Genetic Algorithms to Solve Traveling Salesman Problem and Vehicle Routing Problem

Bianca-Alexandra Ghiorghiu

***Abstract*—This article discusses the use of Genetic Algorithms (GAs) to solve the Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP), two complex optimization problems with significant computational intensity. The article provides an introduction to these problems, their core principles, and the proposed methodology. The methodology section presents the adaptation of Genetic Algorithms used for TSP and VRP, detailing workflows and experimental setups. The results and discussion section evaluates performance metrics across various scenarios, highlighting both the benefits and limitations of the approach. The article concludes with findings, potential improvements, and broader applications, suggesting pathways for future research in optimization problems.**

***Index Terms*—Genetic Algorithms, Optimization, Traveling Salesman Problem, Vehicle Routing Problem**

I. INTRODUCTION

THIS paper explores the use of Genetic Algorithms in solving two challenging optimization problems: the Traveling Salesman Problem and the Vehicle Routing Problem. Both problems, involve finding the most efficient routes under specific constraints. The Travelling Salesman Problem focuses on the shortest path for visiting multiple locations and returning to the origin, while the Vehicle Routing Problem optimizes routes for a fleet of vehicles tasked with multiple deliveries. The results of this project are intended to offer insight on the potential of Genetic Algorithms to enhance decision-making in a variety of areas that face similar routing challenges.

A. *Background on topics challenges*

The Traveling Salesman Problem and the Vehicle Routing Problem are key challenges in optimization due to their NP-hard nature, especially as the problem size increases. The TSP involves finding the shortest route through a set of locations, and its solution complexity grows exponentially with more locations, making exhaustive search methods impractical for large datasets [1].

The Vehicle Routing Problem further complicates matters by involving multiple vehicles with capacity constraints and varying customer demand. The challenges of these problems highlight the necessity for innovative and efficient algorithms capable of handling vast search spaces and adapting to dynamic conditions to find near-optimal solutions [2].

B. *Overview of genetic algorithms and their relevance to the addressed problem*

Genetic Algorithms are a powerful solution to optimization challenges, mimicking natural selection to evolve solutions towards greater fitness. They can efficiently navigate complex problems like TSP and VRP, generating high-quality solutions for large and complex datasets. Genetic Algorithms' flexibility and adaptability make them effective in handling permutations and combinations in routing problems [3]. This exploratory capability is crucial in discovering near-optimal solutions, where exhaustive enumeration of all possible routes is computationally impractical. Genetic Algorithms not only address the inherent complexity of these problems but also provide a scalable and efficient framework, potentially transforming operational strategies in industries reliant on sophisticated routing and scheduling [4].

II. PROBLEM DEFINITION

A. *Description of the specific problem under study*

This project, will explore the application of Genetic Algorithms to solve two optimization problems: the Traveling Salesman Problem and the Vehicle Routing Problem. The project will proceed in a structured manner, starting with TSP as the initial case. Then, investigation will be extended to the more complex VRP. The primary focus will be on assessing the effectiveness of Genetic Algorithms as a solution approach for these optimization challenges.

B. *Objectives and constraints*

The primary objectives of this project are as follows:

- Implement Genetic Algorithms to find solutions for TSP and VRP.
- Investigate the effectiveness of Genetic Algorithms in solving these optimization problems.
- Compare the solutions obtained using Genetic Algorithms with known benchmark solutions or other heuristic methods.

Constraints for this project include:

- The use of Genetic Algorithms as the primary optimization technique.
- The availability of data representing the cities, distances, and customer demands for both TSP and VRP.
- The need to evaluate the quality of solutions in terms of the total distance traveled and the feasibility of routes in the case of VRP.

III. CORE TOPIC

A. *Basics of genetic algorithms, traveling salesman and vehicle routing problem*

*1) **Genetic Algorithms (GAs):*** Genetic Algorithms (GAs) are a class of optimization algorithms inspired by the process of natural selection and evolution. Developed by John Holland in the 1960s, GAs are designed to find approximate solutions to complex optimization and search problems. They are particularly useful when traditional optimization methods struggle due to the high dimensionality or non-linearity of the problem.

The key concepts in Genetic Algorithms are [5]:

Population: In a GA, a population of potential solutions (often called chromosomes or individuals) is maintained. Each individual represents a possible solution to the problem.

Selection: Individuals in the population are selected for reproduction based on their fitness, which measures how well they solve the problem. Better solutions have a higher chance of being selected.

Crossover (Recombination): During reproduction, pairs of individuals exchange information to create new offspring. This mimics the genetic recombination observed in biological evolution.

Mutation: Occasionally, random changes are introduced into the offspring's genetic information to encourage exploration of the solution space.

Termination: The algorithm continues to evolve solutions through multiple generations until a termination criterion is met, such as a maximum number of iterations or a satisfactory solution.

