



HEURÍSTICAS 2

Marcus Ritt

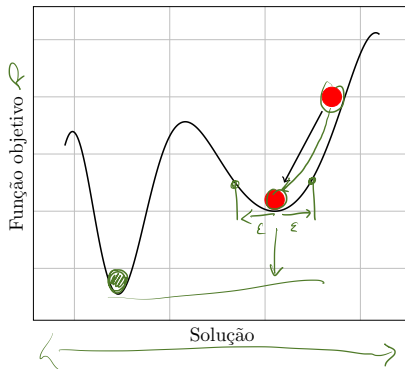
INF 05010 – Otimização combinatória — <2020-06-17
qua>

1. Heurísticas de busca local

Modificação.

- **Desvantagem óbvia:** podemos parar em mínimos locais.
- Exceto: Função objetivo convexa (minimização) ou concava (maximização).
- Técnicas para superar isso baseadas em busca local Busca global
 - Multi-Start, Busca Tabu, Algoritmos Metropolis e Simulated Annealing, Variable neighborhood search

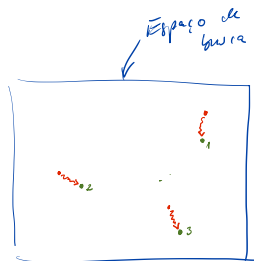
Busca global
Meta-heurísticas



Multiplos inícios

Amostragem

- Gera uma solução aleatória inicial e aplique busca local nesta solução.
- Repita este procedimento por n vezes.
tempo fixo
- Retorne a melhor solução encontrada.



```
Multi_Start( $n$ ) :=  
  { mantém a melhor solução  $s^*$  }  
  repete  $n$  vezes Tempo fixo  
    gera solução aleatória  $s$    
     $s := \text{BuscaLocal}(s)$    
  end repeat  
  return  $s^*$ 
```

- *Problema*: soluções aleatoriamente geradas em geral possuem baixa qualidade.
- **Greedy randomized adaptive search procedure**: método multi-start, em cada iteração
 - Gera soluções com um procedimento guloso-randomizado.
 - Otimiza as soluções geradas com busca local.

GRASP(α , ...) =

```

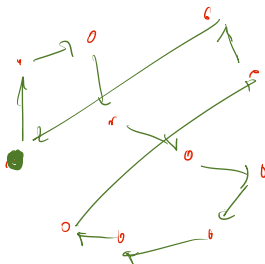
{ a busca mantém a melhor solução encontrada  $s^*$  }
do
   $s :=$  Guloso - Randomizado( $\alpha$ )
   $s :=$  BuscaLocal( $s$ )
  atualiza  $s^*$  caso  $f(s) < f(s^*)$ 
until critério de parada satisfeito
return  $s^*$ 

```

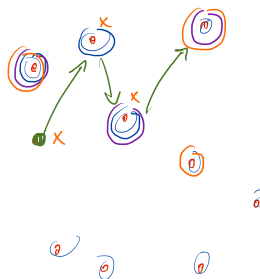
Handwritten annotations:

- A blue arrow points from the word "aleatório" to the "Randomizado(α)" part of the code.
- A red circle highlights the "do" loop.
- A red circle highlights the "atualiza s^* " line.
- Red scribbles and arrows are present on the right side of the code block.

VMP/NN



Alg. guloso randomizado
=> Entre os candidatos seleciona um dos
melhores aleatoriamente (e não o melhor)

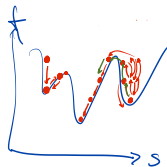


○ CL: candidate list

⊙ RCL: restricted candidate list

vértices mais próximos
 $k \rightarrow$ parâmetro

Passa para qualquer vizinho:
caminhada aleatória.



Minimização.

- Aceita soluções piores
- Critério de Metropolis; aceita com probabilidade

$$\min\{1, e^{-\Delta(s,s')/T}\}$$

com $\Delta(s, s') = f(s') - f(s)$.

