Genetic Algorithms to Solve the Traveling Salesman Problem and the Vehicle Routing Problem

Bianca-Alexandra Ghiorghiu
SIVA

Content

Problem Statement

Objective: Find the best route that visits a set of locations once an returns to the origin point

Best route may mean:

- Minimum cost
- Minimum distance
- Minimum travel time

These optimization problem are essential in various sectors, like **logistics**, **transportation**, and **distribution**, where efficient routing can lead to significant time and cost savings

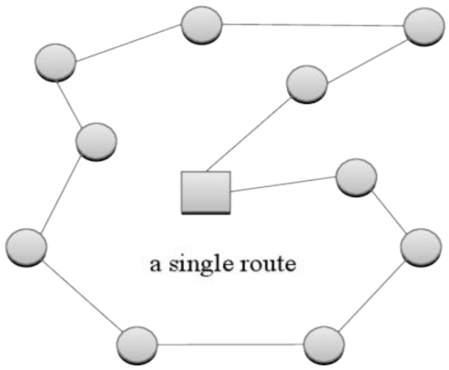
Approach: Genetic algorithms

State of the Art

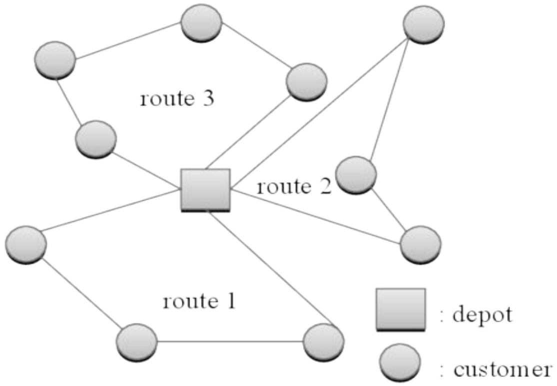
Ant Colony Optimization with Local Search for Dynamic Travelling Salesman Problems [1] The paper introduces an Ant Colony Optimization (ACO) algorithm with a local search operator for dynamic TSP, enhancing its performance and efficiency in solving both symmetric and asymmetric dynamic TSPs.	**Solving The Travelling Salesman Problem by Using Hybrid Simulated Annealing with Tabu Technique [2]** The paper presents a hybrid TSP approach using simulated annealing (SA) and Tabu search, enhancing performance through swap techniques, demonstrating superiority in reducing average percentage deviation from lower bound criteria.	**How to Evaluate Machine Learning Approaches for Combinatorial Optimization: Application to the Travelling Salesman Problem [3]** The paper explores the limitations of Machine Learning (ML) methods in solving TSP and introduces a new metric, the ratio of optimal decisions (ROD), to evaluate the accuracy of ML methods in solving TSP.
A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment [4] The paper introduces a multi-objective vehicle routing optimization algorithm, enhanced from PMOCO, which uses a weight adjustment method and deep reinforcement learning, resulting in a 6% improvement in quality compared to PMOCO with 20 preferences.	**Effective Parallelization of the Vehicle Routing Problem [5]** The study introduces a new technique that combines local search and randomization to solve the Capacitated Vehicle Routing Problem faster and with better accuracy, compared to current GPU implementations.	**Hybrid Ant Colony Optimization Algorithm Applied to the Multi-Depot Vehicle Routing Problem [6]** The article 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. The algorithm outperforms other methods with the smallest average error on benchmark instances.