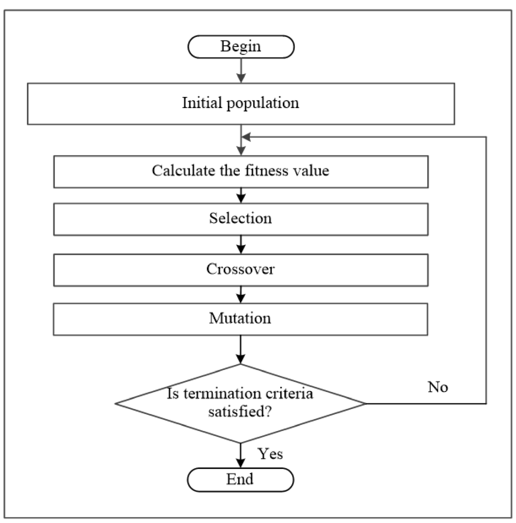

Fig. 1: Genetic Algorithm Diagram [5]

*2) **Travelling Salesman Problem (TSP)**:* The Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem. In this problem, a salesperson is given a list of cities, and the goal is to find the shortest possible route that visits each city exactly once and returns to the starting city [1]. This problem can be mathematically defined as follows [6]:

Given:

- A set of cities C represented by their coordinates in a two-dimensional space.
- A distance or cost function $d(i, j)$ that gives the distance between city i and j.

Find a permutation of cities π that minimizes the total distance traveled, i.e., find π that minimizes:

$$\sum_{i=1}^{n-1} = d(\pi_i, \pi_{i+1}) + d(\pi_n, \pi_1) \quad (1)$$

*3) **Vehicle Routing Problem (VRP)**:* The Vehicle Routing Problem (VRP) is a classic combinatorial optimization problem. In this problem, a fleet of vehicles is tasked with delivering goods to a set of customers from a central depot, aiming to minimize the total transportation cost [2]. This problem can be mathematically defined as follows [7]:

Given:

- A set of customers C, each with a demand for goods and represented by their coordinates in a two-dimensional space.
- A central depot located at coordinates $(0, 0)$.
- A fleet of vehicles with limited capacity, denoted by K.
- A distance or cost function $d(i, j)$ that gives the distance between customer i and customer j.

Find a set of routes for the vehicles, denoted by $\mathcal{R} = \{R_1, R_2, \ldots, R_K\}$, where each route R_k is a permutation of customers π_k such that:

- Each customer is visited exactly once by a single vehicle.
- The total demand of customers in each route does not exceed the vehicle's capacity.
- The total cost, representing the sum of distances traveled by all vehicles, is minimized:

$$\sum_{k=1}^{K} \sum_{i,j \in R_k} d(i, j) \quad (2)$$

The objective is to find the optimal set of routes $\mathcal{R}$ that satisfies these conditions and minimizes the total transportation cost.

B. Advantages of genetic algorithms for traveling salesman and vehicle routing problem

Genetic Algorithms offer several distinct advantages when applied to solving the Traveling Salesman Problem and the Vehicle Routing Problem:

Efficient Exploration: Genetic Algorithms efficiently explore large solution spaces, making them suitable for finding near-optimal solutions in complex scenarios [8].

Flexibility: Genetic Algorithms can adapt to diverse problem constraints and variations encountered in real-world TSP and VRP instances [9].

High Solution Quality: Genetic Algorithms consistently produce high-quality solutions, offering practical value even when finding the absolute optimum is challenging [1].

Multi-Objective Handling: Genetic Algorithms extend easily to multi-objective optimization, addressing conflicting goals in certain vehicle routing problem scenarios [8].

Escaping Local Optima: Genetic Algorithms include mechanisms to escape local optima, ensuring robust solution discovery [8].

Parallelization Potential: Genetic Algorithms can be parallelized, reducing solution times for large-scale TSP and VRP instances [9].

C. Previous work

Recent advancements in the application of Genetic Algorithms for solving the traveling salesman problem and the vehicle routing Ppoblem have shown significant promise.

Firstly, when it comes to the traveling salesman problem, Alkafaween, E. A. et al. propose IAM-TSP [10], a new method providing an approximate solution to TSP in polynomial time. It starts by adding four extreme cities to the route, followed by adding each city using a greedy technique. The method evaluates the cost of adding each city to different positions along the route. The resultant route is further improved by employing local constant permutations.

Mavrovouniotis M., Müller F. M., Yang S. paper [11] presents a memetic Ant Colony Optimization (ACO) algorithm with a local search operator for dynamic TSP. The algorithm is designed to address both symmetric and asymmetric dynamic TSPs. The integration of local search operators significantly improves the performance of ACO, showing efficiency in solving dynamic TSPs.

Phu-Ang A. introduces a hybrid approach [12] based on simulated annealing (SA) for TSP. It embeds Tabu search in SA and uses two swap techniques to enhance performance. The proposed algorithm shows effectiveness over other methods in terms of average percentage deviation from the lower bound criteria.

An article by François A. et al. addresses the limitations of Machine Learning (ML) approaches for solving TSP [13]. It proposes a new metric, ratio of optimal decisions (ROD), for evaluating the accuracy of ML methods in solving TSP. The paper investigates the impact of a search procedure plugged inside an ML model on performances, providing insights into the current state of ML approaches for TSP.

