Genetic algorithm report.

By Sakharov Dmitrii, Vladyslav Dobush.

Introduction

Genetic algorithms (GAs) are a class of optimization techniques inspired by natural evolution. These algorithms are widely used for solving complex optimization problems by evolving a population of candidate solutions over successive generations.

In this study, we investigate the effects of population size, mutation rate, and the use of the crossover operator on the efficiency of genetic algorithms. Our goal is to determine the optimal parameter settings to evolve the phrase "HELLO WORLD." By analyzing these variables, we aim to understand how they influence the performance and convergence rate of GAs for this specific problem.

Influence of size of the population on the number of generations needed to find a solution

Increasing the population size leads to a decrease in the number of generations required to find a solution. This implies that with a larger population, the algorithm identifies the solution more quickly. For example, with a population size of 50, it took about 50 generations to find the solution, but with a population size of 1000, it only took about 7 generations. This significant reduction in the number of generations indicates the positive impact of a larger population on the efficiency of the genetic algorithm.

A larger population size helps the genetic algorithm find solutions more quickly. This is because a bigger population has more diverse individuals, which makes it easier to find the right solution. In our experiments, we observed that increasing the population size consistently reduced the number of generations required to converge to the solution. A diverse population means that the algorithm has more potential solutions to work with, increasing the likelihood of finding the optimal solution faster. This diversity is crucial for maintaining genetic variation within the population, which enhances the algorithm's ability to explore the solution space effectively.

Overall, having a larger population size is beneficial for the genetic algorithm. It increases the diversity of the population, which improves the chances of finding the optimal solution more efficiently. This is illustrated by the graph (Appendix, FIGURE 1) showing the relationship between population size and the number of generations needed to find the solution.

By providing a broader range of genetic material, larger populations allow the genetic algorithm to explore more possible solutions and avoid getting stuck in local optima. Therefore, increasing the population size is a straightforward and effective way to enhance the performance of genetic algorithms.

Influence of mutation rate on the number of generations needed to find a solution

Increasing the mutation rate has a complex impact on the number of generations required to find a solution. This implies that with an optimal mutation rate, the algorithm identifies the solution more quickly. For example, with a mutation rate of 0.010, it took about 55 generations to find the solution, but with a mutation rate of 0.050, it only took about 16 generations, the same situation with a mutation rate of 0,100. However, at a higher mutation rate of 0.200, it took about 36 generations. This variation indicates the significant impact of the mutation rate on the efficiency of the genetic algorithm.

An optimal mutation rate helps the genetic algorithm find solutions more quickly. This is because an appropriate mutation rate introduces necessary diversity into the population, preventing premature convergence to suboptimal solutions. In our experiments, we observed that increasing the mutation rate up to a certain point consistently reduced the number of generations required to converge to the solution. A balanced mutation rate means that the algorithm can maintain diversity without disrupting good solutions too frequently, increasing the likelihood of finding the optimal solution faster. This balance is crucial for maintaining genetic variation within the population, which enhances the algorithm's ability to explore the solution space effectively.

Overall, having an optimal mutation rate is beneficial for the genetic algorithm. It increases the diversity of the population without introducing excessive randomness, which improves the chances of finding the optimal solution more efficiently. This is illustrated by the graph (Appendix, FIGURE 2) showing the relationship between mutation rate and the number of generations needed to find the solution.

After conducting the experiments we find out that the optimal mutation rate with the size of the population of 100 individuals is from 0,050 to 0,100.

By providing a controlled amount of genetic variation, an optimal mutation rate allows the genetic algorithm to explore more possible solutions and avoid getting stuck in local optima. Therefore, selecting an appropriate mutation rate is a straightforward and effective way to enhance the performance of genetic algorithms.

Effect of excluding the crossover operator

The inclusion of the crossover operator significantly reduces the number of generations required to find a solution in genetic algorithms. With a population size of 100 and a mutation rate of 0.075, the algorithm took an average of 16 generations with crossover, compared to 112 generations without it.

