Trabalho prático I:

Métodos de Pesquisa da Computação Natural

Elaborado por

Cátia Machado nº 64223

Felícia Madureira nº 65120

Inteligência Artificial

Paulo Moura Oliveira

# INTRODUÇÃO

No âmbito da unidade curricular de Inteligência Artificial, foi-nos proposto o trabalho prático nº 1 que visa a aplicação dos Métodos de Pesquisa de Computação Natural.

É nos dada uma função bidimensional para estudo e, através dos algoritmos de Simulated Annealing (SA) e Particle Swarm Optimization (PSO), iremos procurar encontrar o seu mínimo global.

Na parte inicial deste trabalho, serão introduzidos alguns conceitos relativos a estes algoritmos, bem como algumas das suas aplicações práticas. Posteriormente, serão apresentados os resultados e gráficos obtidos que nos permite concluir qual o mínimo da função em estudo.

Em anexo, serão apresentados os códigos efetuados no contexto deste trabalho.

# SIMULATED aNNEALING

O Simulated Annealing é um algoritmo motivado por uma analogia ao recozimento de metais. Simula esse processo através da diminuição gradual da temperatura do sistema até esta convergir a um estado estável, congelado.

Utilizado para resolver problemas NP-completos, tais como o problema do caixeiro viajante. É uma técnica que muito dificilmente encontra a solução ótima, mas é frequentemente uma solução boa, mesmo na presença de algum ruído.

Resolve o problema de ficarmos presos num mínimo local, através do uso de duas técnicas: Algoritmo Metropolis e diminuição da temperatura.

 e^(-DeltaD/T)>R(0,1), Na primeira técnica, algumas soluções que não diminuem o caminho percorrido, são aceites para explorar o espaço de soluções. Isto acontece segundo o critério:

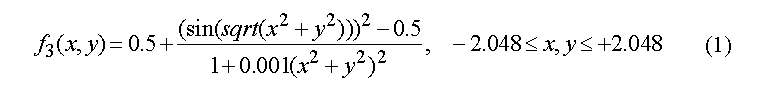
∆D é a mudança da distância (negativa para boa troca, positiva para má troca); T é uma temperatura sintética e R(0,1) é um numero aleatório no intervalo [0,1].

D é uma função custo. Se T é elevado, serão aceites muitas trocas más e uma grande parte do espaço de soluções é visitada. Objetos a ser trocados podem ser escolhidos aleatoriamente, existindo, no entanto, outras técnicas.

Na segunda, após muitas trocas e observando que a função de custo diminui muito lentamente, diminui-se a temperatura, limitando o número de más trocas. Se a temperatura for diminuída muitas vezes, pode-se optar por aceitar somente boas trocas e encontrar o mínimo local.

Existe uma outra estratégia, mais rápida, designada por threshold acceptance. Nesta estratégia, todas as trocas boas são aceites, assim como qualquer troca má que aumente a função custo menos do que um limite fixo. Posteriormente, o limite é reduzido temporariamente. Isto elimina exponenciação e geração de números aleatórios.

# Resolução do Problema com Simulated Annealing

* 1. Problema: Encontrar o mínimo global da função:
  2. Solução implementada – Simulated Annealing codificado na ferramenta MATLAB
  3. Resultados expectáveis

A função apresenta 4 mínimos locais, localizados em (-2,-2),(2,2),(-2,2),(2,-2)

O mínimo global encontrado deverá estar localizado em (0,0)

