Genetic Algorithm: Implementation and Optimization

Pagliuca Simone
Nuclear Engineering - Politecnico di Milano

Academic Year: 2024-2025

1 Introduction

The objective of this project is to explore the implementation and optimization of a Genetic Algorithm (GA) solver to minimize the Styblinski-Tang function. This report details the approach taken for coding the GA solver, a parametric study on its performance, and the results obtained. The Styblinski-Tang function, with a known global minimum, serves as a benchmark for evaluating the accuracy and efficiency of the algorithm.

2 Code Description

The Styblinski-Tang function is used as the fitness function for this problem. Since the GA solver is designed to minimize the function, the ”fittest individual” is the one with the minimum value.

The algorithm consists of the following steps. Except for initialization, each step offers two possible approaches that can be selected individually and will be discussed in detail later.

Initialization:

The population is initialized by randomly choosing elements within given boundaries.

Selection:

- *Tournament Selection:* For each place in the population, a subset of the population of given dimension is randomly chosen, the fittest individual among the subset is chosen.
- *Roulette Selection:* For each place in the population, a subset of the population of given dimension is randomly chosen, probability-weighted selection based on fitness values determines the survoivor.

Crossover:

- *Punnet Crossover:* This is something i made up, for each element of the population there is a given probability that we randomly select two parents, combine their x and y components, evaluate the 4 possible outcomes and selects the best one.

- *Bit Crossover:* For each element of the population there is a given probability that we randomly select two parents then swap a randomly chosen portion of bits from the first 20 digits of the floating-point binary representations (this avoids changing the orders of magnitude).

Mutation:

- *Redefine Mutation:* For each element of the population there is a given probability to reinitialize its variables.

- *Bit Mutation:* For each element of the population there is a given probability to flip a random bit of a gene's binary representation (again, this effects only the mantissa of the floating rapresentation).

3 Parametric Study

A parametric study was conducted to evaluate the impact of all parameters on solver performance. The first step focused on assessing accuracy, as the analytical solution is known. Accuracy is defined as follows:

$$a = 1 - |Sol_{GA} - Sol_{analytical}| \tag{1}$$

3.1 Overall View

Figure 1 shows that as the population size increases, the algorithm's accuracy improves. Certain combinations stand out as particularly reliable, including TBB (Tournament - Bit Crossover - Bit Mutation), TBR (Tournament - Bit Crossover - Redefine Mutation), and TPR (Tournament - Punnett Crossover - Redefine Mutation).

For the following analysis, only combinations with an accuracy of $a > 0.99$ will be considered.