Traveling Salesman Problem vs. Vehicle Routing Problem

Travelling Salesman Problem: One vehicle can deliver all orders

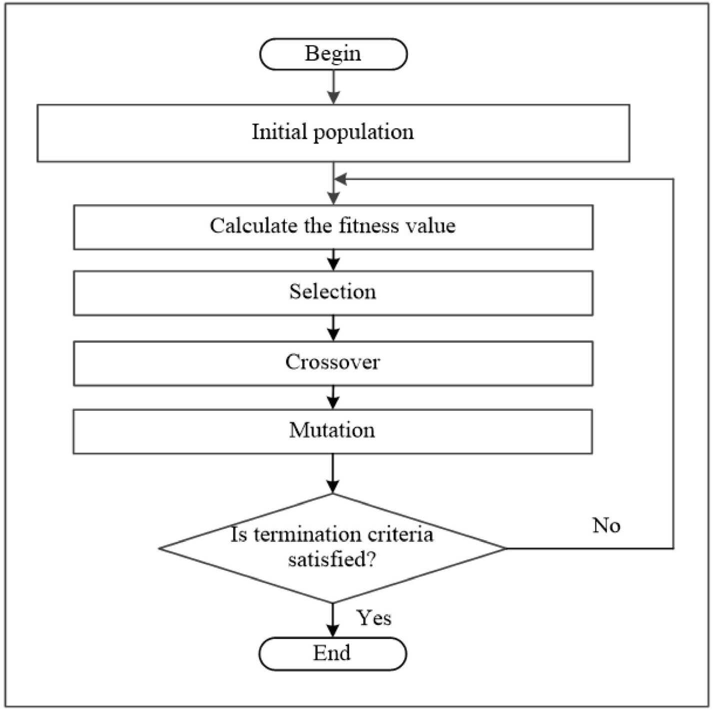


Vehicle Routing Problem: Extends TSP by considering multiple vehicles, routing and constraints such as: vehicle capacity, delivery time, etc.



Genetic Algorithms - Concept

Genetic Algorithms (GAs) are search heuristics that mimic the process of natural selection to solve optimization and search problems.



Methodology - TSP

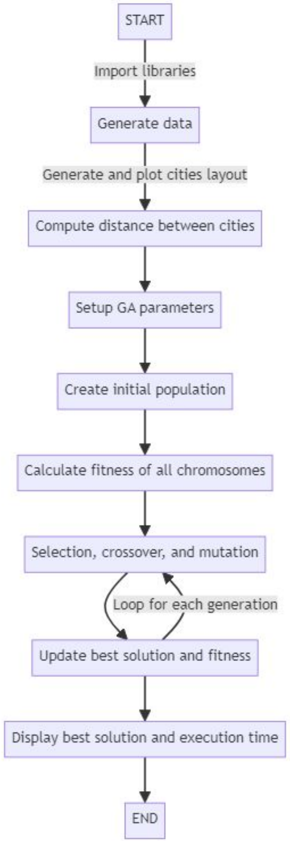
1. **Initialization:** Import necessary libraries

2. **Data generation:**
   a. Generate and plot random city coordinates within a defined range
   b. Calculate the Euclidean distances between all pairs of cities, stored in a distance matrix

3. **Genetic algorithm setup:**
   a. Define a fitness function to evaluate routes (total distance)
   b. Create a population of random routes (chromosomes)
   c. Set GA parameters like population size, mutation, and crossover probabilities

4. **Evolution process:**
   a. Perform selection, crossover, and mutation operations on the population
   b. Continuously update the population with new generations of route
   c. Track and store the best route and its fitness score across generations

5. **Results:**
   a. Display the best solution (shortest route) and the total execution time
   b. Plot the best route

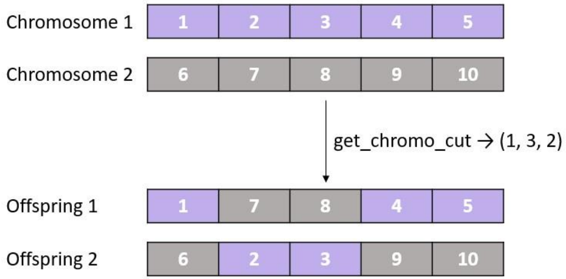


Methodology - VRO

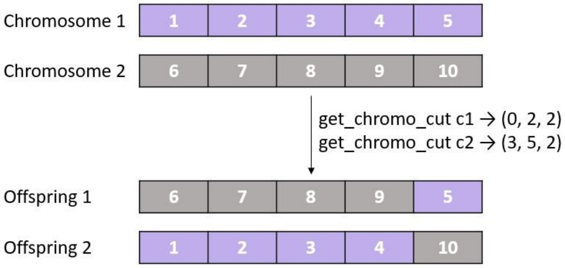
The VRO workflow closely mirrors the TSP: initialization → data generation → genetic algorithm setup → evolution process, but introduces a new step: **custom function definition**

