Problem:

Given a set X of real numbers and a constant c, find a subset $S \subseteq X$ that minimizes $f(X,c) = c|S| + \sum_{y \in (X-S)} (y - \bar{Y})^2$ where $\bar{Y} = \frac{1}{|X - S|} \sum_{y \in (X-S)} y$.

The objective function $f(X,c)$ may be interpreted as follows: The cost of suppressing any element from X is c. The first term in the objective function is the cost of suppressing a subset of elements S. The second term in the objective function is the sum of squared deviation of the unsuppressed points remaining in X from their mean. The goal is to identify a subset S of elements to suppress from X such that this objective function is minimized.

The task is to use a genetic algorithm (GA) to try and identify an optimal solution for the following instance of this problem:
c = 1,000 and
X={64, 92, 39, 97, 52, 85, 93, 30, 1, 53, 74, 87, 2, 24, 42, 89, 60, 95, 12, 79, 35, 27, 9, 18, 19, 32, 31, 94, 63, 36, 28, 41, 34, 72, 48, 46, 40, 25, 26, 84}.

Solution:

A Genetic Algorithm was used to identify the minimum value of f(X,c). The data structure used to encode the chromosome was using 40 (size of X) bits for each chromosome; each bit represented an item in X. For every item that was suppressed the value of the corresponding bit was set to 0. Any item that stayed in X had the corresponding bit value of 1.

The fitness function used in the GA algorithm is 1/f(X,c), where:

$f(X,c) = c|S| + \sum_{y \in (X-S)} (y - \bar{Y})^2$ where $\bar{Y} = \frac{1}{|X - S|} \sum_{y \in (X-S)} y$.

The higher the fitness function the better the fittness for that chromosome.
The initial selection strategies that were used in the GA consisted of two methods, the first method was a complete random operation to fill the values in the population with ones or zeros. The second method was to bias against removing bigger values in X with a certain percentage. Details of the results are mentioned later in the report, but the conclusion was that the biased method was disruptive when compared to the best result of the randomization.

The first replacement strategy that was used was to select the fittest of the new old and the new generation. The second strategy was to select 80% of the old and the new population and 20% randomly from the left ones. Overall there was little or no improvement when using both techniques.

The genetic operators that were used were cross over and mutation operators. Each operator was applied with a certain probability to every chromosome (Cross Over) and every bit (Mutation). Different probabilities were tried for each one operator. Detailed results are displayed later in the report. Overall, it was found that higher crossover probabilities yield better results, while higher mutation probability usually is disruptive.

The minimum value that was identified by the GA that was created by the author of this report was (1/ 19473.8461538462).

The values that were suppressed using this algorithm are “92, 97, 85, 93, 1, 74, 87, 2, 89, 95, 79, 94, 72, 84”.
The values that were reported (X-S) are “64, 39, 52, 30, 53, 24, 42, 60, 12, 35, 27, 9, 18, 19, 32, 31, 63, 36, 28, 41, 34, 48, 46, 40, 25, 26”.
Different population sizes were used, and the number of generations evolved to reach the best result in each run was calculated. Detailed results are shown in the following section.

Detailed Results

In order to gauge how each change will affect the GA, the “Success Rate” metric will be used. Success Rate is a metric that indicates what is the percentage of time did the GA reach the best solution -that is fitness of 1/ 19473.8461538462- when running the GA with the same parameters for certain number of runs, 50 runs was the number of runs that was used in this report.

The following sections present the results for each attempt to quantify the effect of the GA parameters on the success rate and also the effect of initializing the population on the success rate.

Effect of Mutation Probability and Population Size on Success rate for Fixed Crossover Probability

By varying the mutation probability the following results were obtained for a population size of 1000, 500, and 100. The crossover probability was fixed at 80%.

