

Exercício 02

Renan Salles de Freitas
CPE 723 - Otimização Natural

19 de março de 2018

Exercício 1.a. Temos que a distribuição de probabilidade de $X(1)$ é:

$$\mathbf{p}_1 = M\mathbf{p}_0$$

E ainda:

$$\begin{aligned}\mathbf{p}_2 &= M\mathbf{p}_1 = M^2\mathbf{p}_0 \\ \mathbf{p}_n &= M^n\mathbf{p}_0\end{aligned}$$

Portanto:

$$\begin{aligned}\mathbf{p}_3 &= M^3\mathbf{p}_0 \\ \mathbf{p}_3 &= \begin{bmatrix} 0.3328 \\ 0.3344 \\ 0.3328 \end{bmatrix}\end{aligned}$$

Exercício 1.b. Supondo que estamos no estado $X(t)$, construímos uma lista com os possíveis próximos estados, considerando a distribuição de probabilidade, conforme a matriz de transição de estados M : $X(t+1) = [X(t) \ 0 \ 1 \ 2]$. Sorteamos um índice de zero a quatro com o MatLab e atualizamos $X(t+1)$. Observe que, dessa forma, a transição para o estado o estado atual sempre possui probabilidade 0.5 e os outros estados possuem probabilidade 0.25.

$$\begin{aligned}X(0) &= 1 \\ \text{list} &= [0 \ 1 \ 2 \ 1] \\ r &= \text{randi}(4) = 3 \\ X(1) &= \text{list}(r) = 2\end{aligned}\tag{1}$$

$$\begin{aligned}X(1) &= 2 \\ \text{list} &= [0 \ 1 \ 2 \ 2] \\ r &= \text{randi}(4) = 3 \\ X(2) &= \text{list}(r) = 2\end{aligned}\tag{2}$$

$$\begin{aligned}
 X(2) &= 2 \\
 \text{list} &= [0 \quad 1 \quad 2 \quad 2] \\
 r &= \text{randi}(4) = 4 \\
 X(3) &= \text{list}(r) = 2
 \end{aligned}
 \tag{3}$$

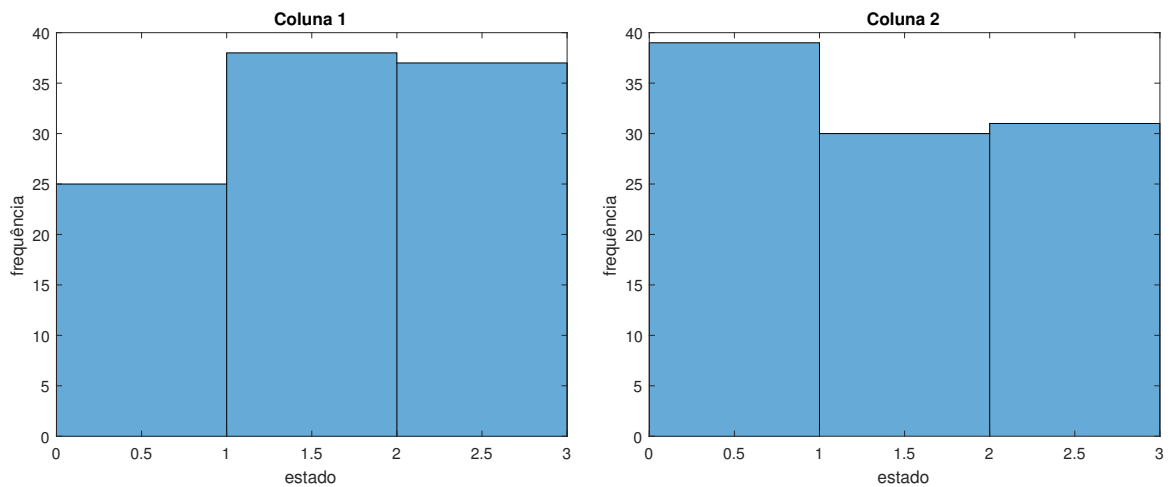
Exercício 1.c. Código MatLab abaixo:

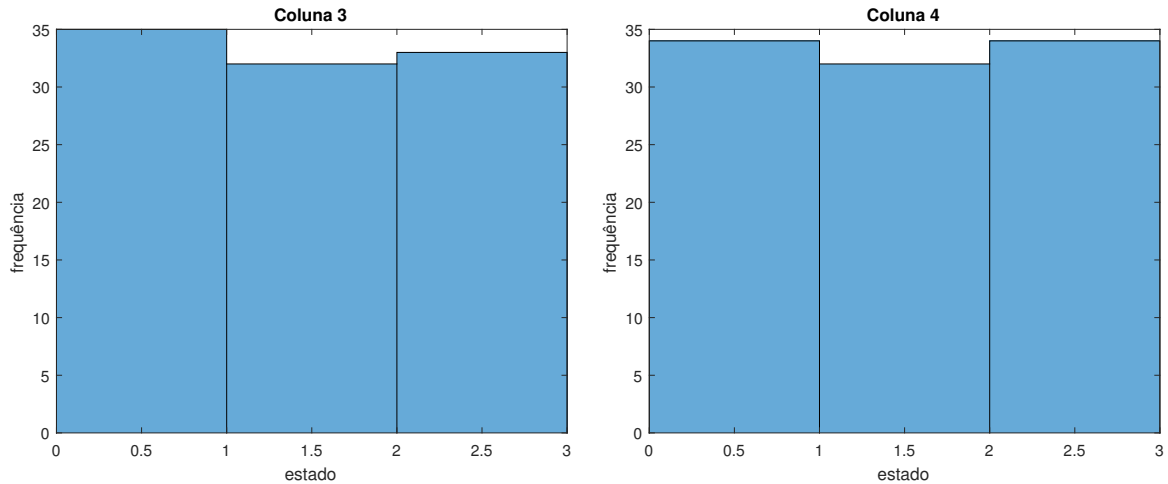
```

1 clear all
2 clc
3
4 init = [0 1 2];
5 n = 100;
6 x = zeros(n,4);
7 for i = 1:n
8     x(i,1) = init(randi(3));
9     for j = 2:4
10        list = [0 1 2 x(i,j-1)];
11        r = randi(4);
12        x(i,j) = list(r);
13    end
14 end

```

Exercício 1.d. Os histogramas estão representados abaixo:





O código MatLab para calcular as probabilidades está abaixo:

```

1 M = [ 0.5 0.25 0.25;
2       0.25 0.5 0.25;
3       0.25 0.25 0.5];
4
5 p0 = [sum(x(:,1)==0)/100 sum(x(:,1)==1)/100 sum(x(:,1)==2)/100]';
6 p1 = [sum(x(:,2)==0)/100 sum(x(:,2)==1)/100 sum(x(:,2)==2)/100]';
7 p2 = [sum(x(:,3)==0)/100 sum(x(:,3)==1)/100 sum(x(:,3)==2)/100]';
8 p3 = [sum(x(:,4)==0)/100 sum(x(:,4)==1)/100 sum(x(:,4)==2)/100]';
9
10 p0g = [1/3 1/3 1/3]';
11 p1g = M*p0g;
12 p2g = M*p1g;
13 p3g = M*p2g;

```

Sabemos que o estado inicial é equiprovável para os três estados :

$$\mathbf{p}_0 = [0.3333 \quad 0.3333 \quad 0.3333]$$

E ainda:

$$\mathbf{p}_1 = M\mathbf{p}_0 = [0.3333 \quad 0.3333 \quad 0.3333]$$

$$\mathbf{p}_n = M^n\mathbf{p}_0 = \mathbf{p}_0 = [0.3333 \quad 0.3333 \quad 0.3333]$$

Calculamos as probabilidades pela frequência do histograma e obtemos:

$$\mathbf{p}_0 = [0.25 \quad 0.38 \quad 0.37]$$

$$\mathbf{p}_1 = [0.39 \quad 0.30 \quad 0.31]$$

$$\mathbf{p}_2 = [0.35 \quad 0.32 \quad 0.33]$$

$$\mathbf{p}_3 = [0.34 \quad 0.32 \quad 0.34]$$

Vale observar que, conforme aumentamos o número de iterações, o estado se aproxima para o estado estacionário $\mathbf{p}_n = [0.333 \quad 0.333 \quad 0.333]$, autovetor da matriz M .