Custom crossover

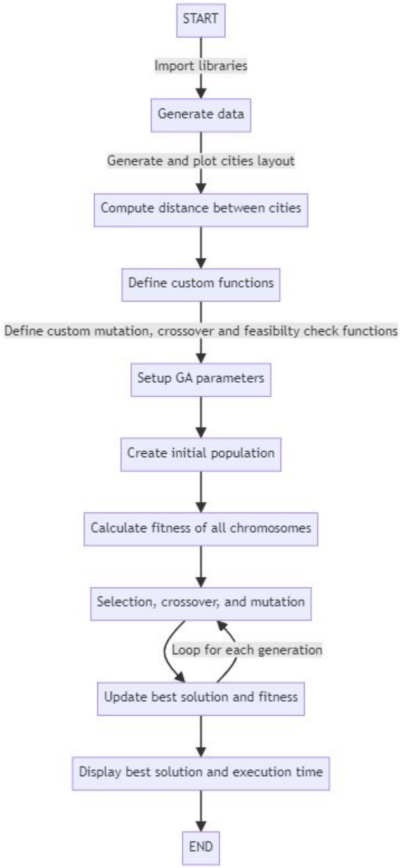
- **get_chromo_cut:** Generates random cut positions within a predefined range
- **partial_crossover:** Performs partial crossover between two chromosomes, exchanging gene segments while maintaining gene order
- **swap_genes:** Swaps a segment of genes between two chromosomes, defined by specified cut positions
- **crossover:** Performs a complete crossover operation between two chromosomes, utilizing partial crossover and gene swapping



partial_crossover function



crossover function

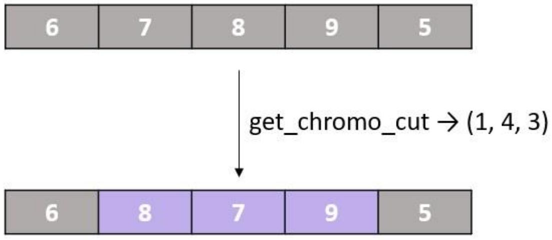


Methodology - VRO

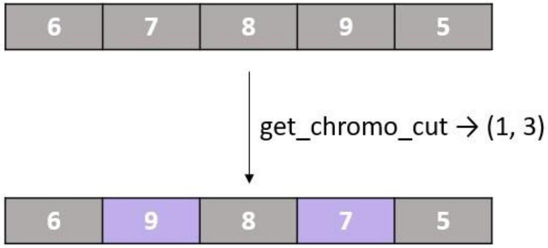
The VRO workflow closely mirrors the TSP: initialization ➔ data generation ➔ genetic algorithm setup ➔ evolution process, but introduces a new step: **custom function definition**

Custom mutation

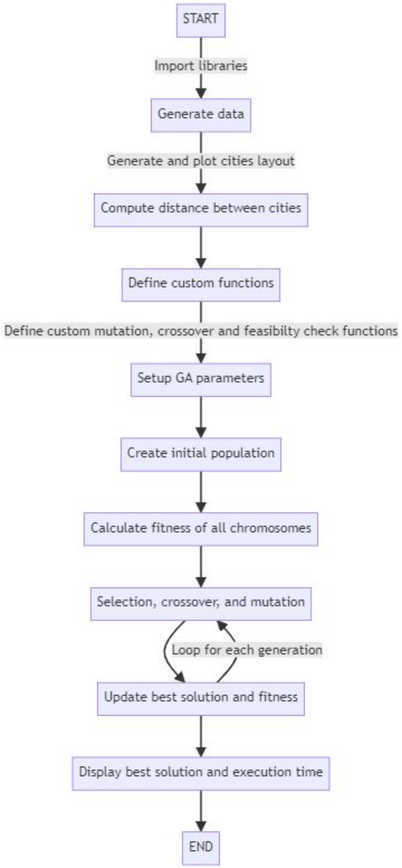
- **swap_gene:** Randomly selects two genes within a chromosome (schedule or vehicle assignment) and swaps their positions,
- **shuffle_gene:** Randomly selects a range of genes within a chromosome and shuffles their positions
- **mutation:** Randomly chooses between swap_gene and shuffle_gene mutations to apply to the given chromosome