In terms of vehicle routing problem, Wang L., et al. proposes a multi-objective vehicle routing optimization algorithm improved from PMOCO. It incorporates a weight adjustment method in PMOCO [14], allowing it to adapt to different approximate Pareto fronts and find solutions with better quality. The method is trained through deep reinforcement learning and shows about a 6% improvement compared with PMOCO with 20 preferences.

Muniasamy, R. P. et al.proposed a novel technique combining local search and randomization for solving the Capacitated Vehicle Routing Problem (CVRP) faster with reasonable accuracy, even on large problem instances [15]. The method is experimentally compared with state-of-the-art GPU implementations, demonstrating its efficacy. The sequential and shared-memory parallel implementations are significantly faster than GPU-parallel Genetic Algorithms while also achieving superior solution quality.

Stodola P. discusses a hybrid Ant Colony Optimization algorithm applied to the Multi-Depot Vehicle Routing Problem (MDVRP). It combines bio-inspired techniques with simulated annealing principles and explores the search space with deterministic local optimization [16]. The algorithm outperforms other methods with the smallest average error on benchmark instances.

These studies collectively highlight the diverse approaches and significant advancements in using Genetic Algorithms to solve TSP and VRP, showcasing their potential in real-world applications.

IV. Proposed Methodology

A. Adaptation of genetic algorithms to solve travelling salesman and vehicle routing problem

1) Traveling Salesman Problem:

Representation: In this implementation, each individual (solution) is represented as a sequence of city indices, denoting the order in which the cities are visited.

Initialization: The initial population is generated with random permutations of city indices, ensuring diverse starting points for the algorithm.

Fitness Function: The fitness function calculates the total (euclidean) distance of the tour. The aim being to minimize this distance.

Selection: The code uses tournament selection for choosing parents for reproduction. In this method, a fixed number (tournament size, here 3) of individuals are randomly selected from the population, and the best among them (based on fitness) is chosen. This process is repeated until the offspring population is filled.

Crossover: The algorithm employs Partially Matched Crossover (PMX) for generating new offspring. In PMX, segments from two parent tours are chosen and swapped between them. This swap respects the order and position of cities in the segment, ensuring that each city appears only once in the offspring. PMX is particularly well suited for TSP as it ensures valid routes in the offspring while combining features of both parent tours.

Mutation: The mutation operation method randomly shuffles the order of cities within a tour. It is applied to each offspring based on a predefined probability. The extent of shuffling is controlled by an individual probability parameter (indpb), which dictates the likelihood of each city in the tour being repositioned. This technique introduces significant diversity into the population, aiding in the exploration of the solution space and helping to prevent the algorithm from getting stuck in local optima.

Termination: The algorithm terminates after a set number of generations or if a satisfactory solution is found.

2) Vehicle Routing Problem:

Representation: Each chromosome represents a possible solution to the VRO and consists of two parts: a schedule of cities (clients) to visit and a corresponding vehicle assignment for each city. The schedule part lists the sequence in which cities are visited, and the vehicle part indicates which vehicle is used for visiting each city.

Initialization: The initial population is generated with the chromo_create function, which randomly assigns cities to vehicles and shuffles the visit order. This ensures a diverse range of initial solutions.

Fitness Function: The chromo_eval function calculates the total distance traveled by all vehicles. Each route's distance is computed using the calc_route_cost function, summing up the distances between consecutive cities and including the return to the depot.

Selection: The selection process uses tournament selection to choose individuals for breeding, ensuring that fitter individuals have a higher chance of being selected.

Crossover: The crossover function combines the genetic information of two parent chromosomes to create offspring. It uses a custom crossover technique involving partially matching genes in the schedule part and swapping genes in the vehicle assignment part.

Mutation: The mutation function introduces variability in the population. It randomly chooses between two mutation types: swap_gene, which swaps two genes in the chromosome, and shuffle_gene, which shuffles a segment of genes. This helps in exploring new solutions and preventing premature convergence.

Termination: The algorithm terminates after a set number of generations or if a satisfactory solution is found.

B. *Workflow description*

1) ***Traveling Salesman Problem:***

Import Libraries: Essential libraries such as numpy, matplotlib, seaborn, sklearn, and deap are imported for various functionalities ranging from data handling to visualization and genetic algorithm implementation.

Generate Data: The project involves generating random coordinates for n cities and plotting them. A distance matrix between these cities using Euclidean distance is also computed.

Function Definition: Functions are created for generating chromosomes (tours) and evaluating their fitness based on the total distance of the tour.

Solving with Genetic Algorithms: The genetic algorithm parameters are set, including population size, number of generations, and probabilities for crossover and mutation. An initial population of potential solutions is generated, and their fitness is evaluated. The population undergoes evolution through selection, crossover, and mutation, with continuous tracking of the best solution in each generation.

Results and Visualization: The optimization process is visualized by plotting the best fitness values over generations. The final solution, which is the best tour found, is printed and visualized on a plot, showing the path connecting all the cities.

2) ***Vehicle Routing Problem:***

Import Libraries: Starts by importing the necessary libraries including numpy, matplotlib, seaborn, sklearn, scipy and deap.

