Abeer Khan
ak05419
Habib University

Travelling Salesman Problem using EA

CS 451 - Computational Intelligence : Spring 2022
Sunday, February 13, 2022

Contents

Problem 1

Travelling Salesman: Problem Formulation

The travelling salesman problem is a well-known optimization problem that when given a list of cities/destinations, involves finding the shortest and most efficient route that visits each destination exactly once. An optimal solution could be located using meta-heuristic optimization techniques.
For this problem, the Qatar dataset was used for finding the optimal solution using an Evolutionary Algorithm. The dataset is comprised of 194 cities/nodes - with each city having it's set of x, and y co-ordinates. Using this, an adjacency list is created to construct the many permutations of possible routes that visit each destination in this instance exactly once. The algorithm generates a random population of possible routes initially, which is then used for parent selection to generate offsprings, based on some parent selection scheme. After a mutation operator is applied onto an offspring, a fitness function is used to calculate the total distances of every chromosome in the population. Based on some survivor selection scheme, the fitness values are evaluated to choose the individuals that will become a part of the population that will be carried onto the next generation. This is continued for a period of time, based on the parameters specified, and the best smallest distance found so far would be the output. The algorithm is implemented using a Python class.

Chromosome Representation

In our population, a chromosome for this problem would represent the many permutations of possible routes that visit each destination exactly once. Each destination in our data-set has its set of x and y-co-ordinates. To calculate the distance between two destinations, we apply the Euclidean distance formula.

Fitness Function

The fitness function determines the total distance between each destination in each chromosome in the population. The fitness value of the chromosome will be considered high, if the total distance of the chromosome is small. As the value of the total distance grows smaller, the fitness value of the chromosome increases.

Parameters Used

The following parameters are implemented in our program with some constant values that ensure fair comparison between all iterations of our algorithm. They can be changed to observe the behavior of our algorithm:

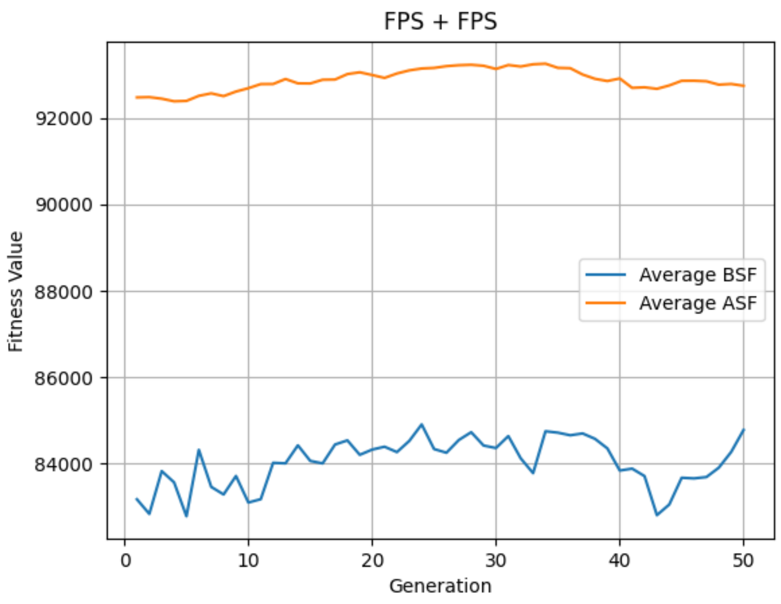
- Population size: 300
- Number of offspring to be produced in each generation: 500
- No. of generations: 50
- Mutation rate: 0.7
- No of Iterations: 10

Problem 1 continued on next page. . .