shuffle_gene function



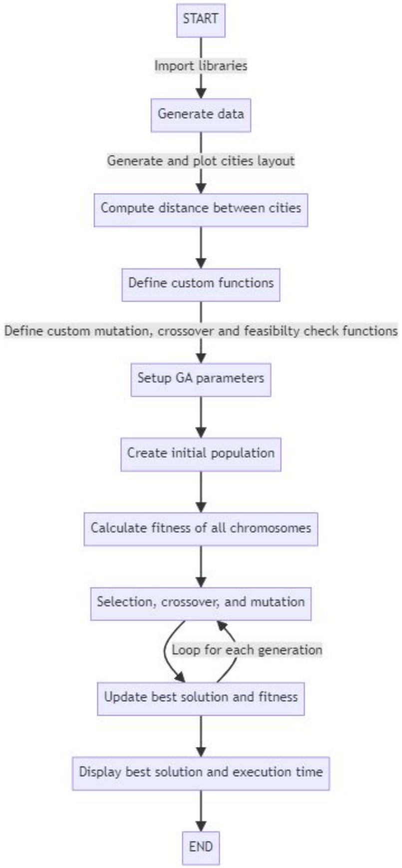
swap_gene function



Methodology - VRO

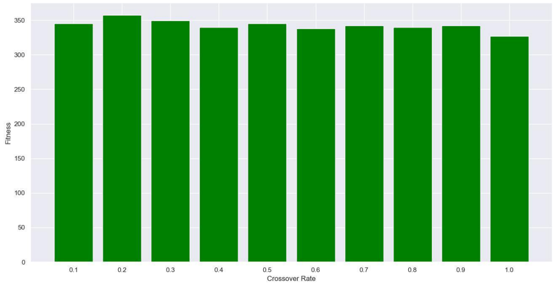
The VRO workflow closely mirrors the TSP: initialization → data generation → genetic algorithm setup → evolution process, but introduces a new step: **custom function definition**

Feasibility: Iterates through the vehicles and their payload assignments, attempting to redistribute cargo to eliminate excess payload. The function terminates when there is no excess payload for any vehicle, ensuring a feasible solution for the given problem.

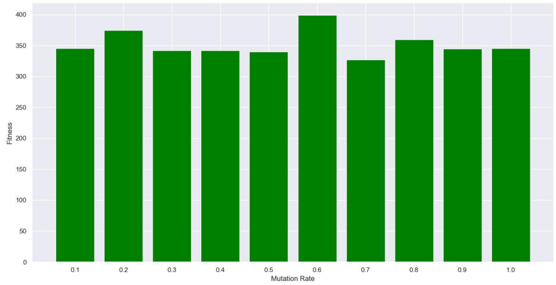


Experiments - TSO

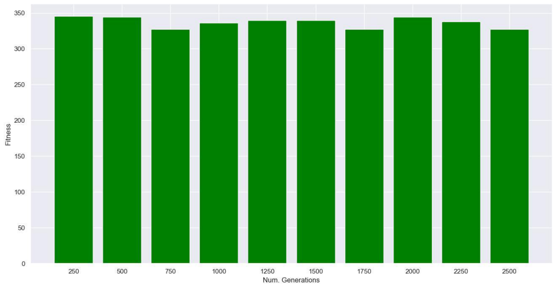
Crossover Rate



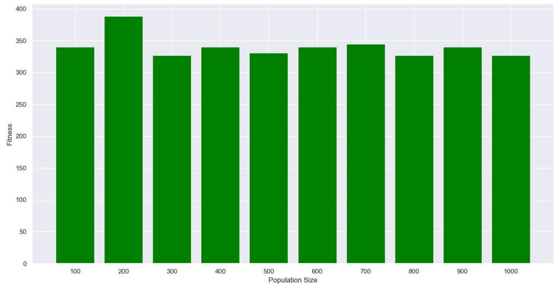
Mutation Rate



No. Generations



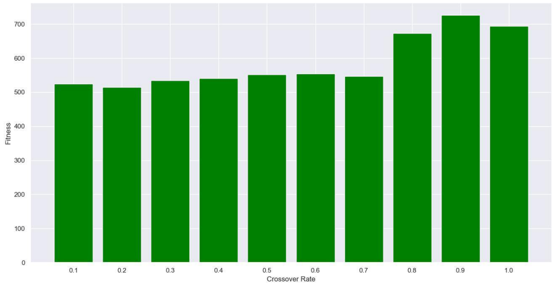
Population Size



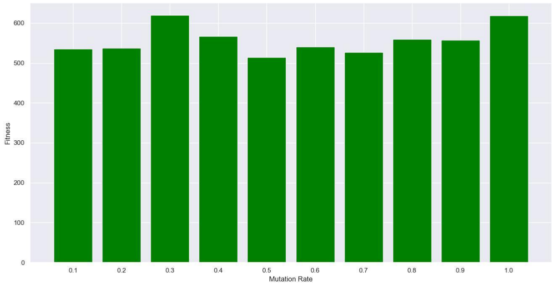
Crossover Rate	40%
Mutation Rate	70%
No. Generations	750
Population Size	300

