**MATLAB Assignment 1**

ECEN-674

Prepared By,

Mrinmoy Sarkar

Bannar Id: 950-363-260

Date: 2/22/2018

**Solution 1:**

**MATLAB OUTPUT:**

****

**Remark:** The issue concerning the relative importance of mutation and crossover can be viewed at a higher level. Mutation serves to create random diversity in the population, while crossover serves as an accelerator that promotes emergent behavior from components. The meta-issue, then, is the relative importance of diversity and construction. For the GA community, this is also related to the balance between exploration and exploitation. So, both these operations together will increase the probability of finding optimal solution.

**MATLAB Code:**

|  |
| --- |
| %% solution1.m  % author: Mrinmoy sarkar  % date: 2/16/2018  % email: msarkar@aggies.ncat.edu      Pcc=[0 .2 .5 .7 .9 1];  Pmm=[0 .001 .005 .009 .01 .09];  for kk=1:2  for ii=1:length(Pcc)    pop\_size = 100;  str\_len = 20;  pop = round(rand(pop\_size,str\_len));  if kk==1  pc = Pcc(ii);  pm = 0.001;  else  pc=0.7;  pm=Pmm(ii);  end    gen\_no = 20;    fit = @fitness1;    fit\_of\_gen = zeros(pop\_size,gen\_no);    for i=1:gen\_no  fit\_of\_gen(:,i)=fit(pop);  pop = reproduction(pop,fit);  pop = cross(pop,pc);  pop = mutation(pop,pm);  end    avg\_fit = mean(fit\_of\_gen);  max\_fit = max(fit\_of\_gen);  if kk==1  subplot(121)  else  subplot(122)  end  plot(avg\_fit)  hold on    %plot(max\_fit)  %hold on  ylim([min(avg\_fit)-3,max(max\_fit)+3])  %maxfit = max(avg\_fit)  end  if kk==1  xlabel('generation')  ylabel('average fitness')  legend('Pc=0,Pm=.001','Pc=.2,Pm=.001','Pc=.5,Pm=.001','Pc=.7,Pm=.001','Pc=.9,Pm=.001','Pc=1,Pm=.001')  else  xlabel('generation')  ylabel('average fitness')  legend('Pc=.7,Pm=0','Pc=.7,Pm=.001','Pc=.7,Pm=.005','Pc=.7,Pm=.009','Pc=.7,Pm=.01','Pc=.7,Pm=.09')  end  end |

**Solution 2:**

**MATLAB OUTPUT:**

****

**MATLAB Code:**

|  |
| --- |
| %% solution2.m  % author: Mrinmoy sarkar  % date: 2/16/2018  % email: msarkar@aggies.ncat.edu    Pcc=[0 .2 .5 .7 .9 1];  Pmm=[0 .001 .005 .009 .01 .09];  popSize=[100 500 1000 1500 2000 3000];  for kk=1:3  for ii=1:length(Pcc)  if kk==1  pop\_size = popSize(ii);  pc = 0.7;  pm = 0.001;  elseif kk==2  pop\_size = 100;  pc = Pcc(ii);  pm = 0.001;  else  pop\_size = 100;  pc = 0.7;  pm = Pmm(ii);  end  str\_len = 20;  pop = round(rand(pop\_size,str\_len));    gen\_no = 100;    fit = @fitness2;    fit\_of\_gen = zeros(pop\_size,gen\_no);    for i=1:gen\_no  fit\_of\_gen(:,i)=fit(pop);  pop = reproduction(pop,fit);  pop = cross(pop,pc);  pop = mutation(pop,pm);  end    avg\_fit = mean(fit\_of\_gen);  max\_fit = max(fit\_of\_gen);  if kk==1  subplot(321)  plot(avg\_fit)  xlabel('generation')  ylabel('average fitness')  title('max fitness vs gen for varying Pop size')  hold on  subplot(322)  plot(max\_fit)  xlabel('generation')  ylabel('maximum fitness')  title('max fitness vs gen for varying pop size')  hold on  elseif kk==2  subplot(323)  plot(avg\_fit)  xlabel('generation')  ylabel('average fitness')  title('avg fitness vs gen for varying Pc')  hold on  subplot(324)  plot(max\_fit)  xlabel('generation')  ylabel('maximum fitness')  title('max fitness vs gen for varying Pc')  hold on  else  subplot(325)  plot(avg\_fit)  xlabel('generation')  ylabel('average fitness')  title('avg fitness vs gen for varying Pm')  hold on  subplot(326)  plot(max\_fit)  xlabel('generation')  ylabel('maximum fitness')  title('max fitness vs gen for varying Pm')  hold on  end  ylim([min(avg\_fit)-.01\*min(avg\_fit),max(max\_fit)+.1\*max(max\_fit)])  %maxfit = max(avg\_fit)  end  end |

