utad

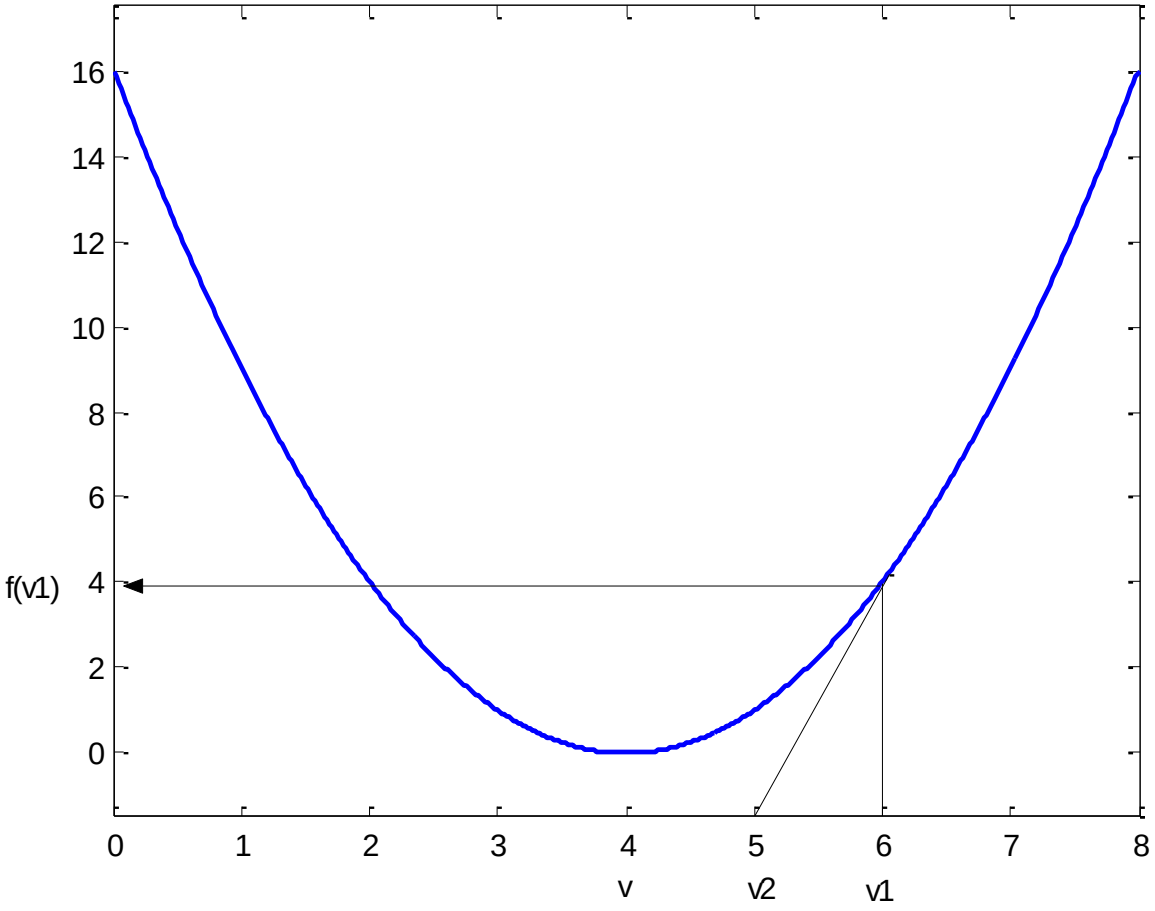
Inteligência Artificial

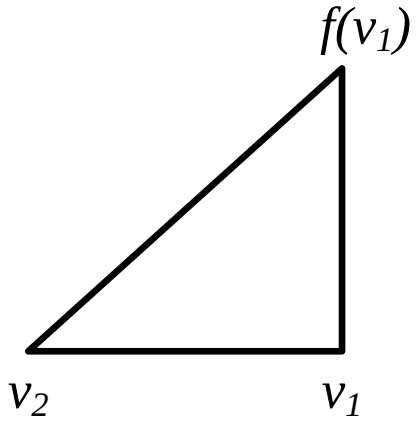
Métodos de Pesquisa (*Search*) – Parte II

Paulo Moura Oliveira
Departamento de Engenharias
Gabinete F2.15, ECT-1
UTAD
email: oliveira@utad.pt

Método de Newton-Raphson

- ✓ Considere o seguinte exemplo. Considerando a solução atual v1 como obter v2?



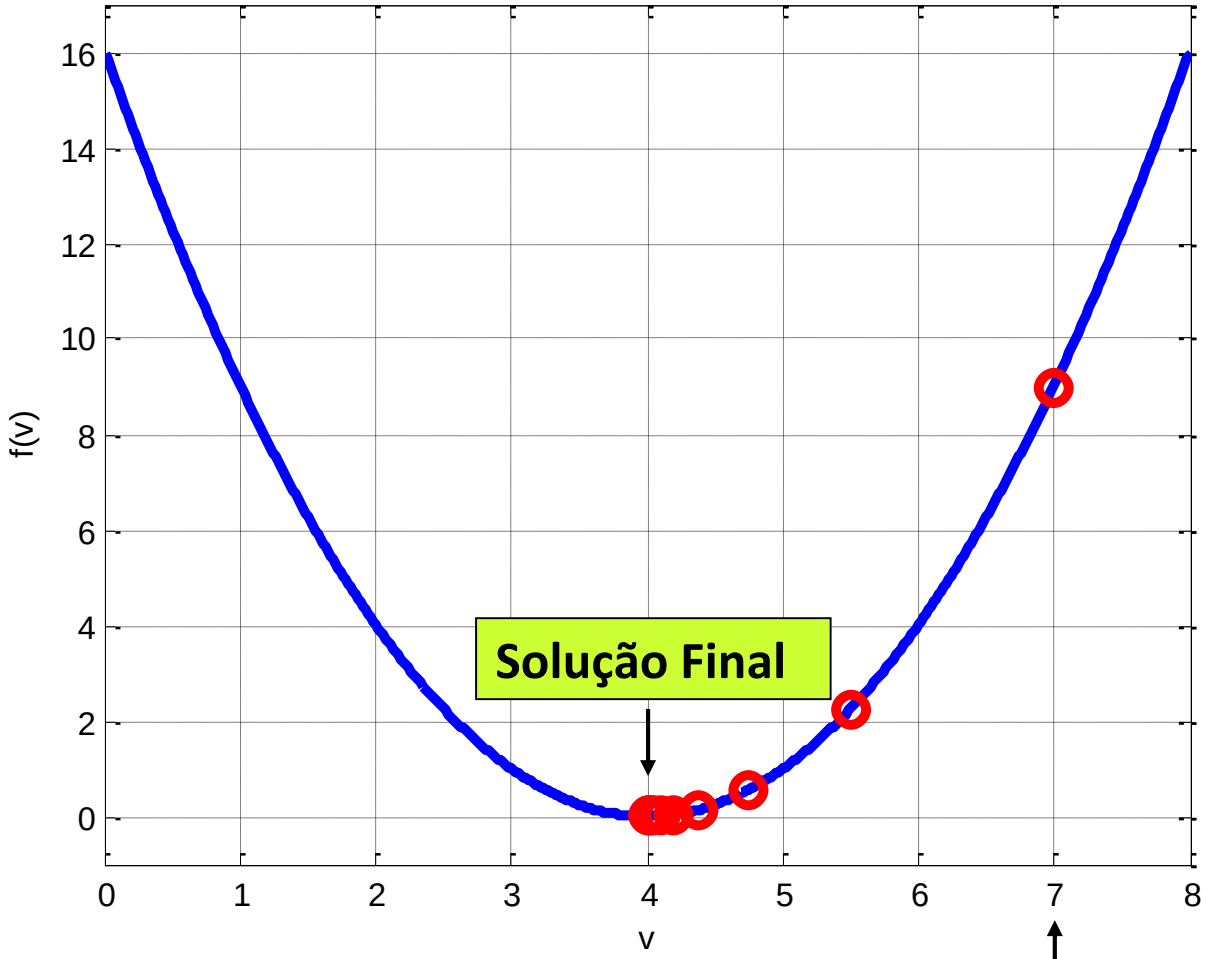


$$f'(v_1) = \frac{f(v_1)}{(v_1 - v_2)}$$

$$\therefore v_1 - v_2 = \frac{f(v_1)}{f'(v_1)} \Rightarrow v_2 = v_1 - \frac{f(v_1)}{f'(v_1)}$$

$$\boxed{v_{k+1} = v_k - \frac{f(v_k)}{f'(v_k)}}$$

Método de Newton-Raphson



Newton-Raphson (10 Iterações)

k=2	*f(k)=2.2500*	*df(k)=3.0000*	*v(k)=5.5000*
k=3	*f(k)=0.5625*	*df(k)=1.5000*	*v(k)=4.7500*
....			
k=10	*f(k)=0.0000*	*df(k)=0.0117*	*v(k)=4.0059*