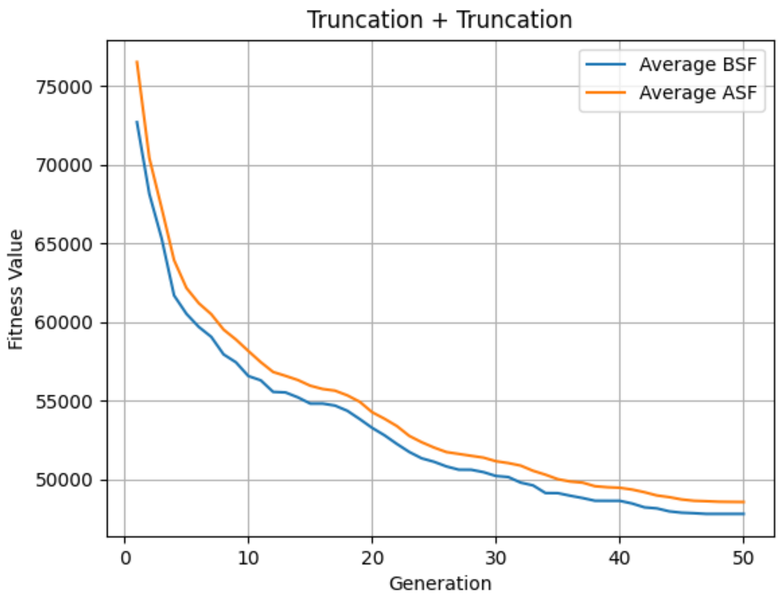
Results: Single Combination

We repeated the EA process for the following single combinations of parent and survivor selections respectively:

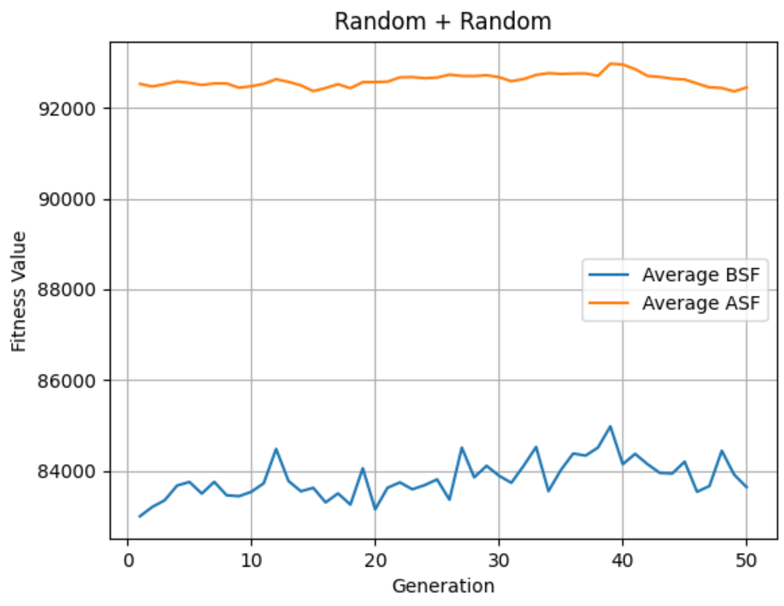
- FPS and FPS:



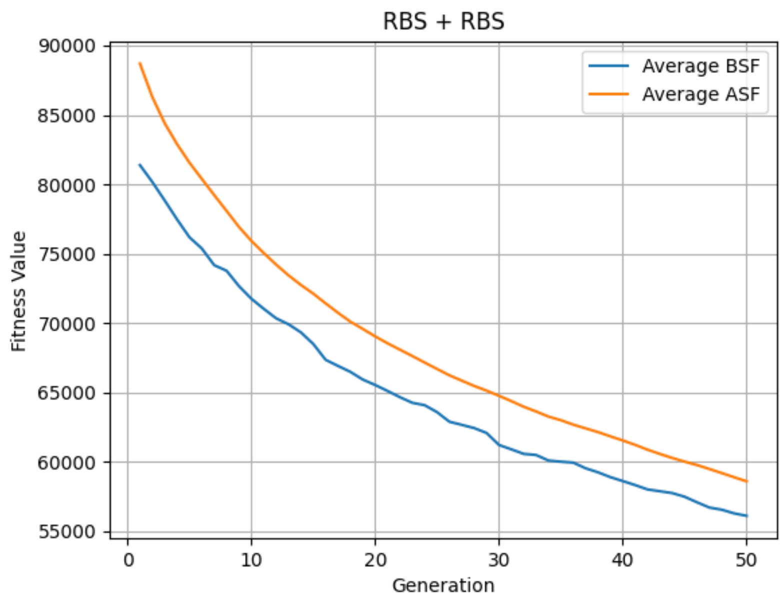
- Truncation and Truncation:



- Random and Random:



- RBS and RBS:

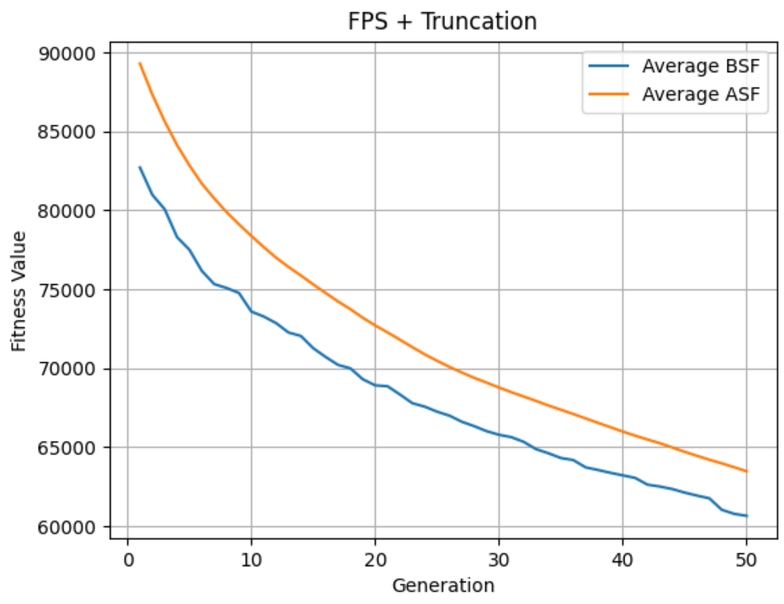


Problem 1 continued on next page. . .

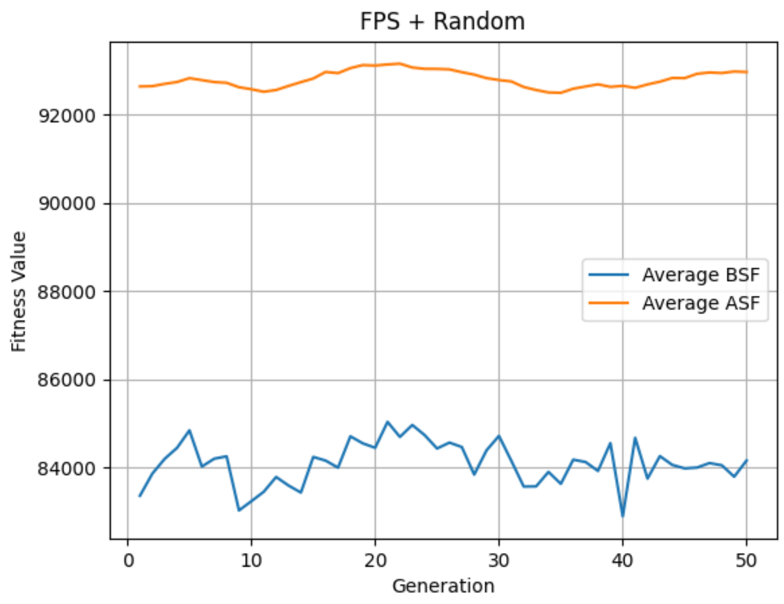
Results: Different Combinations (FPS)

We repeated the EA process for the following combinations of parent and survivor selections respectively:

- FPS and Truncation:

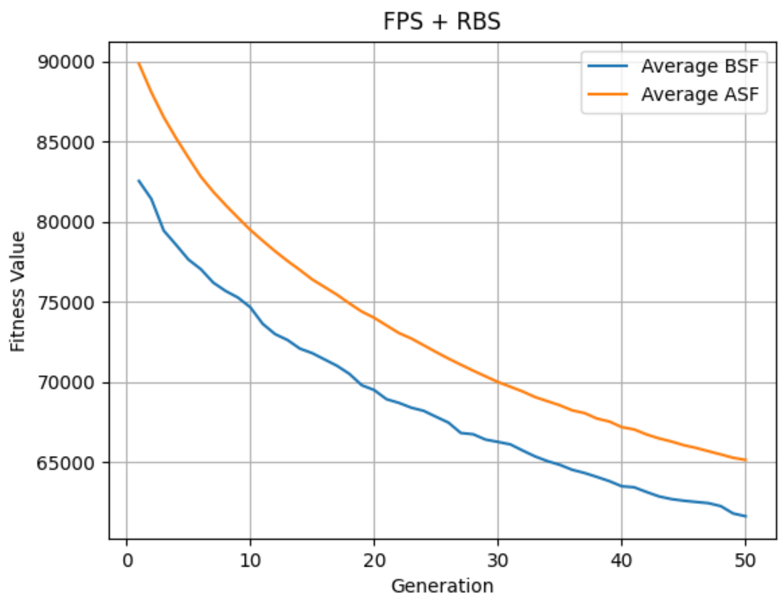


- FPS and Random:



Problem 1 continued on next page. . .

