

Jack Gallagher
Artificial Intelligence
3/28/2024

Assignment 5: Optimum Design for an I-beam

Goal: use the GA package in Google Colab (R) to perform multi-objective engineering design optimization for a load-bearing I-beam

To start off my analysis, I decided to run both functions individually in google colab through Genetic Algorithm in order to determine the fitness value individually. After running the function for the cross section area, I noticed that it tended to favor lower x values and that its fitness value would normally fall between -50 and -100. In the following specific instance, figure 1 showcases a fitness value of -62.11.

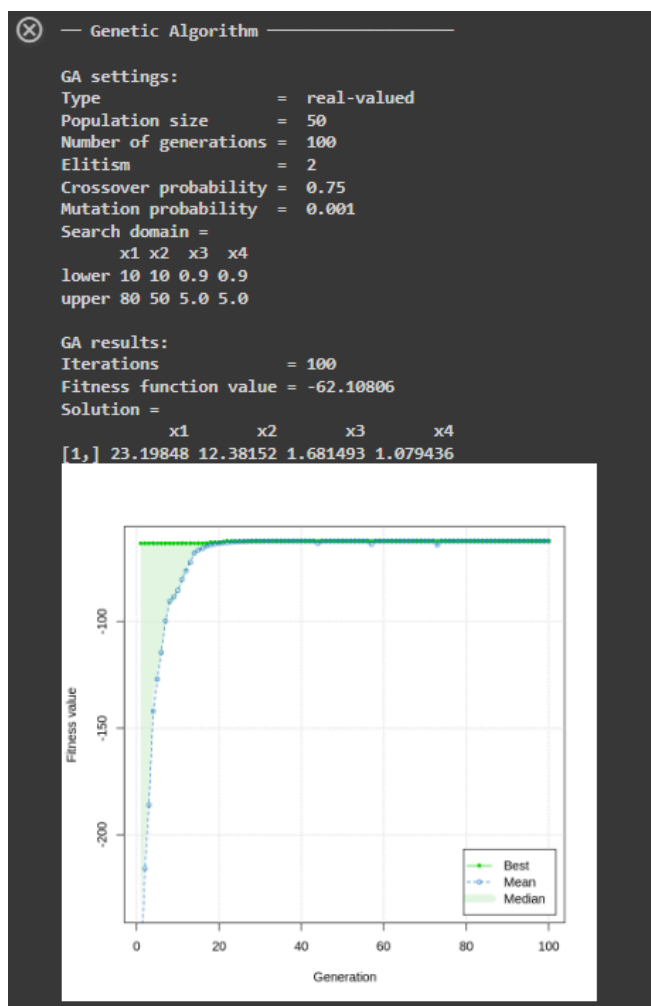


Figure 1: Cross Section Area genetic algorithm results (function 1)

When tested in the genetic algorithm, the Static Deflection held an opposite effect on x values which was to be expected. These values tended to favor the higher x values and held a fitness line value between 0 and -15. Figure 2 showcases a fitness line of -.008 with low x values.