Generate Data: The process involves generating data for n cities, which includes a depot and n-1 clients, along with setting the number of vehicles and their payload capacity. This step also includes plotting the cities layout and computing a Euclidean distance matrix between the cities.

Define Useful Functions: Functions are created for chromosome creation, evaluating chromosome fitness, crossover and mutation operations, and checking the feasibility of the solutions.

Solving with Genetic Algorithms: Genetic algorithm parameters such as population size, number of generations, crossover probability, and mutation probability are established. An initial population of potential solutions is generated, and their fitness is evaluated. The population undergoes evolution through selection, crossover, mutation, and feasibility checks, with continuous tracking of the best solution in each generation.

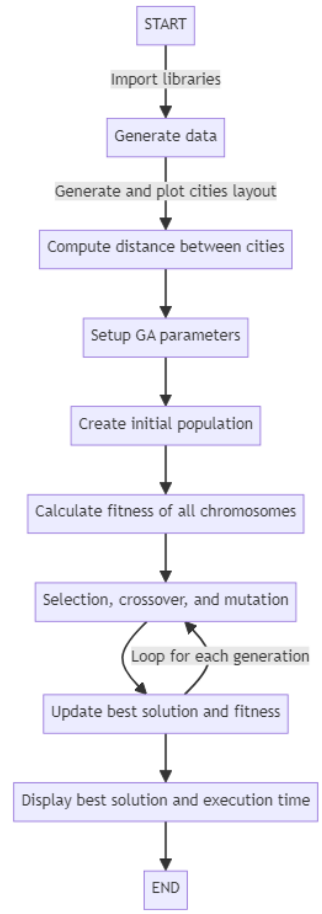

Fig. 2: TSP Workflow

Results and Visualization: The optimization process is illustrated by plotting the best fitness values across generations. The final solution, which is the best set of vehicle routes found by the algorithm, is printed and visualized on a plot, showcasing the optimized paths connecting the depot to the clients.

V. IMPLEMENTATION

A. *Setup & datasets description*

1) ***Traveling Salesman Problem:*** For this project, it was chosen to work with a dataset of 15 cities. These cities were generated within a specified coordinate range, using the make_blobs function from the sklearn.datasets module, ensuring a realistic spatial distribution. Each city was assigned a unique identifier, and their coordinates were mapped in a dictionary for easy access.

Then, a distance matrix was computed, using the distance.cdist function from the scipy.spatial module, which calculates the Euclidean distance between each pair of cities. This matrix serves as a fundamental component for the evaluation

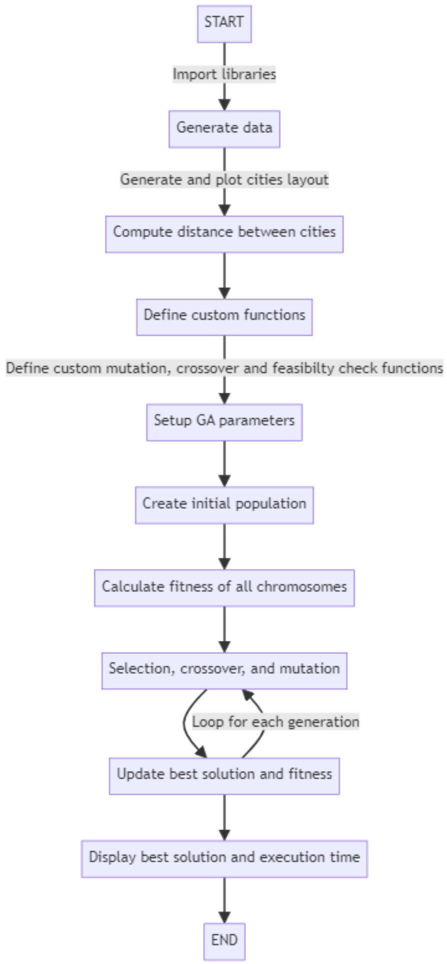

Fig. 3: VRO Workflow

of the genetic algorithm, as it allows the calculation of the total travel distance for any given sequence of cities.

*2) **Vehicle Routing Problem**:* For VRP, the dataset was extended to 21 cities, including a central depot and 20 client locations. Additional parameters such as the number of vehicles (4) and the payload capacity of each vehicle (7 units) were introduced. They are important for simulating a more complex real-world delivery scenario.

Similar to the TSP setup, the make_blobs function was used to generate random city coordinates. The cities were again assigned unique identifiers, and their coordinates were stored in dictionaries. One key difference in the VRP dataset was the categorization of cities into a depot and client locations, influencing the route planning and payload distribution among the vehicles.

*B. **Parameter settings***

*1) **Traveling Salesman Problem**:* In a series of tests, the optimal parameters for a genetic algorithm were determined for a specific problem context. The first test, seen in Fig. 4 revealed that a population size of 300 yielded the highest fitness. In the second test, it was found that the best performance is achieved at 750 generations, as seen in Fig. 5. The third test (Fig. 6) showed that a crossover rate of 0.4, or 40%, strikes a balance between exploring new solutions and retaining good existing ones. Lastly, in the fourth test, which can be observed in Fig 7, a high mutation rate of 0.7 was identified as the best setting, indicating that a significant amount of random variation is required in the problem space to find the optimal solution compared to typical settings.