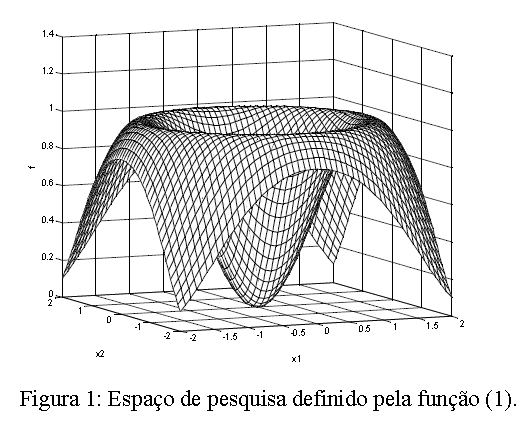
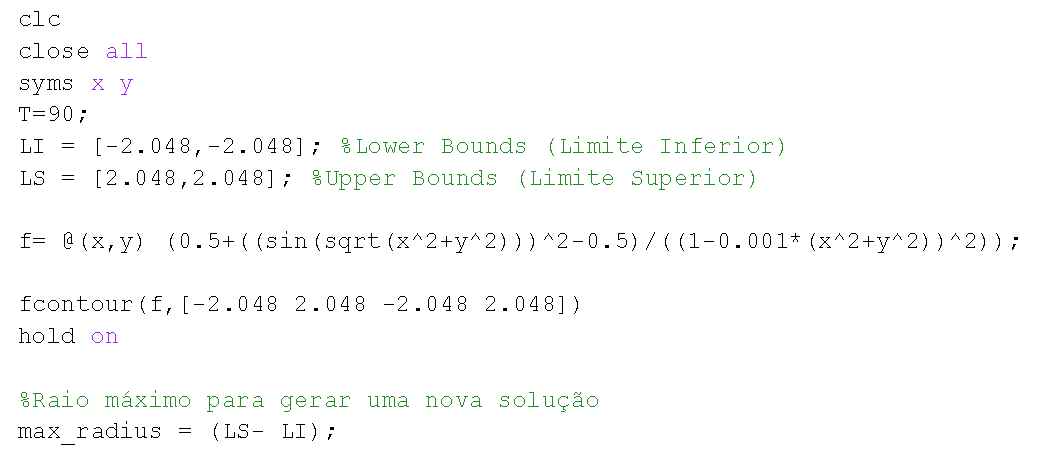
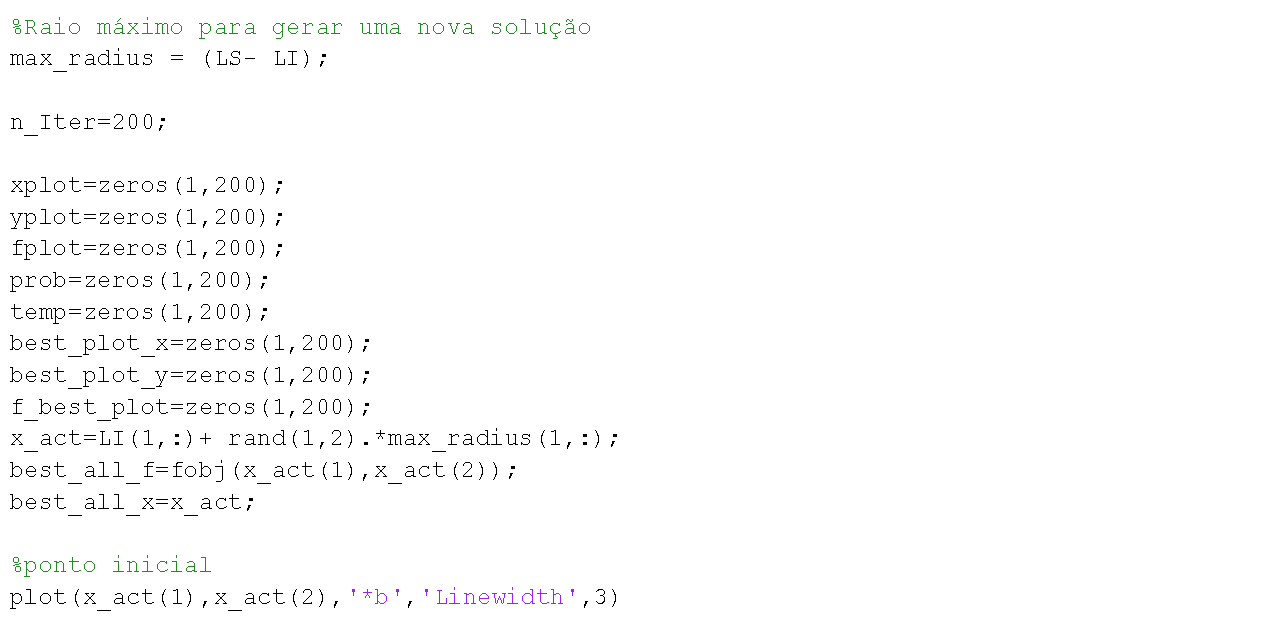


Figura 1 - Espaço de pesquisa definido pela função (1)

* 1. Análise do código elaborado
     1. Inicialização



Nesta fase inicial, inicializa-se a temperatura a 90 graus (testaram-se vários valores iniciais de temperatura, sendo este o que produzir gráficos e resultados mais eficazes); definem-se os limites superiores e inferiores do gráfico onde se vai desenhar a função (estabelece-se a escala da figura), declara-se a função proposta; desenha-se a mesma com recurso à função *fcontour*, obtendo-se o gráfico da função

