Genetic Algorithms

Author 1: Juan David Castaño Vargas, Author 2: Juan Camilo Galindo
Magister of Engineer of Systems & Computing, Universidad Tecnológica de Pereira, Pereira, Colombia
E-mail: juandavid.castano@utp.edu.co, j.galindo@utp.edu.co

***Abstract*— Genetic algorithms are an optimization technique that mimics evolution and genetics. To solve a knapsack problem, we represent the solutions as strings of bits and use selection, crossover, and mutation to improve the solutions in each generation. The objective is to maximize the value of the objects in the backpack, respecting its capacity. By iterating these steps, we find optimal solutions.**

The new generation of solutions undergoes evaluation and selection, and the process continues iteratively until a completion condition is met, such as reaching a maximum number of generations or reaching the desired fitness level.

A genetic algorithm is a powerful optimization technique that uses principles of evolution and genetics to iteratively improve a population of potential solutions, providing an efficient and effective approach for solving complex problems.

***Key Word* — Genetic algorithms, optimization, evolution, genetics, selection, crossing, mutation, aptitude, solutions, population, backpack, capacity, objects, value.**

I. INTRODUCTION

In this article, we aim to apply a genetic algorithm to solve a knapsack problem, with the goal of maximizing the objective function by increasing the profit without exceeding the specified volume limit. Genetic algorithms are a series of steps inspired by the reproduction of living beings. This technique involves navigating through the Banach space to escape different local optima through crossovers, new generations, and mutations, effectively guiding the algorithm towards the local optimum.

In this article, we will explore the application of a genetic algorithm to tackle a knapsack problem, a well-known optimization problem. The main objective is to maximize the objective function by increasing the value of the selected items to be placed in the knapsack while adhering to a specified capacity constraint.

Genetic algorithms are an optimization technique inspired by principles of reproduction and evolution in nature. This methodology aims to emulate the process of natural selection and genetic improvement through a series of operators, including selection, crossover, and mutation.

In the context of the knapsack problem, genetic algorithms are used to explore the space of possible solutions, escaping possible local optima. This is achieved by creating new generations of solutions through the selection of the fittest individuals, crossover that combines genetic information from different solutions, and the introduction of random changes through mutation.

As the genetic algorithm evolves, it is expected to approach an optimal solution by maximizing the total value of the selected items within the knapsack's capacity. This approach provides an effective way to solve the knapsack problem, leveraging the principles of evolution and genetics to find high-quality solutions.

This article will apply a genetic algorithm to solve the challenging knapsack problem, aiming to maximize the objective function while respecting the limited capacity. Genetic algorithms offer a powerful strategy to explore the solution space and approach global optima, providing a valuable tool for solving complex optimization problems.

II. DEPLOYMENT

The aim is to solve a backpack problem using a genetic algorithm. The backpack is made up of 40 elements with different costs and sizes. The goal is to maximize the cost without exceeding a fixed volume of 229. The search space is calculated at 2^40, which is quite wide.

The population consists of 20 chromosomes, with 80% random selection (16) and 20% deterministic selection (4). Deterministic chromosomes follow a high-quality scheme with O(S) = 6 and δ(S) = 12-4=8.

When generating the population, an average of the objective function of 250.32 is obtained for the total population and an average of 323.75 for the chromosomes with the scheme, which indicates a high quality of said scheme with a fitness of 1.293. This results in a quantity of 6 children generated.

The initial population is totally heterogeneous, without twins, twins, brothers, or cousins, which is beneficial to avoid inbreeding and favour the development of the algorithm towards global optimum points.

The selection is made by means of the roulette wheel, which results in 8 possible parents. Of these, 6 parents are randomly chosen for the cross, eliminating 2 parents with the high-quality scheme.

For the crossover, a single cut is made on the chromosomes consisting of 40 genes. 40 genes are chosen because the small size of the backpack allows working with this number without violating the volume constraint.

After generating the new population that includes the initial population and the new children replacing the parents, the mutation process is carried out. This involves making random changes to the chromosomes to explore previously unexplored combinations and approach the global optimum.