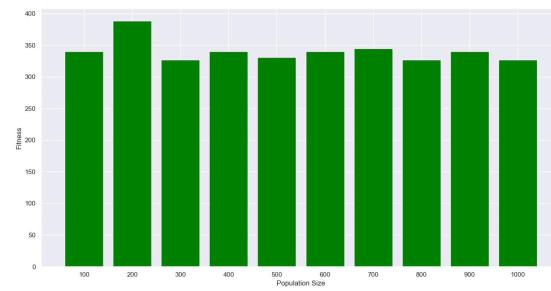

Fig. 4: TSP Population Size Test

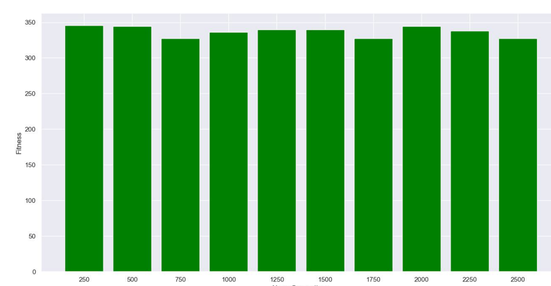

Fig. 5: TSP Number of Generations Test

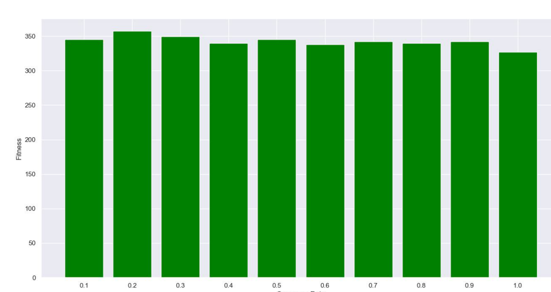

Fig. 6: TSP Crossover Rate Test

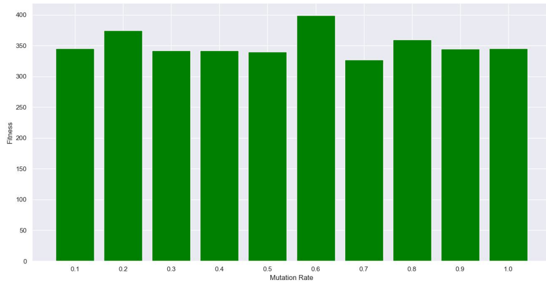

Fig. 7: TSP Mutation Rate Test

*2) **Vehicle Routing Problem**:* In a series of tests to optimize the genetic algorithm for a specific problem context, the following parameters were determined as optimal. The first test (Fig. 8) showed that a population size of 400 yields the best results. In the second test, pictured in Fig. 9, it was found that running the algorithm for 1250 generations allows for adequate evolution towards an optimal solution. Also, the optimal crossover rate was identified as 0.2 in Fig. 10. Lastly, a mutation rate of 0.5, was considered optimal (Fig. 11), suggesting that a higher level of random variation in the solution is beneficial for this problem.

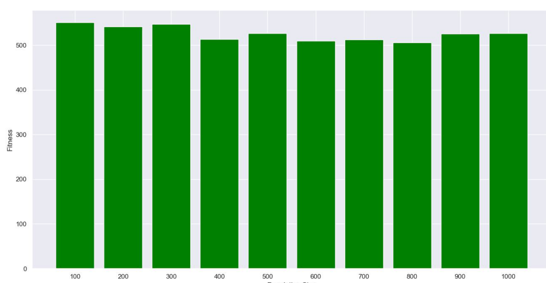

Fig. 8: VRO Population Size Test

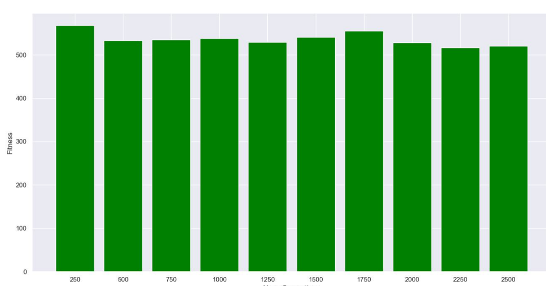

Fig. 9: VRO Number of Generations Test

C. *Comparison with traditional methods in the field*

The Traveling Salesman Problem and the Vehicle Routing Problem have long been studied in the field of combinatorial optimization. Traditional methods for solving these problems