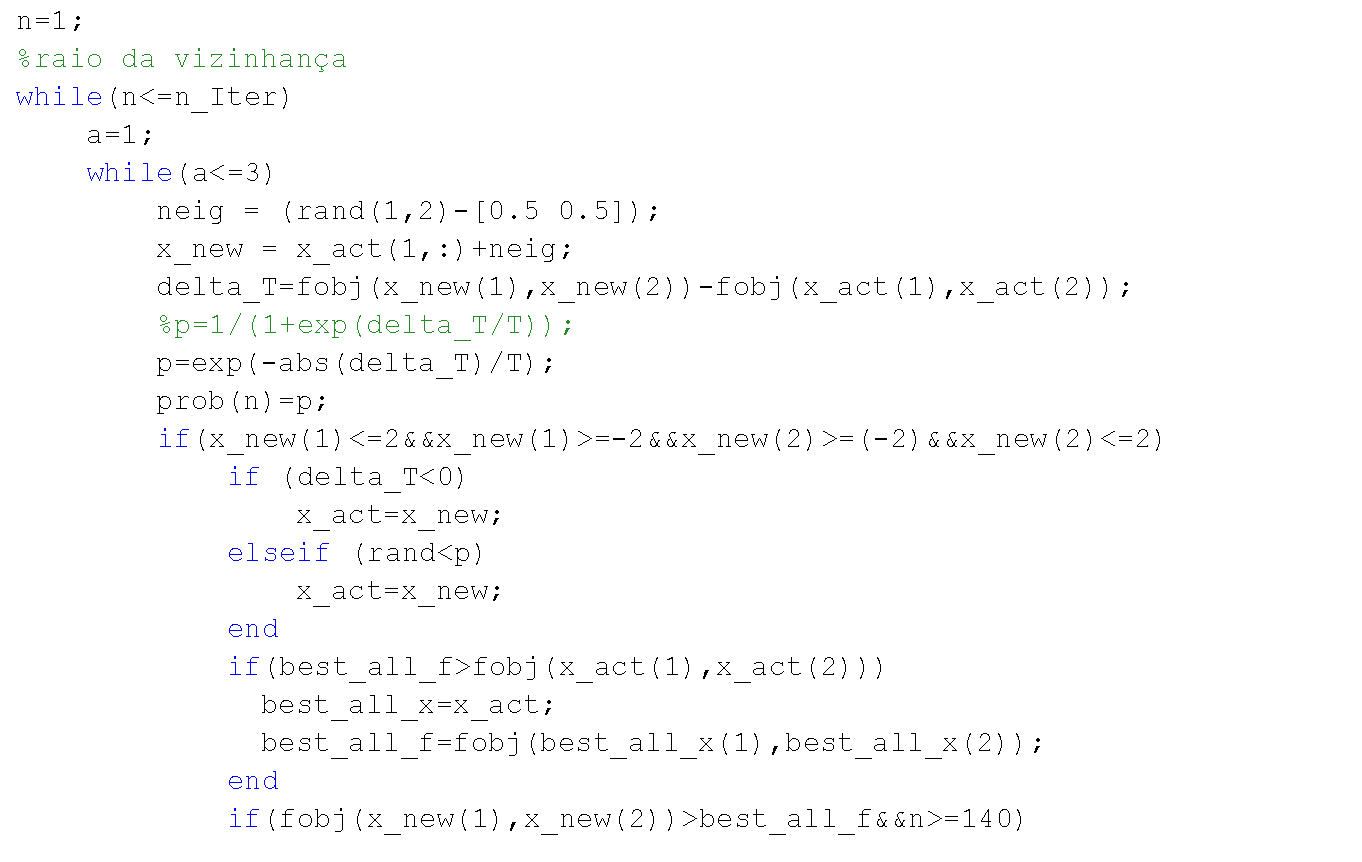
Para os pontos não saírem dos limites estabelecidos, define-se um max\_radius em que os pontos se podem movimentar. Estabelece-se um n\_Iter=200, para sair do ciclo ao atingir as 200 iterações.

