# Project 1

Find the minimized and maximized value of the function:

Discuss the effect of:

1. Population Size
2. Bit Length
3. Generation Number
4. Crossover Rate
5. Mutate Rate
6. Elite

## Results and discussion

The effects of different initial parameters on the results of the method are shown in figures as the graphs of the fittest value vs generation. The graphs in each table contain both results for maximizing and minimizing given function, respectively. The effect of different parameter values is discussed below each table.

1. Population Size

| Population Size = 30 | Population Size = 100 |
| --- | --- |
|  |  |
|  |  |

Comparing the graphs it is evident that for both maximizing and minimizing given function larger population size helps the method converge faster.

It is because the space is sampled more densely and therefore genetic algorithm is initialized with rich set of solutions. Such rich set of solutions is advantageous for getting out of the local extrema because crossover selection and mutation selection work require a variety of solutions.

It is also visible in the graphs that the fittest value for almost 100 generations doesn’t change significantly for the smaller population size that is larger population size gives more opportunities to perform crossover and mutation and a bigger part of the space can be explored in the search.

1. Bit Length

| Bit Length= 30 | Bit Length= 60 |
| --- | --- |
|  |  |
|  |  |

Bit length does not seem to have dominating influence on genetic algorithm. Neither the solution quality nor the performance of genetic algorithm seem to differ noticeably when varying bit length.

Larger bit length will naturally give the solution with greater resolution. However since this parameter does not have big impact on the performance, it seems that it is waste of computational time to increase the bit length beyond the resolution required.

1. Generation Number

| Generation number = 100 | Generation number = 300 |
| --- | --- |
| C:\Users\dsp520\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.jpg | C:\Users\dsp520\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2.jpg |
| C:\Users\dsp520\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.jpg | C:\Users\dsp520\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.jpg |

Generation number naturally has no influence on how algorithm acts, however it is very important to select the generation number large enough to let algorithm converge. It is visible in the graph of minimum of function when generation number was set to 100 – it seems that algorithm could perhaps still find lower value because the final result was the fittest value only in three last generations (97-100).

1. Crossover Rate

| Crossover rate = 0.3 | Crossover rate = 0.7 |
| --- | --- |
|  |  |
|  |  |

Larger crossover rate gave better answers for both minimizing and maximizing the function. Crossover helps the algorithm to converge more towards local extremum of the function and therefore helps to obtain more accurate answer (with better resolution). This is the reason why smaller crossover rate takes off the local extremum faster after initialization of genetic algorithm in the graphs above.

1. Mutation Rate

| Mutation rate = 0.02 | Mutation rate = 0.08 |
| --- | --- |
|  |  |
|  |  |

Mutation rate helps the algorithm avoid getting stuck in local extrema. As expected, larger mutation rate helped the algorithm obtain new values and it is visible in the graphs above that new fittest values appeared more often when the mutation rate was larger.

However, in case of maximizing the function, algorithm initialized with the lower mutation rate performed better overall (the final result was significantly better). Therefore, the important conclusion is that while mutation is a mechanism that prevents algorithm to fall into local extrema, it does not guarantee better answers.

1. Elite

| Elite = 0 | Elite = 1 |
| --- | --- |
|  |  |
|  |  |

The comparison of maximizing the function in this case shows the importance of elitism in genetic algorithm. Even though in search of maximized value much larger value (by the order of magnitude of 1) was obtained with no elitism, the overall result is poor, because the value did not make it to new population.

By carrying over the best genes to the next generation, elitism ensures that the quality of solution will not decrease over the generation.

## Code

function PI = GA\_fitfunP2max(chro)

x = chro(1);

y = chro(2);

z = chro(3);

q = (x^2+y^2+z^2+2\*x-10)/(cos(y^2)-sin(z^2)+x\*y);

PI = q;

gene\_no=3;

range=[-40 -50 -60;

40 50 60];

fitfcn='GA\_fitfunP2max';

generation\_no=200;

crossover\_rates = [0.7; 0.9];

mutate\_rates = [0.02; 0.08];

elites = [0; 1];

popu\_sizes = [30; 100];

bit\_lengths = [30; 60];

t=1:generation\_no;

figure\_no = 1;

for ps\_index=1:length(popu\_sizes)

popu\_size = popu\_sizes(ps\_index);

bit\_length = bit\_lengths(2);

crossover\_rate = crossover\_rates(1);

mutate\_rate = mutate\_rates(2);

elite = elites(2);