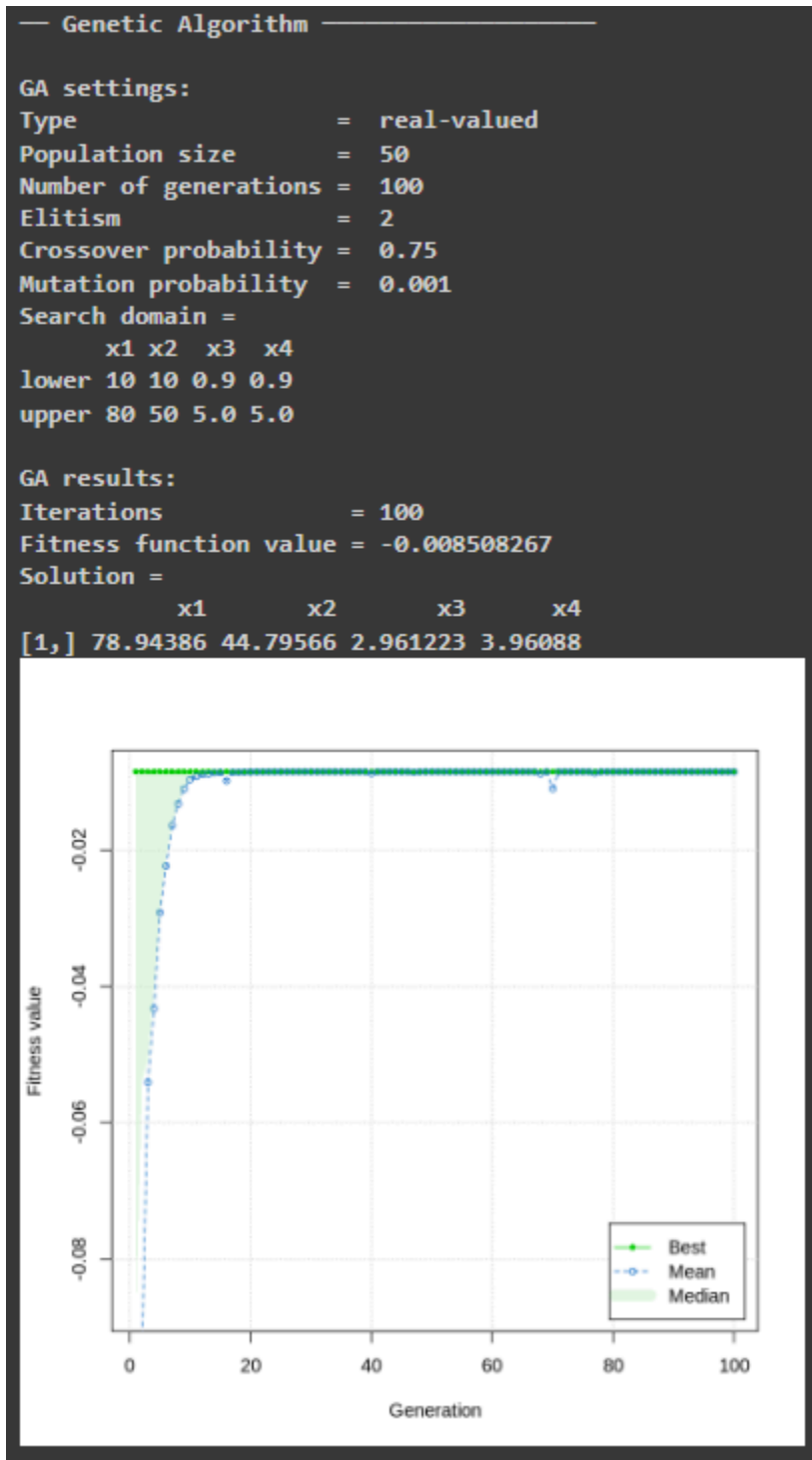


Figure 2: Static Deflection genetic algorithm results (function 2)

Weighted functions are used to help us determine a combined minimized fitness when both functions are run through the genetic algorithm together. This is due to the fact that one side is minimized with low x values and the other is minimized with high x values. When looking at these weights we are able to get a fitness line closer to 0 when more weight is focused on Static while more weight on Cross leads to a higher negative number for the fitness value. The x values also hold true to being higher when more weight is placed on Static and less when placed on Cross.