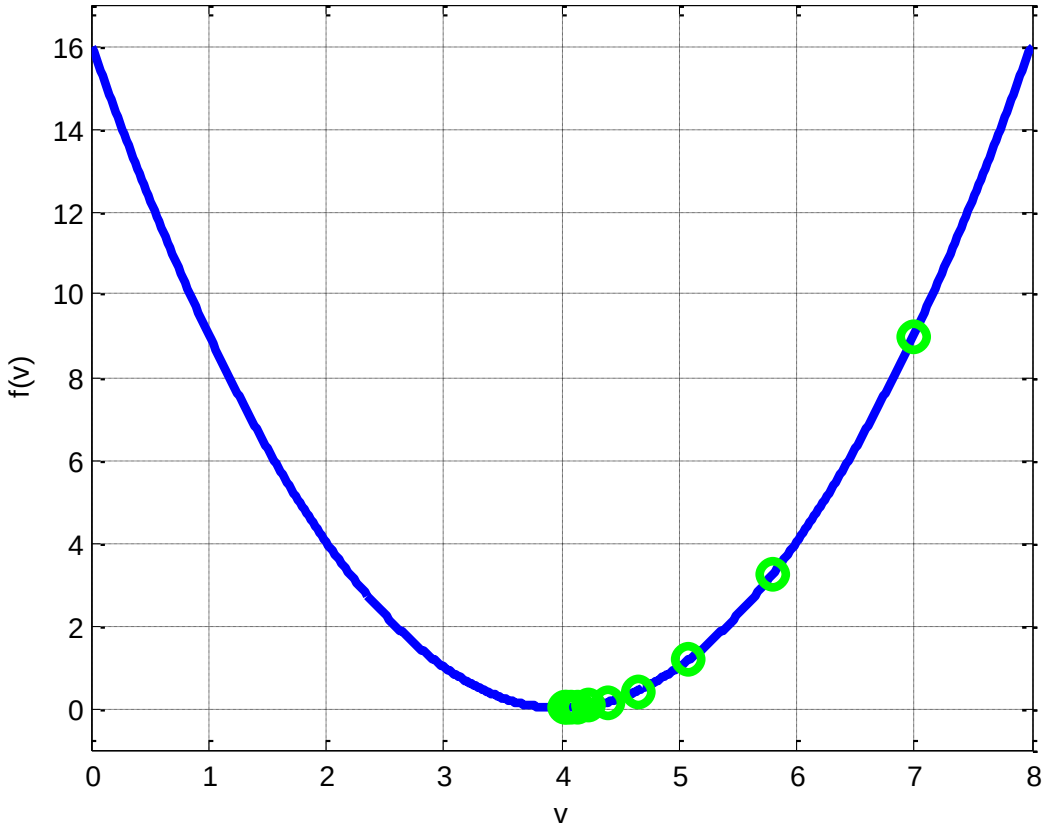
Método do Gradiente Descendente

- ✓ O método de Newton-Raphson pode ser de aplicação complexa. Em alguns casos, como o do veículo autónomo tem de ser simplificado

$$v_{k+1} = v_k - \alpha \, f'(v_k)$$

Gradient Descent (Alfa=0.2)

k=1	*v(k)=7.0000*
k=2	*v(k)=5.8000*
k=3	*v(k)=5.0800*
k=4	*v(k)=4.6480*
k=5	*v(k)=4.3888*
k=6	*v(k)=4.2333*
k=7	*v(k)=4.1400*
k=8	*v(k)=4.0840*
k=9	*v(k)=4.0504*
k=10	*v(k)=4.0302*



Caminhada Aleatória (Random Walk)

- ✓ Movimentos em qualquer direção têm a mesma probabilidade.
- ✓ Cada passo é independente do anterior.

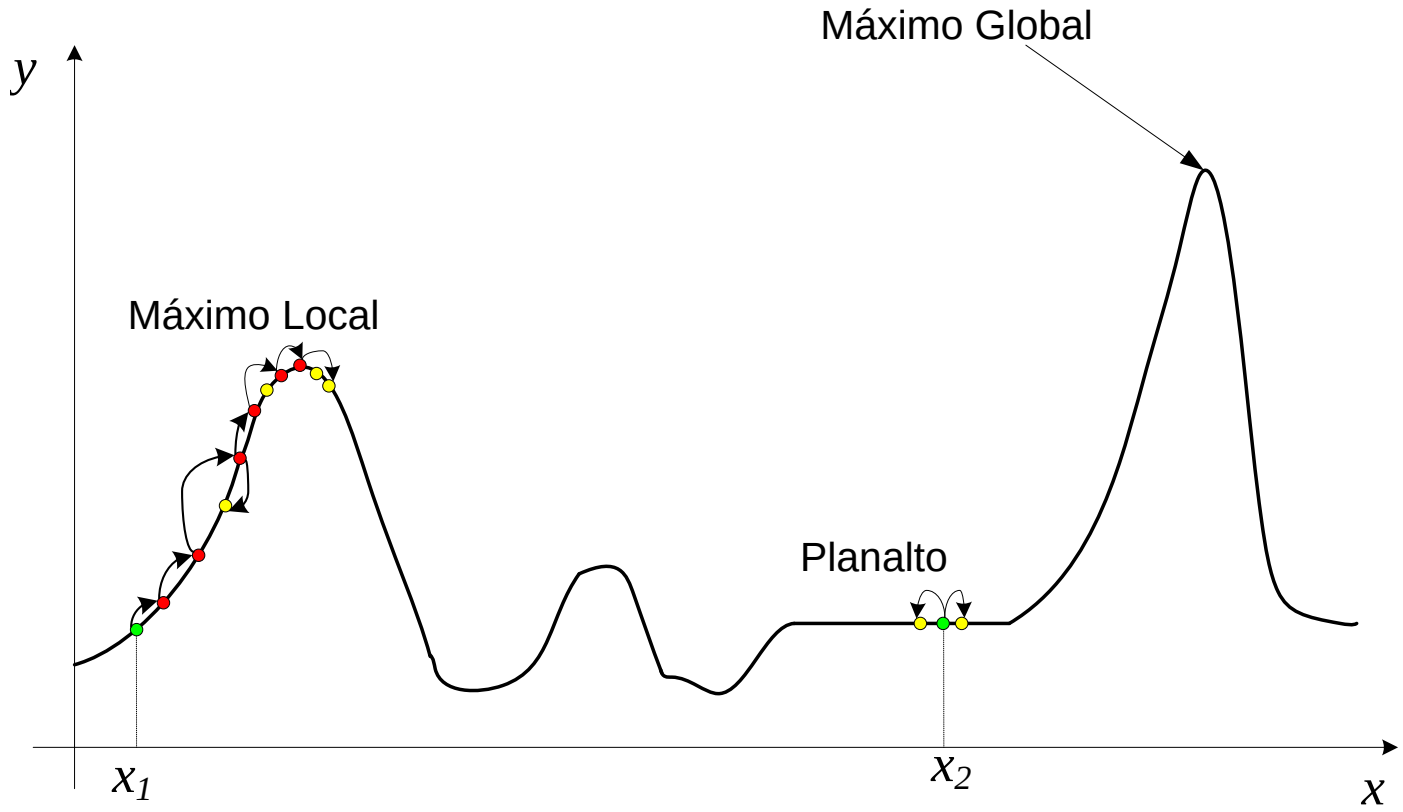
✘ A Random Search Algorithm:

1. *Selecionar um estado inicial aleatoriamente.*
2. *Fazer alteração local do estado atual.*
3. *Repetir passo 2 até que o estado objetivo seja atingido (ou o tempo se esgote…)*

- ✓ Tópicos relacionados:
  - Movimento Browniano (ver https://en.wikipedia.org/wiki/Brownian_motion)

Método da Subida da Colina
(*Hill-Climbing Method*)

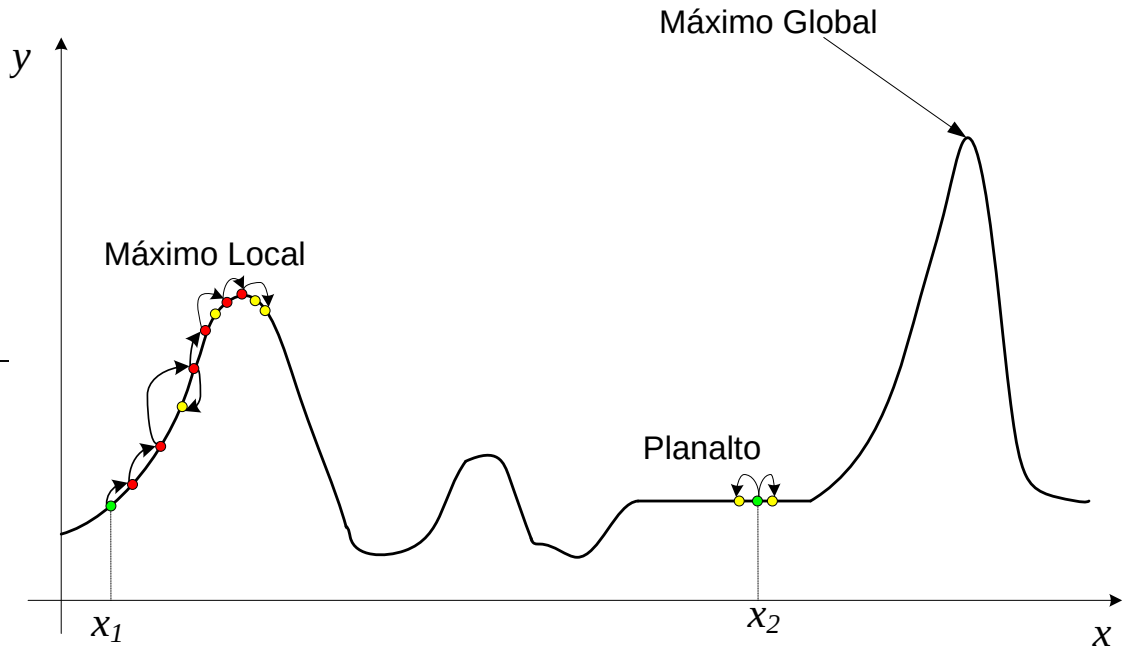
- Problema: Considerando um ponto inicial da pesquisa gerado aleatoriamente no espaço de pesquisa como chegar ao máximo global?