$s \xrightarrow{\quad} s' \in N(s)$

T: parâmetro (temperatura) > 0 .

$$\Delta \leq 0 : f(s') \leq f(s)$$

$\Rightarrow$ melhora
 $\Rightarrow$ prob. aceitar $s' \in 1$

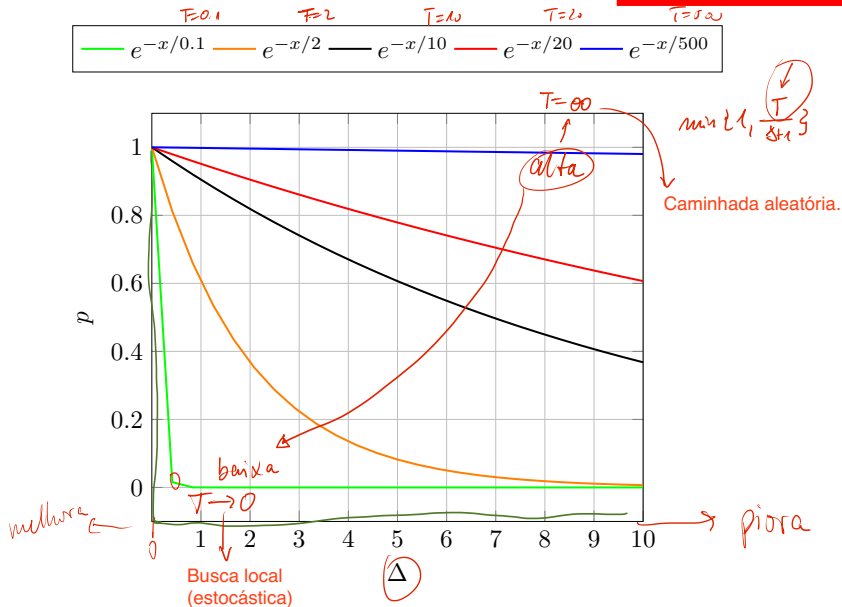
$$\Delta > 0 \quad e^{-\Delta/T} < 1$$

$$T=1 ; D=1 : \overline{e^{-1} \approx 1/3} \quad e^{-2} \approx 1/9$$

HEURÍSTICAS DE BUSCA LOCAL

Distribuição de Boltzmann

HEURÍSTICAS 2



$\text{Metropolis}(s, T) = \{ \text{manter melh. sol. } s^* \}$

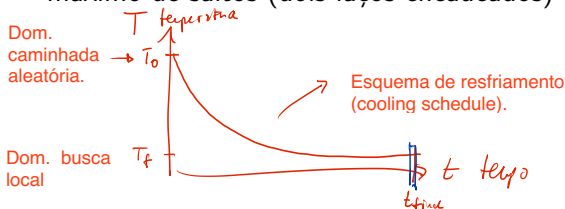
gd. inicial
Temp.

```

do
  seleciona  $s' \in \mathcal{N}(s)$  aleatoriamente
  seja  $\Delta := f(s') - f(s)$ 
  if  $\Delta \leq 0$  then
    atualiza  $s := s'$ 
  else
    atualiza  $s := s'$  com probabilidade  $e^{-\Delta/T}$ 
  end if
until critério de parada satisfeito
return  $s^*$  (#iter., tempo)

```

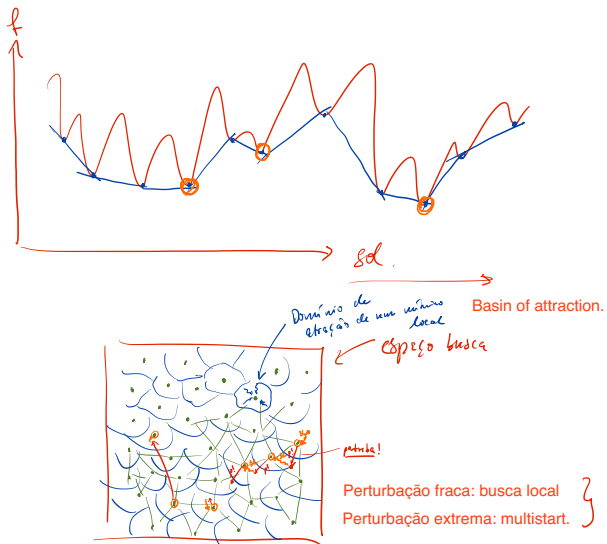
- Simula um processo de recozimento.
- Recozimento: processo da física que aquece um material a uma temperatura bem alta e resfria aos poucos, dando tempo para o material alcançar seu estado de equilíbrio
- Recozimento simulado: parte de uma alta temperatura e baixa gradualmente. Para cada temperatura, permite um número máximo de saltos (dois laços encadeados)