Experiments - VRO

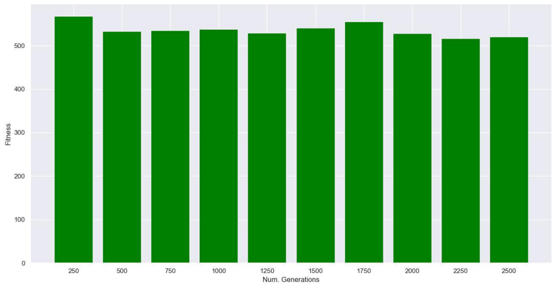
Crossover Rate



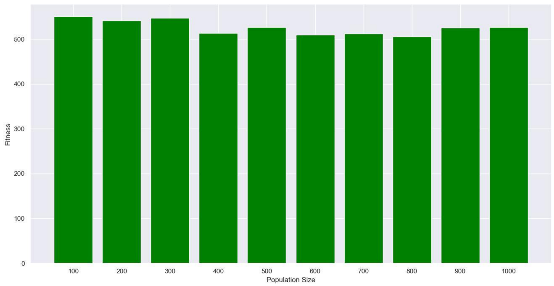
Mutation Rate



No. Generations



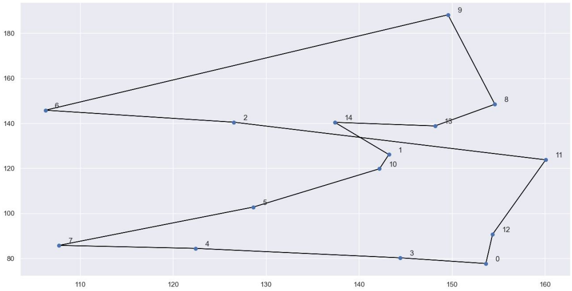
Population Size



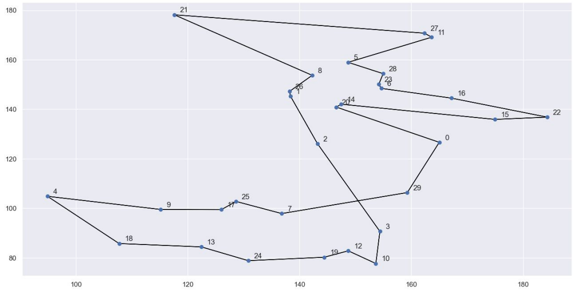
Crossover Rate	20%
Mutation Rate	50%
No. Generations	1250
Population Size	400

Results - TSP

15 cities

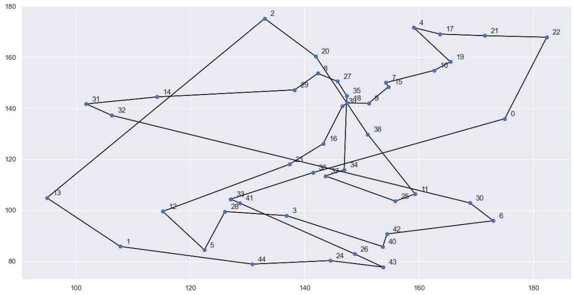


30 cities



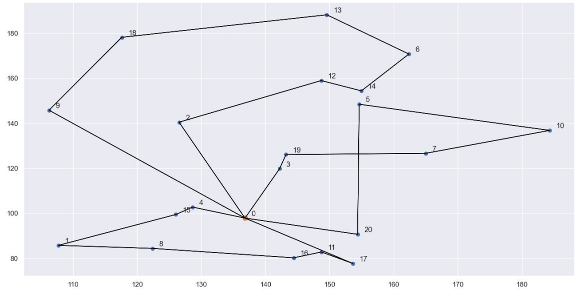
Cities	Fitness	Time [s]
15	334	23.61
30	452	34.37
45	756	44.82