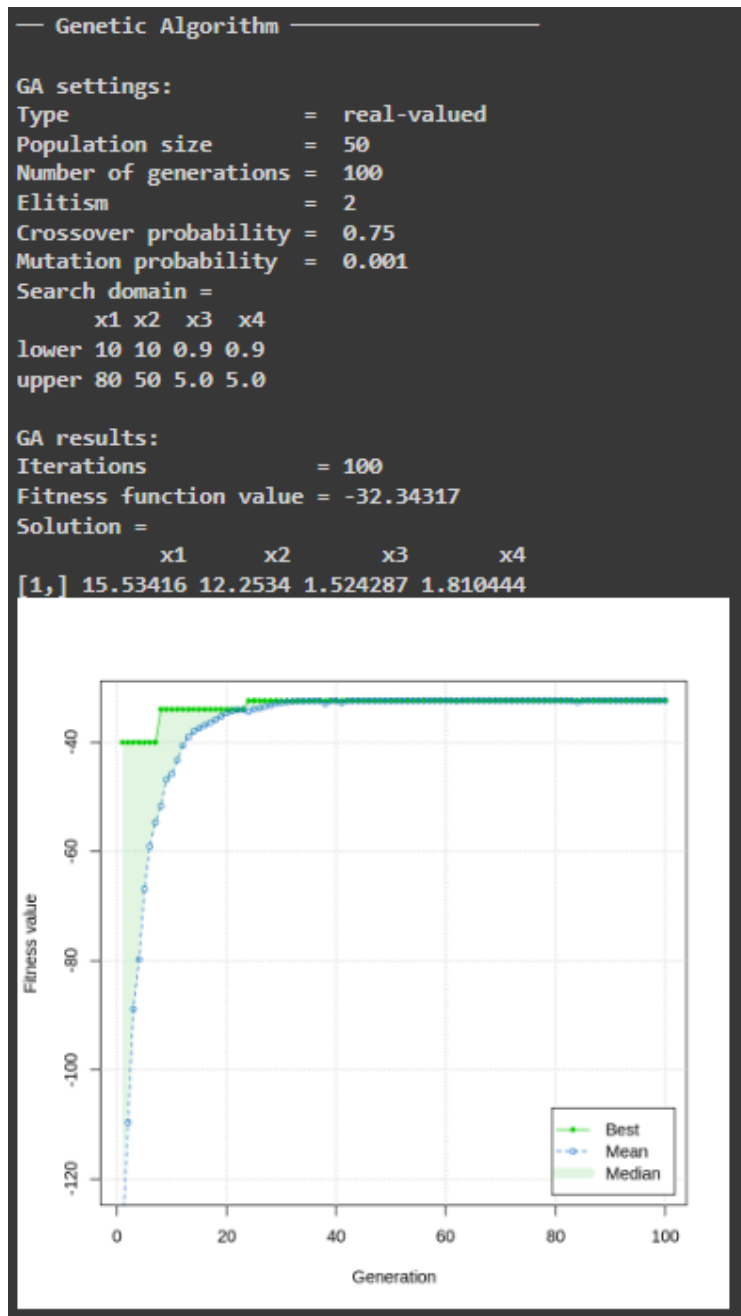


Figure 3: Cross weight .5 Static weight.5

— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -10.18133
Solution =
x1 x2 x3 x4
[1,] 28.92626 12.99891 2.068986 1.735879