SimulatedAnnealing(s , T_0 , r , I) := temperature length
 $T := T_0$ $T \leq T_f$
 repeat até sistema “esfriado”
 repeat I vezes (manter a temp. constante por I iterações)
 seleciona $s' \in \mathcal{N}(s)$ aleatoriamente
 seja $\Delta := f(s') - f(s)$
 if $\Delta \leq 0$ then
 $s := s'$
 else
 $s := s'$ com probabilidade $e^{-\Delta/T}$
 end if
 end repeat
 $T := rT$ $r \in (0, 1)$ $T_i = T_0 \cdot r^i$
end repeat
return s

Handwritten notes:
 - vmp (above I)
 - $Autopls$ (vertical line next to the inner loop)
 - $2-opt$ (with arrow pointing to the acceptance condition)
 - $r \in (0, 1)$ (with arrow pointing to the cooling factor)
 - $T_i = T_0 \cdot r^i$ (with arrow pointing to the cooling schedule)

Geometric cooling schedule
 Esquema de resfriamento geométrica.



Busca local iterada(s) :=
repete (n vezes, tempo)
 aplica busca local
 aplica perturbação



k passos aleatórios na vizinhança

$k=0 \rightarrow \text{BL}$

$k \rightarrow \infty \rightarrow \text{Multi-start}$

Iterated greedy algorithm (IGA)

Remove l elementos aleatórios
Insere os elementos via alg. guloso randomizado
novamente.

Busca gulosa iterada: busca local iterada com um algoritmo guloso randomizado na perturbação.

```
BuscaTabu()=  
  Inicialização:  
     $s := S_0$ ;  $f^* := f(s_0)$ ;  $s^* := s_0$  ;  $T := \emptyset$   
  while critério de parada não satisfeito  
     $s' := \text{seleciona } s' \in \tilde{\mathcal{N}}(s) \text{ com } \min f(s)$   
    if  $f(s) < f^*$  then  
       $f^* := f(s)$ ;  $s^* := s$   
    end if  
    insira movimento em T (a lista tabu)  
  end while
```

- Introduz uma memória de curta duração ("lista tabu")
- Ideia: "lembrar" soluções visitadas recentemente e evitar ("tabu")
-


```
BuscaTabu() =  
  Inicialização:  
     $s := S_0$ ;  $f^* := f(s_0)$ ;  $s^* := s_0$  ;  $T := \emptyset$   
  while critério de parada não satisfeito  
     $s' := \text{seleciona } s' \in \tilde{\mathcal{N}}(s) \text{ com } \min f(s)$   
    if  $f(s) < f^*$  then  
       $f^* := f(s)$ ;  $s^* := s$   
    end if  
    insira movimento em T (a lista tabu)  
  end while
```

Variable neighborhood search (VNS)

$\{ N_{i,1} : N_1$
 $\{ N_{i,2} : N_2$
 $\vdots$
 $N_{i,k} : N_k$

$\nwarrow \nearrow$ min. loc. diferentes


```
Movimento( $s, s', k$ ) :=  
  if  $f(s') < f(s)$  then  
     $s := s'$   
     $k := 1$  ← volta h7. 1  
  else  
     $k := k + 1$   
  end if  
  return ( $s, k$ )
```

```

VNS( $s, \{\mathcal{N}_i\}$ ) =
  until critério de parada satisfeito
     $k := 1$ 
    while  $k \leq m$  do
      seleciona vizinho aleatório  $s' \in \mathcal{N}_k(s)$  { shake }
       $s'' := \text{BuscaLocal}(s')$ 
       $(s, k) := \text{Movimento}(s, s'', k)$ 
    end until
  return  $s$ 

```

Handwritten annotations:

- A red arrow points from the k in $\mathcal{N}_k(s)$ to the k in $\text{Movimento}(s, s'', k)$.
- The word "shake" is circled in red.
- The expression $s \rightarrow s''$ is written in red.
- The numbers 123 are written in red with a curved arrow pointing from 1 to 2 to 3.