MATLAB Assignment 2

ECEN-674

Prepared By,

Mrinmoy Sarkar

Bannar Id: 950-363-260

MATLAB output:

>> lcs

trial no.: 1

trial no.: 2

trial no.: 3

trial no.: 4

trial no.: 5

trial no.: 6

trial no.: 7

trial no.: 8

trial no.: 9

trial no.: 10

000### 0 5262 14

001### 1 3316 15

01#1## 1 4640 27

01#0## 0 3433 33

10##1# 1 4564 24

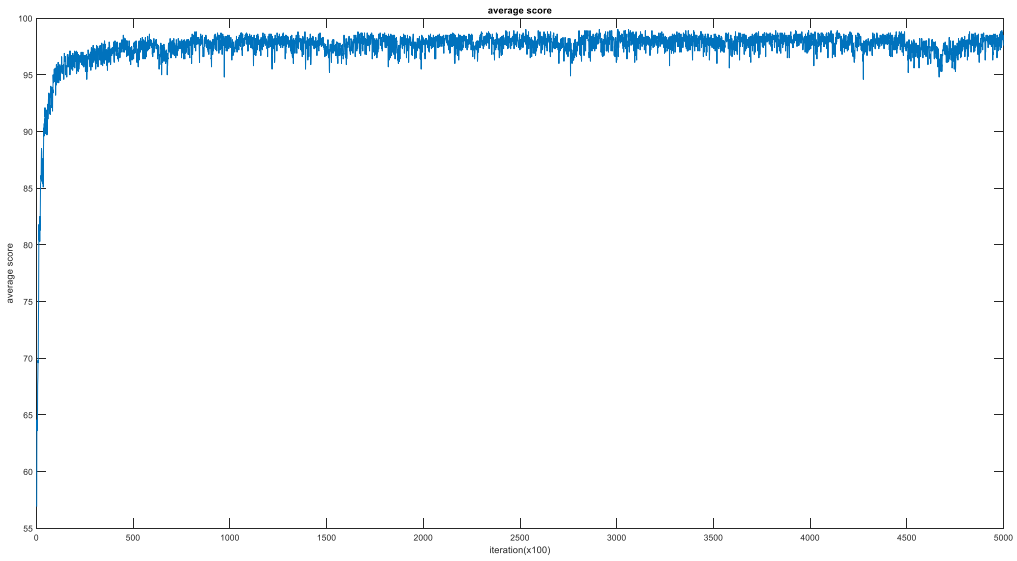
10##0# 0 4114 23

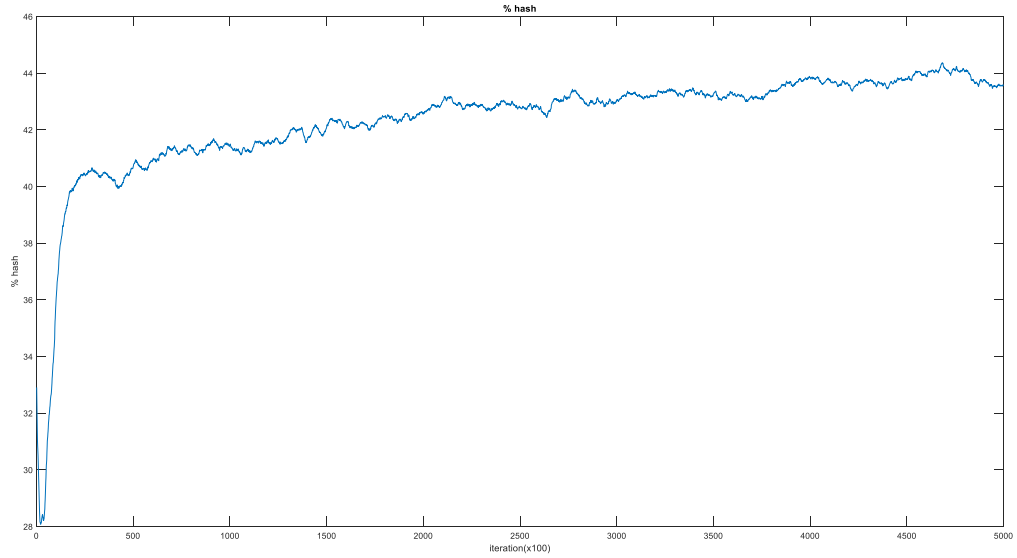
11###1 1 3646 19

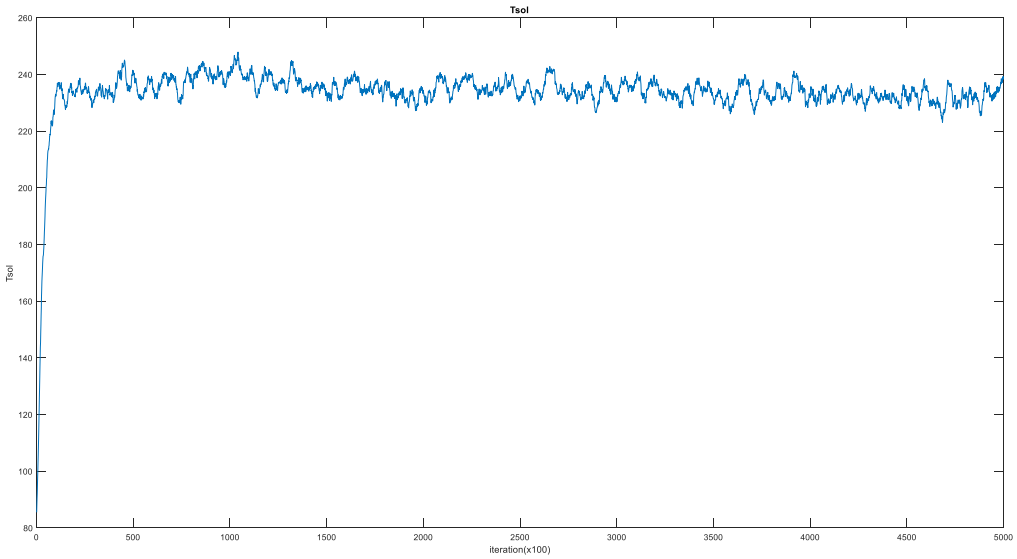
11###0 0 1129 22

Classifier	Action	Total strength	No. of copy
000###	0	5262	14
001###	1	3316	15

01#1##	1	4640	27
01#0##	0	3433	33
10##1#	1	4564	24
10##0#	0	4114	23
11###1	1	3646	19
11###0	0	1129	22







MATLAB Code:

```
clear all;
close all;

A = 2;
D = 2^A;
Out = 1;
n = 400;
l=A+D+Out;
S=100;
R=1000;
hash = 7;
total_ite = 500000;
Cext = 0.05;
Ctax = 0.8;
Cbid = 0.1;
C = 8;
Pc = 0.6;
Pm = 0.001;
no_of_trial = 10;
```

```
env = getenvironment(A,D);
init_pop = generate_pop(n,l,hash,S);
pop = init_pop;
for trial=1:no_of_trial
    pop = init_pop;%generate_pop(n,l,hash,S);
    ite=0;
    indx1=1;
    total_corect = 0;
    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_score1(indx1) = total_corect;
                percent_hash1(indx1) =
count_hash(pop,hash);
                Tsol1(indx1) =
noOfCorrectSol(pop,hash);
                indx1 = indx1+1;
                total_corect = 0;
            end
            if mod(ite,5000)==0
                sol_count = count_sol(pop);
            end
        end
    end
    if trial == 1
        avg_score = avg_score1;
        percent_hash = percent_hash1;
        Tsol = Tsol1;
```

```
    else
        avg_score = avg_score + avg_score1;
        percent_hash = percent_hash + percent_hash1;
        Tsol = Tsol + Tsol1;
    end
    fprintf("trial no.: %d \n", trial);
end
sol_count = count_sol(pop);
final_solution = sol_count(1:8,:);
final_solution(:,end+1) =
bi2de(final_solution(:,1:2),'left-msb');
final_solution = sortrows(final_solution,10);
for i=1:8
    for j=1:9
        if final_solution(i,j) == hash
            fprintf('#')
        else
            if j==7 || j==8 || j==9
                fprintf('\t')
            end