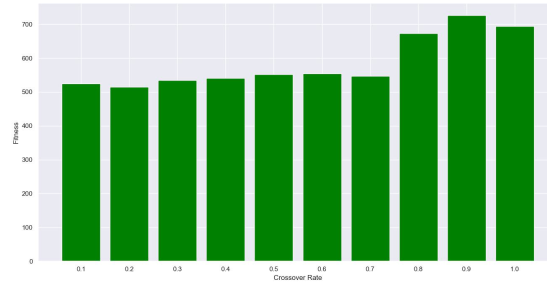

Fig. 10: VRO Crossover Rate Test

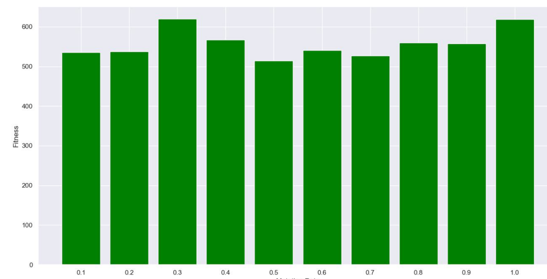

Fig. 11: VRO Mutation Rate Test

include exact algorithms, heuristics, and metaheuristics. Exact algorithms, such as dynamic programming and branch-and-bound, guarantee an optimal solution but are computationally expensive, especially for large instances. Genetic Algorithms are stochastic and provide near-optimal solutions efficiently, making them ideal for large-scale instances. Also, they offer a compromise between solution quality and computational resources, while heuristics like the Nearest Neighbor or Clarke-Wright Savings Algorithm are simple and quick but may not consistently yield high-quality results. Genetic Algorithms, on the other hand, use a population-based approach, exploring a wider search space and potentially discovering better solutions. They have the advantage of adaptability and can improve solutions over time through generations. Metaheuristic methods like Simulated Annealing and Tabu Search also offer a compromise between computational efficiency and solution quality, but GAs offer parallelism and are highly customizable, with various operators and parameters that can be fine-tuned for specific problem instances.

VI. Results and Discussion

A. *Performance metrics and evaluation criteria*

*1) **Performance Metrics**:* The primary performance metric in TSP is the total traveled distance, the goal being to minimize the distance covered while visiting each city exactly once and returning to the starting point. In contrast, VRO focuses on the total route cost, which factors in distance or time and considers vehicle capacities and client demands in the overall cost calculation. Furthermore, the computational time is a critical aspect, especially for larger TSP or VRP instances, as efficient solutions must provide the shortest path in the least amount of computational time to meet optimization objectives effectively.

*2) **Evaluation Criteria**:* In assessing solutions for TSP and VRO, several essential factors come into play. The foremost is the solution's quality, measured by total distance or cost. Next, we evaluate the solution's consistency: its ability to deliver reliable, high-quality results across various applications and problem types. Scalability is also a key consideration; the solution should perform effectively not just in smaller scenarios but also in larger, more complex ones. Finally, we assess the computational efficiency of the solution, focusing on how it manages computer resources like processing time and memory usage.

B. Analysis of results in different scenarios

*1) **Traveling Salesman Problem**:* Figures 12, 13, and 14 depict the results of TSP testing with 15, 30, and 45 cities, respectively. Additionally, table I provides a summary of the fitness and execution time performance metrics for each scenario

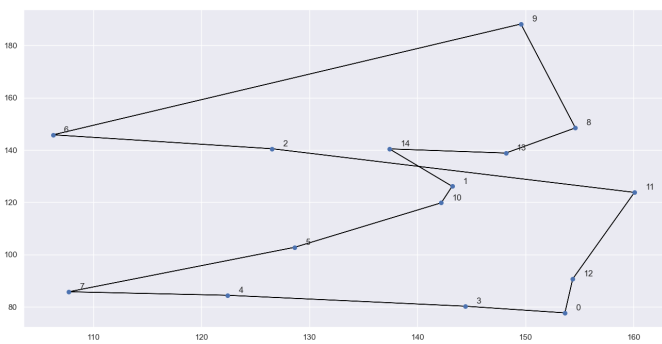

Fig. 12: TSP (15 cities)

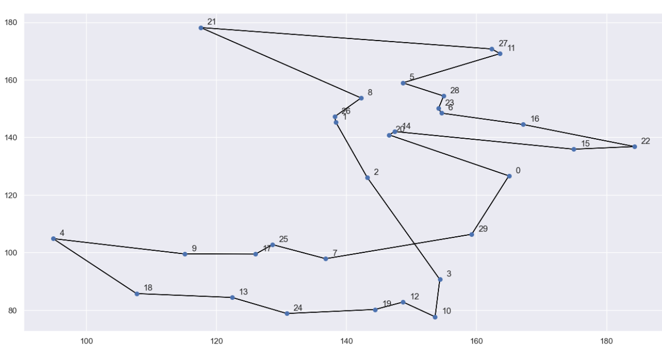

Fig. 13: TSP (30 cities)

Cities	Fitness	Time [s]
15	334	23.61
30	452	34.37
45	756	44.82

TABLE I: TSP Results

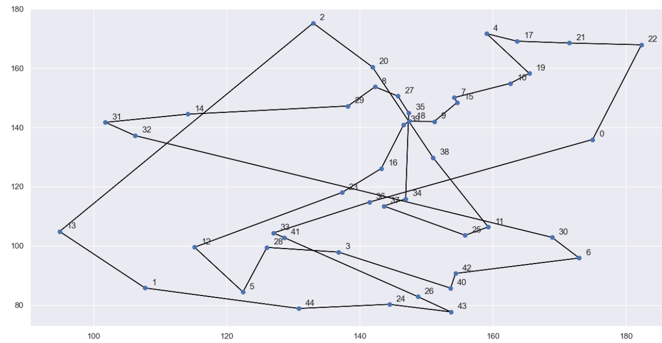

Fig. 14: TSP (45 cities)

*2) **Vehicle Routing Problem**:* Figure 15 illustrates the baseline scenario, involving 21 cities, 4 vehicles, and a payload of 7. In figure 16, we examine the scenario where the number of cities is expanded to 28, while figure 17 delves into the case with a reduced vehicle count of 3. Lastly, figure 18 portrays the scenario with an increased payload of 10. Furthermore, table II provides a comprehensive summary of the results, presenting the fitness and execution time data for each respective scenario.