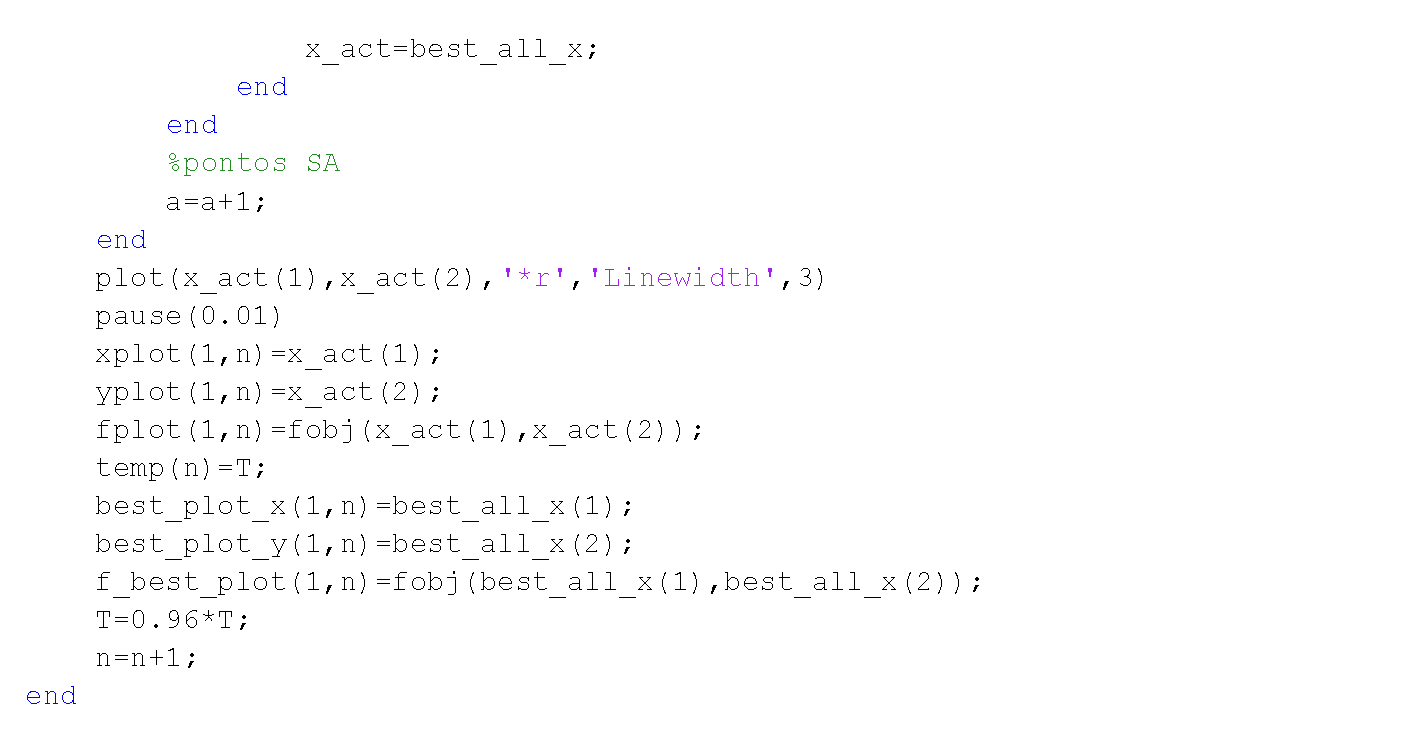
De modo às variáveis que armazenam valores atuais da localização dos pontos não mudarem de tamanho a cada iteração, inicializam-se como arrays de tamanho fixo (1 linha e 200 colunas – vetor linha), com as entradas todas a zero.

Inicializa-se aleatoriamente o x\_act (posição atual da partícula), obtendo-se um valor aleatório com x entre (-2,2) e y entre (-2,2), em que x\_act(1) corresponde ao valor da coordenada x, e x\_act(2) corresponde ao valor da coordenada y. Neste momento, como só temos um único ponto, o seu valor corresponde ao melhor até agora encontrado, pelo que se atribui o valor do objeto e da sua imagem às variáveis best\_all\_x e best\_all\_f, respetivamente.

Seguidamente, desenha-se com o comando plot, a localização do ponto inicial no gráfico, a azul.

* + 1. Ciclos interno e externo





No ciclo interno, que é repetido 3 vezes para cada iteração do ciclo externo, define-se uma vizinhança, neig, atualiza-se o x atual, obtendo-se o valor do novo x. Atualiza-se a temperatura, que é a diferença da imagem de x\_new e x\_act

Atualiza-se o valor da probabilidade com que se vai aceitar uma solução ‘pior’ e armazena-se esse valor de probabilidade para colocar no gráfico de variação da probabilidade.