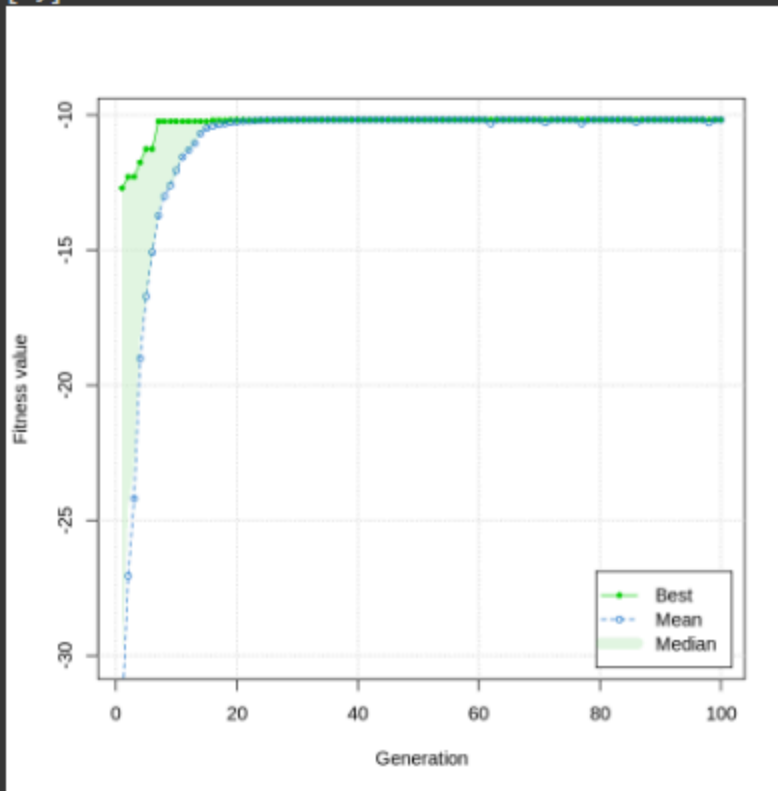


Figure 4: Cross weight .1 Static weight .9

— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -14.796
Solution =
x1 x2 x3 x4
[1,] 17.48864 14.85767 1.635419 1.483152

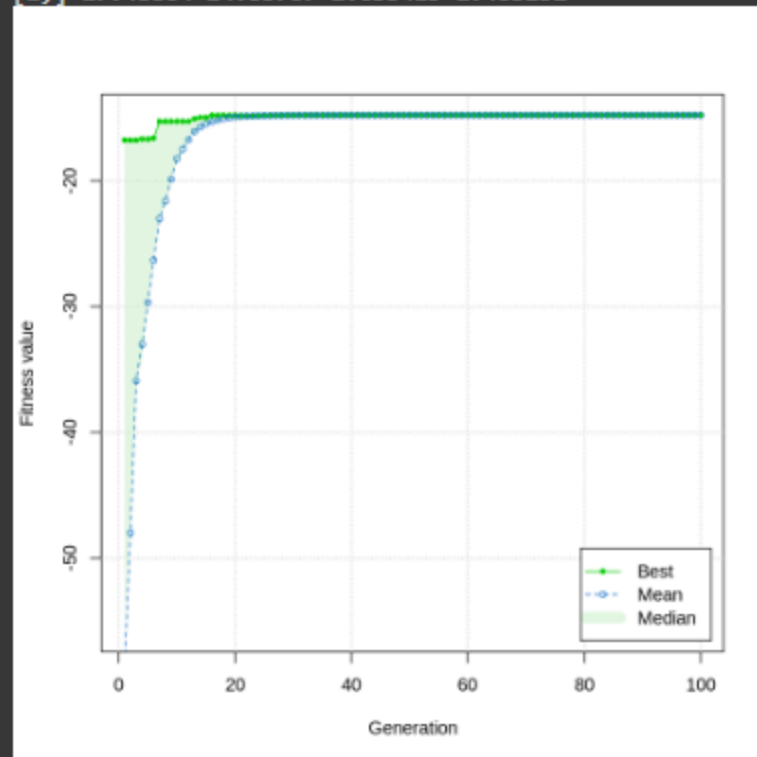


Figure 5: Cross weight .2 Static weight .8

— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -15.75073
Solution =
x1 x2 x3 x4
[1,] 20.66709 13.85692 1.147199 0.9931729