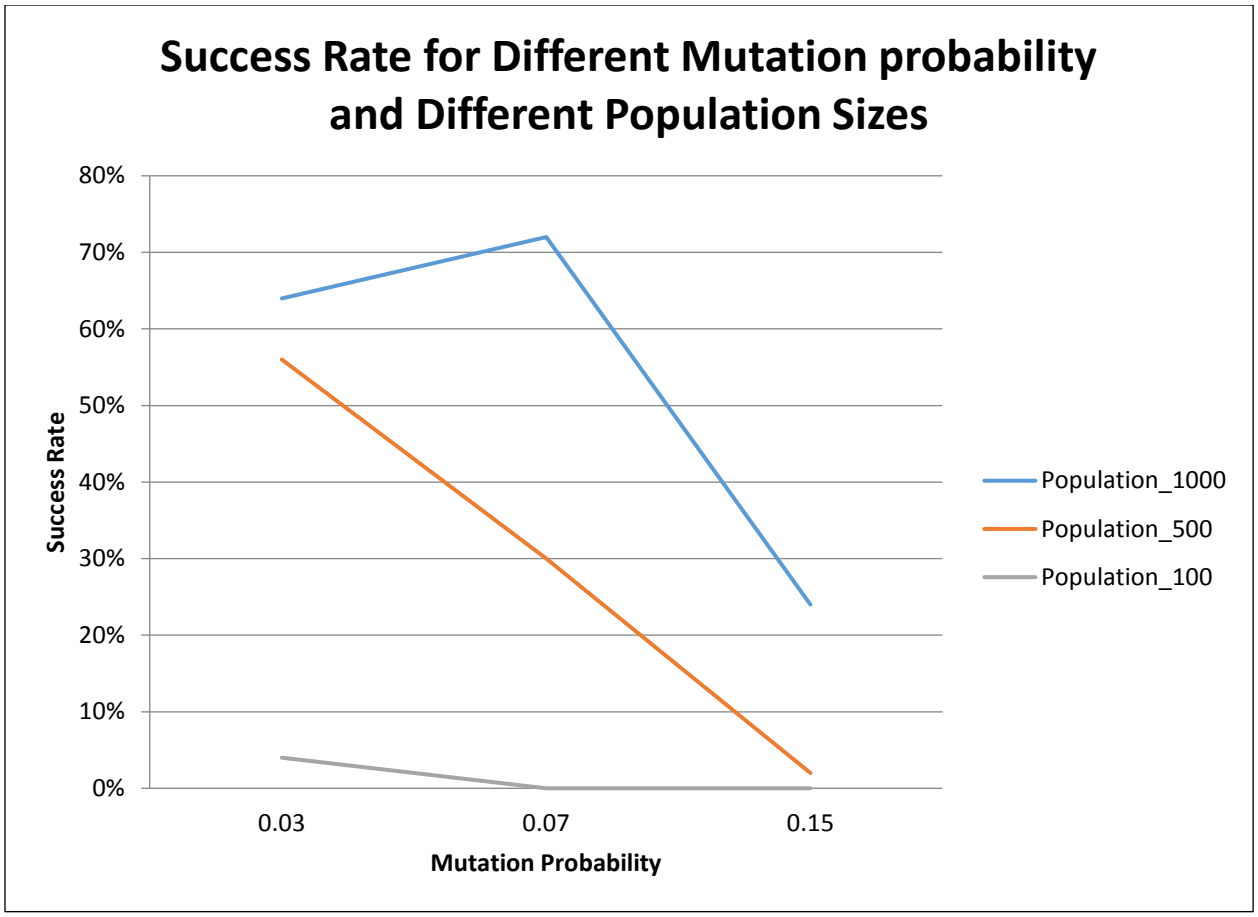

Figure 1 Effect of Mutation and Population Size on Success Rate

As seen, for small population sizes when the mutation increases the success rate decrease. As the Population size grows, some scenarios can tolerate higher mutation probability such as the case of mutation of 7% and population size of 1000. But eventually as the mutation probability increases the success rate will eventually decrease. On the other hand, the higher the population size the better the Success Rate.

Effect of Population size with fixed Mutation probability and Fixed Crossover probability on success Rate

To elaborate more on the effect of the population size on the success rate, the mutation probability was fixed to 3% and the crossover probability was fixed to 80% and the success rate was displayed as a function of population size as shown below.

