LAB 2

Bio-Inspired Computing (COSC 527)

Azizul Zahid
Graduate Student
ID: 000673586
EECS (Computer Engineering), UTK

Introduction:

The lab explores using Evolutionary Algorithms (EAs) to solve problems in computer systems. Specifically, it demonstrates how EAs can effectively find optimal solutions using a fitness function. The LEAP-EC python library is used to simulate 30 generations of evolution, where each generation is iterated 20 times with different parameter combinations. The experiment tests four different parameters, each with four different values: population sizes (25, 50, 75, and 100), mutation probabilities (0, 0.01, 0.03, and 0.05), uniform crossover probabilities (0, 0.1, 0.3, and 0.5), and tournament sizes (2, 3, 4, and 5).

The default parameter values are population size = 50, mutation probability = 0.01, crossover probability = 0.3, and tournament size = 2. The experiment aims to find a genome represented by a binary string of 40 values that equals a string of all 1s. During the selection stage, the EA uses a fitness function to assess the genome.

Experimental results:

All the data generated from the evolutionary algorithm using the LEAP library function and different parameter combinations are stored in the "***Main_data.csv***" file. To illustrate the relationship between different parameters, we used a set of default values: 50 for population size, 0.01 for the mutation probability, 0.1 for the crossover probability, and 2 for the tournament size. Each of the graphs below shows the fitness values (average and best) for each combination of three different parameters, with one parameter in a variable state.