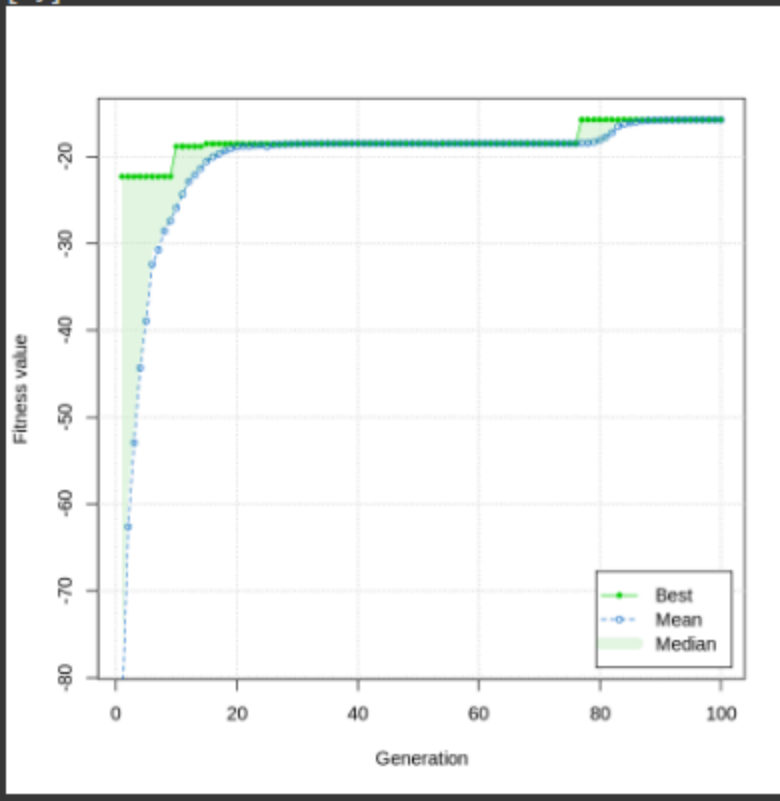


Figure 6: Cross weight .3 Static weight .7

— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -25.25176
Solution =
x1 x2 x3 x4
[1,] 19.05541 12.9177 1.061752 1.723416

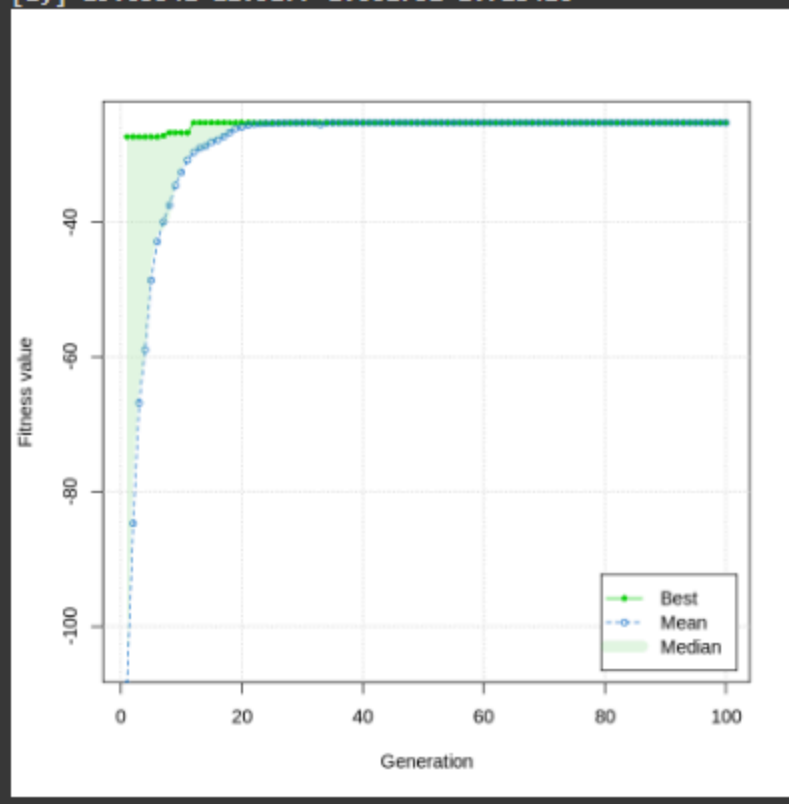


Figure 7: Cross weight .4 Static weigh .6

```

— Genetic Algorithm —

GA settings:
Type           = real-valued
Population size = 50
Number of generations = 100
Elitism        = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
    x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:
Iterations           = 100
Fitness function value = -34.83366
Solution =
    x1      x2      x3      x4
[1,] 13.5167 14.10759 1.297262 1.503401