Método da Subida da Colina

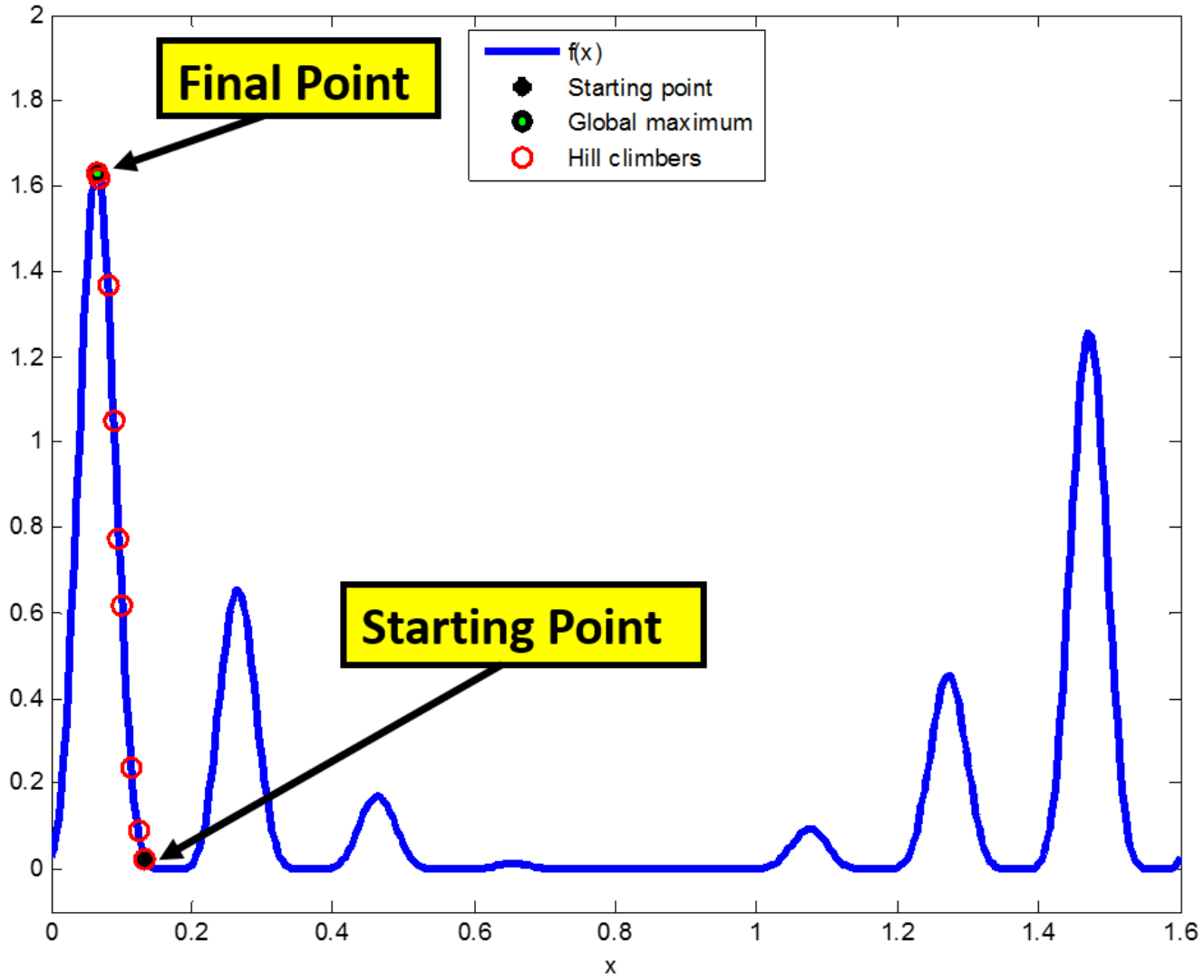
- Um algoritmo para a subida da colina (maximização):

```
t=0      % Contador de Iterações
inicializar x(t) aleatoriamente no espaço
while(!(critério de término))
    gerar uma nova solução x(t+1) aleatoriamente
    if  f(x(t+1))>f(x(t))
        x(t)=x(t+1)
        f(x(t))= f(x(t+1))
    end if
    t=t+1
end while
```



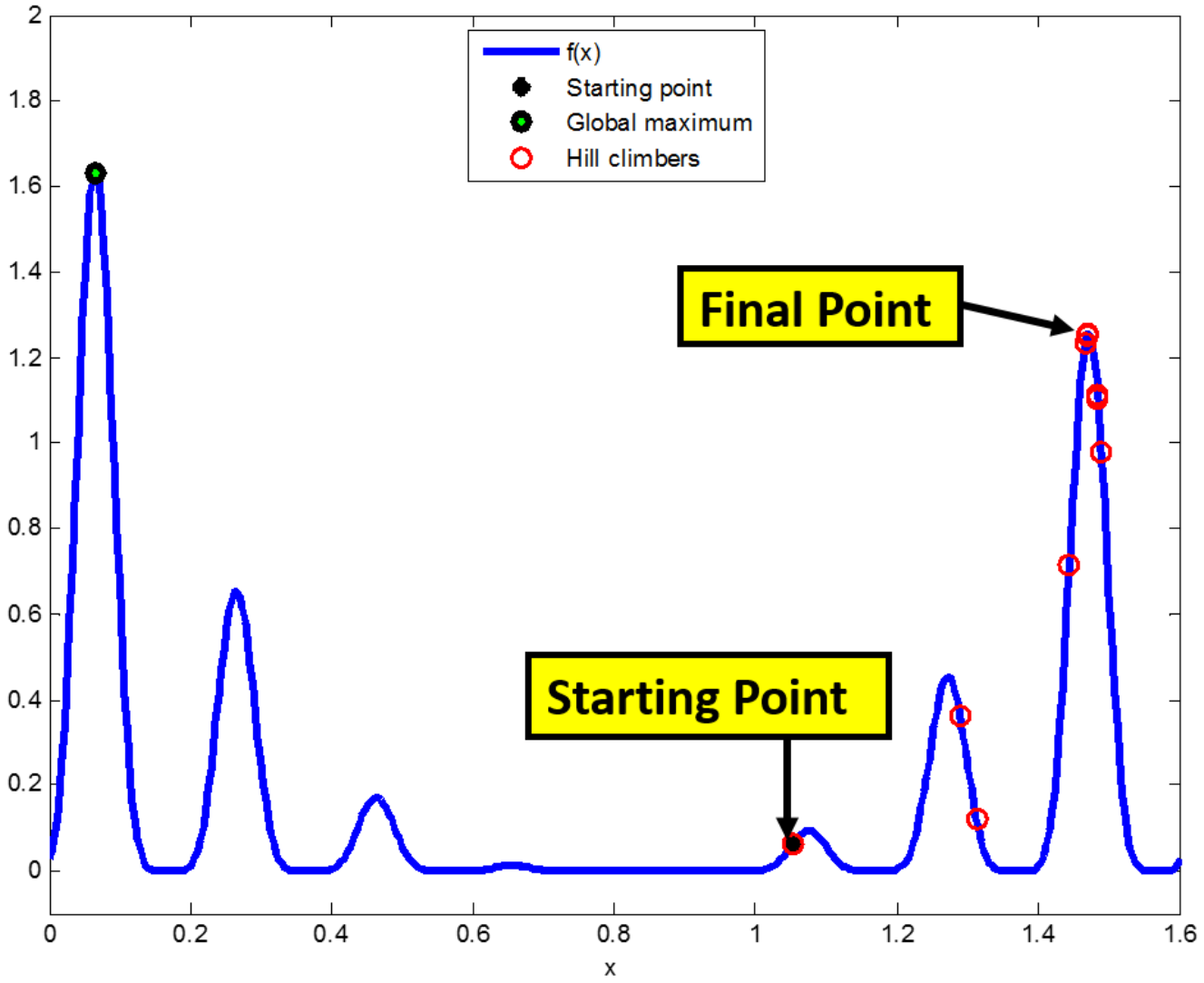
Método da Subida da Colina

- ✓ Exemplo de uma execução do algoritmo com sucesso:



Método da Subida da Colina

✓ Exemplo de uma execução do algoritmo com insucesso:

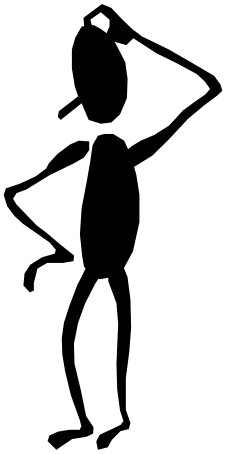


Método da Subida da Colina

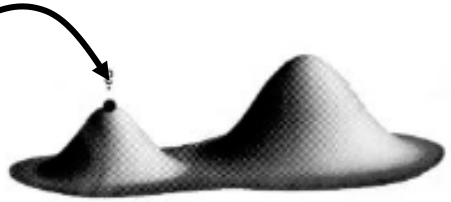
- ✓ Questões relevantes:

Como é que sabemos que estamos num máximo (ou mínimo) global e não num local?

