

Apêndice 3

```
nsim = 1000; % numero de repetições
nint = 5000; % numero de jogadas
p = 100; % tamanho da população
tx = 1; % taxa de substituição

M = [3 0;5 1]; % matriz do jogo
Mr = [0.5 1;-1 0]; % matriz das reputações

R = zeros(p,1); % matriz de reputação.
Per = rand(p,1); % probabilidades de cooperar ou não cooperar de cada indivíduo.
P = zeros(p,1); % vetor de pontos por jogador
%SP = zeros(nint,1); % vetor de pontos por rodada
pes = rand(p,1); % pesos da media ponderada entre as probabilidades de escolher e a probabilidade de cooperar dada por pela

c=[];
vint = (1:nint)';
vestab = zeros(nsim,1);
meanper = zeros(nsim,1);

co = [1,1]; % chute inicial para o ajuste exponencial
coe = [1,1]; % chute inicial para o ajuste exponencial
opts = optimset('Display','off');

for n = 1:nsim

    SP = zeros(nint,1); % vetor de pontos por rodada
    SR = zeros(nint,1);

    tr = randperm(p,tx*p); % substituição da população
    R(tr) = zeros(tx*p,1);
    Per(tr) = rand(tx*p,1);
    P(tr) = zeros(tx*p,1); % vetor de pontos por jogador

    pesa = rand(tx*p,1);

    if (n>1&&c(2)~=0)
        pesa = (1/c(2))*log(c(2).*pesa/c(1)+1); % Distribuição da nova geração
    end

    pes(tr) = pesa;

    for i = 1:nint

        js = randperm(p,2); % escolha dos jogadores 1 e 2

        ea = rand(1,1);
        eb = rand(1,1);
```

```

ta = 1/(1+exp(-0.5*R(js(2)))); %% probabilidade de 1 cooperar pela reputação de 2
tb = 1/(1+exp(-0.5*R(js(1)))); %% probabilidade de 2 cooperar pela reputação de 1

ma = (1-pes(js(1)))*Per(js(1))+(pes(js(1)))*ta; % media ponderada para a probabilidade de 1 cooperar
mb = (1-pes(js(2)))*Per(js(2))+(pes(js(2)))*tb; % media ponderada para a probabilidade de 2 cooperar

ta = ma; % troca de variavel
tb = mb; % troca de variavel

if ((ea<ta)&&(eb<tb))

    P(js) = P(js)+M(1,1);
    R(js) = R(js)+Mr(1,1);

    SR(i+1) = SR(i)+2*Mr(1,1);
    SP(i+1) = SP(i)+2*M(1,1);

elseif((ea>=ta)&&(eb>=tb))

    P(js) = P(js)+M(2,2);
    R(js) = R(js)+Mr(2,2);

    SR(i+1) = SR(i)+2*Mr(2,2);
    SP(i+1) = SP(i)+2*M(2,2);

elseif((ea<ta)&&(eb>=tb))

    P(js(1)) = P(js(1))+M(1,2);
    R(js(1)) = R(js(1))+Mr(1,2);
    P(js(2)) = P(js(2))+M(2,1);
    R(js(2)) = R(js(2))+Mr(2,1);

    SR(i+1) = SR(i)+Mr(1,2)+Mr(2,1);
    SP(i+1) = SP(i)+M(1,2)+M(2,1);

elseif((ea>=ta)&&(eb<tb))

    P(js(1)) = P(js(1))+M(2,1);
    R(js(1)) = R(js(1))+Mr(2,1);
    P(js(2)) = P(js(2))+M(1,2);
    R(js(2)) = R(js(2))+Mr(1,2);

    SR(i+1) = SR(i)+Mr(1,2)+Mr(2,1);
    SP(i+1) = SP(i)+M(1,2)+M(2,1);

end

end

```

```

SR = SR(2:end); SP = SP(2:end);

A = [ones(size(R)) R]; b = A'*P; A = A'*A; coe = A\b;

fun = @(x)(P-x(1).*exp(x(2).*pes)); % ajuste exponencial
coe = lsqnonlin(fun,coe,[],[],opts); % ajuste exponencial

[nind,xind] = hist(pes,10); % ajuste exponencial
yind = nind.*(coe(1)*exp(coe(2)*xind)); % para o ajuste exponencial
fun = @(x)(yind-x(1).*exp(x(2).*xind)); % ajuste exponencial
co = lsqnonlin(fun,co,[],[],opts); % ajuste exponencial
alf = co(2)/(co(1)*(exp(co(2))-1)); % ajuste exponencial
c = [co(1)*alf co(2)]; % ajuste exponencial
vestab(n) = SP(end)/(nint+1); % ajuste exponencial

meanper(n,1) = mean(pes);

end

disp(['Retorno medio por jogada => ' num2str(SP(end)/(nint+1))])
disp(['Reputacao media por jogada => ' num2str(SR(end)/(nint+1))])
disp(' ')

f1 = figure(1);
subplot(2,3,1);
plot(vint,SR./vint)
title('Reputação média por jogada')

subplot(2,3,2);
plot(R,P,'*')
title('Reputação vs Pontos')
hold on
plot([min(R) max(R)],coe(1)+coe(2)*[min(R) max(R)])
hold off

subplot(2,3,3);
plot(vint,SP./vint)
title('Ganho Médio')

subplot(2,3,4);
plot(pes,P,'*')
title('Perfil vs Pontos')

subplot(2,3,5);
plot(xind,yind,'*',linspace(0,max(xind),100),co(1)*exp(co(2).*linspace(0,max(xind),100)))
hold on
plot(linspace(0,max(pes),100),coe(1)*exp(co(2).*linspace(0,max(pes),100)))
hold off

```

```
f2 = figure(2);  
subplot(2,1,1);  
plot(1:nsim,vestab,'--*')  
title('Ganho Médio por Simulação')  
  
subplot(2,1,2);  
plot(1:nsim,meanper)  
title('Perfil Médio por Simulação')
```