MATLAB Assignment 2

ECEN-674

Prepared By,

Mrinmoy Sarkar

Bannar Id: 950-363-260

**MATLAB output:**

****

**A1 A0 D0 D1 D2 D3 Action Total No. of**

**(#=7) strength Copy**

0 0 0 7 7 7 0 5227 19

0 0 1 7 7 7 1 5198 23

0 1 7 0 7 7 0 5155 24

0 1 7 1 7 7 1 10207 26

1 0 7 7 0 7 0 10321 27

1 0 7 7 1 7 1 5168 22

1 1 7 7 7 0 0 4900 25

1 1 7 7 7 1 1 5003 25

0 0 0 7 0 7 0 124 7

1 1 0 7 7 7 0 0 1

0 0 7 1 7 7 1 0 1

1 1 7 0 7 7 0 0 7

0 1 7 0 7 7 1 0 3

1 0 7 7 7 0 0 0 5

0 0 0 7 7 0 0 0 6

1 1 7 7 7 7 1 0 5

0 0 0 0 7 7 0 0 1

1 0 0 7 7 7 0 0 6

0 1 7 7 1 7 1 0 3

1 0 7 1 7 7 1 0 3

1 1 7 7 0 7 0 0 6

0 0 1 7 7 1 1 0 4

0 1 7 1 7 1 1 0 2

0 0 0 7 7 7 1 0 5

0 1 7 0 7 1 1 0 4

1 0 7 7 1 7 0 0 3

0 0 7 7 1 7 1 0 3

0 0 1 7 0 7 0 0 2

0 1 7 0 1 7 1 0 5

1 0 7 7 1 0 0 0 4

0 0 1 7 7 7 0 0 4

1 1 7 7 7 7 0 0 6

0 1 7 1 7 0 0 0 5

1 0 7 7 7 1 1 0 5

1 0 7 0 7 7 0 0 4

0 0 7 7 7 1 1 0 1

0 1 7 7 7 1 1 0 6

0 0 7 7 0 7 0 0 9

1 0 7 7 7 7 0 0 7

0 1 7 7 7 0 0 0 2

1 0 7 7 0 1 1 0 1

1 0 7 7 1 1 1 0 4

0 1 7 7 0 7 0 0 3

1 0 1 7 7 7 1 0 5

1 0 7 7 7 7 1 0 5

0 0 1 7 7 0 0 0 2

0 0 0 7 7 1 1 0 5

0 1 7 1 1 7 1 0 1

0 1 7 1 0 7 0 0 3

1 0 7 7 0 0 0 0 1

0 0 1 7 1 7 1 0 1

0 1 1 7 7 7 1 0 1

0 0 0 1 7 7 1 0 2

0 0 0 7 1 7 1 0 2

0 1 1 0 0 0 0 0 3

1 1 7 7 1 7 1 0 3

0 1 7 0 1 1 0 0 4

0 1 7 7 7 7 1 0 1

1 1 0 1 0 0 0 0 1

0 0 1 0 1 1 0 0 1

1 0 7 7 0 7 1 0 1

0 0 1 1 1 0 1 0 1

0 0 0 0 0 7 0 0 1

1 1 7 0 1 7 0 0 1

1 0 7 0 1 7 0 0 1

1 1 7 1 7 7 1 0 2

0 1 0 0 0 0 1 0 1

0 1 7 0 7 0 0 0 1

0 1 7 0 1 1 1 0 1

0 1 7 0 0 7 0 0 3

0 0 1 0 0 7 0 0 1

0 1 7 0 1 7 0 0 2

0 0 0 0 7 1 0 0 1

1 0 0 0 1 1 1 0 1

1 0 7 0 1 0 0 0 1

1 0 7 1 1 0 0 0 1

0 0 0 1 1 7 1 0 1

1 1 1 7 7 1 1 0 1

0 1 7 0 0 1 1 0 1

1 0 1 7 1 1 0 0 1

1 0 7 0 0 0 0 0 1

**MATLAB Code:**

|  |
| --- |
| clear all;  close all;      A = 2;  D = 2^A;  Out = 1;    dec = (0:(2^(A+D)-1))';  bin = de2bi(dec,'left-msb');  env = zeros(size(bin,1),size(bin,2)+1);  env(1:size(bin,1),1:size(bin,2)) = bin;  for i=1:D  ind1 = (i-1)\*2^D+1;  ind2 = i\*2^D;  env(ind1:ind2,end)=bin(ind1:ind2,A+i);  end  env = env(randperm(size(env, 