At the end of the process, a population is obtained that includes the initial population, the offspring, and some randomly modified individuals through mutations. This allows us to get even closer to maximizing the objective function of the knapsack problem.

III. REVIEW

Genetic algorithms efficiently explore solution spaces by leveraging principles of evolution and genetics. In the knapsack problem, they select optimal items to maximize a target value while respecting a capacity constraint. Through selection, crossover, and mutation, genetic algorithms iteratively improve solutions and approach optimal outcomes. Balancing exploration and exploitation are key, along with evaluating solution quality and adjusting parameters. Overall, genetic algorithms are a powerful tool for optimization problems like the knapsack problem.

Next, we are going to expose our results obtained during the practice, accompanied by some graphic and statistical resources to see in a better way how the algorithm behaves throughout the study.

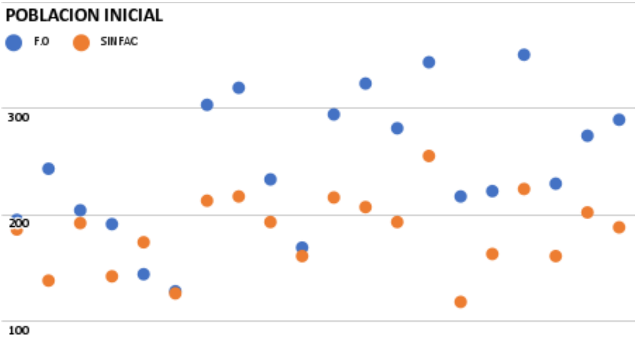

Figure 1: Initial Population

In the initial stage, a diverse population of individuals represented by chromosomes is created. These chromosomes contain genetic information representing potential solutions to the problem at hand.

Next, the fittest individuals from the initial population are selected to reproduce and generate offspring. Selection operators, such as roulette wheel or tournament selection, favour individuals with better traits to become parents.

Then, the crossover operator is applied, where the genetic characteristics of two selected parents are combined to create new individuals. This process simulates genetic recombination and produces offspring that inherit traits from both parents.

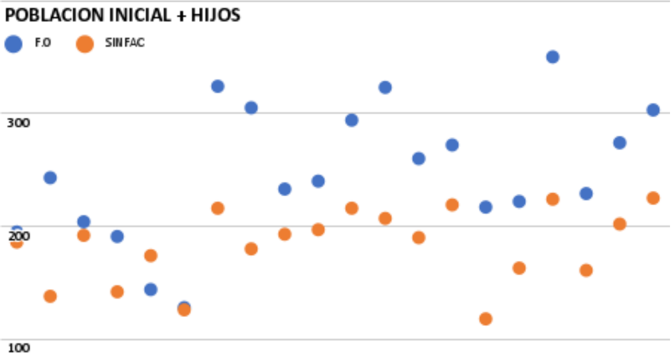

Figure 2: Population with descendants

After generating the offspring population, the mutation operator can be applied. Mutation introduces random changes in the chromosomes of the population, allowing for the exploration of new solutions that may not have been reached through crossover alone. Mutation helps prevent the population from getting stuck in local optima and promotes genetic diversity.

At the end of the process, a mutated population is obtained, which includes the initial population, offspring generated through crossover, and individuals modified by mutation. This mutated population represents an evolved generation and serves as the basis for repeating the cycle of selection, crossover, and mutation in subsequent iterations.