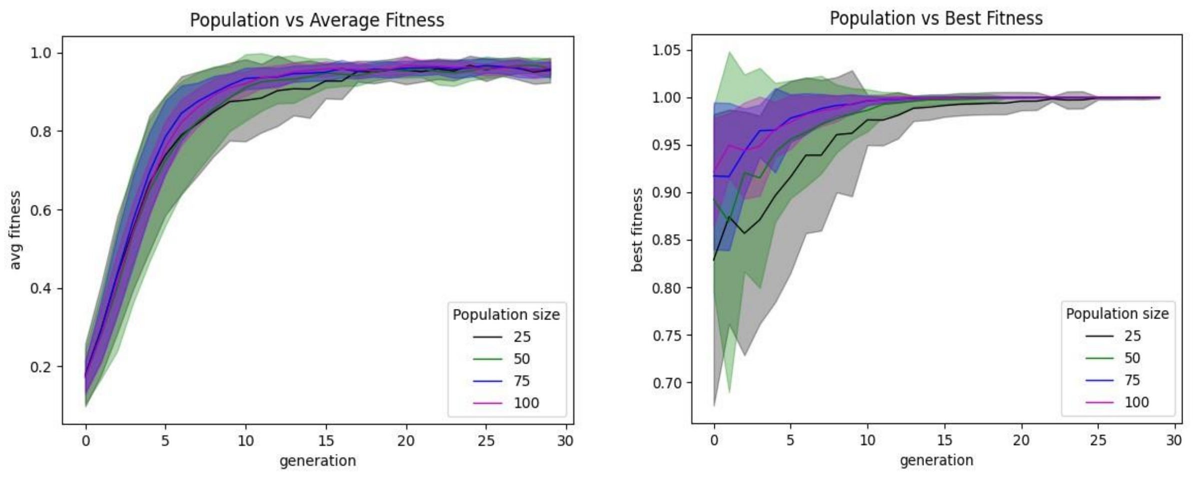

Figure 1: Population vs Fitness

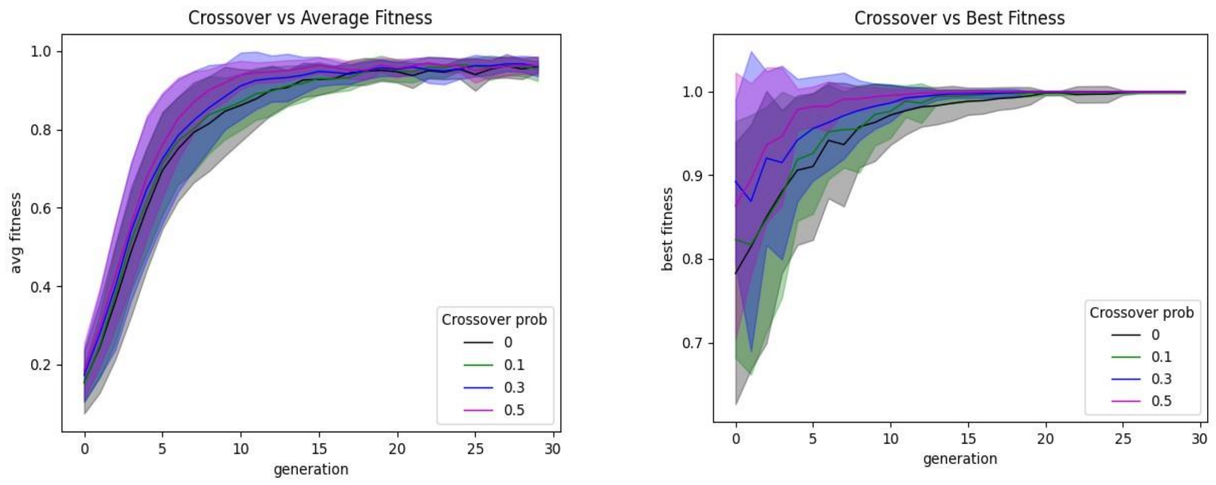

Figure 2: Mutation vs Fitness

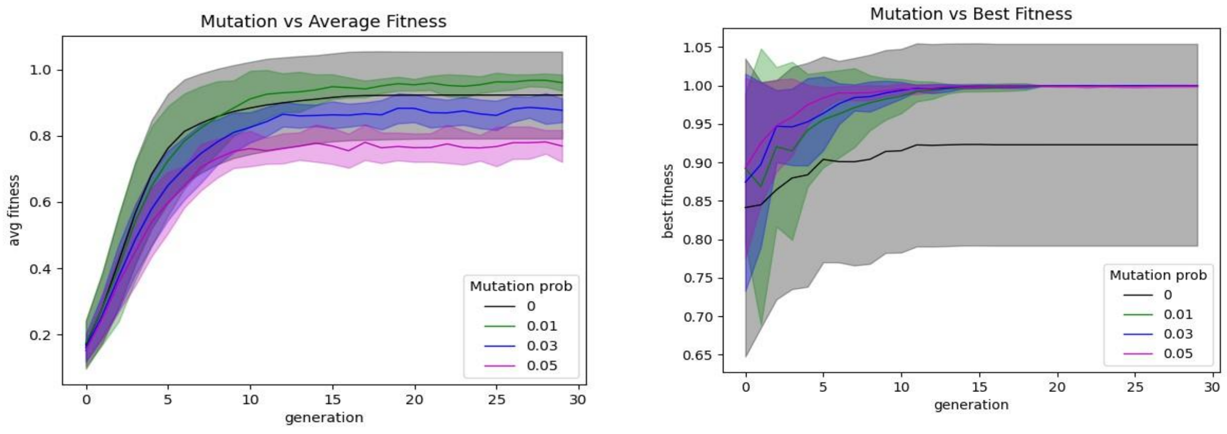

Figure 3: Crossover vs Fitness

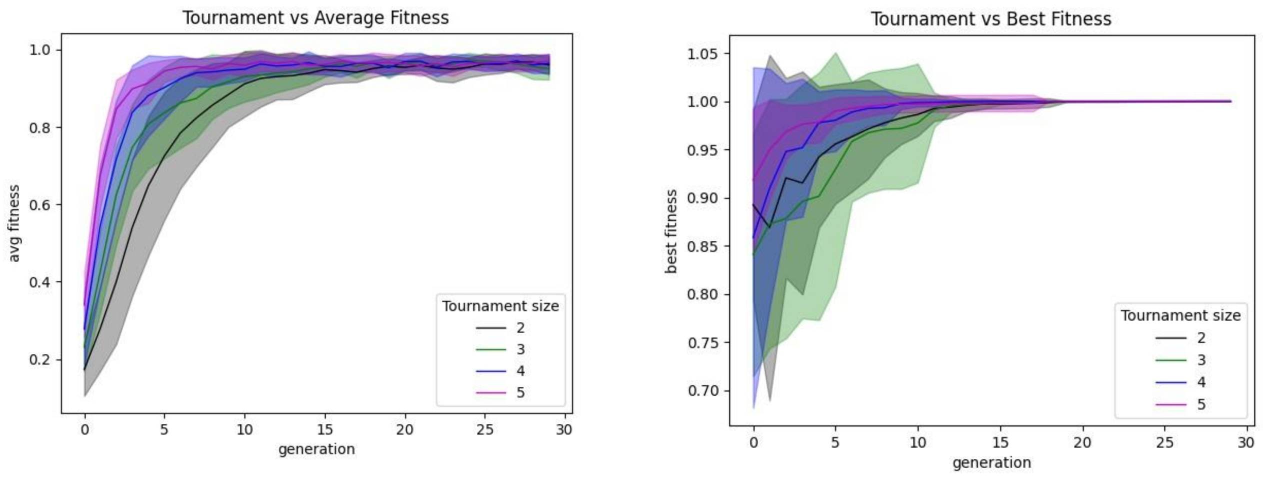

Figure 4: Tournament size vs Fitness

Discussion:

From **all the graphs** related to **the average and best fitness** achieved for different parameters (**Figure 1, 2, 3 & 4**), it can be concluded that the population size, crossover probability, and tournament size do not have a **significant impact** on the fitness achieved. No matter how much these parameters are changed within the specified ranges, the generational population progression does not show much variation or a different trend. Although the tournament size makes the population achieve the desired fitness a little bit **faster** than the other parameters. On the other hand, the **mutation probability** has a significant impact on the fitness level. During the initial generation, a lower mutation rate has a faster fitness rate, but after that, the generation with the mutation rate of **0.01** has a higher fitness level than the others. Additionally, for different mutation probabilities, the average and best fitness plot shows **distinguishable differences** from the others.

Selection pressure does have an **influence on performance**. In the tournament vs. average fitness plot (**Figure 4**), more selection pressure makes generational populations achieve best fitness faster. Additionally, from **Figure 5**, we can infer that there are no solutions for lower tournament sizes, such as 2 and 3. All the solutions occurred when tournament sizes are 4 and 5, and mostly occurred when it was 5.