Como é evitamos ficar presos num máximo (ou mínimo) local ?

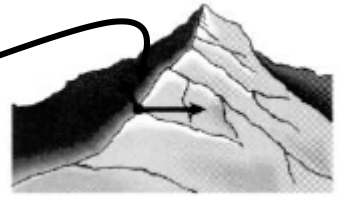


Exemplo de um máximo local



Exemplo de um planalto

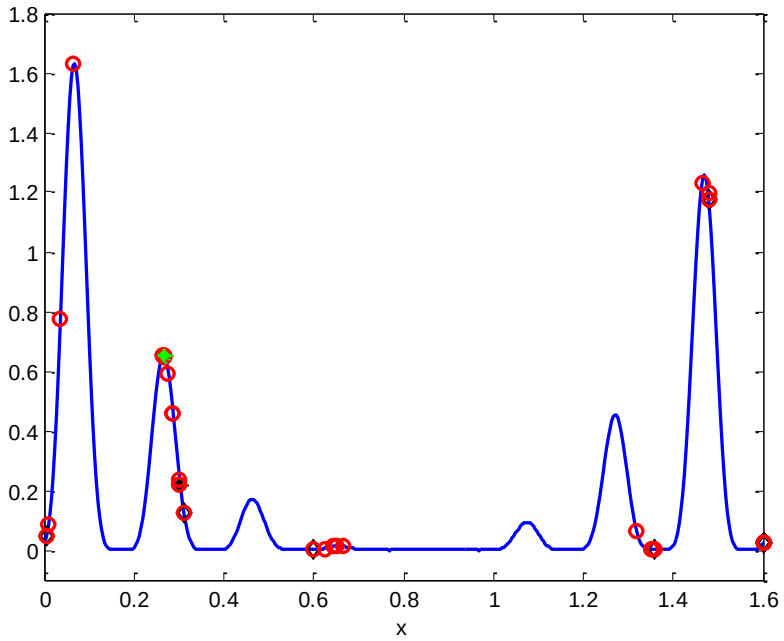
Exemplo de um cume

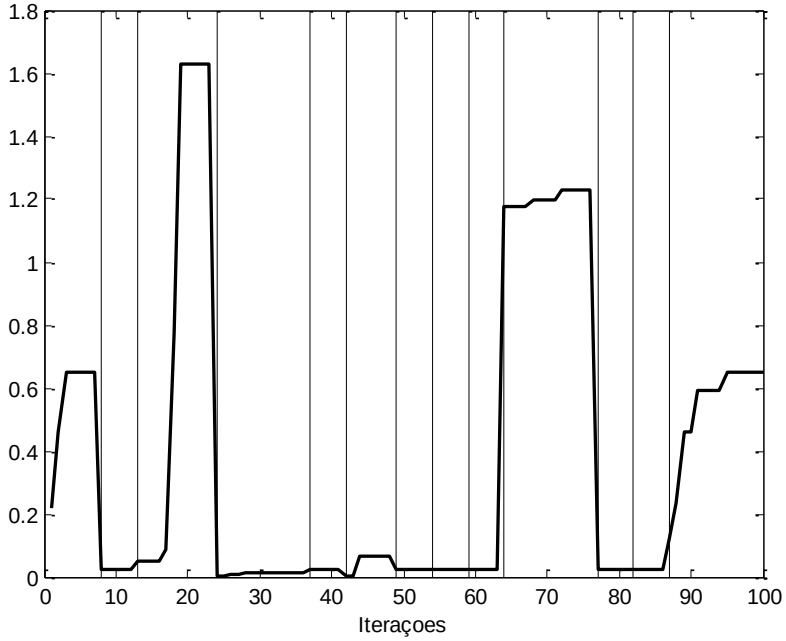


Subida da Colina com Re-inicialização Múltipla (*Hill-Climbing with Multiple Re-Start*)

Porque não repetir a pesquisa da subida da colina várias vezes?

Qual o critério a utilizar para re-inicializar a pesquisa?





100 iterações 12 re-inicializações

Subida da Colina com Re-inicialização Múltipla

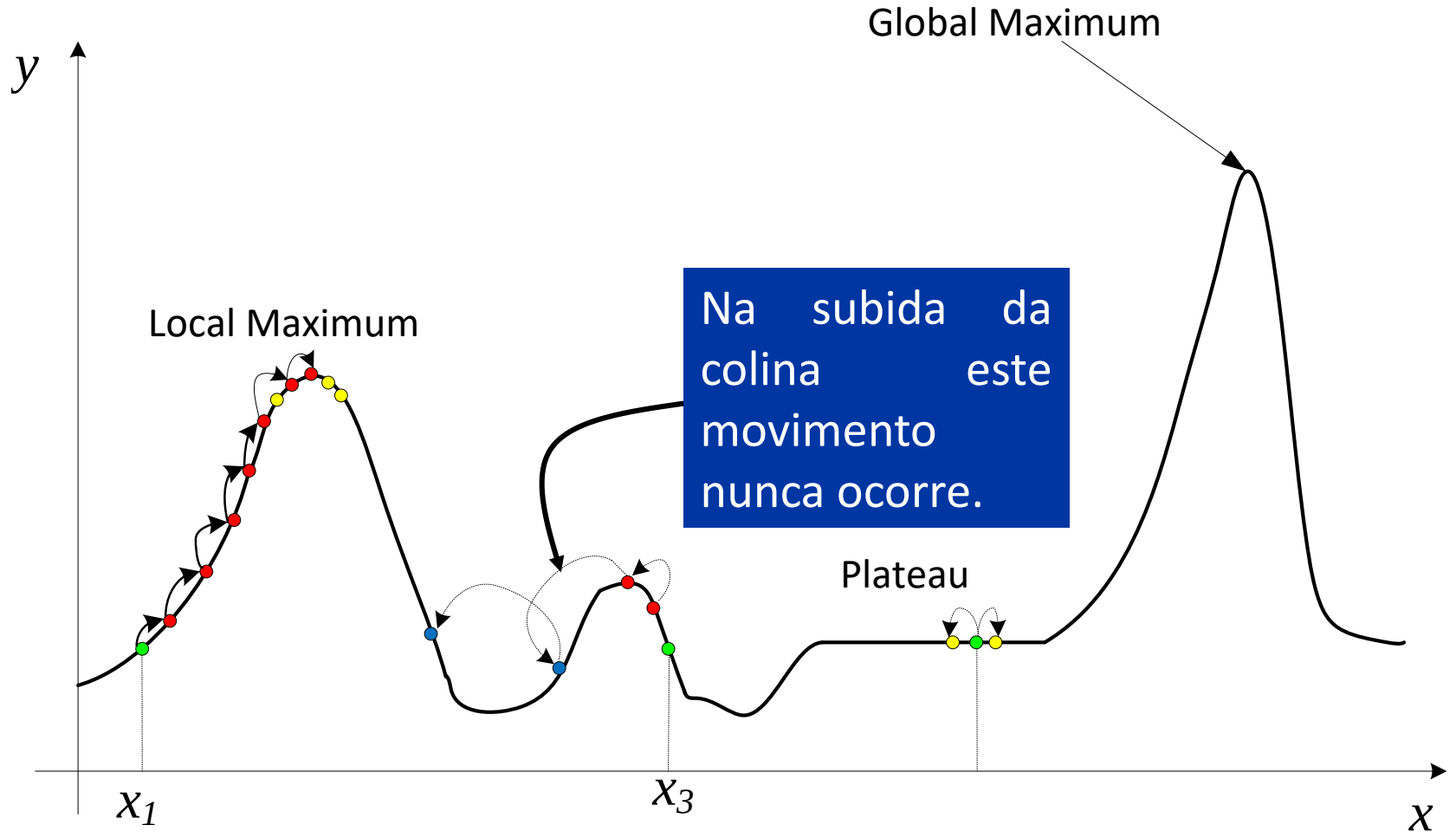
- Um algoritmo para a subida da colina com re-inicialização múltipla:

- Maximização

```
t = 0
inicializar x(t) aleatoriamente
best_x=x(t)
best_f=f(t)
while(!(critério de término))
   gerar uma nova solução aleatória x_new
   if  f(x_new) > f(x(t))
       x(t)= x_new
       f(x(t))=f(x_new)
      if  f(x(t)) > f(best_x)
          best_x=x(t)
          best_f=f(x(t))
      end if
   end if
   determinar critério de estagnação
   if pesquisa estagnada
      inicializar x(t) aleatoriamente
      atualizar f(x(t))
   end if
   t = t + 1
end while
```

Que critério de estagnação utilizar?