Exercício 2.a.

$$M = \begin{bmatrix} 0 & \frac{1}{4}e^{-3} & \frac{1}{4}e^{-2} & \frac{1}{4}e^{-4} & \frac{1}{4}e^{-1} \\ \frac{1}{4}(3 - e^{-1} - e^{-2} - e^{-3}) & \frac{1}{4} & \frac{1}{4}(2 - e^{-1} - e^{-2}) & \frac{1}{4}e^{-1} & \frac{1}{4} \\ \frac{1}{4}e^{-1} & \frac{1}{4}e^{-2} & \frac{1}{4} & \frac{1}{4}e^{-2} & \frac{1}{4} \\ \frac{1}{4}e^{-2} & \frac{1}{4}e^{-1} & \frac{1}{4}(4 - e^{-1} - e^{-2} - e^{-3} - e^{-4}) & \frac{1}{4}e^{-3} & \frac{1}{4} \\ \frac{1}{4}e^{-2} & \frac{1}{4}e^{-1} & \frac{1}{4}e^{-3} & \frac{1}{4} & \frac{1}{4}(1 - e^{-1}) \end{bmatrix}$$

O código MatLab para calcular as probabilidades está abaixo:

```

1 % Exercício 2a
2 k = (1/4);
3 ex1 = exp(-1);
4 ex2 = exp(-2);
5 ex3 = exp(-3);
6 ex4 = exp(-4);
7
8 M = [0 k*ex3 k*ex2 k*ex4 k*ex1;
9      k k*(3-ex1-ex2-ex3) k k*ex1 k;
10     k k*ex1 k*(2-ex1-ex2) k*ex2 k;
11     k k k k*(4-ex1-ex2-ex3-ex4) k;
12     k k*ex2 k*ex1 k*ex3 k*(1-ex1)];
13
14 %{
15 M =
16
17         0         0.0124         0.0338         0.0046         0.0920
18     0.2500     0.6117         0.2500         0.0920         0.2500
19     0.2500     0.0920         0.3742         0.0338         0.2500
20     0.2500     0.2500         0.2500         0.8572         0.2500
21     0.2500     0.0338         0.0920         0.0124         0.1580
22
23 %}
```

Exercício 2.b. Coonstruimos uma lista com os possíveis estados: $\text{list} = [1 \ 2 \ 3 \ 4 \ 5]$. Utilizamos o MatLab para fornecer um número inteiro randômico (distribuição uniforme) entre 1 e 4: $\text{randi}(4)$. Removemos o estado atual da lista: $\text{list} = [2 \ 3 \ 4 \ 5]$. O número randômico será o índice do elemento que utilizaremos como próximo candidato para estado $X(1)$:

$$\text{randi}(4) = 4$$

$$\text{list} = [2 \ 3 \ 4 \ 5]$$

$$X(1) = \text{list}(4) = 5$$

$$J(5) < J(1)$$

$$X(1) = 5$$

(4)

$$\begin{aligned}
&\text{randi}(4) = 4 \\
&\text{list} = [1 \quad 2 \quad 3 \quad 4] \\
&X(2) = \text{list}(4) = 4 \\
&J(4) < J(5) \\
&X(2) = 4
\end{aligned} \tag{5}$$

$$\begin{aligned}
&\text{randi}(4) = 3 \\
&\text{list} = [1 \quad 2 \quad 3 \quad 5] \\
&X(3) = \text{list}(3) = 3 \\
&\text{rand} = 0.0975 > \frac{1}{4}e^{-0.2/0.1} = 0.0338 \\
&X(3) = 4
\end{aligned} \tag{6}$$

$$\begin{aligned}
&\text{randi}(4) = 2 \\
&\text{list} = [1 \quad 2 \quad 3 \quad 5] \\
&X(4) = \text{list}(2) = 2 \\
&\text{rand} = 0.5469 > \frac{1}{4}e^{-0.1/0.1} = 0.092 \\
&X(4) = 4
\end{aligned} \tag{7}$$