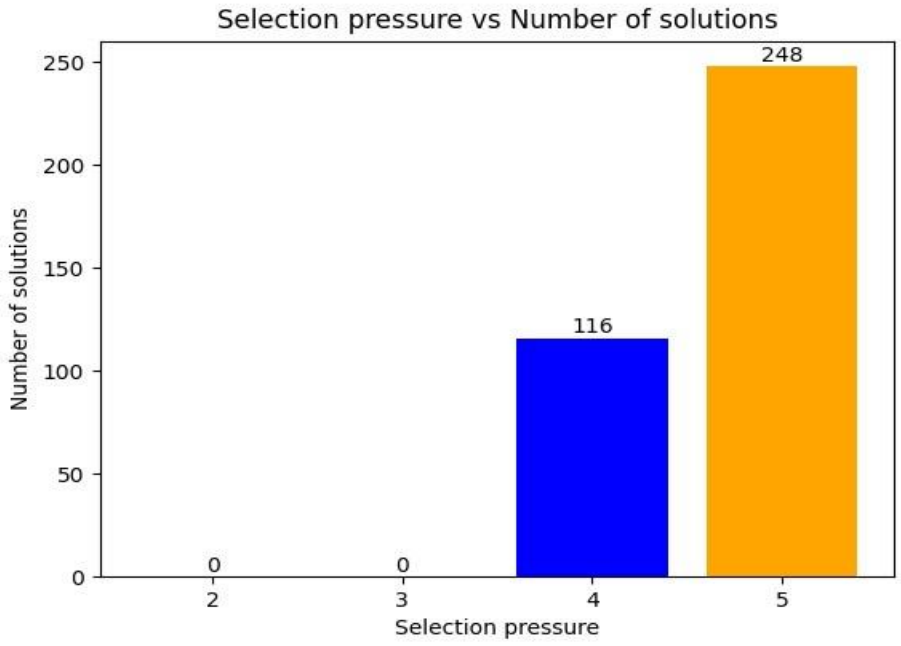

Figure 5: Selection pressure vs Solutions

Mutation and crossover are both important parameters for getting the best fitness through evolution. But mutation has more influence on getting the best fitness than crossover (**Figure 6**). In the experiment, no solution was found without mutation. On the other hand, three solutions were found without crossover.

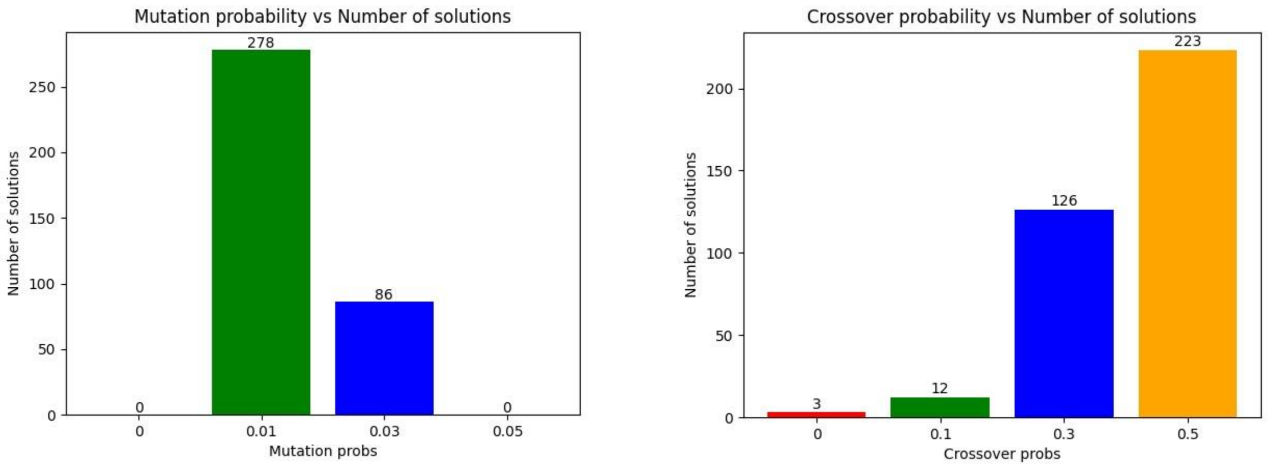

Figure 6: Mutation, Crossover and Solutions

The results show a strong **correlation** between mutation and crossover probabilities. Interestingly, there were only three solutions found (as shown in **Figure 7**) when there were no probabilities for either mutation or crossover. However, almost all the solutions were found for any mutation and any crossover probabilities.