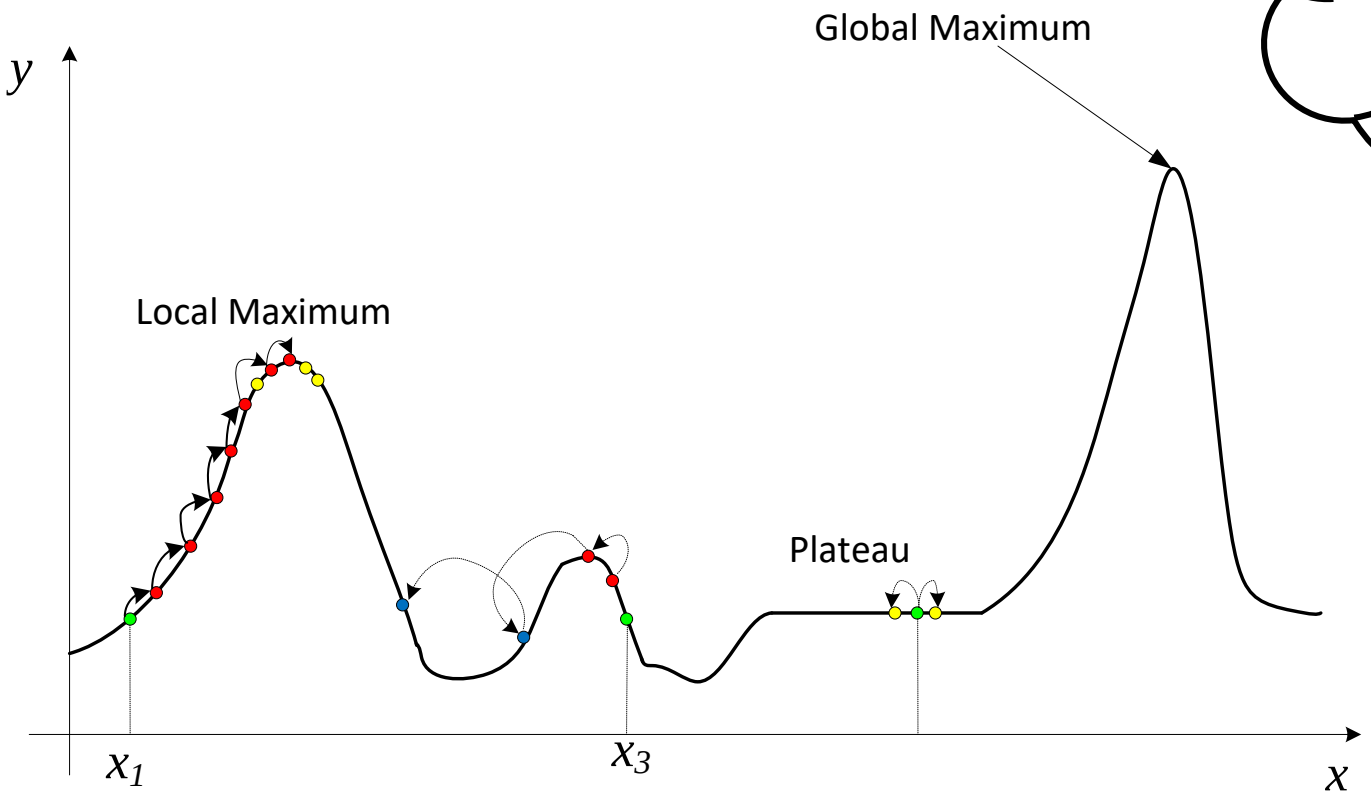
Métodos de Pesquisa Probabilísticos

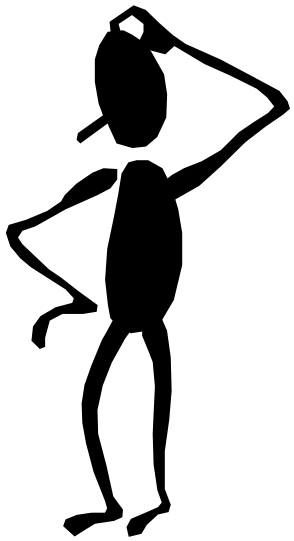


Tal como na vida, na pesquisa, por vezes é preciso dar um passo atrás antes de dar dois ou mais passos para a frente.

Métodos de Pesquisa Probabilísticos

Como fazer com que a pesquisa seja **normalmente ascendente**, mas que **ocasionalmente descendente**?

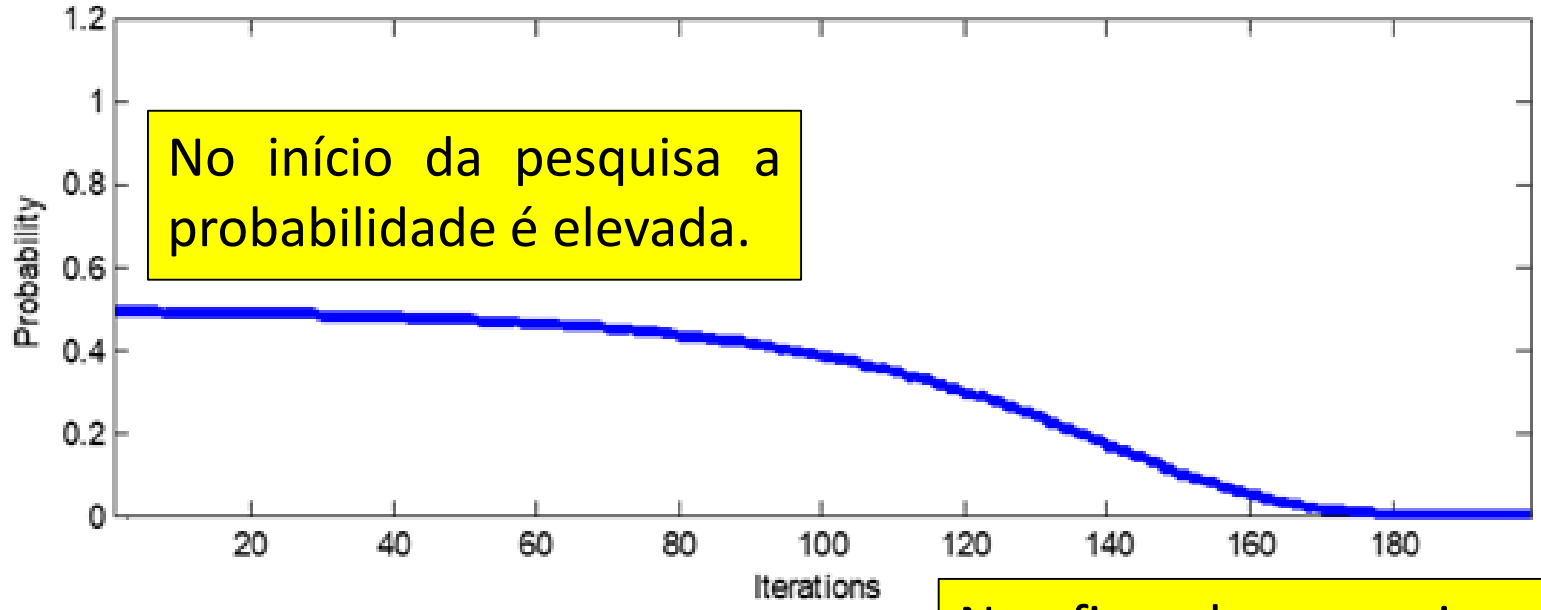




Métodos de Pesquisa Probabilísticos

Função de Probabilidade

$$p(t) = \frac{1}{1 + e^{y}}$$



No início da pesquisa a probabilidade é elevada.

No fim da pesquisa a probabilidade é nula.

Simulated Annealing (SA)

- ✓ **A inspiração natural** da técnica do *Simulated Annealing (SA)* vem de um processo metalúrgico conhecido como *annealing*.

- ✓ Neste procedimento metalúrgico os metais são aquecidos a alta temperatura de forma a obter um estado líquido e depois arrefecido lentamente para evitar ruturas no material.

- ✓ Método proposto em 1983 por Kirkpatrick et al. baseado num algoritmo desenvolvido por Metropolis (1953).

- *Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. (pp. 671-680). Science.*
- *Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. Technical Report*

Simulated Annealing (SA)

- ✓ Estabelece-se uma analogia entre a temperatura do metal e um parâmetro ajustável do algoritmo SA: T- Temperatura.

A ideia consiste em fazer com que a probabilidade de aceitar uma solução pior que a solução atual seja:

- Maior no início da pesquisa e
- Gradualmente tender para zero no fim da pesquisa.