Exercício 2.c. Fazemos $[v, a] = \text{eig}(M)$. O vetor invariante é autovetor correspondente ao autovalor 1, dividido pela soma de seus elemento:

$$v = \begin{bmatrix} 0.0117 \\ 0.2341 \\ 0.0861 \\ 0.6364 \\ 0.0317 \end{bmatrix}$$

Exercício 2.d. Calculando os fatores de Boltzmann, obtemoos:

$$\begin{aligned}
B &= [e^{-5} \quad e^{-2} \quad e^{-3} \quad e^{-1} \quad e^{-4}] \\
B &= [0.0067 \quad 0.1353 \quad 0.0498 \quad 0.3679 \quad 0.0183]
\end{aligned}$$

Dividindo o vetor por sua soma, obtemos:

$$B = [0.0117 \quad 0.2341 \quad 0.0861 \quad 0.6364 \quad 0.0317]$$

Que é o mesmo vetor invariante.

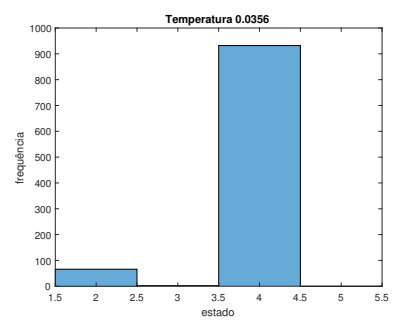
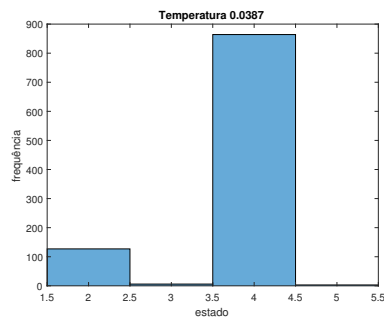
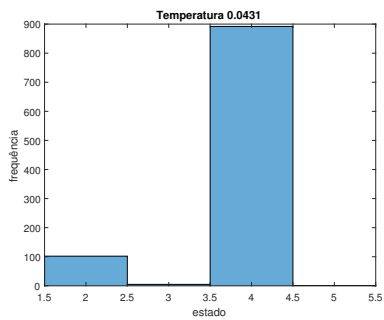
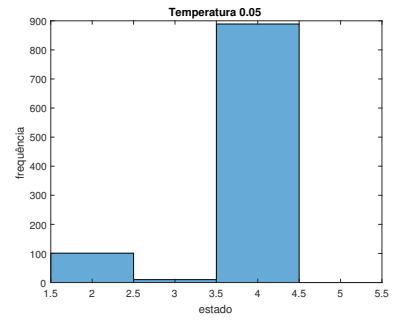
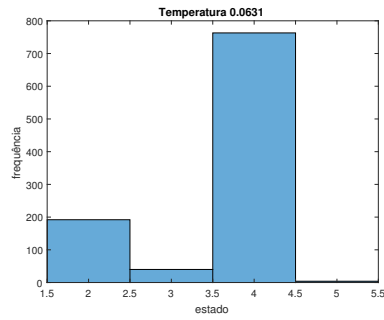
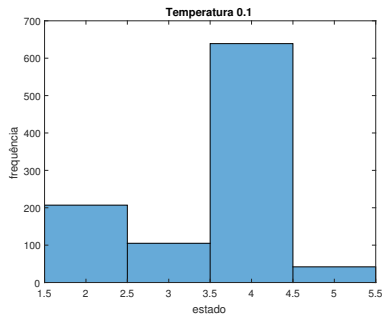
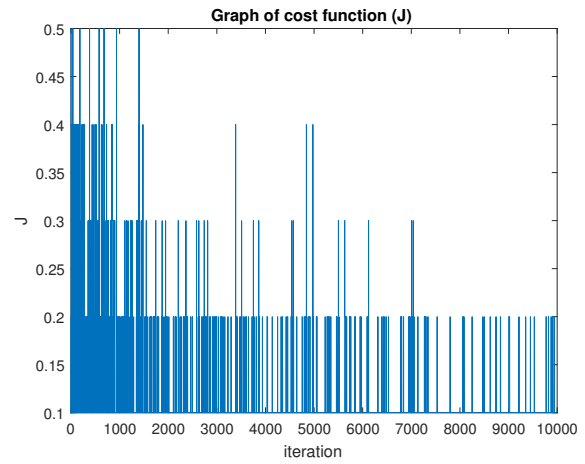
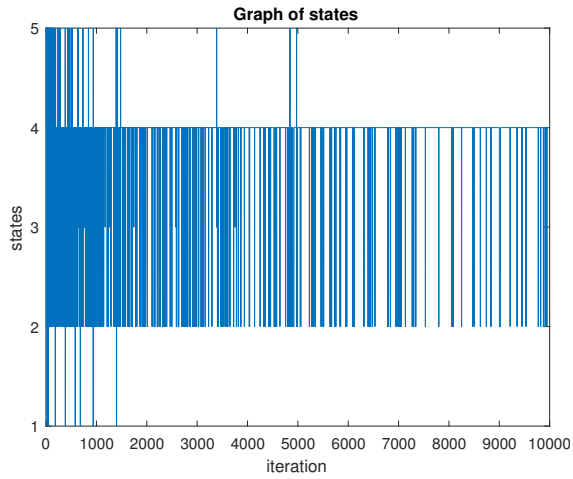
Exercício 2.e. O código MatLab para SA está abaixo:

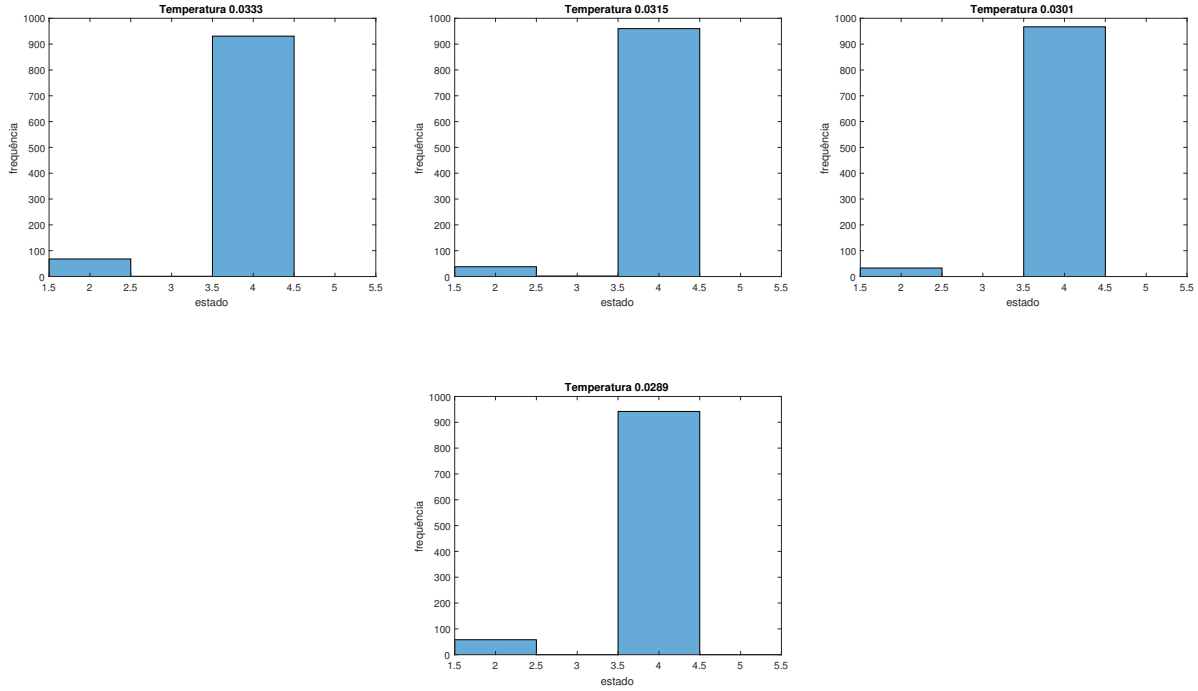
```
1 % Exercício 2e
2
3 clear all
4 clc
5
6 x0 = 1;
7 number_of_iteration = 1000;
8 number_of_temperature_iteration = 10;
9 T = [0.1 0.0631 0.05 0.0431 0.0387 0.0356 0.0333 0.0315 0.0301
      0.0289];
10
11 x = zeros(number_of_iteration * number_of_temperature_iteration, 1);
12 x(1) = x0;
13
14 costs = [0.5 0.2 0.3 0.1 0.4];
15 J = @(n) costs(n);
16
17 B = @(n,t) exp(-n/t);
18
19 counter = 1;
20 counter_temperatures = 1;
21 Jmin = J(x(counter));
22 xmin = x(counter);
23 while counter_temperatures <= number_of_temperature_iteration
24     while counter < number_of_iteration * counter_temperatures
25         states = 1:5;
26         states(x(counter)) = [];
27         r = randi(4);
28         xk = states(r);
29         Jxk = J(xk);
30         dJ = Jxk - J(x(counter));
31         counter = counter + 1;
32         if dJ < 0
33             x(counter) = xk;
34         else
35             a = rand;
36             if B(dJ,T(counter_temperatures)) > a
37                 x(counter) = xk;
38             else
39                 x(counter) = x(counter - 1);
40             end
41         end
42         if Jxk < Jmin
43             Jmin = Jxk;
44             xmin = xk;
45         end
46     end
47 end
```

```

46     end
47     counter_temperatures = counter_temperatures + 1;
48 end
49
50 histogram(x(1:1000),[1 2 3 4 5])

