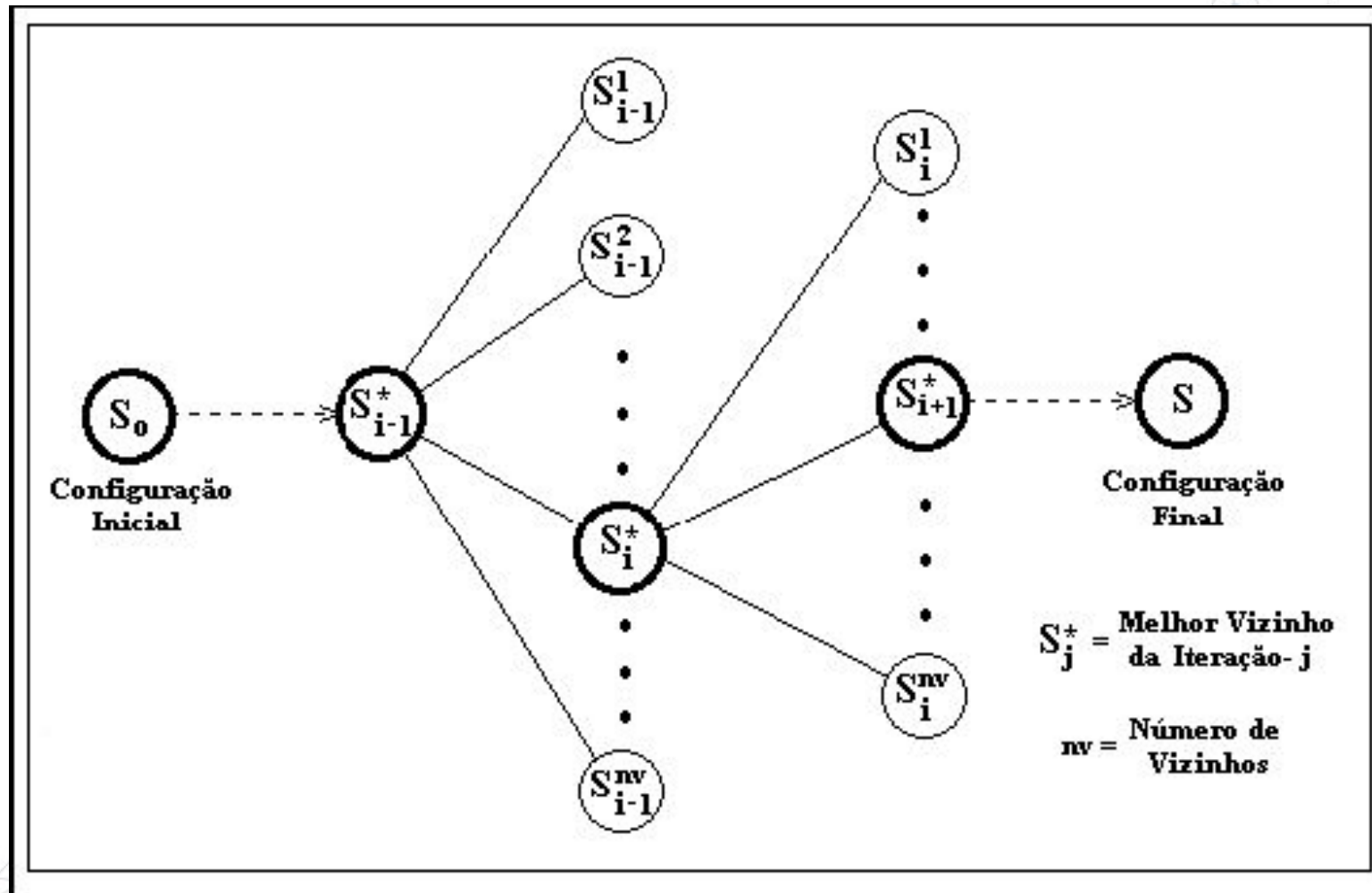
*Guia um procedimento de busca  
heurística local para explorar o espaço  
da solução além da otimização local.*

Um dos principais componentes da Tabu Search é  
o uso da memória adaptativa, o que cria um  
comportamento de pesquisa mais flexível.

