NOVA IMS Information Management School

Capacitated vehicle routing problem optimization using genetic algorithms

https://github.com/Joana-Goncalves/CVRP

Mikołaj Jarząbkowski | 20230529

Rodrigo Silva | 20230536

Nicolau Dulea | 20230544

Joana Gonçalves | 20230977

Tiago Fernandes | 20230988

Computational Intelligence For Optimization

Master's Degree in Data Science and Advanced Analytics

2023/2024

Table of contents

Statement of contribution:

Discussion of the problem: Dulea, Fernandes, Gonçalves, Jarząbkowski, Silva.

Designing the study: Gonçalves, Jarząbkowski.

Data collection and preparation: Gonçalves.

Writing the code: Dulea, Fernandes, Gonçalves, Jarząbkowski, Silva.

Writing the report: Gonçalves, Jarząbkowski.

Discussion of the Results: Dulea, Fernandes, Gonçalves, Jarząbkowski, Silva.

1. Introduction

The Vehicle Routing Problem (VRP) is an optimization problem, which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?", generalising the Travelling Salesperson Problem (TSP). The earliest known reference to the VRP is attributed to Dantzig and Ramser in 1959. Their work focused on finding optimal routes for delivering goods from a central depot to a set of customers using a fleet of vehicles, while minimising the total transportation cost [1].

Since its introduction, the VRP has become a fundamental problem in the field of operations research and combinatorial optimization, being relevant to various real-world routing and logistics challenges. The VRP has numerous variants, each with its own set of constraints and objectives, including the Capacitated Vehicle Routing Problem (CVRP). In this variant, the vehicles have a limited carrying capacity of the goods that must be delivered.

This project aims to solve the CVRP instance A-n32-k5. This is an instance of the CVRP with 32 customers (fig 1) to be served by 5 vehicles, each with capacity of 100, taken from the Augerat [2] dataset. The goal is to reach global optimum (784) using a Genetic Algorithm (GA).

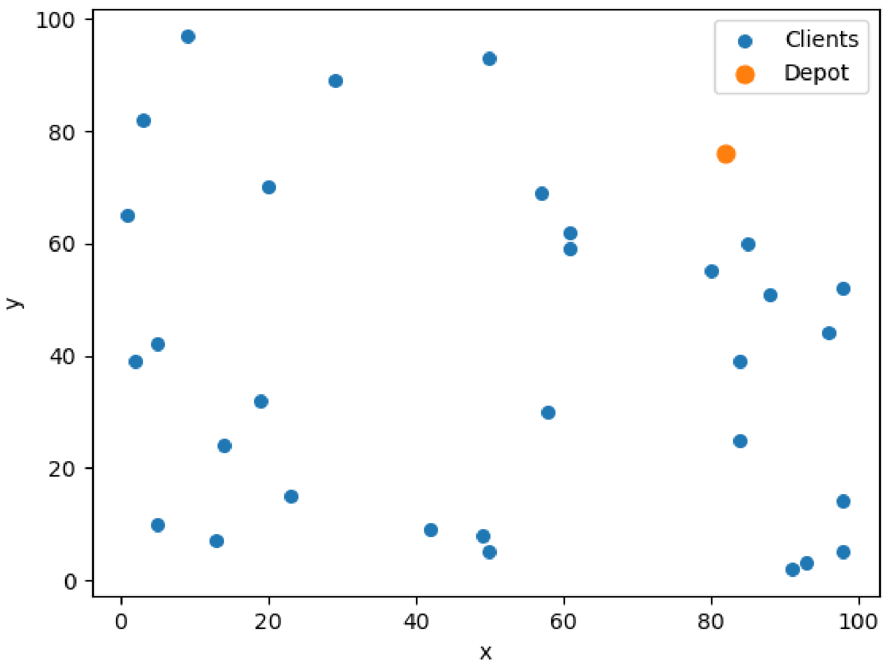

Fig 1. Coordinates of clients and depot of the A-n32-k5 instance.