The crossover operator combines genetic material from different individuals to create new offspring. This recombination process introduces new genetic variations and enhances diversity within the population. By mixing genes from parents, the crossover operator increases the chances of producing individuals with better fitness, thereby accelerating the search for optimal solutions, what is shown on the graph (Appendix, FIGURE 3) .

This substantial difference in convergence rates highlights the efficiency gained by using the crossover operator, which helps combine genetic material from different individuals, increasing diversity and enhancing the algorithm's ability to find optimal solutions more quickly.

Effect of removal mutations

The exclusion of the crossover operator significantly hampers the efficiency of genetic algorithms. With a population size of 100 and a mutation rate of 0, the algorithm failed to find the correct solution even after 10,000 generations. This failure is due to the initial population lacking the necessary genetic diversity.

The crossover operator is crucial as it combines genetic material from different individuals to create new offspring, introducing genetic variations and enhancing population diversity. Without crossover, the algorithm relies solely on the initial genetic pool and mutations, which is insufficient for effective exploration of the solution space.

The inability to find the correct solution after 10,000 generations without crossover highlights its critical role. The crossover operator increases the chances of producing fitter individuals, accelerating the search for optimal solutions. Its absence results in stagnation, as the algorithm cannot effectively combine and improve genetic traits.

The optimum parameters for the given task

The optimum parameters for the given task involve the inclusion of the crossover operator, a mutation rate ranging between 0,050 and 0,100 , and a population size of 1000. These settings ensure efficient convergence, significantly reducing the number of generations required to find the correct solution. The crossover operator enhances genetic diversity and accelerates the search for optimal solutions.