Energia da nova solução $f(x\_new)$

$$\Delta E(t) = E\_new - E(t)$$

Energia solução corrente (melhor até ao momento)

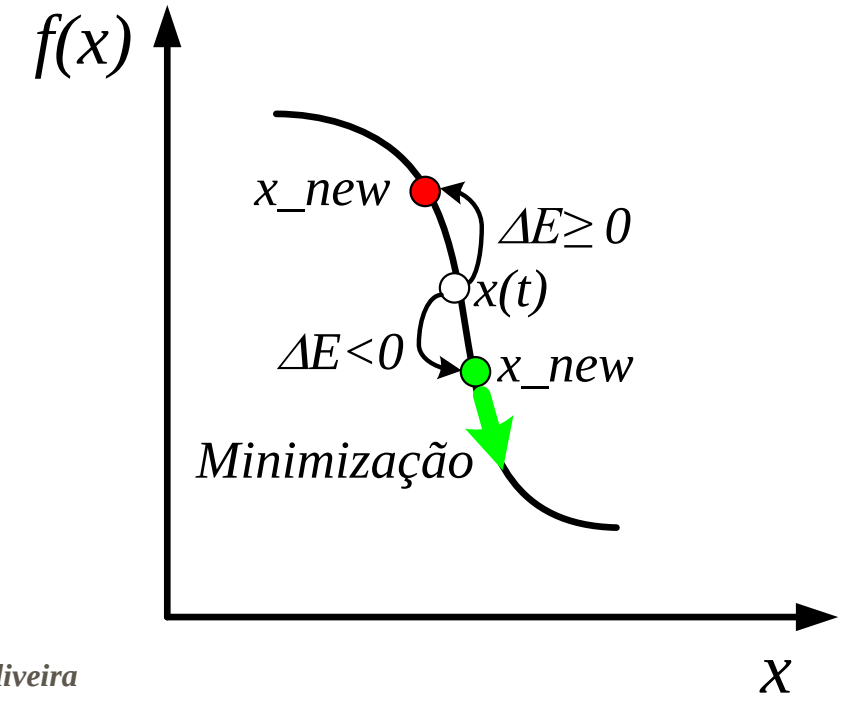
$f(x(t))$

Simulated Annealing (SA)

Energia da nova solução $f(x\_new)$

$$\Delta E(t) = E\_new - E(t)$$

Energia solução corrente (melhor até ao momento) $f(x(t))$



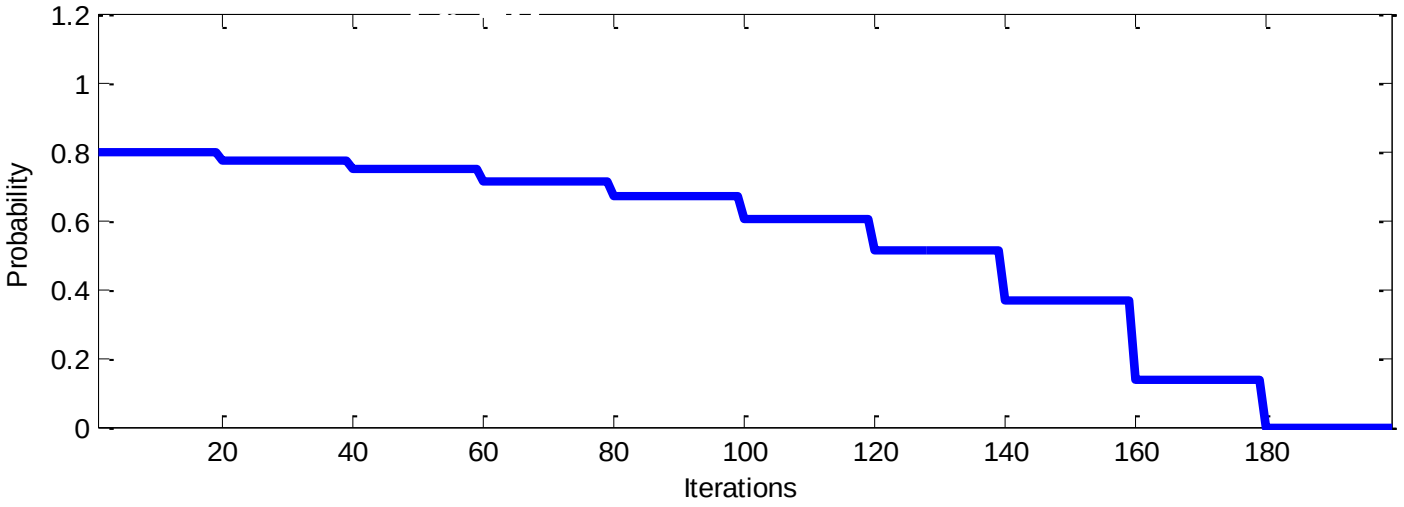
Simulated Annealing (SA)