Se estivermos dentro dos limites, e a diferença de temperatura for negativa, então aceita-se a nova solução – é uma solução de melhoria – aceita-se sempre. Caso contrário, se uma probabilidade aleatória for menor que a probabilidade de aceitação atual, então aceita-se a solução ‘pior’.

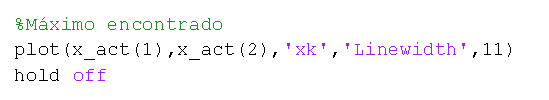
Se o valor atual for menor que o melhor valor até agora encontrado, então o valor atual passa a ser o melhor valor. Se já estivermos na iteração 140 e o valor atual for ‘pior’ que o melhor valor até agora, então retrocedemos para esse valor e recomeçamos a calcular a partir desse local. Isto é feito para evitar ficar preso em máximos locais.

Saindo do ciclo interno e colocando-se no ciclo externo, faz-se plot da posição atual da partícula, armazenam-se os valores atuais nas matrizes xplot, yplot e fplot, temp, best\_plot\_x e best\_plot\_y para posterior elaboração dos gráficos.

Atualiza-se a temperatura para uma diminuição de 0.04% a cada iteração.



Desenha-se o ponto máximo encontrado no gráfico, com um X a bold e a preto.



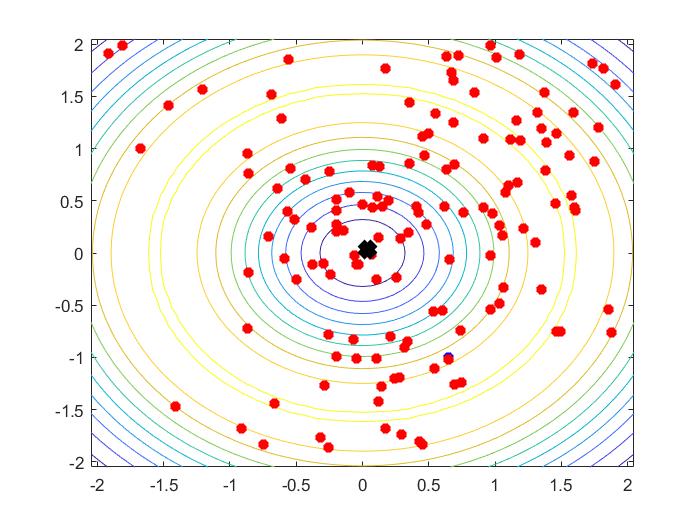
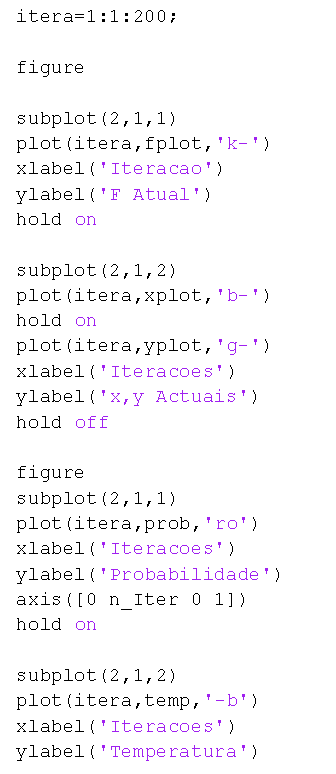
Estes comandos permitem obter o seguinte gráfico:

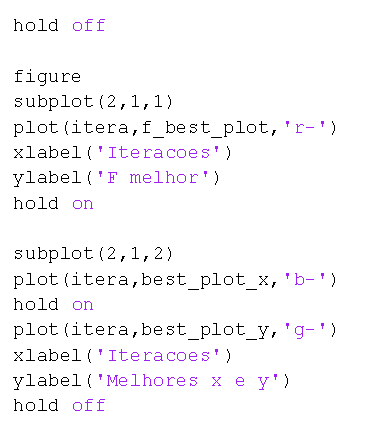
Figura 2 - Distribuição final da aplicação do SA

* Pontos vermelhos – Pontos atuais
* Cruz preta – Solução final encontrada

Como pretendido, os pontos percorrem a região, tentando alcançar o mínimo global; caso se encontrem num mínimo local, fazem ‘subidas na colina’ para escapar do mesmo. Reparamos que os pontos se ‘escapam’ dos extremos, onde se encontram os mínimos locais, atingindo o ponto (0,0) onde se encontra o mínimo global.