Appendix

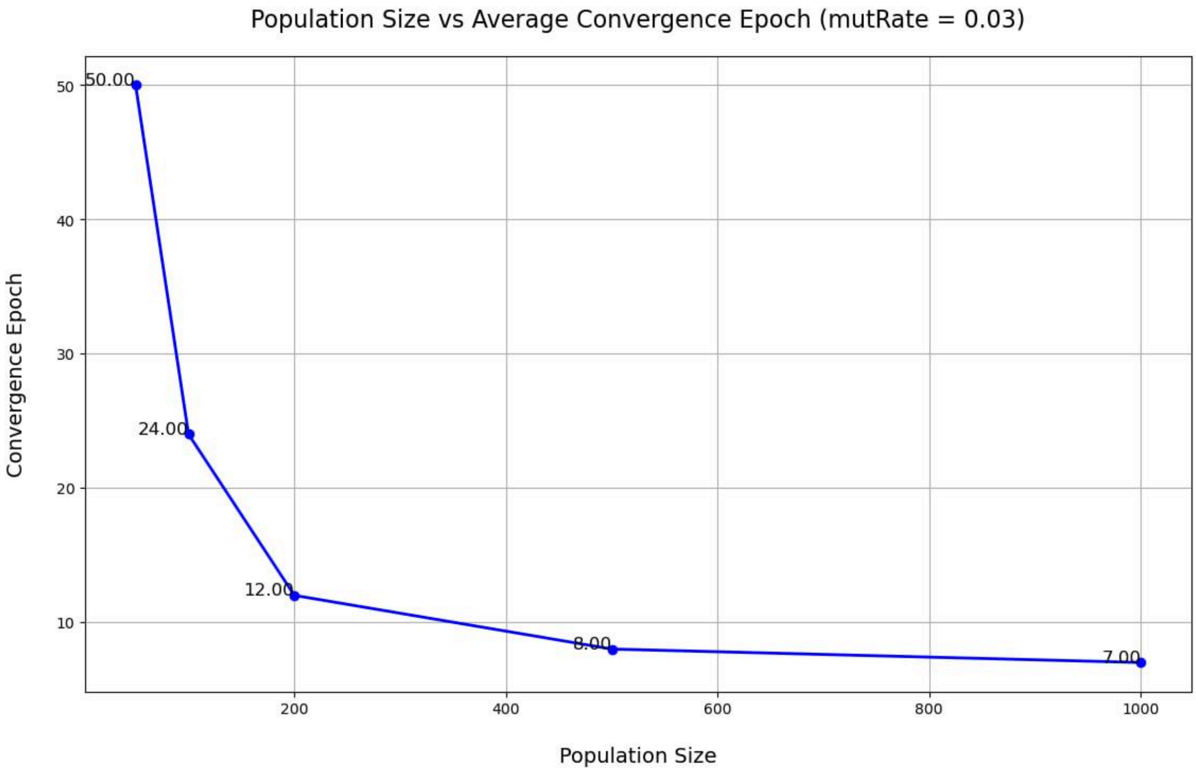

FIGURE 1 Line chart demonstrating the effect of population size on the number of generations needed to find the correct solution. All points on the graph are the average value of numbers of generations after conducting 30 experiments.

DATA:

population = 50, needed 50 generations;
population = 100, needed 24 generations;
population = 200, needed 12 generations;
population = 500, needed 8 generations;
population = 1000, needed 7 generations.

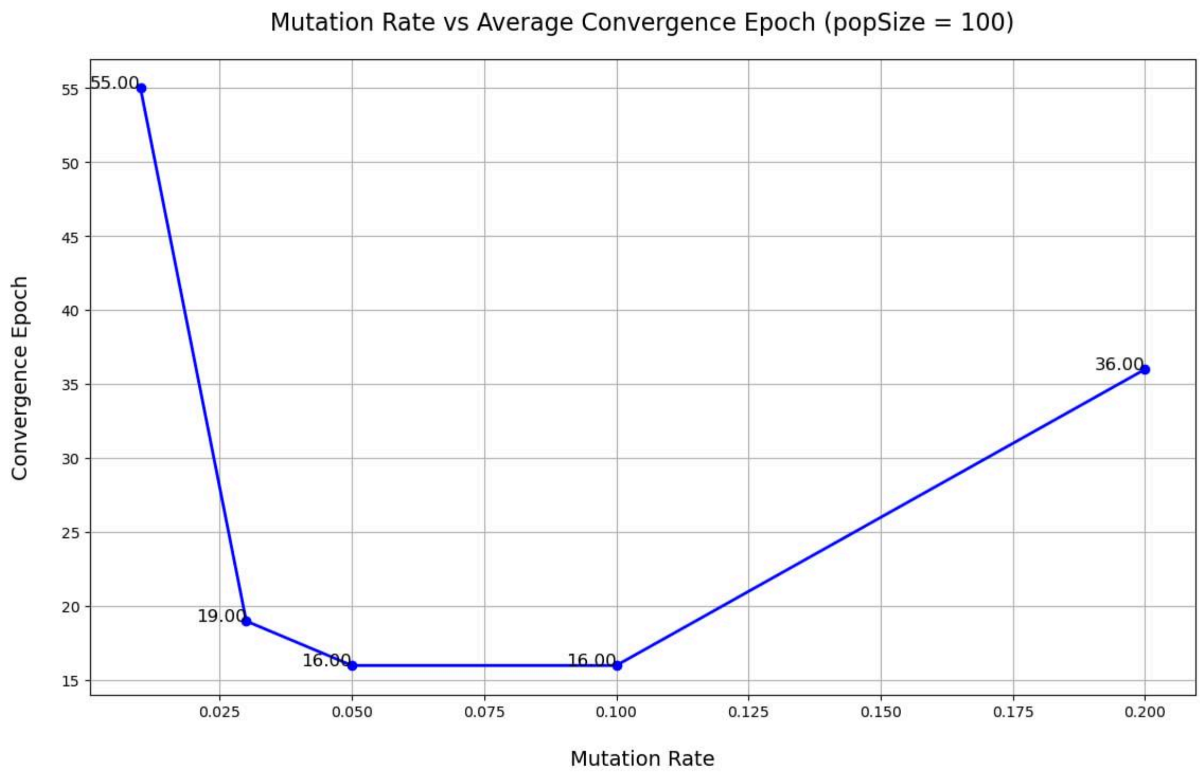

FIGURE 2 Line chart demonstrating the effect of mutation rate on the number of generations needed to find the correct solution. All points on the graph are the average value of numbers of generations after conducting 30 experiments.