            fprintf('%d',final_solution(i,j))
        end
    end
    fprintf('\n')
end

figure(1)
plot(avg_score/no_of_trial)
title('average score')
xlabel('iteration(x100)')
ylabel('average score')
figure(2)
plot(percent_hash/no_of_trial)
title('% hash')
xlabel('iteration(x100)')
ylabel('% hash')
figure(3)
plot(Tsol/no_of_trial)
title('Tsol')
xlabel('iteration(x100)')
```

```
ylabel('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) %|| (sum(pop(i,1:end-
1)==pop(mai,1:end-1)) == length(pop(mai,1:end-1)))
        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 new_pop = ga(pop,pc,pm,hash)
new_pop = pop;
[p1,p2] = select(pop);
child = cross(pop,pc,p1,p2);
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,:);
new_pop(i1,end)=(2/3)*new_pop(p1,end);
new_pop(i2,end)=(2/3)*new_pop(p2,end);
end
```

```
function [p1,p2]=select(pop)
f = pop(:,end);
f = f/sum(f);
f = cumsum(f);
tem = find((rand<=f)==1);
p1 = tem(1);
while 1
    tem = find((rand<=f)==1);
    p2 = tem(1);
    if p1~=p2
        break;
    end
end
end
```

```
function new_pop = cross(pop,pc,p1,p2)
l = size(pop,2)-2;
k = randi(l-1,1,1);
new_pop = pop([p1,p2],:);
if rand <= pc
    new_pop(1,:) = [pop(p1,1:k) pop(p2,k+1:end)];
    new_pop(2,:) = [pop(p2,1:k) pop(p1,k+1:end)];
end
new_pop(1,end) = (1/3)*(pop(p1,end)+pop(p2,end));
new_pop(2,end) = (1/3)*(pop(p1,end)+pop(p2,end));
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=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
```

```
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 pop = generate_pop(n,l,hash,S)
pop = zeros(n,l+1);
pop(:,end) = pop(:,end) + S;
alphabet=[0,1,hash];
for i=1:n
    for j=1:l
        if j==l
            pop(i,j) = alphabet(randi(2,1));
        else
            pop(i,j) = alphabet(randi(3,1));
        end
    end
end
end
```

```
function env = getenvironment(A,D)
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)), :);
end
```

```
function msg=env()
msg=round(rand(1,7));
msg(end) = msg(3+bi2de(msg(1:2),'left-msb'));
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 n=noOfHash(classifier,hash)
n=0;
for i=1:length(classifier)-2
    if hash == classifier(i)
        n = n+1;
    end
end
end
```