* + 1. Traçado dos gráficos a interpretar





Com estes comandos é possível obter 3 imagens, cada uma com dois gráficos. O output é apresentado a seguir:

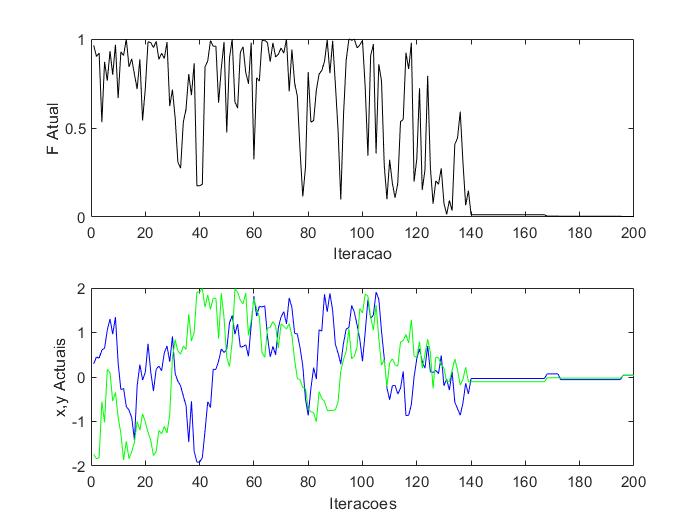


Figura 3 - Posições atuais da particula

No Grafico 1, ?????

No gráfico 2, obtiveram-se os resultados pretendidos. Neste caso, o x atual (a azul) começa em cerca de 0.2, assume valores superiores e inferiores ao longo do tempo, com variações cada vez menores e estabiliza perto do 0. O y actual (a verde), começa em cerca de -1,7, a amplitude de variação da sua posição vai diminuindo à medida que o seu valor se aproxima do 0.

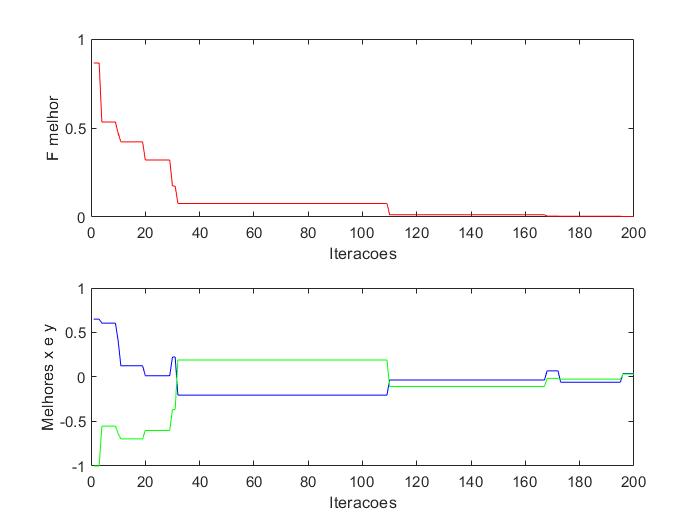


Figura 4 - Melhores posições alcançadas pela partícula

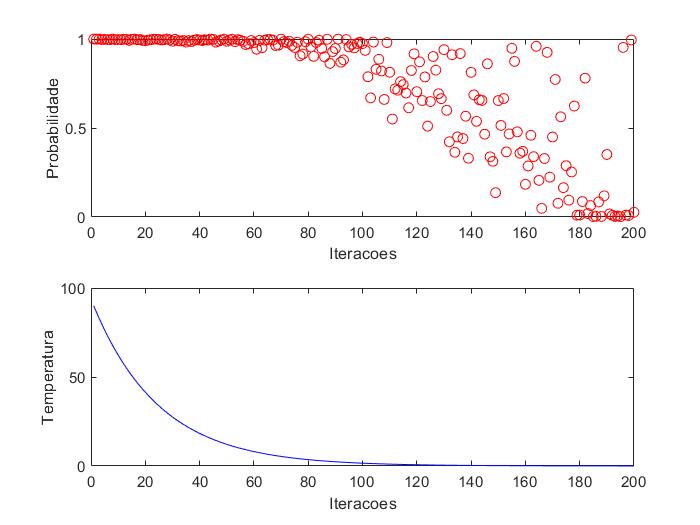


Figura 5 - Variação da probabilidade e temperatura

## Notas Finais

Apresentar notas finais do trabalho.

### Bibliografia

Apresentar uma lista numerada de todas as referências utilizadas incluindo URLs com data de consulta.