```

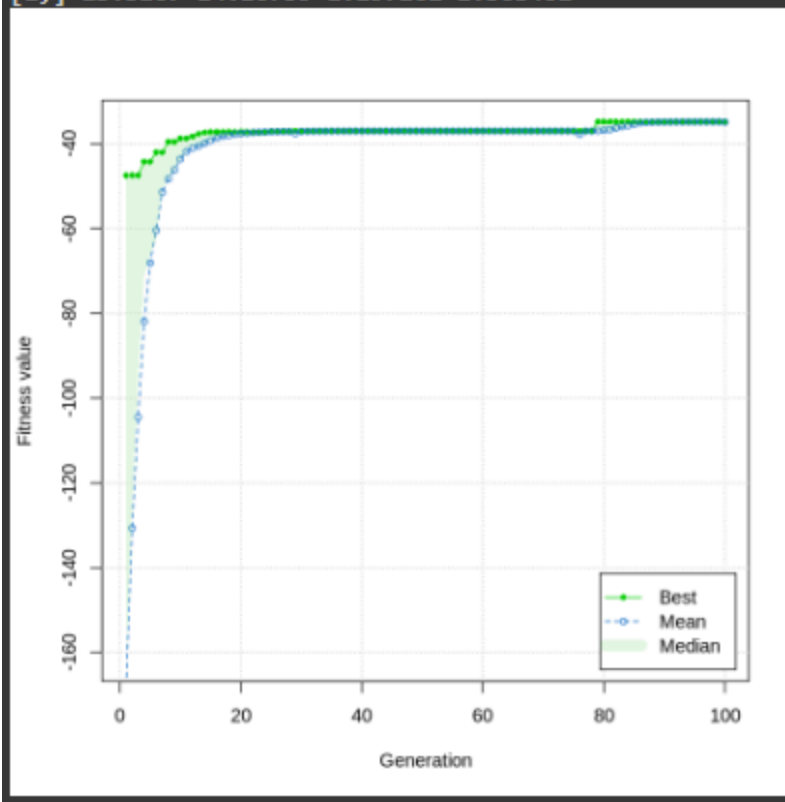


Figure 8: Cross weight .6 Static weight .4

— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -45.49598
Solution =
x1 x2 x3 x4
[1,] 12.89417 20.11088 1.74862 1.122385