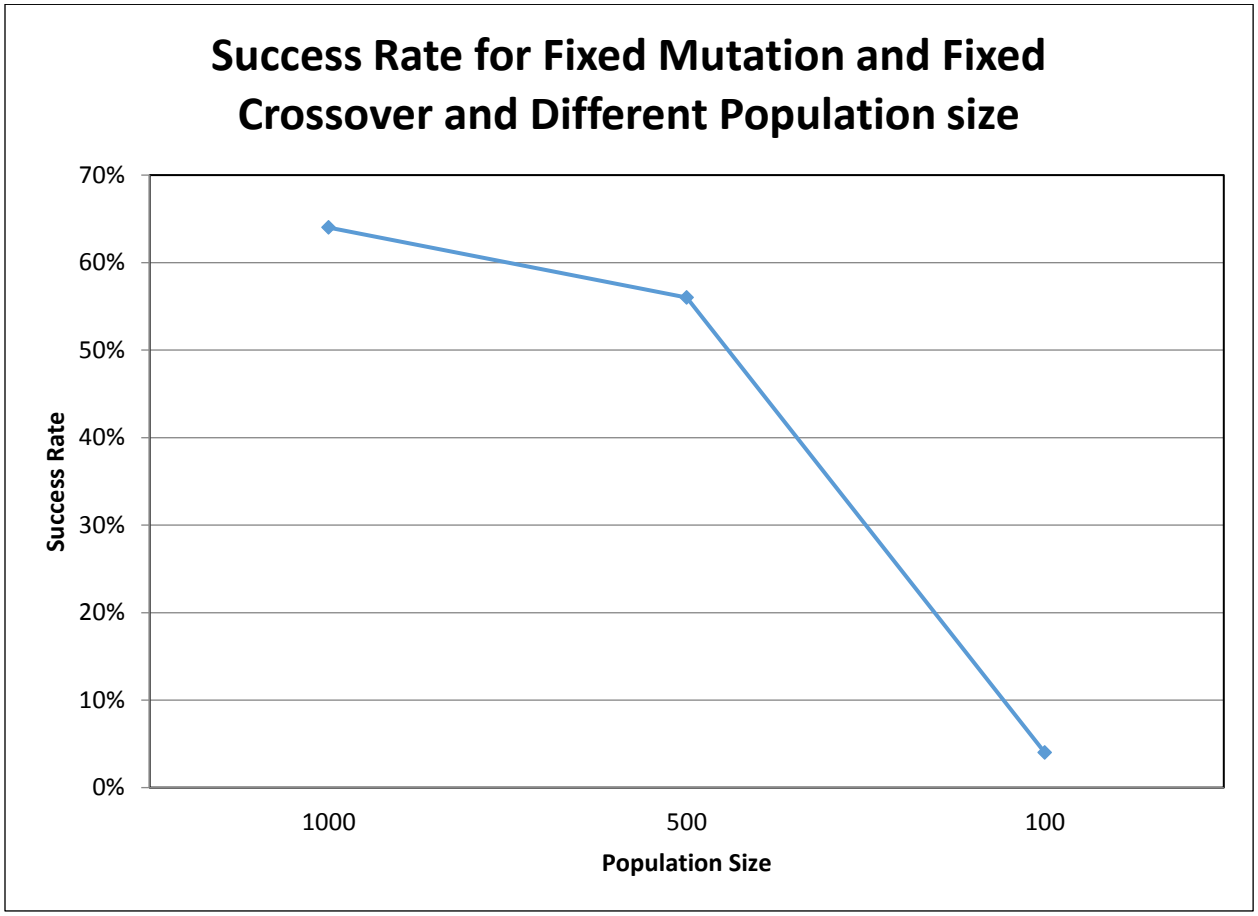

Figure 2 Effect of Population Size on success Rate

As seen, when the population size increases it can tolerate more mutation and provide better results.

Effect of Crossover Probability with Fixed Mutation probability and Fixed Population size on success Rate

As can be seen from the graph below, the higher the Crossover probability it yielded better the Success Rate.