DATA:

mutation rate = 0,010, needed 55 generations;
mutation rate = 0,030, needed 19 generations;
mutation rate = 0,050, needed 16 generations;
mutation rate = 0,100, needed 16 generations;
mutation rate = 0,200, needed 36 generations.

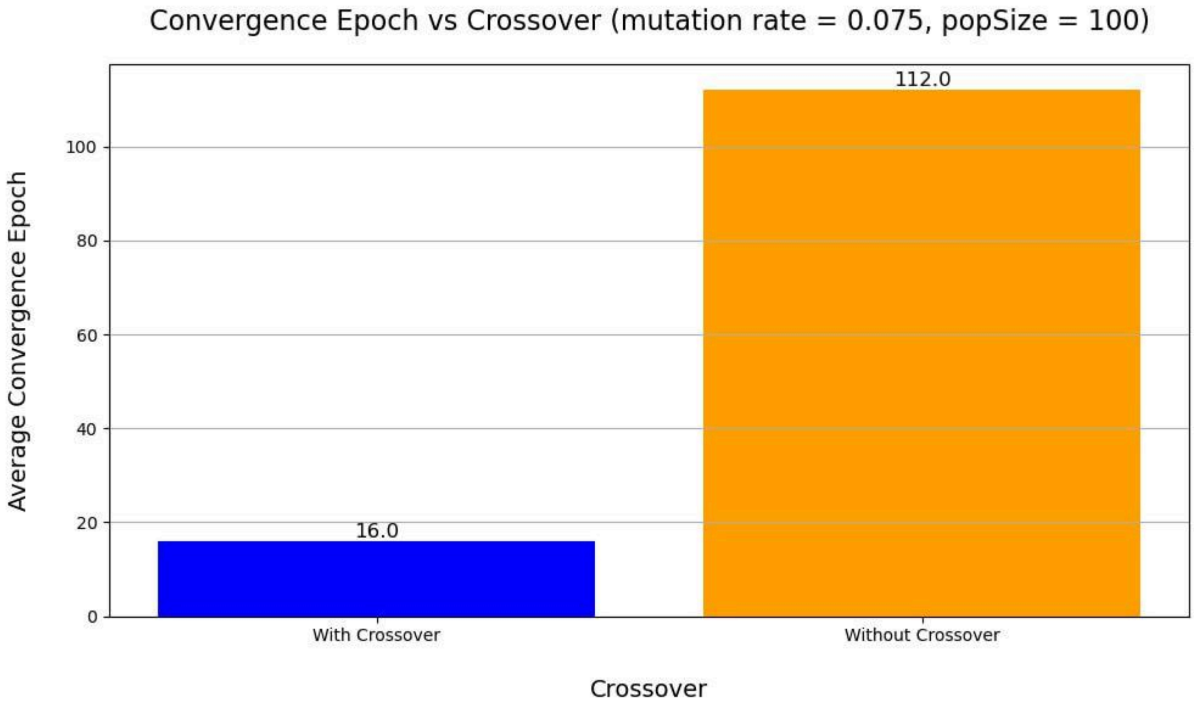

FIGURE 3 Bar chart demonstrating the effect of excluding the crossover operator on the number of generations needed to find the correct solution. All points on the graph are the average value of numbers of generations after conducting 30 experiments.

DATA:

with CrossOver = 16 generations (average value after 30 experiments);
without CrossOver = 112 generations (average value after 30 experiments);
population size = 100 individuals;
mutation rate = 0,075.