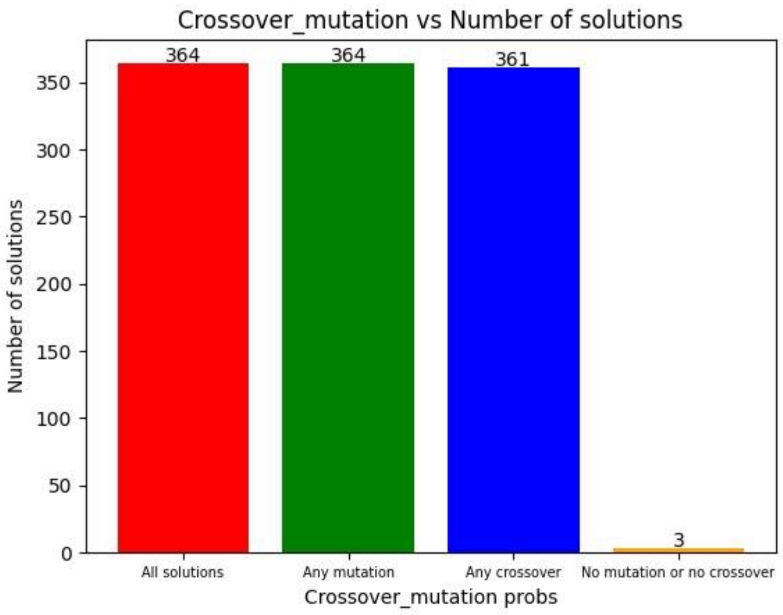

Figure 7: Correlation between mutation and crossover

Regarding the **performance** (faster saturation to maximum values), different **population sizes** do not show significant differences. But we can see in **Figure 8**, population size 75 has a relatively higher rate of convergence than the other sizes.

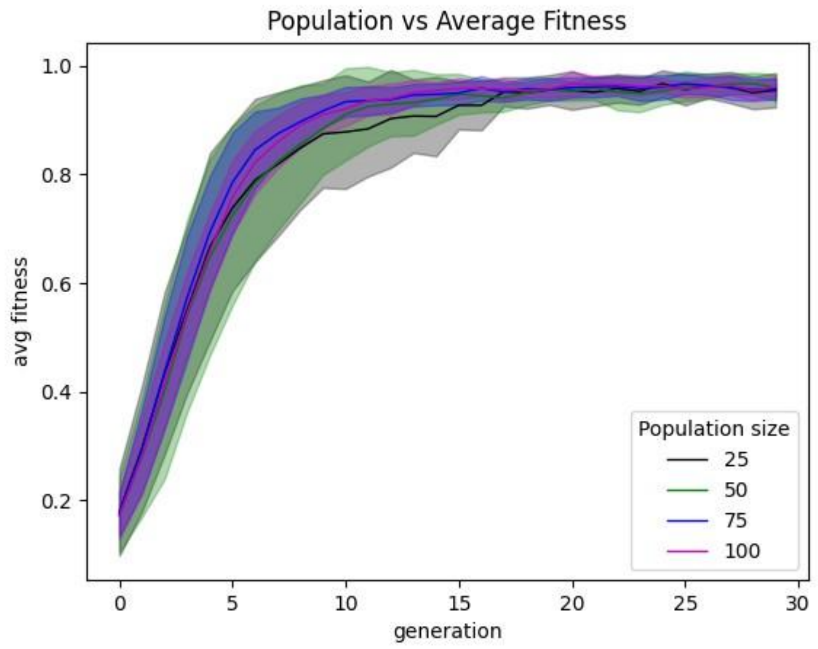

Figure 8: population vs Average fitness

In **Figure 9**, the impact of different parameters on the **diversity of the populations** is investigated. For both “population vs. diversity” and “crossover vs. diversity” plots, the trend is the same meaning that higher population and crossover values make the populations more diverse. Although this was also the same for “mutation vs. diversity” plot, the main difference is that mutation has a much higher impact on diversity than population and crossover. In our data, when the mutation rate is 0, diversity is also 0 most of the time. On the other hand, with the highest mutation rate, diversity was the highest, meaning very little similarity among the generated populations. Moreover, regarding tournament size, higher selection size creates the possibility of getting a similar population in the next generation. That’s why we got less diversity when the selection pressure was higher, and more diversity when the selection pressure was low.