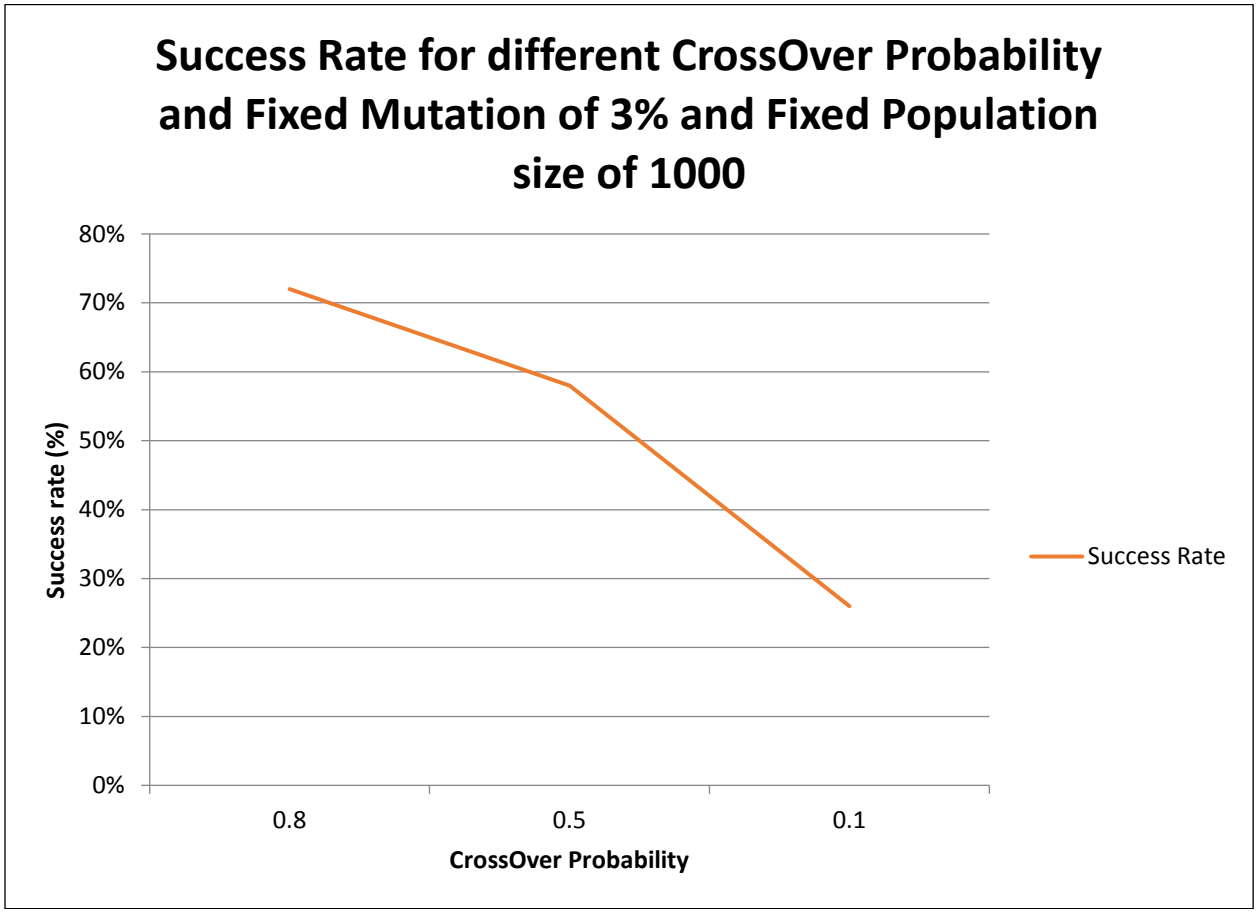

Figure 3 Effect of Crossover on success Rate

Effect of choosing a different way to populate the values for the initial population

By using a bias technique that deletes the large numbers randomly using a certain percentage another set of data was evaluated. This was done assuming that the large numbers are the ones that will control the fitness function based on examining the fitness function. Fixed values for Mutation (7%) and Crossover (80%) probabilities were used.