- FPS and RBS:

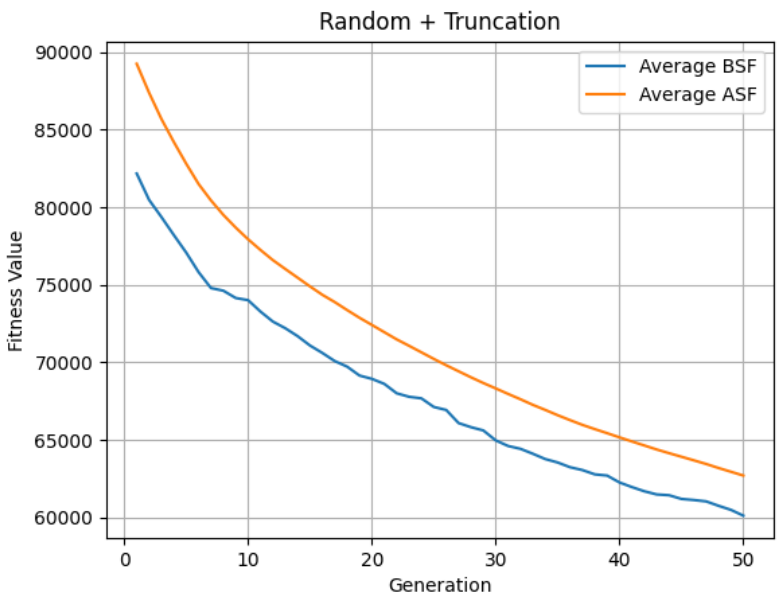


Problem 1 continued on next page. . .

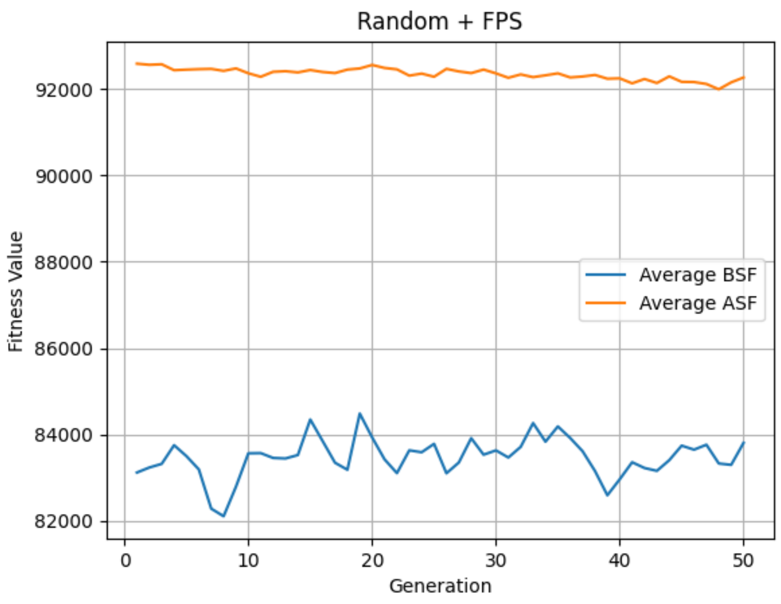
Results: Different Combinations (Random)

We repeated the EA process for the following combinations of parent and survivor selections respectively:

- Random and Truncation:

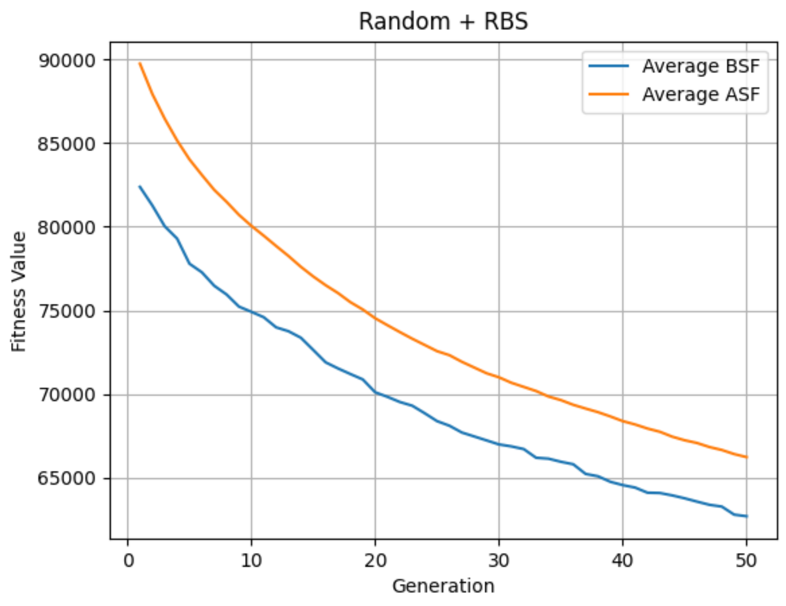


- Random and FPS:



Problem 1 continued on next page. . .

- Random and RBS:

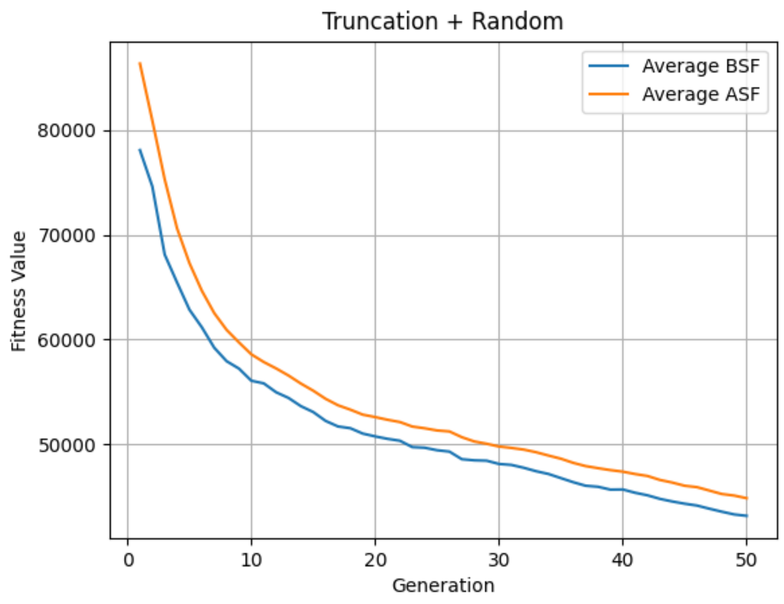


Problem 1 continued on next page. . .

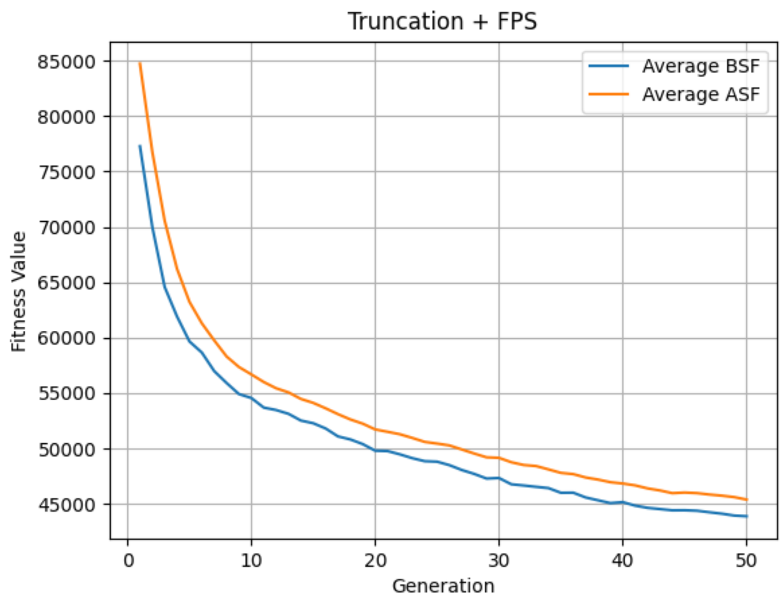
Results: Different Combinations (Truncation)

We repeated the EA process for the following combinations of parent and survivor selections respectively:

- Truncation and Random:

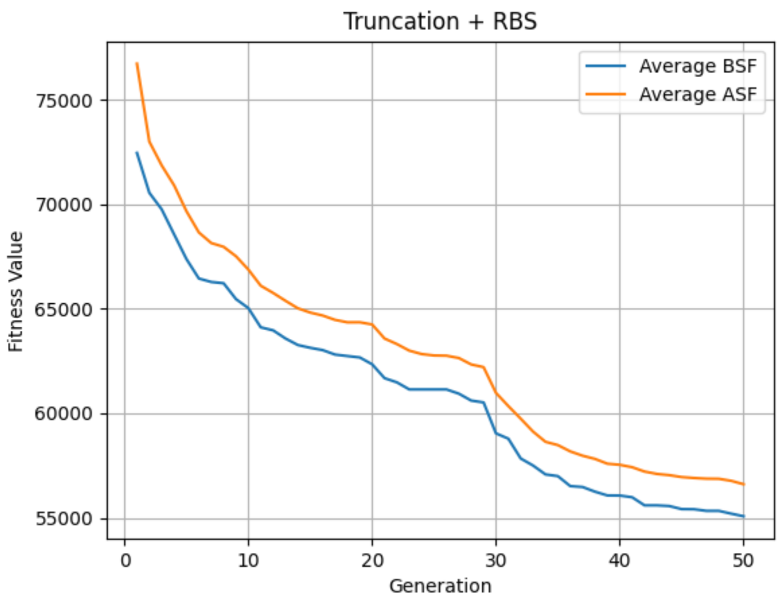


- Truncation and FPS:



Problem 1 continued on next page. . .

- Truncation and RBS:

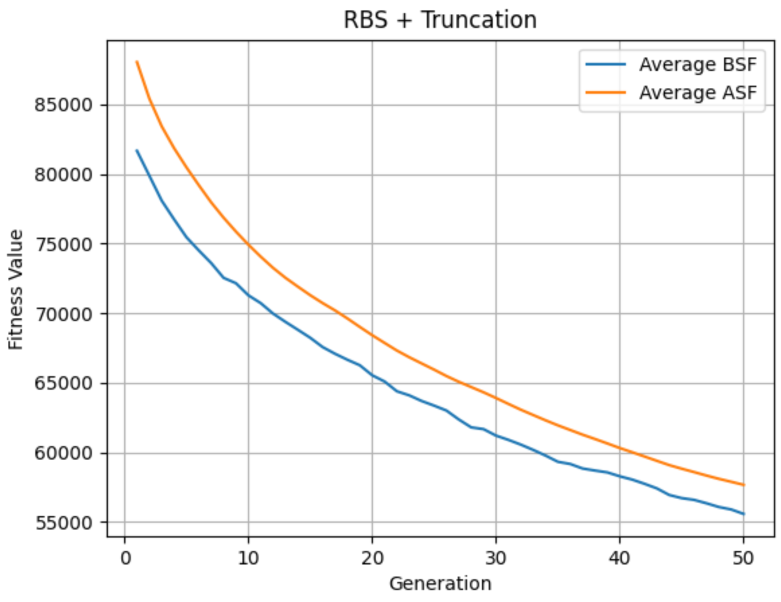


Problem 1 continued on next page. . .

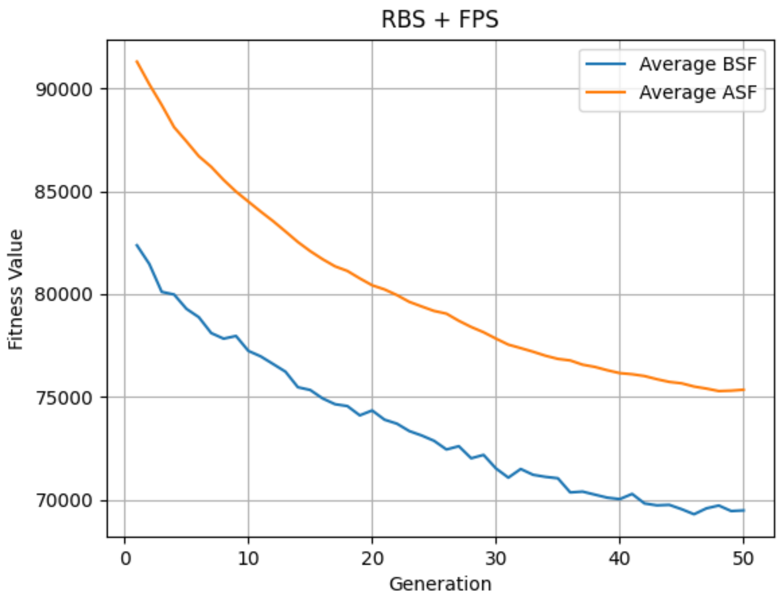
Results: Different Combinations (RBS)