45 cities

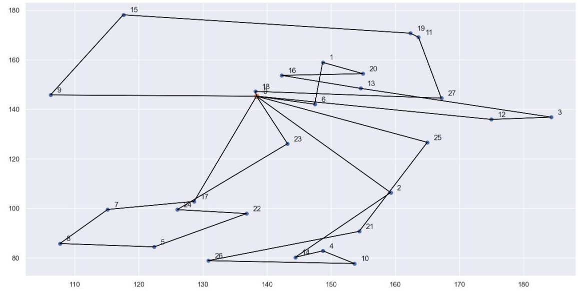


Results - VRO

21 cities, 4 vehicles, 7 payload

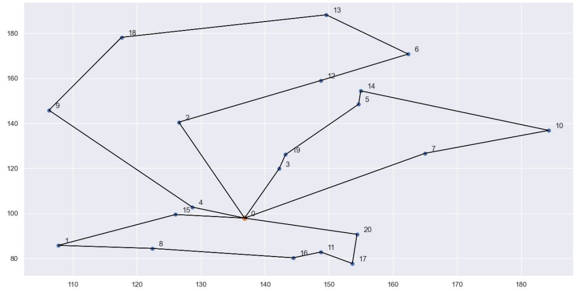


28 cities, 4 vehicles, 7 payload

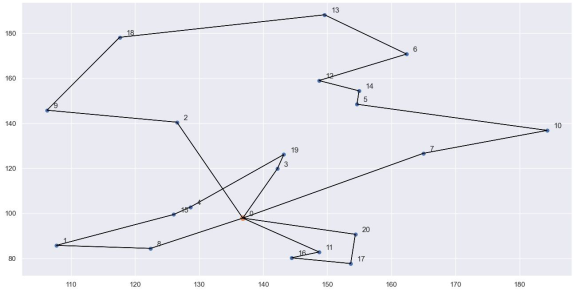


Cities	Vehicles	Payload	Fitness	Time [s]
21	4	7	536	59.01
28	4	7	684	65.46
21	3	7	509	50.67
21	4	10	453	61.44

21 cities, 3 vehicles, 7 payload



21 cities, 4 vehicles, 10 payload



Future Work

- Optimize algorithm parameters: mutation rates, crossover rates, population sizes, etc.
- Combine genetic algorithms with other methods like simulated annealing.
- Improve selection methods to maintain diversity and prevent early convergence
- Speed up computations using parallel processing, especially for large datasets
- Develop complex crossover and mutation techniques for specific optimization problems
- Improve constraint handling in vehicle optimization
- Multi-Objective Optimization

Conclusion

- Genetic algorithms provide an effective method for solving complex optimization problems like TSP and VRP, demonstrating adaptability and efficiency.

- **Traveling Salesman Problem:** As cities increase, fitness and computation time both rise significantly ➔ algorithm's effectiveness in adapting to larger problem sizes.

- **Vehicle Routing Problem:** Adding more cities makes both the fitness and time go up, while cutting down on vehicles makes both fitness and time go down. Increasing the payload, it decreases the fitness, but increases time go up.

- Future work could explore: integrating advanced machine learning techniques, the improvement of algorithm parameters or selection/crossover/mutation mechanism, parallel processing or multiple-object optimization.

Bibliography

1. Mavrovouniotis, M., Müller, F. M., & Yang, S. (2016). Ant colony optimization with local search for dynamic traveling salesman problems. *IEEE transactions on cybernetics*, *47*(7), 1743-1756.
2. Phu-Ang, A. (2022, September). Solving The Travelling Salesman Problem by Using Hybrid Simulated Annealing with Tabu Technique. In *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)* (pp. 61-64). IEEE.
3. François, A., Cappart, Q., & Rousseau, L. M. (2019). How to evaluate machine learning approaches for combinatorial optimization: Application to the travelling salesman problem. *arXiv preprint arXiv:1909.13121*.
4. Wang, L., Song, C., Sun, Y., Lu, C., & Chen, Q. (2023). A Neural Multi-Objective Capacitated Vehicle Routing Optimization Algorithm Based on Preference Adjustment. *Electronics*, *12*(19), 4167.
5. Muniasamy, R. P., Singh, S., Nasre, R., & Narayanaswamy, N. S. (2023, July). Effective parallelization of the vehicle routing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 1036-1044).
6. Stodola, P. (2020). Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem. *Natural Computing*, *19*(2), 463-475.