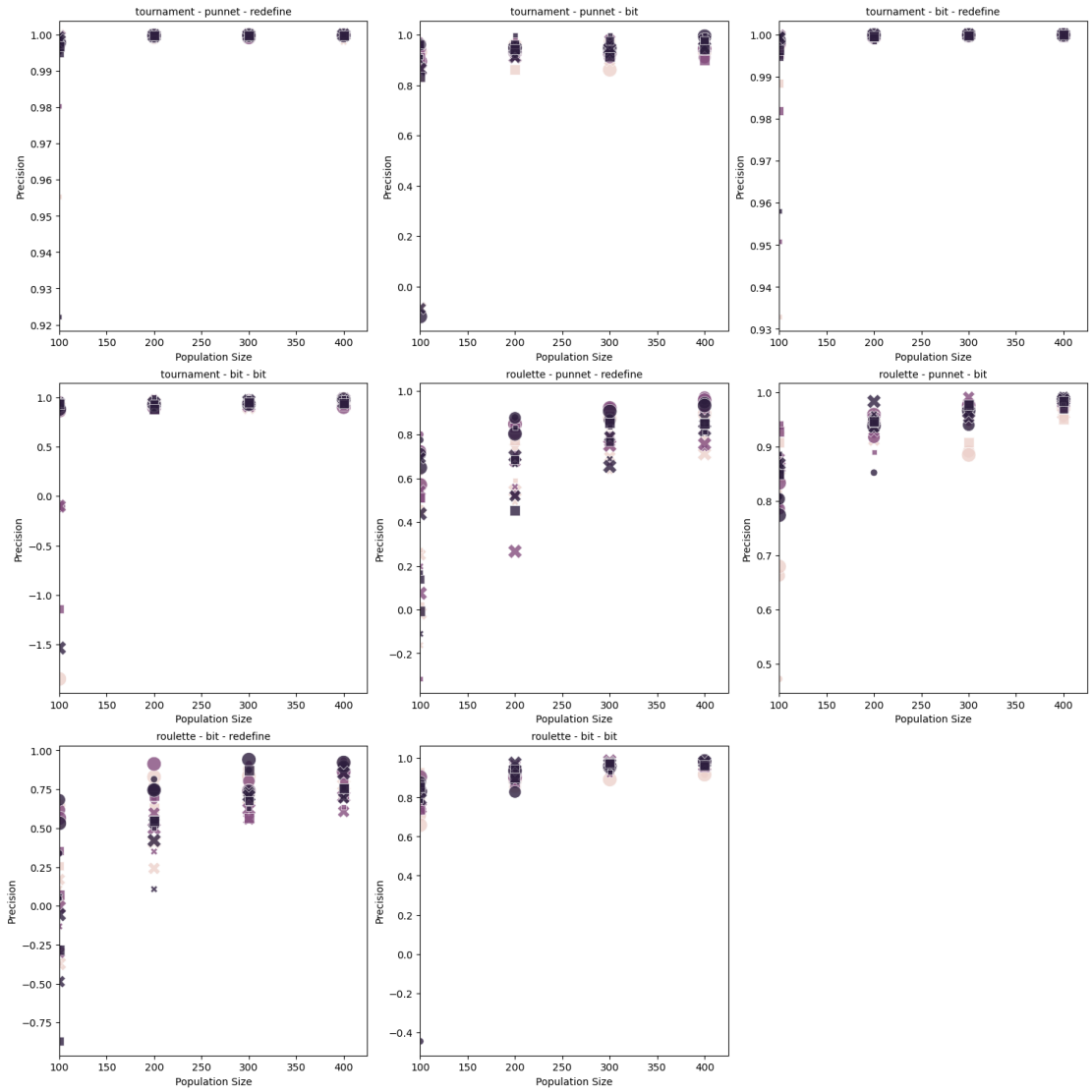
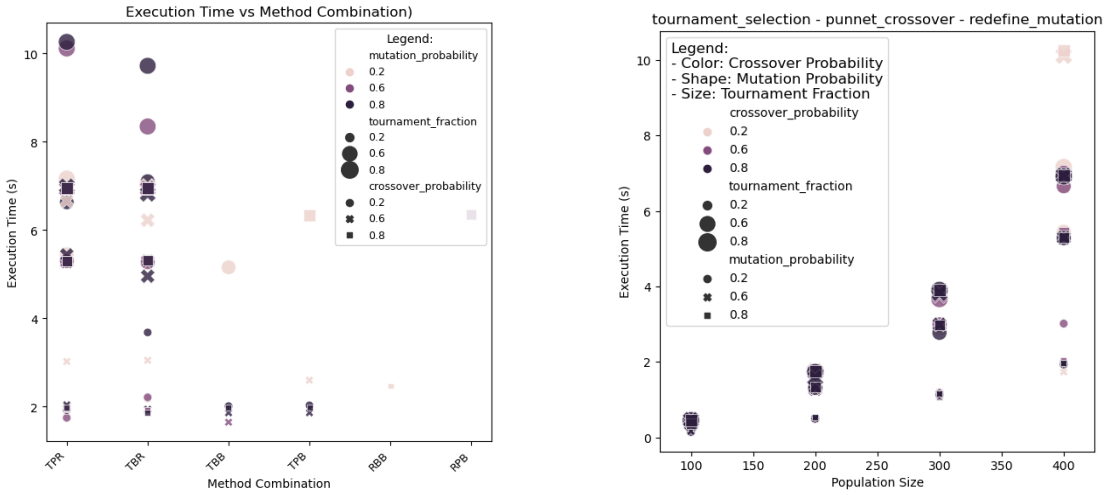


Figure 1: Precision vs Population size for all combinations analyzed

3.2 Execution Time

Another critical parameter for solver performance is the execution time. As expected, execution time increases with population size. Interesting is the formation of ”clusters” that become more distinct as the population grows, as shown in Figure 2b for the TPR method. Based on this observation, the analysis will proceed by focusing on the largest population sizes and the top 5% combinations for execution time.

It is also worth noting that larger populations tend to produce higher accuracy, as illustrated in Figure 3.



(a) Execution Times

(b) TPR Example

Figure 2: Execution time for all combinations with $a > 0.99$ based on methods 2a and execution time influenced by all parameters for the TPR method in particular 2b.

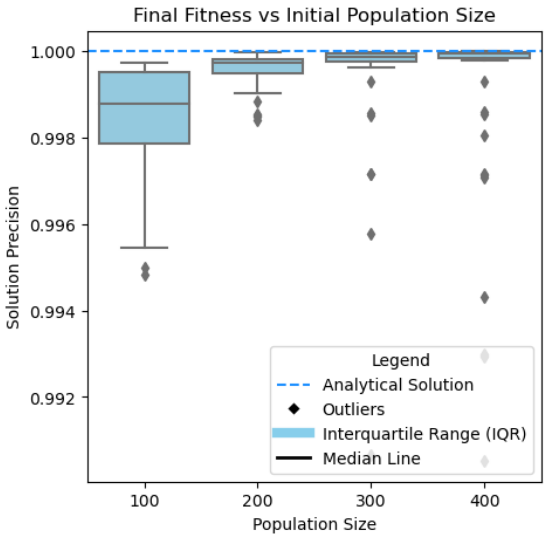
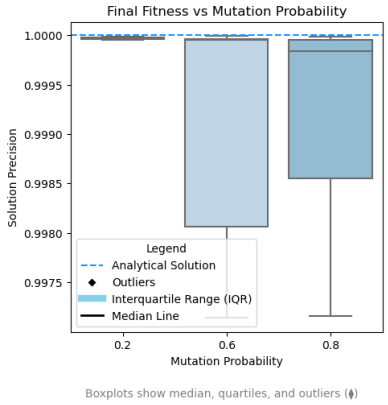