We repeated the EA process for the following combinations of parent and survivor selections respectively:

- RBS and Truncation:

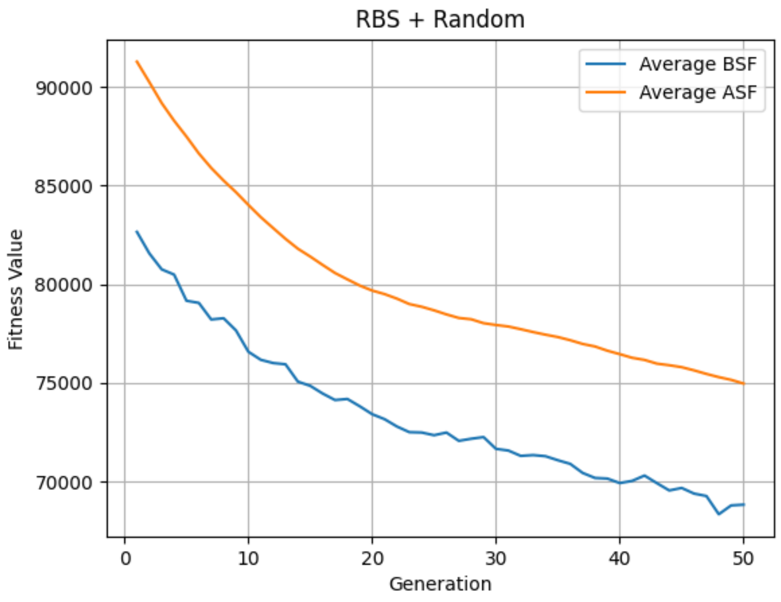


- RBS and FPS:



Problem 1 continued on next page. . .

- RBS and Random:



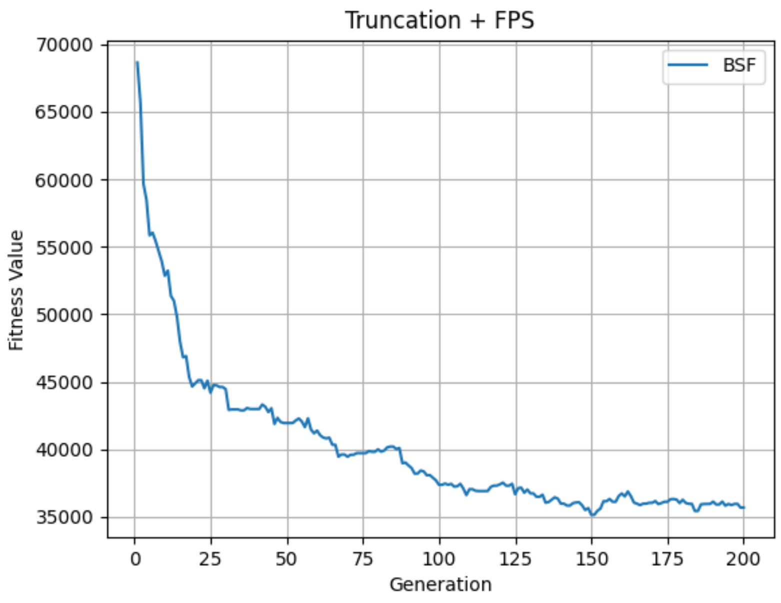
Problem 1 continued on next page. . .

Analysis and Findings

Analysis of the graphs leads to the following conclusions:
a) The Truncation and FPS parent and survivor selection scheme combination yields the best results when compared to other combinations of selection schemes. It converges to a total distance of around forty-five thousand. Furthermore, there is a very small difference between the averages of the average fitness values and average best fitness values. According to this, we can conclude that Truncation and FPS would be the best parent and survivor selection scheme combination for this problem.

b) Certain parent and survivor selection scheme combinations such as FPS and FPS, Random and Random, FPS and Random, and Random and FPS, do not yield very good results. Furthermore, there is a very large difference between the averages of the average fitness values and average best fitness values. They converge to very high values of total distance, which is not a good solution for this problem as we need to find the shortest overall distance.

After concluding that the Truncation and FPS parent and survivor selection scheme combination can yield the best results, the generation parameter was changed from 50 to 200 to observe how small the best fitness value can become over a period of time. The best fitness value generated was calculated to be at 35693.


Truncation + FPS
BSF
Fitness Value
Generation