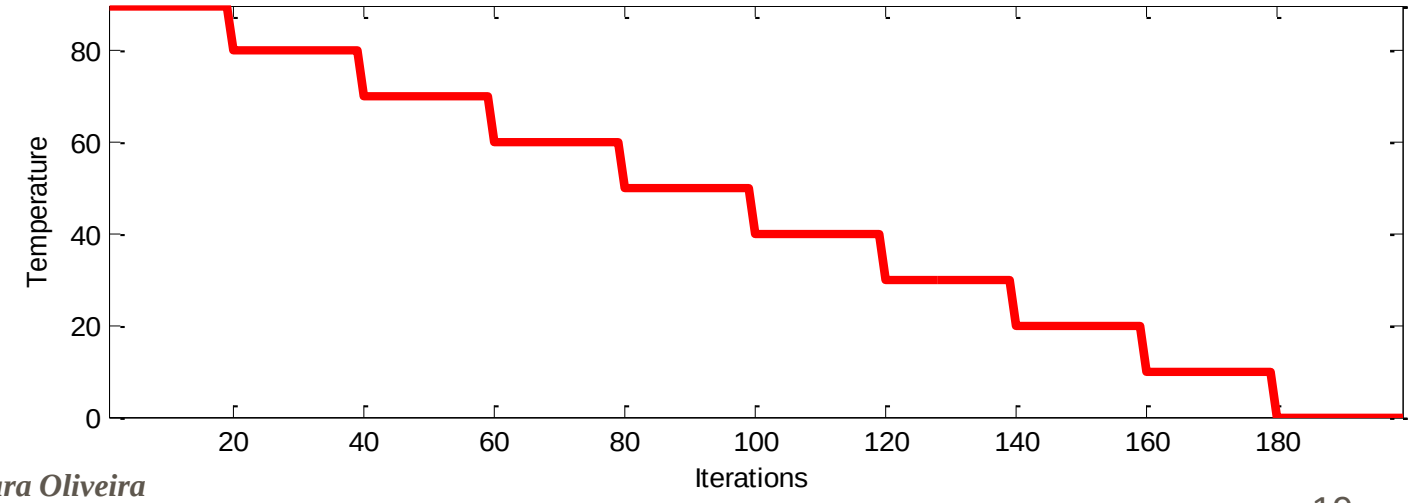
Função de Probabilidade

$$p(t) = e^{\frac{-\Delta E(t)}{T}}$$

Iter.	T
1	90
20	80
40	70
60	60
80	50
100	40
120	30
140	20
160	10
180	0

Exemplo: $\Delta E=20$ (const.) $T_{init}=90$

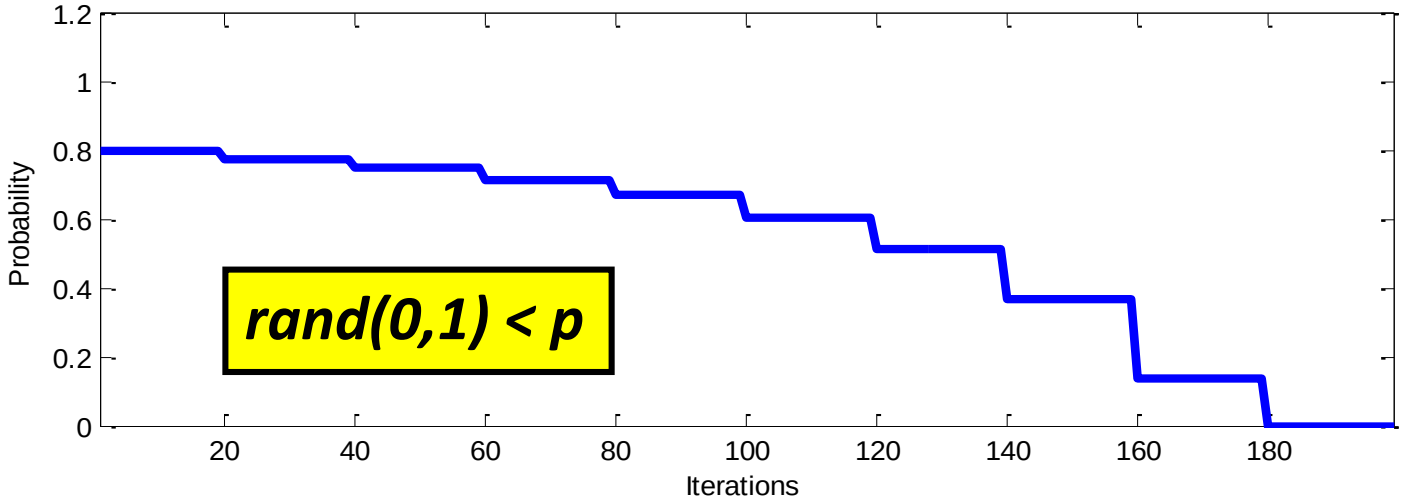


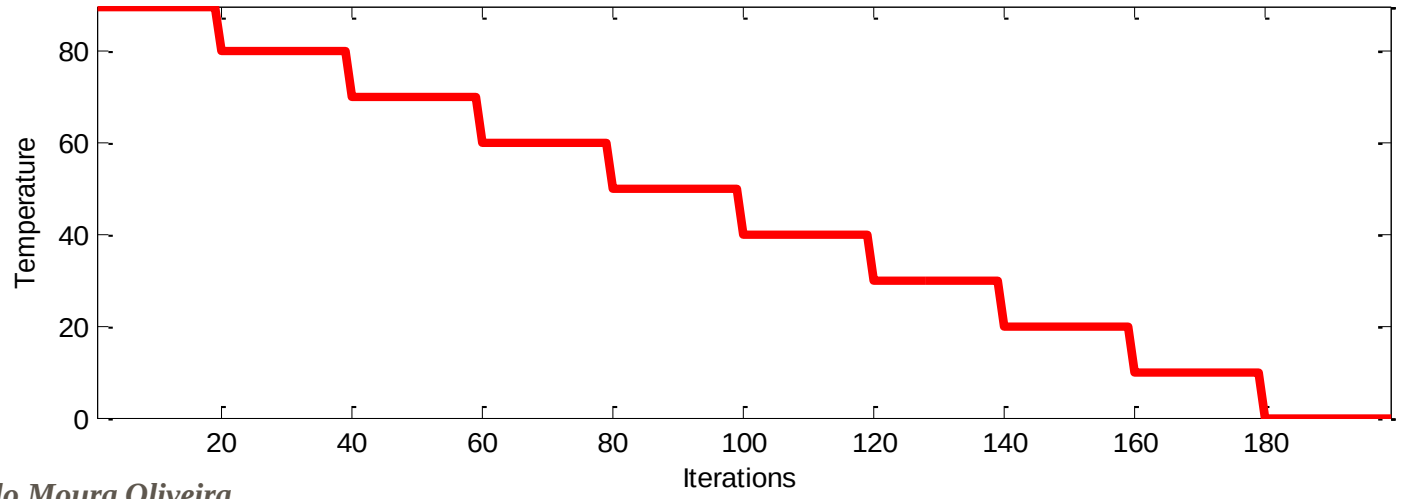


Simulated Annealing (SA)

Decisão Probabilística

- ✓ Com base numa comparação entre um número aleatório entre 0 e 1.

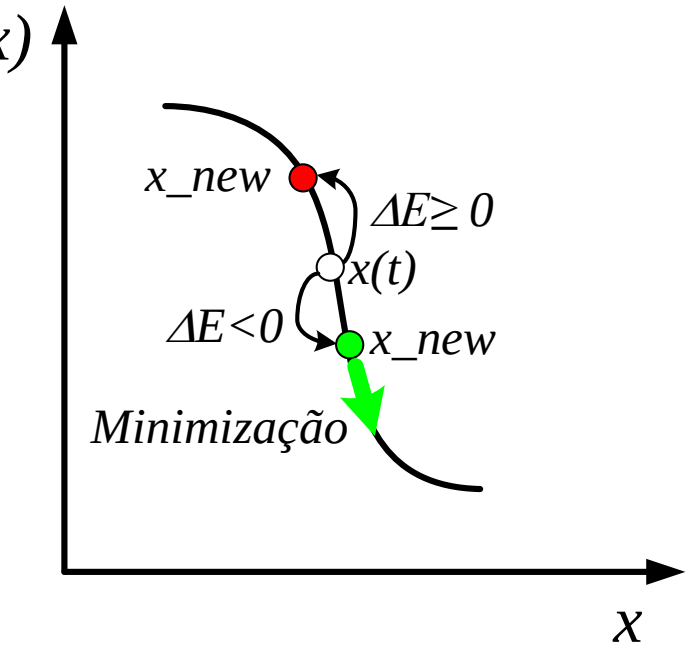




Simulated Annealing (SA)

Algoritmo:

```
t = 0
T=Tmax
inicializar x(t) aleatoriamente
while(!(termination criterion))
    n=1;
    while n<=Tit
       inicializar x_new aleatoriamente
       ΔE=…
        p=….
       if ΔE= f(x_new)- f(x(t)) <0    % Descida da colina
          x(t)=x_new
          f(x(t))=f(x_new)
       elseif rand(0,1)< p     % Subida da Colina
          x(t)=x_new
          f(x(t))=f(x_new)
       end if
       n=n+1
    end while
    T= Tnew     % Temperatura diminuída
    t = t + 1
end while
```



Melhoria

Pioria

Simulated Annealing (SA)

- ✓ O sucesso da aplicação do SA depende do ajustamento correto de uma série de aspetos para cada aplicação, tais como

1. O valor da temperatura inicial
2. A relação de decaimento da temperatura utilizada
3. A lei probabilística utilizada
4. O número de repetições da pesquisa para cada valor da temperatura.

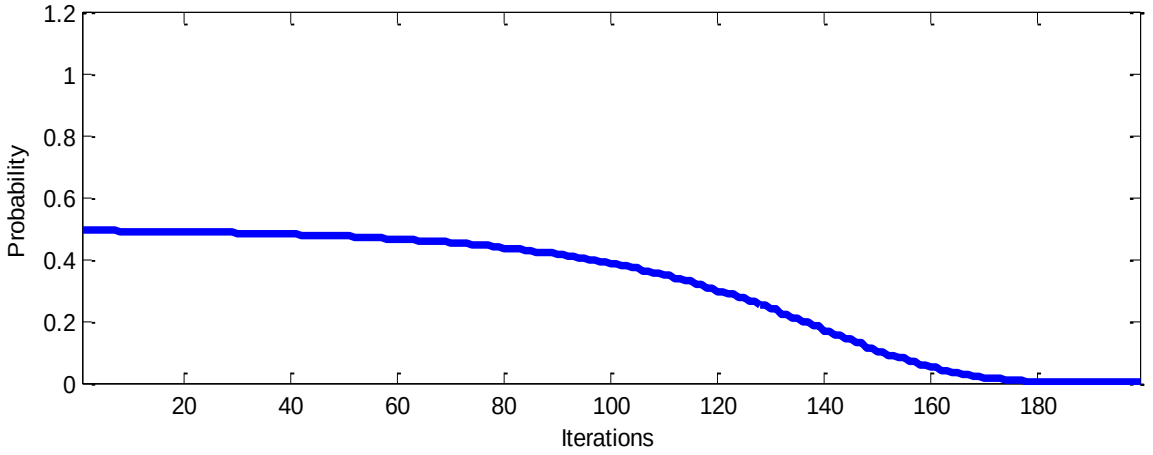
Simulated Annealing (SA)

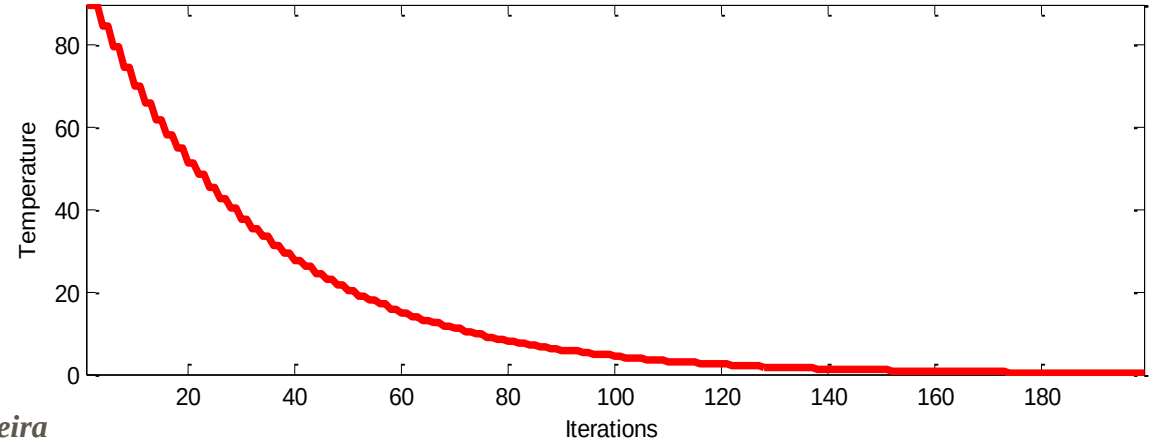
Exemplo: $\Delta E=2$ (const.) $T_{init}=90$ $T_{fin}=0$

$$p(t) = \frac{1}{1+e^{\frac{(E\_new - E(t))}{T}}} = \frac{1}{1+e^{\frac{\Delta E(t)}{T}}}$$

$Tnew=0.94\ T$

Perfil





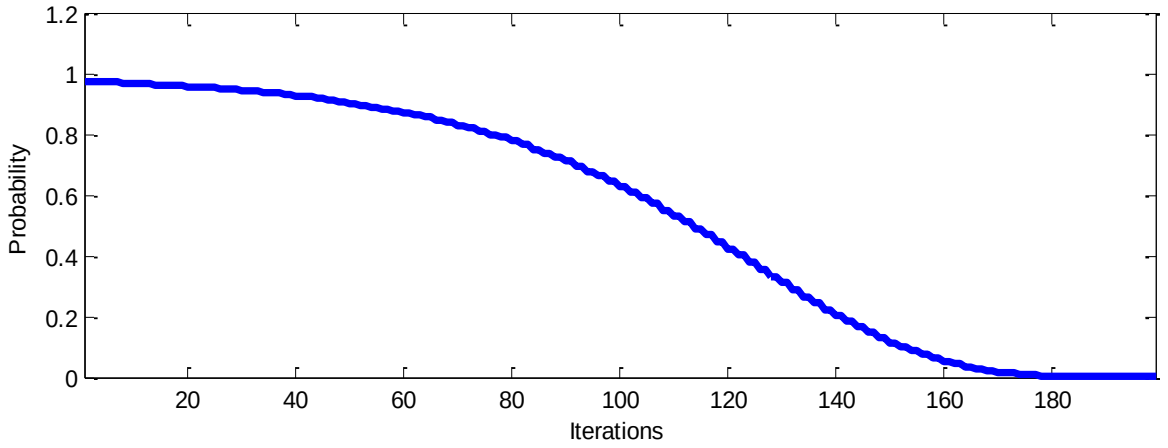
Simulated Annealing (SA)