1)), :);      n = 400;    pop = zeros(n,A+D+Out+1);  S=100;  R=1000;  pop(:,end) = pop(:,end) + S;  hash = 7;    alphabet=[0,1,hash];    for i=1:n  for j=1:A+D+Out  if j==A+D+Out  pop(i,j) = alphabet(randi(2,1));  else  pop(i,j) = alphabet(randi(3,1));  end  end  end    total\_ite = 60000;  Cext = 0.005;  Ctax = 0.85;  Cbid = 0.1;  C = 8;  ite = 0;  Pc = 0.6;  Pm = 0.001;  total\_corect = 0;  indx1=1;  while ite <= total\_ite  if mod(ite,10000)==0  env = env(randperm(size(env, 1)), :);  end  for i=1:size(env,1)  msg = env(i,:);  m = match(msg,pop,hash);  [pop,corect] = clearinghouse(pop,m,Cext,Ctax,Cbid,R,msg,hash);  ite = ite+1;  total\_corect = total\_corect + corect;  if mod(ite,25)==0  pop = ga(pop,Pc,Pm,hash);  end  if mod(ite,100)==0  avg\_score(indx1) = total\_corect;  percent\_hash(indx1) = count\_hash(pop,hash);  Tsol(indx1) = noOfCorrectSol(pop,hash);  indx1 = indx1+1;  total\_corect = 0;  end  if mod(ite,5000)==0  sol\_count = count\_sol(pop);  end  end  end    figure(1)  subplot(311)  plot(avg\_score)  title('average score')  subplot(312)  plot(percent\_hash)  title('percent hash')  subplot(313)  plot(Tsol)  title('Tsol') |

|  |
| --- |
| function y = match(msg,pop,hash)  n = size(pop,1);  y = ones(n,1);  for j=1:n  classifier = pop(j,:);  for i=1:length(msg)-1  if msg(i) ~= classifier(i) && classifier(i) ~= hash  y(j) = 0;%no match  break;  end  end  end  end |

|  |
| --- |
| function [y,correct] = clearinghouse(pop,m,Cext,Ctax,Cbid,R,msg,hash)  correct = 0;  n = size(pop,1);  Ebid = zeros(n,1);  for i=1:n  if m(i) == 1  Ebid(i) = pop(i,end)\*Cbid + randn - 0.5;  end  end  [ma, mai]=max(Ebid);  for i=1:n  if (i==mai)  if pop(i,length(msg)) == msg(end)  pop(i,end) = (1-Cext-Ctax-Cbid)\*pop(i,end)+R\*(1+8\*noOfHash(pop(i,:),hash)/6);  correct = 1;  else  pop(i,end) = (1-Cext-Ctax-Cbid)\*pop(i,end);  end  elseif m(i) == 1  pop(i,end) = (1-Cext-Ctax)\*pop(i,end);  else  pop(i,end) = (1-Cext)\*pop(i,end);  end  end  y = pop;  end |

|  |
| --- |
| function n=noOfHash(classifier,hash)  n=0;  for i=1:length(classifier)-2  if hash == classifier(i)  n = n+1;  end  end  end |