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['popu\_size = ', num2str(popu\_size)]);

maxfitness=upper;

[maximum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation,1)

maximum\_y=BEST\_popu(generation,2)

maximum\_z=BEST\_popu(generation,3)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:')

title({['Maximum of f(x) for popu size = ', num2str(popu\_size)],['Max f = ',num2str(maximum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

for bls\_index=1:length(bit\_lengths)

popu\_size = popu\_sizes(2);

bit\_length = bit\_lengths(bls\_index);

crossover\_rate = crossover\_rates(1);

mutate\_rate = mutate\_rates(1);

elite = elites(2);

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['bit\_length = ', num2str(bit\_length)]);

maxfitness=upper;

[maximum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation,1)

maximum\_y=BEST\_popu(generation,2)

maximum\_z=BEST\_popu(generation,3)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:')

title({['Maximum of f(x) for bit length = ', num2str(bit\_length)],['Max f = ',num2str(maximum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

gene\_no=3;

range=[-40 -50 -60;

40 50 60];

fitfcn='GA\_fitfunP2max';

generation\_nos=[100 300];

crossover\_rates = [0.7; 0.9];

mutate\_rates = [0.02; 0.08];

elites = [0; 1];

popu\_sizes = [30; 100];

bit\_lengths = [30; 60];

figure\_no = 1;

for gn\_index=1:length(generation\_nos)

popu\_size = popu\_sizes(2);

bit\_length = bit\_lengths(2);

crossover\_rate = crossover\_rates(1);

mutate\_rate = mutate\_rates(2);

elite = elites(2);

generation\_no=generation\_nos(gn\_index);

t=1:generation\_no;

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['generation\_nos = ', num2str(generation\_no)]);

maxfitness=upper;

[minimum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation,1)

maximum\_y=BEST\_popu(generation,2)

maximum\_z=BEST\_popu(generation,3)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:');

title({['Maximum of f(x) for gen. number = ', num2str(generation\_no)],['Max f = ',num2str(minimum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

popu\_size=100;

bit\_length=60;

gene\_no=3;

range=[-40 -50 -60;

40 50 60];

fitfcn='GA\_fitfunP2max';

generation\_no=200;

crossover\_rates = [0.3; 0.7];

mutate\_rates = [0.02; 0.08];

elites = [0; 1];

t=1:generation\_no;

figure\_no = 1;

for cr\_index=1:length(crossover\_rates)

crossover\_rate = crossover\_rates(cr\_index);

mutate\_rate = mutate\_rates(2);

elite = elites(2);

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['CR = ', num2str(crossover\_rate)]);

maxfitness=upper;

[maximum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation,1)

maximum\_y=BEST\_popu(generation,2)

maximum\_z=BEST\_popu(generation,3)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:')

title({['Maximum of f(x) for CR = ', num2str(crossover\_rate)],['Max f = ',num2str(maximum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

for mr\_index=1:length(mutate\_rates)

mutate\_rate = mutate\_rates(mr\_index);

crossover\_rate = crossover\_rates(1);

elite = elites(2);

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['MR = ', num2str(mutate\_rate)]);

maxfitness=upper;

[maximum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:')

title({['Maximum of f(x) for MR = ', num2str(mutate\_rate)],['Max f = ',num2str(maximum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

for e\_index=1:length(elites)

mutate\_rate = mutate\_rates(2);

crossover\_rate = crossover\_rates(1);

elite = elites(e\_index);

[popu, popu\_real, fitness, upper, average, lower, BEST\_popu]...

=GA\_genetic(popu\_size, bit\_length, gene\_no, range, fitfcn, ...

generation\_no, crossover\_rate, mutate\_rate, elite);

disp(['Elite = ', num2str(elite)]);

maxfitness=upper;

[maximum\_f,generation]=max(maxfitness)

maximum\_x=BEST\_popu(generation,1)

maximum\_y=BEST\_popu(generation,2)

maximum\_z=BEST\_popu(generation,3)

figure(figure\_no);

figure\_no = figure\_no + 1;

plot(t,maxfitness,'\*:')

title({['Maximum of f(x) for Elite = ', num2str(elite)],['Max f = ',num2str(maximum\_f)],['Generation no = ',num2str(generation)]});

xlabel('Generation')

ylabel('f(x)')

saveas(gcf,strcat(num2str(figure\_no-1),'.jpg'));

end

Subroutines for minimizing the function are analogical to the subroutines above.