Figure 3: Precision vs Population size for all combinations analyzed

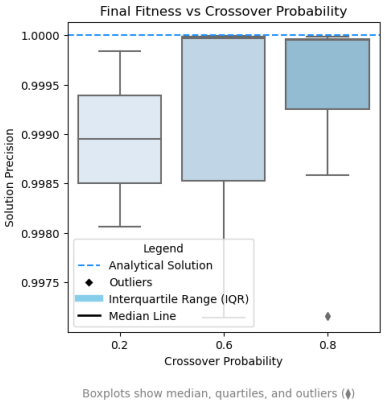
3.3 Parameter Effects

The impact of input parameters on the results can now be examined in more detail. All the configurations with high precision and reasonably fast execution times share a tournament fraction of 20%.
As shown in Figure 4a, lower mutation probabilities reduce variation in the solution but

can slightly compromise precision. This suggests that a finer analysis of this parameter towards low values might reveal an optimal balance.
For crossover, Figure 4b illustrates that higher probabilities improve solution quality. Compared to $p_C = 0.8$, lower probabilities exhibit smaller deviations but reduced precision. Increasing the probability to 0.6 improves precision but introduces slightly larger deviations. Further exploration of higher values could provide deeper insights into the optimal setting for this parameter.



(a) Precision vs Mutation Probability



(b) Precision vs Crossover Probability

Finally, we can now see a comparison between the best combinations so far.

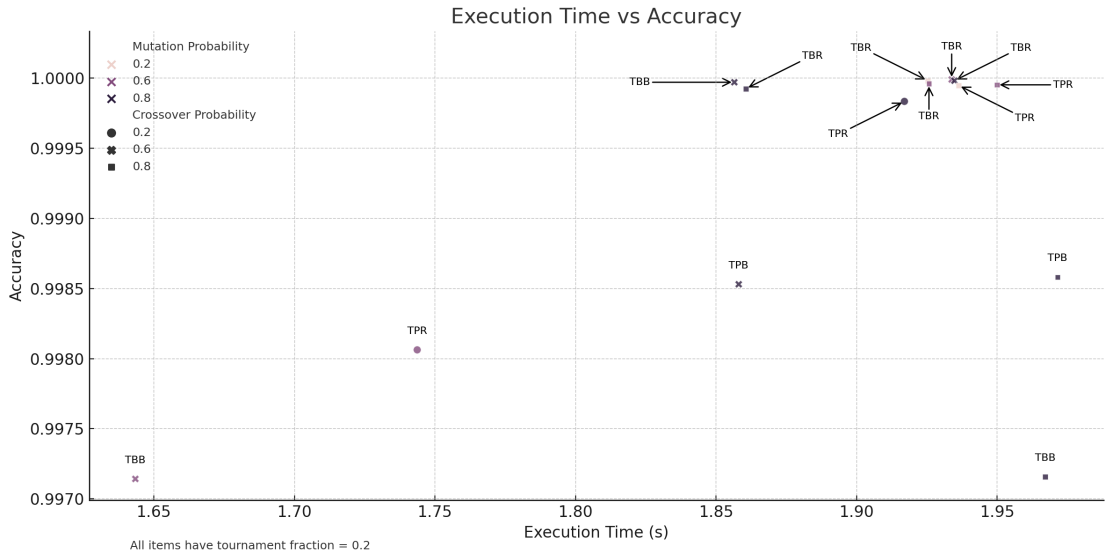


Figure 5: Execution time versus accuracy for optimal configurations.

4 Results and Conclusion

For the final run, the model providing the best balance between accuracy and execution time was chosen from the configurations tested. The selected parameters are as follows:

- Population size: 400 individuals.
- Crossover probability: 0.6.
- Mutation probability: 0.6.
- Selection: Tournament selection.
- Crossover: Bit crossover.
- Mutation: Redefine mutation.

The solver consistently converged to solutions near the global minimum of -78.3323 ± 0.0003 with the following statistics:

- Average solution coordinates: $x = -2.9035 \pm 0.0011$, $y = -2.9036 \pm 0.0011$.
- Average execution time: 1.92 ± 0.13 seconds.

The algorithm achieved high accuracy and efficiency, converging reliably near the global minimum. The findings highlight the importance of careful parameter tuning to balance precision and execution time. Future work could further refine these results by exploring additional parameter values or applying the algorithm to more complex problems.