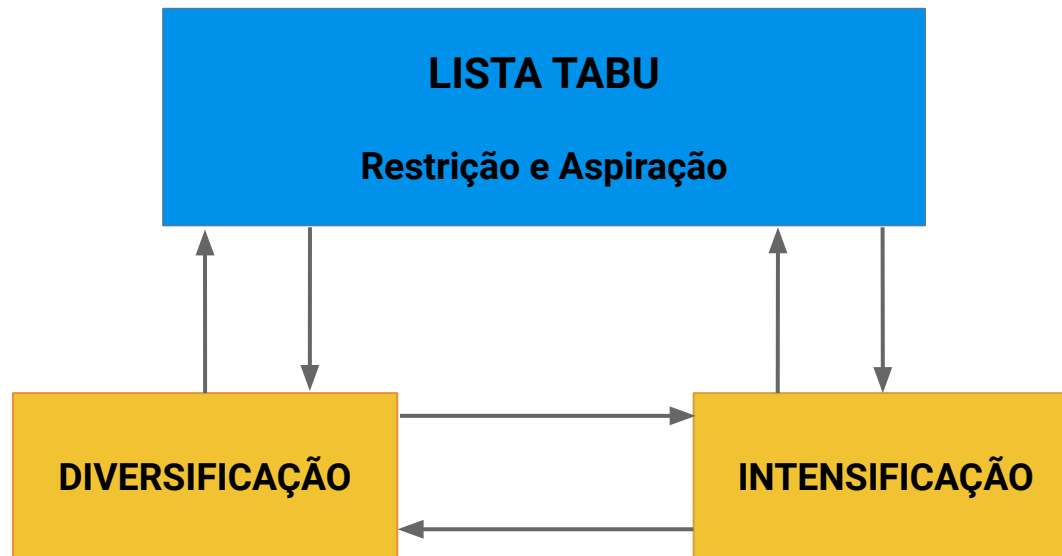
## Intuições

- ◎ A Busca Tabu é um procedimento adaptativo de busca local que explora continuamente a vizinhança, evitando retornar a um ótimo local já visitado;
- ◎ Utiliza uma **Lista Tabu** que é uma estrutura de memória que “guarda” o histórico da evolução do processo de busca;
- ◎ Partindo de uma solução inicial “move-se”, a cada iteração, para a melhor solução na vizinhança.

## Ilustração da busca



# Principais componentes



## Lista tabu

- ⊙ Representa regiões de busca já visitadas;
- ⊙ Armazena soluções, atributos de soluções ou movimentos;
- ⊙ Pode ter tamanho fixo ou dinâmico em cada iteração;
- ⊙ É atualizada a cada iteração;

# Tipos de memorização

## ◎ Short-term

- lista de soluções recentemente consideradas. Se uma solução potencial aparecer na lista de tabu, ela não pode ser revisada até a menos que seja “aspirada”.

## ◎ Intermediate-term

- Regras de intensificação destinadas a favorecer a busca em áreas promissoras do espaço de busca.

## ◎ Long-term

- Regras de diversificação que impulsionam a busca para novas regiões.

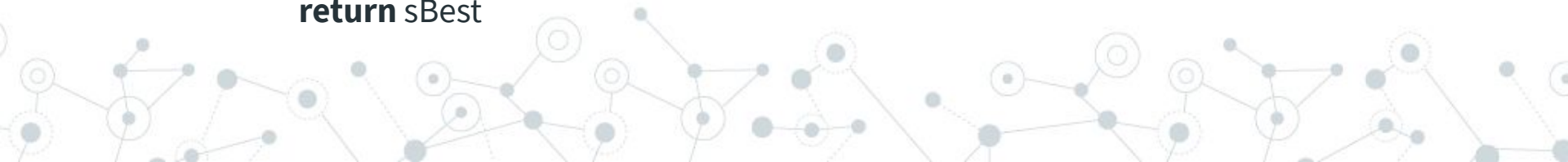
## Critério de aspiração

- ◎ Condição em que um “movimento” torna-se permitido mesmo estando na lista tabu:
  - Objetivo;
  - Direção da busca;
  - Influência.



## Algoritmo simplificado da busca tabu

```
sBest ← s0, tabuList ← [], tabuList.push(s0)
while (not stoppingCondition())
    sNeighborhood ← getNeighbors(bestCandidate)
    bestCandidate ← sNeighborhood.firstElement
    for (sCandidate in sNeighborhood)
        if ( (not tabuList.contains(sCandidate)) and
            (fitness(sCandidate) > fitness(bestCandidate)) )
            bestCandidate ← sCandidate
    if (fitness(bestCandidate) > fitness(sBest))
        sBest ← bestCandidate
    tabuList.push(bestCandidate)
    if (tabuList.size > maxTabuSize)
        tabuList.removeFirst()
return sBest
```





# Obrigado!

## Perguntas?

**altinoneto@inf.ufg.br**

**I4Soft**

Intelligence for Software