Fig. 15: VRO (21 cities, 4 vehicles, 7 payload)

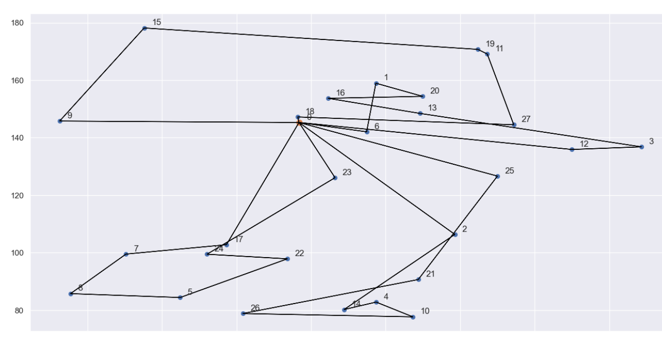

Fig. 16: VRO (28 cities, 4 vehicles, 7 payload)

Cities	Vehicles	Payload	Fitness	Time [s]
21	4	7	536	59.01
28	4	7	684	65.46
21	3	7	509	50.67
21	4	10	453	61.44

TABLE II: VRO Results

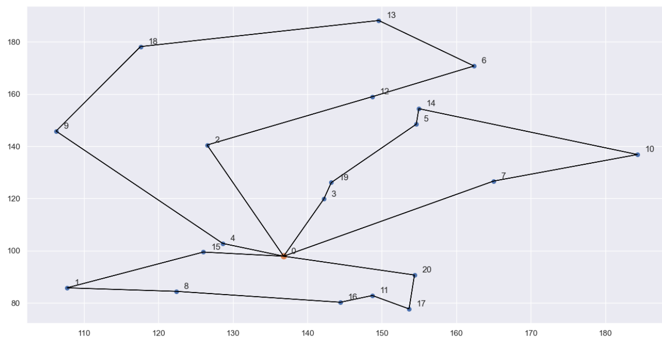

Fig. 17: VRO (21 cities, 3 vehicles, 7 payload)

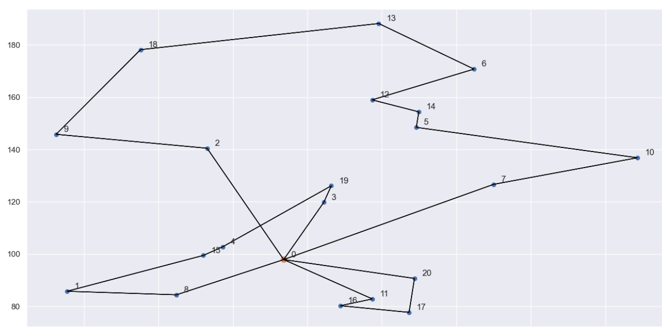

Fig. 18: VRO (21 cities, 4 vehicles, 10 payload)

C. *Benefits and limitations*

*1) **Benefits**:*

Scalability: Genetic Algorithms effectively scale with problem size, as seen in the moderate increase in fitness values despite the rising number of cities in the TSP and VRO results.

Adaptability: Genetic Algorithms can adapt to complex constraints, evident in the VRO results where changes in payload and vehicle numbers still produces efficient solutions.

Time Efficiency: The time required to find solutions increases modestly with problem complexity, suggesting efficient computation.

*2) **Limitations**:*

Time Constraints: More complex scenarios significantly increase computation time, which may be a limitation for larger datasets.

Potential for Suboptimal Solutions: Genetic Algorithms are a heuristic method and may not always find the optimal solution, as indicated by varying fitness values.

Parameter Sensitivity: Genetic Algorithms' performance heavily depends on the setting of parameters like mutation and crossover rates.

Limited Interpretability: Genetic Algorithms do not provide intuitive insights into why certain solutions are chosen, which can be a drawback in scenarios requiring transparency.

VII. Conclusions and Future Work

A. *Summary of findings*

The research found that as the number of cities in the TSP grows, so does the fitness value and calculation time. The fitness value rises from 334 for a 15-city problem to 756 for a 45-city setup, showing the problem's increasing complexity. The computation time increases, from 23.61 seconds for 15 cities to 44.82 seconds for 45 cities, demonstrating the evolutionary algorithm's effectiveness in scaling with problem size.