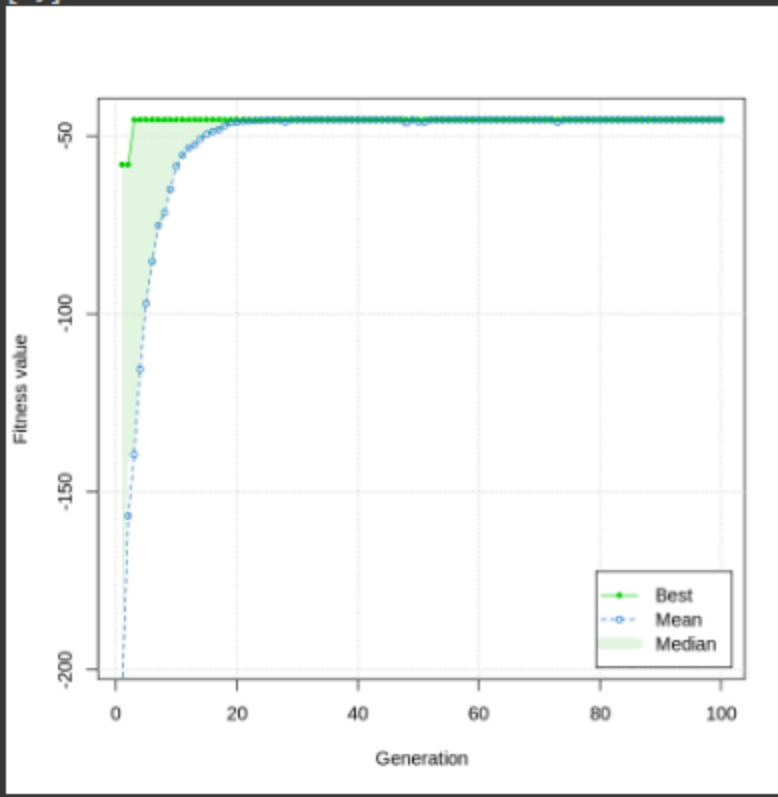
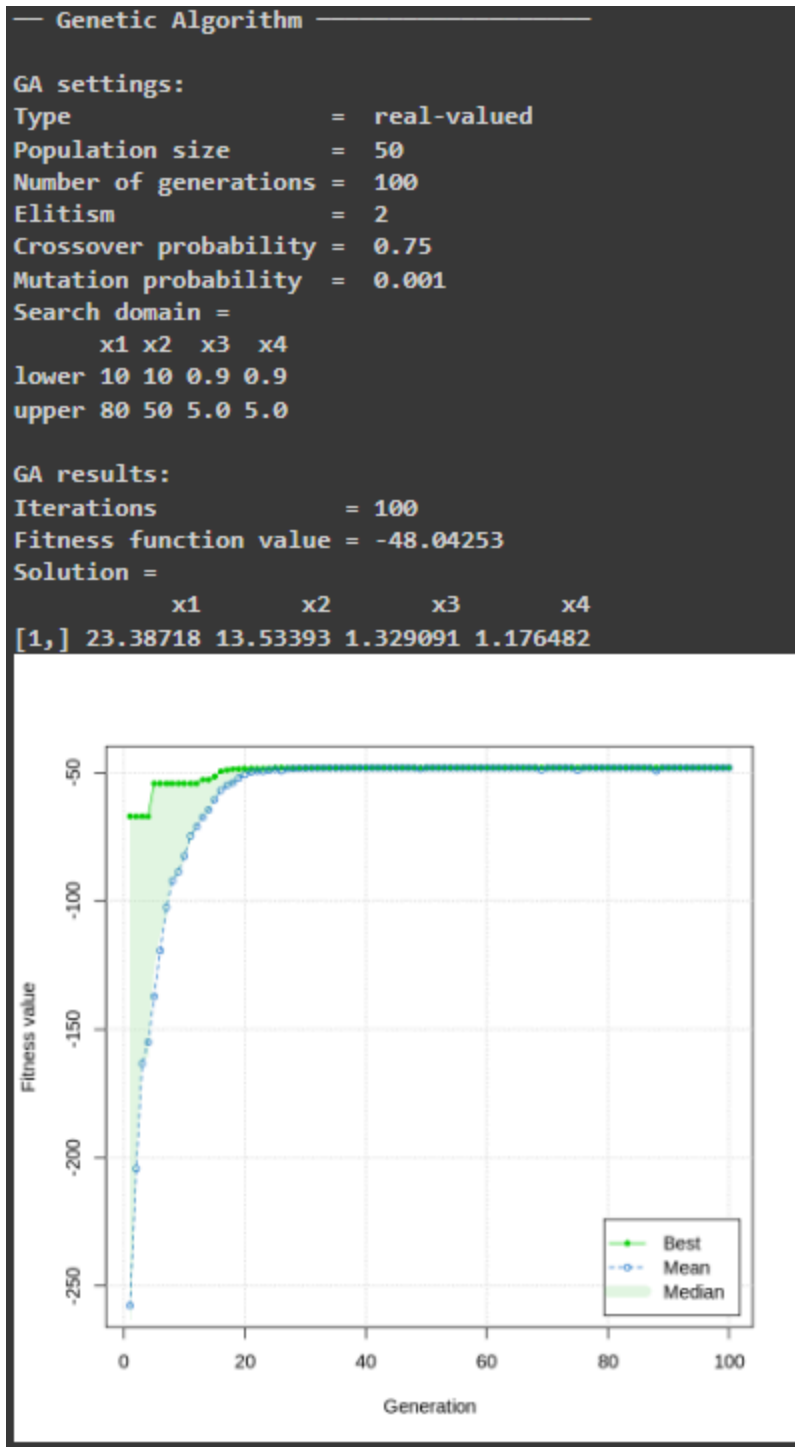