```





As probabilidades foram calculadas e estão abaixo:

$$\begin{aligned}
 v[1] &= [0.007 \quad 0.207 \quad 0.105 \quad 0.639 \quad 0.042] \\
 v[2] &= [0.001 \quad 0.192 \quad 0.04 \quad 0.763 \quad 0.04] \\
 v[3] &= [0 \quad 0.101 \quad 0.01 \quad 0.889 \quad 0] \\
 v[4] &= [0 \quad 0.102 \quad 0.005 \quad 0.892 \quad 0.001] \\
 v[5] &= [0 \quad 0.127 \quad 0.006 \quad 0.864 \quad 0.003] \\
 v[6] &= [0 \quad 0.066 \quad 0.002 \quad 0.932 \quad 0] \\
 v[7] &= [0 \quad 0.068 \quad 0.001 \quad 0.931 \quad 0] \\
 v[8] &= [0 \quad 0.038 \quad 0.002 \quad 0.96 \quad 0] \\
 v[9] &= [0 \quad 0.033 \quad 0 \quad 0.967 \quad 0] \\
 v[10] &= [0 \quad 0.058 \quad 0 \quad 0.942 \quad 0]
 \end{aligned}$$

Inicialmente, a distribuição $v[1]$ se assemelha à probabilidade estacionária (vetor invariante). Porém, conforme resfriamos o sistema (diminuímos a temperatura), o estado converge para o mínimo global.

Exercício 3. Para este exercício vamos utilizar a função Rosenbrock para $N = 10$:

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$$

O mínimo está em $\mathbf{x}_{\min} = \mathbf{1}$ e $J_{\min} = 0$. O código MatLab para SA está abaixo:


```

1  % Exercício 3
2
3  clear all
4  clc
5
6  number_of_variables = 10;
7  number_of_iteration = 5000;
8  number_of_temperature_iteration = 30;
9  epsilon = 0.05;
10 T0 = 1;
11
12 T = T0;
13 x0 = zeros(1, number_of_variables);
14 x = zeros(number_of_iteration * number_of_temperature_iteration, 10);
15 x(1,:) = x0;
16
17 B = @(n,t) exp(-n/t);
18
19 counter = 1;
20 counter_temperatures = 1;
21 Jmin = J(x(counter,:));
22 xmin = x(counter,:);
23 while counter_temperatures <= number_of_temperature_iteration
24     while counter < number_of_iteration * counter_temperatures
25         r = randn(1,number_of_variables);
26         xk = x(counter,:) + epsilon * r;
27         Jxk = J(xk);
28         dJ = Jxk - J(x(counter,:));
29         counter = counter + 1;
30         if dJ < 0
31             x(counter,:) = xk;
32         else
33             a = rand;
34             if B(dJ,T) > a
35                 x(counter,:) = xk;
36             else
37                 x(counter,:) = x(counter - 1,:);
38             end
39         end
40         if Jxk < Jmin
41             Jmin = Jxk;
42             xmin = xk;
43         end
44     end
45     counter_temperatures = counter_temperatures + 1;
46     T = T0 / log2(counter_temperatures + 1);
47 end

```

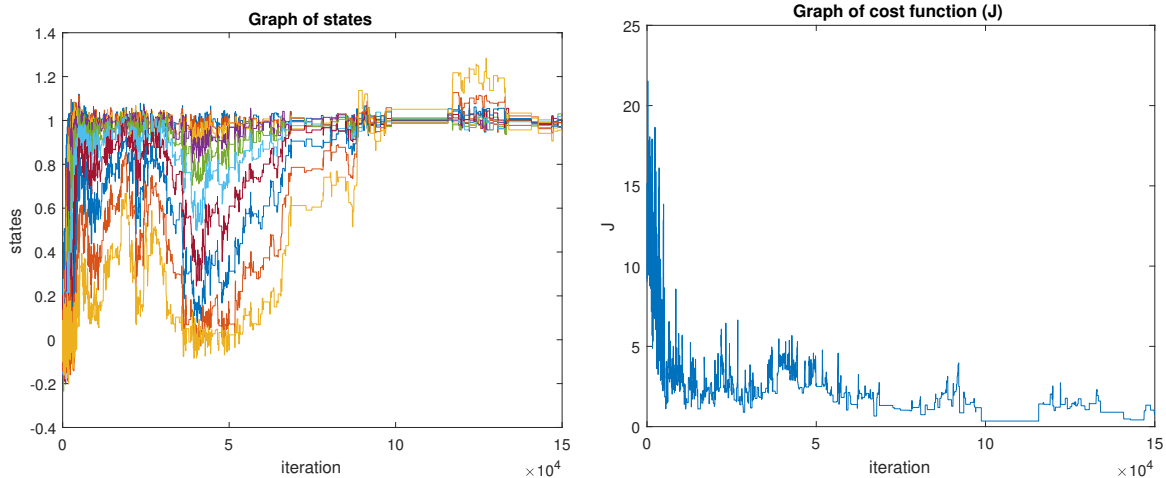
A função custo:

```
1 function y = J(x)
2 n = size(x);
3 y = zeros(n(1),1);
4 for i=1:n(2)-1
5     y = y + 100*(x(:,i+1) - x(:,i).^2).^2 + (1 - x(:,i)).^2;
6 end
```

Encontramos:

$$x_{\min} = \begin{bmatrix} 0.9887 \\ 0.9877 \\ 0.9884 \\ 0.9997 \\ 1.0082 \\ 1.0121 \\ 1.0005 \\ 0.9981 \\ 1.0034 \\ 1.0511 \end{bmatrix}$$

e $J_{\min} = 0.3474$.



Verificou-se que o SA teve certa dificuldade em encontrar o mínimo global pelo grande número de mínimos locais. O algoritmo necessitou de 5000 iterações para temperaturas baixas. Com um número menor de iterações (como 4000) o algoritmo SA não foi capaz de encontrar o mínimo. Além disso, o algoritmo é sensível ao resfriamento (variável Temperatura), pois um resfriamento rápido ou temperaturas iniciais baixas impossibilitaram a convergência.