The Vehicle Routing Optimization (VRO) problem has been extensively studied under a variety of conditions, including variations in city numbers, vehicle counts, and payload capacities, to understand their impacts on routing efficiency. For example, when the number of cities increased from 21 to 28, the fitness function value increased to 684 and the time required escalated to 65.46, indicating more complex routing challenges. Similarly, changes in the number of vehicles and payload capacities have significant impacts. Reducing the number of vehicles while keeping the city count and payload constant resulted in a less optimal routing, as evidenced by an increase in the fitness value; specifically, decreasing the number of vehicles to 3 led to a fitness decrease to 503 and a time reduction to 50.67. On the other hand, increasing the payload capacity improves efficiency; for instance, when the payload was increased to 10, the fitness value decreased to 453, but the time increased to 61.44, suggesting a more efficient routing solution under higher payload conditions.

B. *Potential improvements and enhancements for the proposed methodology*

Fine-Tuning Algorithm Parameters: Adjusting mutation rates, crossover rates, and population sizes for optimal performance.

Hybrid Approaches: Combining Genetic Algorithms with other optimization techniques, such as simulated annealing, for better results.

Enhanced Selection Methods: Implementing more complex selection methods to maintain genetic diversity and avoid premature convergence.

Parallel Processing: Utilizing parallel computing to speed up the genetic algorithm computations, especially for larger datasets.

Advanced Crossover and Mutation Techniques: Exploring novel crossover and mutation strategies specifically customized for the traveling salesman and vehicle optimization problems.

Handling Constraints More Effectively: Developing methods to better manage constraints in vehicle optimization, such as vehicle capacity and delivery time windows.

Multi-Objective Optimization: Extending the algorithms to handle multiple objectives simultaneously, such as minimizing cost while maximizing coverage.

C. *Broader applications in other domains*

*1) **Travelling Salesman Problem**:*

- Logistics and Supply Chain Optimization: Improving route planning for delivery vehicles, reducing transportation costs and time.
- DNA Sequencing: Arranging fragments of DNA in the most efficient order to reconstruct the original sequence.

– Network Design: Planning the layout of cables and wireless networks to ensure the most efficient connections between nodes.
– Tour Planning: Creating optimal itineraries for tourism, covering maximum attractions in the shortest path.
– Astronomical Observations: Planning the sequence of observations for telescopes to minimize the movement and maximize the observation time.
– Robotics: Path planning for robots in manufacturing or warehousing to optimize their movement and task efficiency.

*2) **Vehicle Routing Problem**:*

– Public Transportation Systems: Enhancing bus or train routes and schedules for maximum efficiency and coverage.
– Emergency Response and Disaster Management: Optimizing routes for ambulances, fire trucks, and other emergency services for quickest response times.
– Waste Collection and Management: Streamlining garbage truck routes for efficient city-wide waste management.
– Field Service Management: Planning routes for service personnel like repair technicians or home service providers to minimize travel time and costs.
– Postal and Courier Services: Enhancing delivery routes for postal and courier services to ensure faster and more reliable deliveries.
– Resource Distribution in Humanitarian Aid: Efficiently planning the distribution of resources in disaster relief or humanitarian aid scenarios.

References

[1] G. Bezalel, S. C. Graves. *"The travelling salesman problem and related problems."*, 1978.
[2] B. Kris, K. Ramaekers, I. Nieuwenhuyse. *"The vehicle routing problem: State of the art classification and review."*, 2016.
[3] R. N. Mohd, J. Geraghty. *"Genetic algorithm performance with different selection strategies in solving TSP."*, 2011.
[4] D. Yong, Y. Liu, D. Zhou. *"An improved genetic algorithm with initial population strategy for symmetric TSP."*, 2015.
[5] O. Kramer, *"Genetic Algorithms"*, 2017.
[6] J. Michael, G. Reinelt, and G. Rinaldi. "The Traveling Salesman Problem.", 1995.
[7] T. Paolo, D. Vigo, *The Vehicle Routing Problem*, 2002.
[8] L. Pedro, "Genetic Algorithms for the Traveling Salesman Problem: A Review of Representations and Operators.", 2010.
[9] M. M. Abed, "Solving Vehicle Routing Problem by Using Improved Genetic Algorithm for Optimal Solution.", 2017.
[10] Alkafaween, E. A., Elmougy, S., Essa, E., Mnasri, S., Tarawneh, A. S., Hassanat, A. *IAM-TSP: Iterative Approximate Methods for Solving the Travelling Salesman Problem.*, 2023
[11] Mavrovouniotis, M., Müller, F. M., Yang, S. *Ant colony optimization with local search for dynamic traveling salesman problems*, 2016
[12] Phu Ang, A. *Solving The Travelling Salesman Problem by Using Hybrid Simulated Annealing with Tabu Technique*, 2022
[13] François, A., Cappart, Q., Rousseau, L. M. *How to evaluate machine learning approaches for combinatorial optimization: Application to the travelling salesman problem*, 2019
[14] Wang, L., Song, C., Sun, Y., Lu, C., Chen, Q. *A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment*, 2023
[15] Muniasamy, R. P., Singh, S., Nasre, R., Narayanaswamy, N. S. *Effective parallelization of the vehicle routing problem.*, 2023
[16] Stodola, P. *Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem*, 2020