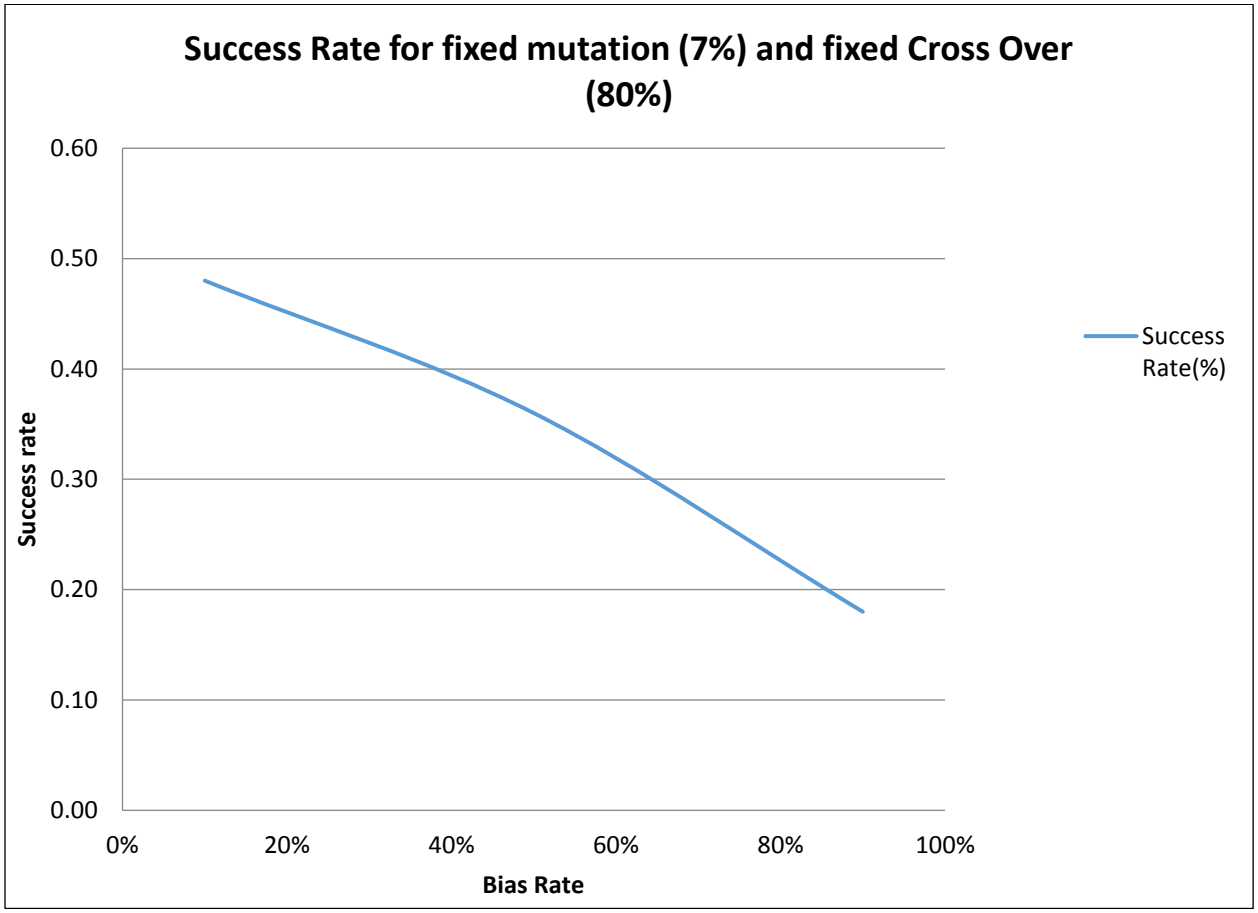

Figure 4 Effect of biasing the population towards certain values

As seen, the Success rate is good, although it is still worse than the best values obtained using random techniques.

Summary:

To summarize the results of the GA algorithm results, it was noticed that higher population sizes helped the GA to achieve higher Success Rate, higher Crossover probabilities are useful, high values for Mutation probability are disruptive as well as using a predefined method to select the initial population.

The C# Source Code is also provided in Program_AI_2.cs files.