Figure 9: Cross weight .7 Static weight .3



— Genetic Algorithm —

GA settings:

Type = real-valued
Population size = 50
Number of generations = 100
Elitism = 2
Crossover probability = 0.75
Mutation probability = 0.001
Search domain =
 x1 x2 x3 x4
lower 10 10 0.9 0.9
upper 80 50 5.0 5.0

GA results:

Iterations = 100
Fitness function value = -78.76492
Solution =
 x1 x2 x3 x4
[1,] 14.72093 13.95607 2.70048 2.112345

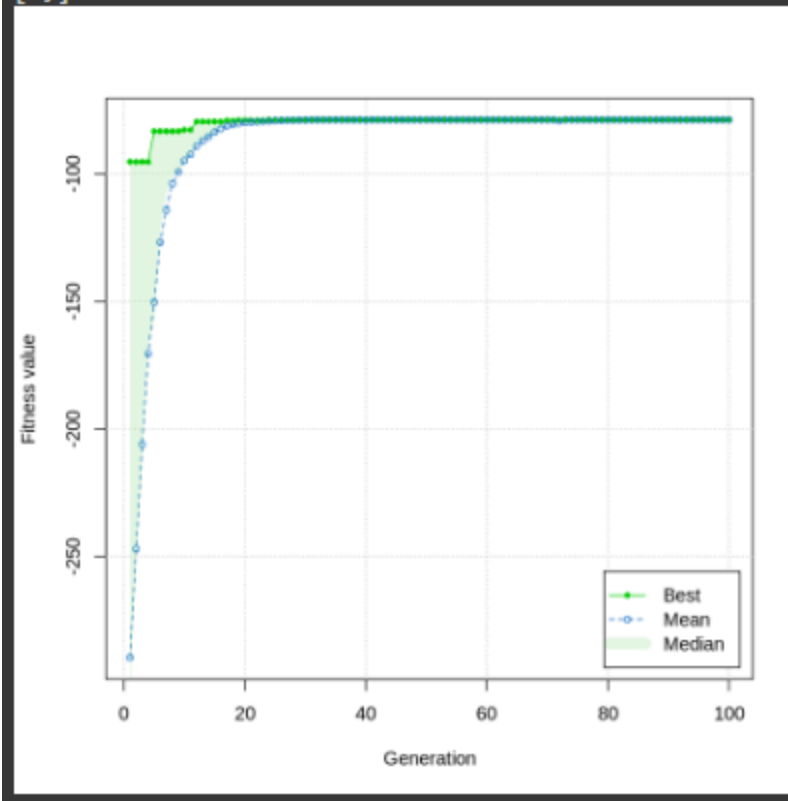


Figure 11: Cross weight .9 Static weight .1

Final Analysis

- Bigger constraint values when weighted on function Static Deflection
- Smaller constraint values when weighted on function Cross Section Area
- Fitness value gets closer to 0 as you increase the weight on the static value equation
- The trend of weighted functions matched well with how the functions performed separately (more weight on static = closer resemblance to solely static function)
- Although some patterns were seen, higher x values didn't necessarily determine the result as each value had its own unique impact in the equations. (example just cause you have a greater x_1 doesn't necessarily mean its static as the other values held impact as well)
- **Following our goal, The weight on Static Deflection should be focused on more for better stability as it got closer to 0 for the fitness value.**