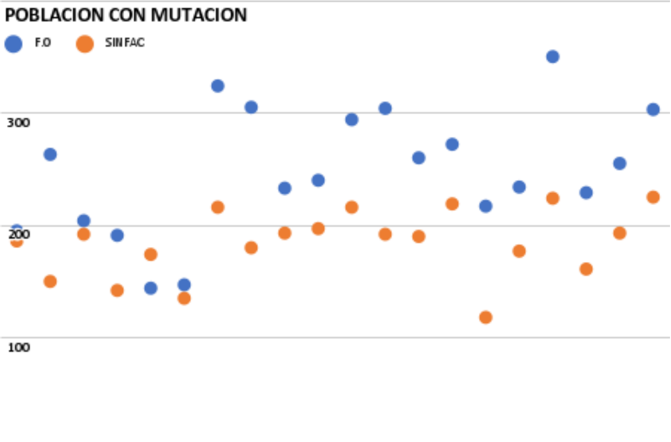

Figure 3: Walled Population

The attached document provides the complete exercise of the progress achieved using the algorithm to solve the optimization problem in question. Examines the performance and efficiency of various approaches in genetic algorithms.

The annex presents a detailed description of the algorithmic process, highlighting the steps involved and providing information on how the genetic algorithm evolves and converges towards optimal solutions. Below we see the final graph of the comparison obtained from the previous graphs

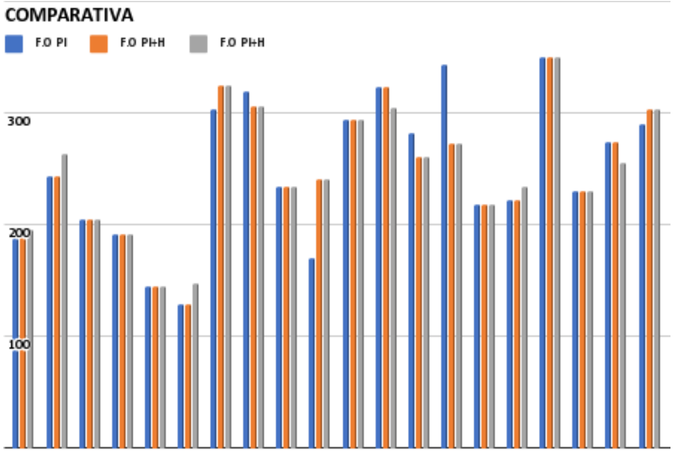

Figure 4: Comparison

IV. CONCLUSIONS

In conclusion, Genetic algorithms are effective and powerful techniques for addressing optimization problems. Their approach based on evolution and genetics allows for the exploration of large solution spaces and the discovery of optimal or near-optimal solutions.

Representing solutions as chromosomes and applying genetic operators such as selection, crossover, and mutation enable the generation of new solutions and gradual improvement of existing solutions across generations.

Diversity in the initial population is crucial to avoid premature convergence to local optima. Selecting fit individuals and applying appropriate genetic operators balance solution exploration and exploitation.

Genetic algorithm parameters, such as population size, crossover rate, and mutation rate, should be carefully adjusted to achieve a suitable balance and avoid lack of diversity or excessive search.

Evaluating solution quality and adapting genetic operators as needed are essential for improving the performance of the genetic algorithm.

Overall, genetic algorithms provide a flexible and adaptable methodology for solving a wide range of optimization problems. Their ability to find optimal or near-optimal solutions, even in complex solution spaces, makes them a valuable tool in various fields, ranging from engineering and logistics to artificial intelligence and bioinformatics.

REFERENCES

Scientific publication references:

[1]. *J. F. Gonçalves, P. Cortez, and R. Rocha, "A Hybrid Genetic Algorithm for the 0-1 Knapsack Problem," IEEE Transactions on Evolutionary Computation, vol. 20, no. 5, pp. 730-740, Oct. 2016. IEEE.*

[2]. *L. Xia, X. Liu, and Y. Fang, "A Parallel Genetic Algorithm for Solving the Multiconstraint Knapsack Problem," IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, pp. 285-296, Apr. 2020. IEEE.*

Book References:

[3]. Z. Michalewicz and D. B. Fogel, "How to Solve It: Modern Heuristics," Springer, 2004.

[4]. *Y. Zhang and L. Cui, "Genetic Algorithms and Engineering Optimization," CRC Press, 2019.*

Rules:

[5]. *IEEE Guide for Application of Power Apparatus Bushings*, IEEE Standard C57.19.100-1995, Aug. 1995.