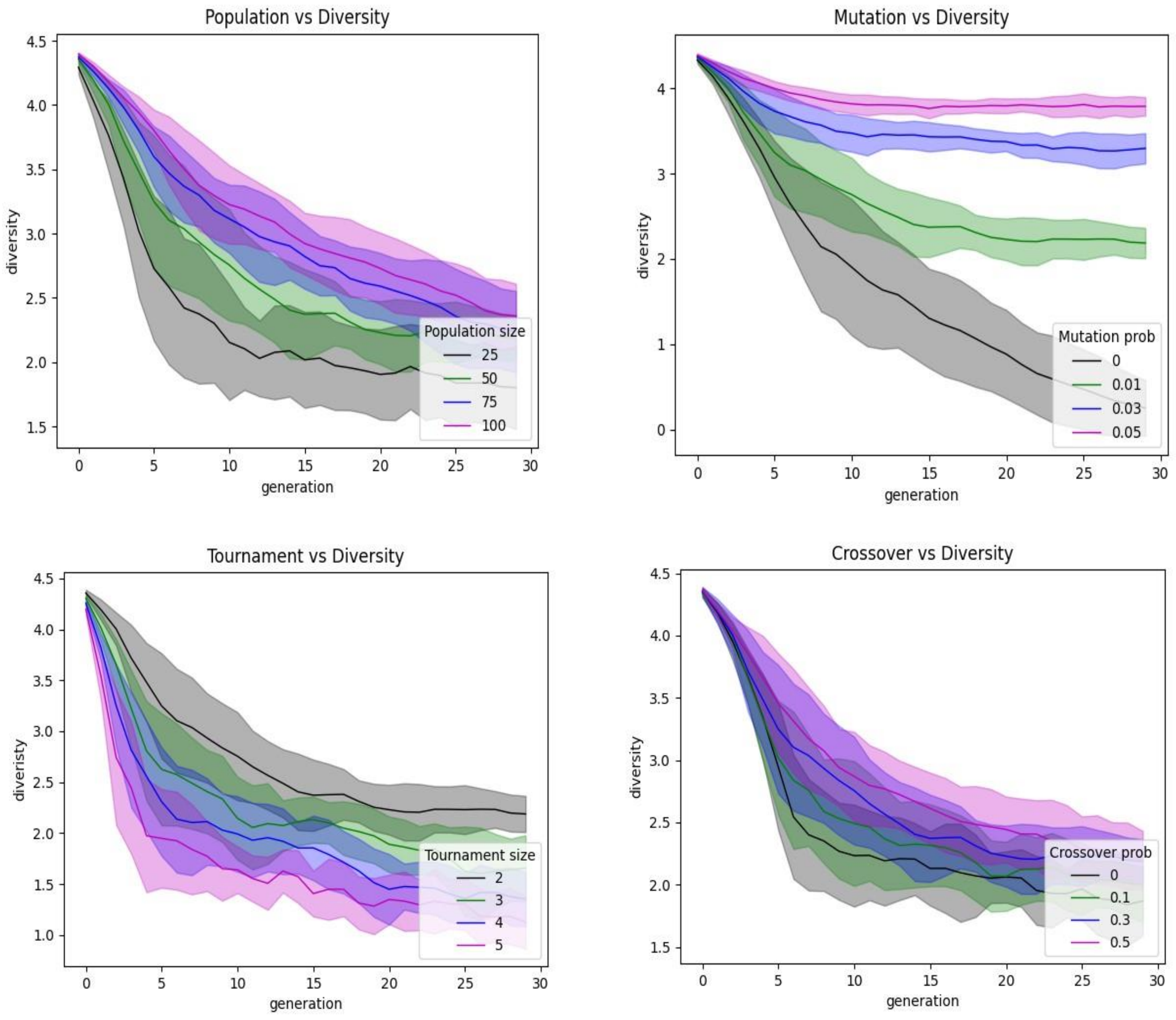

Figure 9: Diversity vs Four different parameters