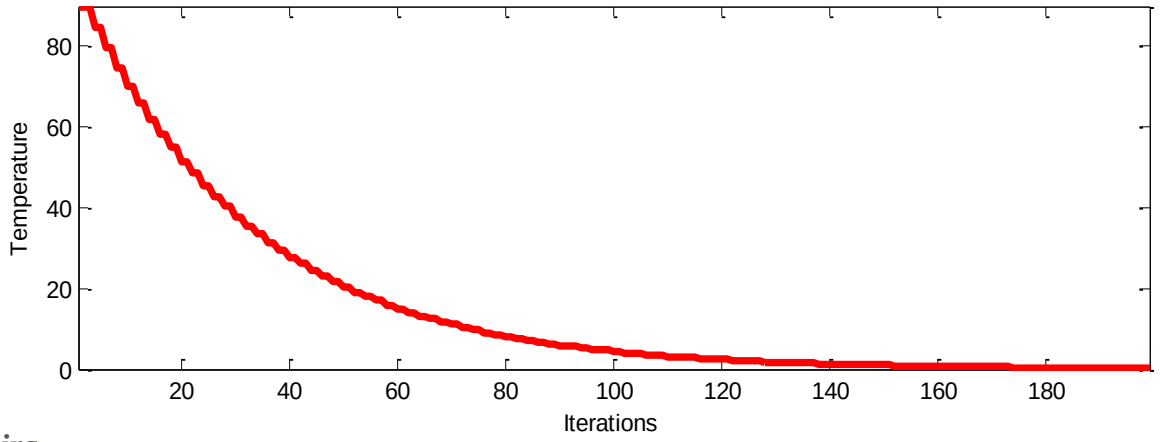
Exemplo: $\Delta E=2$ (const.) $T_{init}=90$

$$p(t) = e^{\frac{-\Delta E(t)}{T}}$$

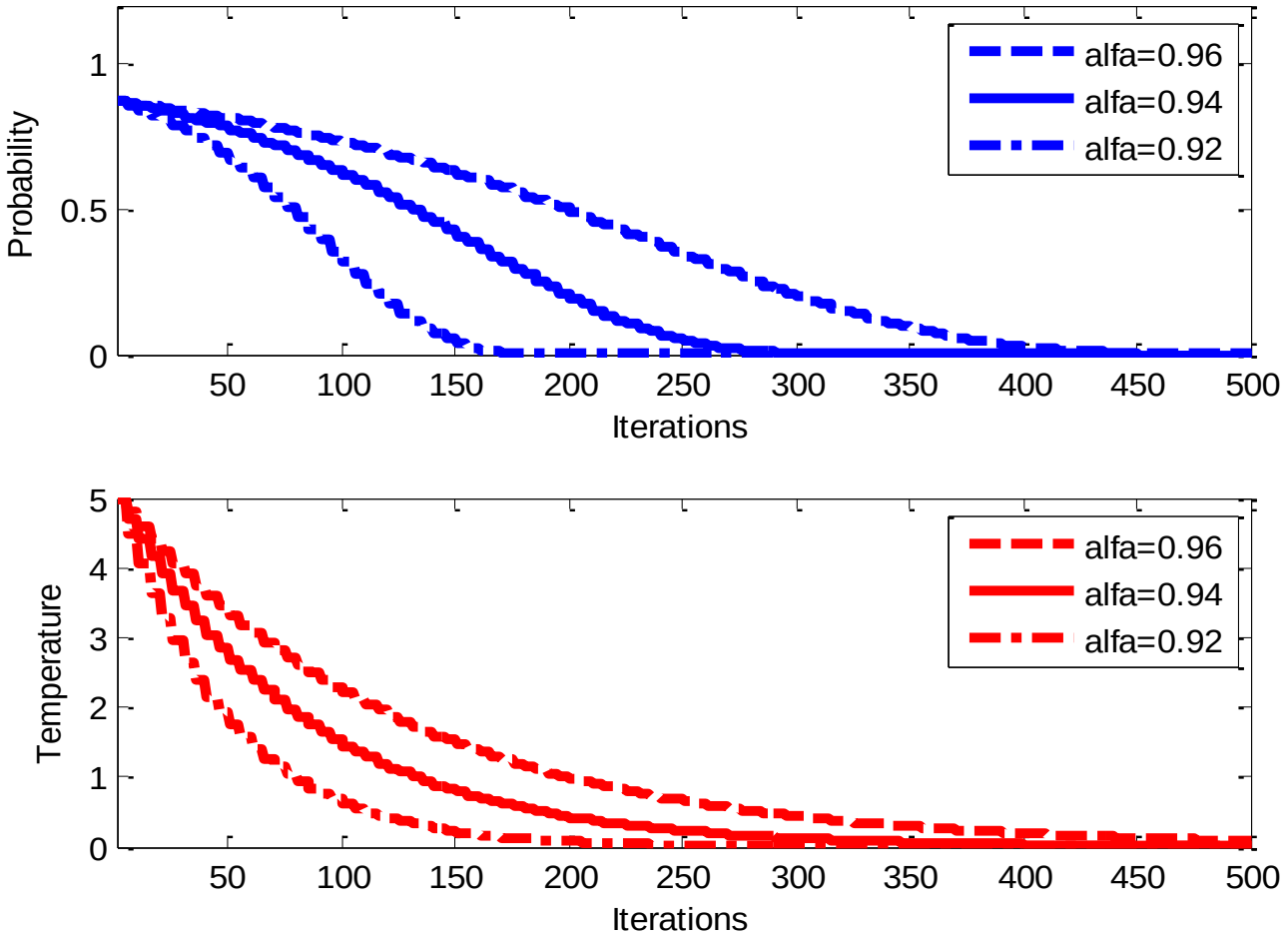
$Tnew=0.94\ T$

Perfil





Simulated Annealing (SA)



Simulated Annealing (SA)

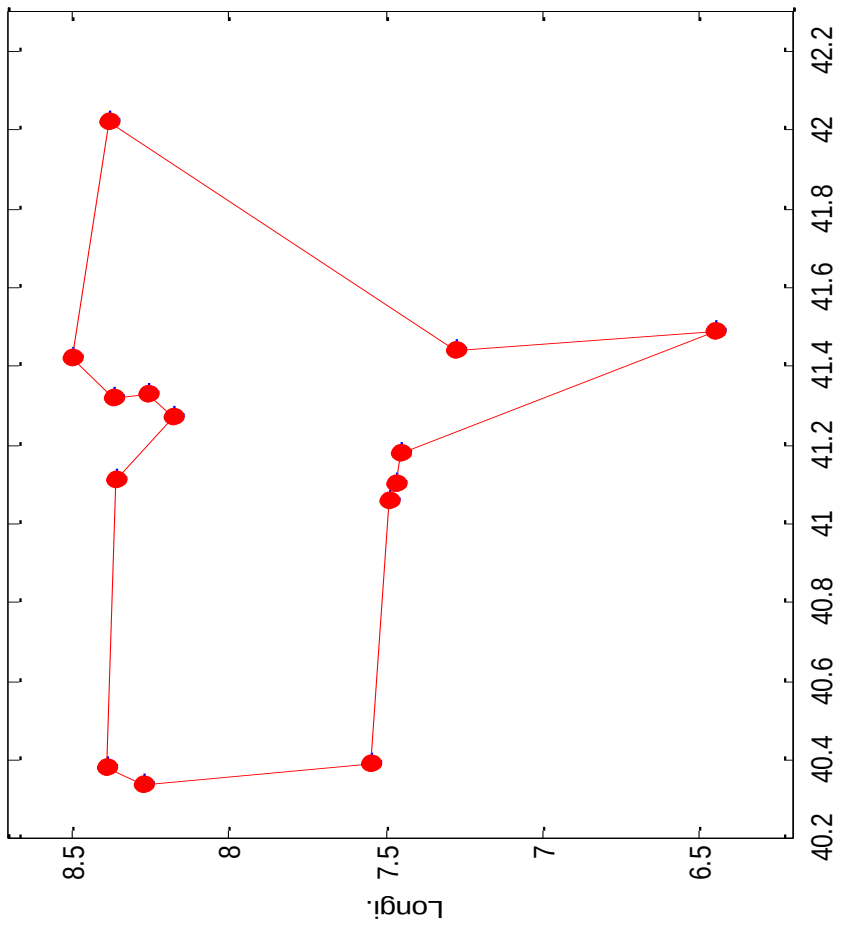
Exemplo: Problema do Caixeiro Viajante (*Traveling Salesman Problem*)

- ✓ Considere que um caixeiro viajante tem de fazer uma rota que inclua cidades que constam no seguinte mapa (com estrela).
- ✓ A única cidade que pode ser visitada duas vezes é a de origem.
- ✓ Qual o percurso que corresponde à distância mínima?

Simulated Annealing (SA)

Exemplo: Problema do Caixeiro Viajante (*Traveling Salesman Probelm*)

- ✓ Será esta solução?



Simulated Annealing (SA)

Exemplo: Problema do Caixeiro Viajante (*Traveling Salesman Probelm*)

- ✓ Ou esta? E com 20,30,40… cidades?