|  |
| --- |
| function new\_pop = ga(pop,pc,pm,hash)  n = size(pop,1);  new\_pop = pop;  f = pop(:,end);  f = f/sum(f);  f = cumsum(f);  tem\_pop = zeros(2,size(pop,2));  k=1;  tindx=zeros(2,1);    tem = find((rand<=f)==1);  i1 = tem(1);  parent1 = pop(i1,:);  while 1  tem = find((rand<=f)==1);  i2 = tem(1);  parent2 = pop(i2,:);  if i1~=i2 % && parent1(end-1) == parent2(end-1)  break;  end  end  child(1,:) = parent1;  child(2,:) = parent2;  if rand <= pc  l = size(pop,2)-2;  s = randi(l-1,1,1);  child(1,:) = [parent1(1:s) parent2(s+1:end)];  child(2,:) = [parent2(1:s) parent1(s+1:end)];  child(1,end) = (1/3)\*(parent1(end)+parent2(end));  child(2,end) = (1/3)\*(parent1(end)+parent2(end));  new\_pop(i1,end)=(2/3)\*new\_pop(i1,end);  new\_pop(i2,end)=(2/3)\*new\_pop(i2,end);  end    child = mutation(child,pm,hash);    ff = new\_pop(:,end);  [mm,i1] = min(ff);  ff(i1) = ff(i1)+max(ff);  [mm,i2] = min(ff);    new\_pop(i1,:)=child(1,:);  new\_pop(i2,:)=child(2,:);  end |

|  |
| --- |
| function new\_pop = mutation(pop,pm,hash)  l = size(pop,2)-2;  alphabet1=[1,hash];  alphabet2=[0,hash];  alphabet3=[0,1];    for i=1:size(pop,1)  for j=1:l  if rand <= pm  if pop(i,j) == 0  pop(i,j) = alphabet1(randi(2,1));  elseif pop(i,j) == 1  pop(i,j) = alphabet2(randi(2,1));  else  pop(i,j) = alphabet3(randi(2,1));  end  end  end  end  new\_pop=pop;  end |

|  |
| --- |
| function y = noOfCorrectSol(pop,hash)    n=size(pop,1);  y = 0;  for i=1:n  classifier = pop(i,:);  sol = 0;  if classifier(1) == 0 && classifier(2) == 0 && classifier(3)==classifier(7)  sol = 1;  elseif classifier(1) == 0 && classifier(2) == 1 && classifier(4)==classifier(7)  sol = 1;  elseif classifier(1) == 1 && classifier(2) == 0 && classifier(5)==classifier(7)  sol = 1;  elseif classifier(1) == 1 && classifier(2) == 1 && classifier(6)==classifier(7)  sol = 1;  elseif classifier(1) == 0 && classifier(2) == hash && classifier(3)==classifier(7) && classifier(4)==classifier(7)  sol = 1;  elseif classifier(1) == hash && classifier(2) == 0 && classifier(3)==classifier(7) && classifier(5)==classifier(7)  sol = 1;  elseif classifier(1) == 1 && classifier(2) == hash && classifier(5)==classifier(7) && classifier(6)==classifier(7)  sol = 1;  elseif classifier(1) == hash && classifier(2) == 1 && classifier(4)==classifier(7) && classifier(6)==classifier(7)  sol = 1;  elseif classifier(1) == hash && classifier(2) == hash && classifier(3)==classifier(7) && classifier(4)==classifier(7) && classifier(5)==classifier(7) && classifier(6)==classifier(7)  sol = 1;  end  y = y+sol;  end    end |

|  |
| --- |
| function y=count\_sol(pop)    k=1;  while ~isempty(pop)  classifier = pop(1,:);  s = sum(classifier(1:7)==pop(:,1:7),2);  indx = find(s==7);  total\_s = sum(pop(indx,end));  total\_copy = length(indx);  pop(indx,:)=[];  classifier(end) = total\_s;  y(k,:)=[classifier total\_copy];  k = k+1;  end  y=sortrows(round(y),8,'descend');    end |

|  |
| --- |
| function y=count\_hash(pop,hash)  n=size(pop,1);  total\_hash = 0;  for i=1:n  classifier = pop(i,:);  total\_hash = total\_hash + noOfHash(classifier,hash);  end  y = (total\_hash\*100)/(6\*n);  end |