**Solution 3:**

**Next-ascent hill climbing(NAHC):**

**MATLAB OUTPUT:**

****

**MATLAB Code:**

|  |
| --- |
| %% nahc.m  % author: Mrinmoy sarkar  % date: 2/16/2018  % email: msarkar@aggies.ncat.edu    no\_of\_run = 10;  no\_of\_iteration = 10000;  pop\_size = 1;  str\_len = 20;  for kkk=1:2  if kkk==1  fit = @fitness1;  else  fit = @fitness2;  end    avg\_fit = zeros(no\_of\_run,100);  for r=1:no\_of\_run  kk = 1;  fit\_mat=zeros(1,no\_of\_iteration);  n=1;  while 1  pop = round(rand(pop\_size,str\_len));  fit\_mat(n) = fit(pop);  if n == no\_of\_iteration  break  end  best\_fit = fit\_mat(n);  if mod(n,100) == 0  avg\_fit(r,kk) = best\_fit;  kk = kk+1;  end  n = n+1;  for i = 1:str\_len  pop(i) = xor(pop(i),1);  fit\_mat(n) = fit(pop);  if fit\_mat(n)>=best\_fit  best\_fit = fit\_mat(n);  else  pop(i) = xor(pop(i),1);  end  if mod(n,100) == 0  avg\_fit(r,kk) = best\_fit;  kk = kk+1;  end  if n == no\_of\_iteration  break  end  n=n+1;  end  end  end    if kkk==1  subplot(211)  plot(mean(avg\_fit))  xlabel('no of generation(equivalent to GA)')  ylabel('average fitness')  title('problem 1')  else  subplot(212)  plot(mean(avg\_fit))  xlabel('no of generation(equivalent to GA)')  ylabel('average fitness')  title('problem 2')  end  end |

**Random-mutation hill climbing(RMHC):**

**MATLAB OUTPUT:**

****

**MATLAB Code:**

|  |
| --- |
| %% rmhc.m  % author: Mrinmoy sarkar  % date: 2/16/2018  % email: msarkar@aggies.ncat.edu    no\_of\_run = 10;  no\_of\_iteration = 10000;  pop\_size = 1;  str\_len = 20;  for kkk=1:2  if kkk==1  fit = @fitness1;  else  fit = @fitness2;  end    avg\_fit = zeros(no\_of\_run,100);  for r=1:no\_of\_run  pop = round(rand(pop\_size,str\_len));  fit\_mat=zeros(1,no\_of\_iteration);  fit\_mat(1) = fit(pop);  best\_fit = fit\_mat(1);  kk = 1;  for n=2:no\_of\_iteration  i = randi(str\_len,1,1);  pop(i) = xor(pop(i),1);  fit\_mat(n) = fit(pop);  if fit\_mat(n)>=best\_fit  best\_fit = fit\_mat(n);  else  pop(i) = xor(pop(i),1);  end  if mod(n,100) == 0  avg\_fit(r,kk) = mean(fit\_mat(n-100+1:n));% best\_fit;  kk = kk+1;  end  end  end    if kkk==1  subplot(211)  plot(mean(avg\_fit))  xlabel('no of generation(equivalent to GA)')  ylabel('average fitness')  title('problem 1')  ylim([0,22])  else  subplot(212)  plot(mean(avg\_fit))  xlabel('no of generation(equivalent to GA)')  ylabel('average fitness')  title('problem 2')  end  end |

**APPENDIX:**

|  |
| --- |
| function f=fitness1(pop)  f=sum(pop,2);  end |

|  |
| --- |
| function f=fitness2(pop)  m = size(pop,1);  l = size(pop,2);  f = zeros(m,1);  for i=1:m  for j=1:l  f(i) = f(i)+pop(i,j)\*2^(l-j);  end  end  end |

|  |
| --- |
| function new\_pop = reproduction(pop,fit)  f = fit(pop);  f = f/sum(f);  f = cumsum(f);  m = size(pop,1);  new\_pop = zeros(size(pop));  for i=1:m  tem = find((rand<=f)==1);  new\_pop(i,:) = pop(tem(1),:);  end |

|  |
| --- |
| function new\_pop = cross(pop,pc)  n = floor(size(pop,1)/2);  l = size(pop,2);  new\_pop = pop;  for i=1:n  k = randi(l-1,1,1);  m = size(pop,1);  mm = 1:m;  i1 = randi(m,1,1);  mm(i1)=[];  i2 = mm(randi(length(mm),1,1));  if rand <= pc  new\_pop(i1,:) = [pop(i1,1:k) pop(i2,k+1:l)];  new\_pop(i2,:) = [pop(i2,1:k) pop(i1,k+1:l)];  end  end  end |

|  |
| --- |
| function new\_pop = mutation(pop,pm)  m = size(pop,1);  l = size(pop,2);  new\_pop = pop;  for i=1:m  for j=1:l  if rand <= pm  new\_pop(i,j) = xor(new\_pop(i,j),1);  end  end  end  end |