2. Methods

Individuals

In this implementation, individuals were formalised as containing 2 chromosomes, the first with genes illustrating customers, and the second representing which vehicle delivered the goods to each client, such as: individual = [[clients], [vehicles]]. Both lists have the same length and clients are ordered according to the route made by each vehicle. It is possible to observe a schematic representation in fig 1. Individuals were always randomly initialised.

26	21	29	22	3	31	30	16	2	9	27	23	6	19	15	17	18	10	8	13	25	1	11	12	5	4	7	28	14	20	24
2	0	4	4	2	0	1	1	2	4	3	2	2	0	4	0	3	4	3	0	4	1	3	1	4	3	0	3	2	4	3

Fig 2. Schematic representation of an individual: clients (top row) and vehicles (bottom row).

Selection

In our project, three selection algorithms were tested: tournament, roulette wheel and rank selection. Tournament selection is suitable for CVRP due to its ability to balance the selection of high-quality solutions with maintaining diversity, ensuring effective exploration of the solution space. For this project, a fixed tournament size of 9 individuals was used. Roulette wheel selection provides a probabilistic selection that maintains diversity by giving each individual a chance proportional to its fitness. Since this algorithm is not suited for minimization problems in its standard version, fitness values needed to be inverted in order to implement it. Rank selection is suitable as it reduces the influence of outliers and ensures uniform selection pressure, promoting consistent improvement across generations.

Crossover

Various crossover operators were employed for both clients and vehicles. For clients, it is necessary to guarantee that every client is visited. Therefore, partially mapped crossover (PMX) and cycle crossover were used. PMX is suitable for CVRP as it preserves relative order and position of elements, ensuring valid offspring by mapping segments from parent solutions. Cycle crossover is effective as it maintains cycles of genes from the parents, preserving strong genetic traits and ensuring diverse and valid solutions.

For vehicles, single-point and two-point crossover were utilized. Single-point crossover is suitable because it creates offspring by splitting the parent chromosomes at a single point, combining the initial segment from one parent with the final segment from the other, thus preserving large continuous sections of routes from both parents. Two-point crossover, on the other hand, is suitable as it combines segments from two parents, promoting diversity while preserving large sections of the routes.

Mutation

Three mutation operators were utilized: swap, shuffle and random resetting. Swap and shuffle mutations were used for both clients and vehicles. Swap mutation is effective for CVRP due to its simplicity, making small changes by swapping two elements, which maintains solution feasibility. Shuffle mutation introduces larger changes by shuffling a subset of the solution, enhancing exploration and avoiding local optima. Random resetting mutation replaces randomly chosen elements with new values, increasing diversity.

Elitism

For all iterations of the GA, elitism was incorporated (2 individuals), to preserve high-quality solutions while still allowing for diverse exploration of the solution space.

Fitness and evaluation

Fitness was calculated as the minimization of the total distance travelled by the vehicles while ensuring feasible solutions. Each solution's total distance is calculated, and vehicle capacity constraints are verified. Infeasible solutions, where demands exceed vehicle payloads, are assigned infinite distance values. This approach ensures the algorithm prioritizes efficient and feasible routes.

Genetic algorithm

For this problem, a population of 200 individuals (including elites) was used, since larger values did not improve results. This population evolved for 3000 generations, since some combinations of parameters needed more time to converge. Crossover and mutation operators were applied with a probability of 0.9 and 0.15, respectively. These values provided the best initial results and are aligned with standard applications of GAs.

The previously described selection algorithms, crossover and mutation operators were tested in several combinations, as detailed in fig 3, resulting in 90 different configurations. To get more robust results, each combination was tested 30 times. At each 100 generations, the best fitness was stored. After the 30 runs, those fitness values were averaged. The average fitness value obtained at the end of 3000 generations was considered the representing fitness obtained with each combination for the algorithm.

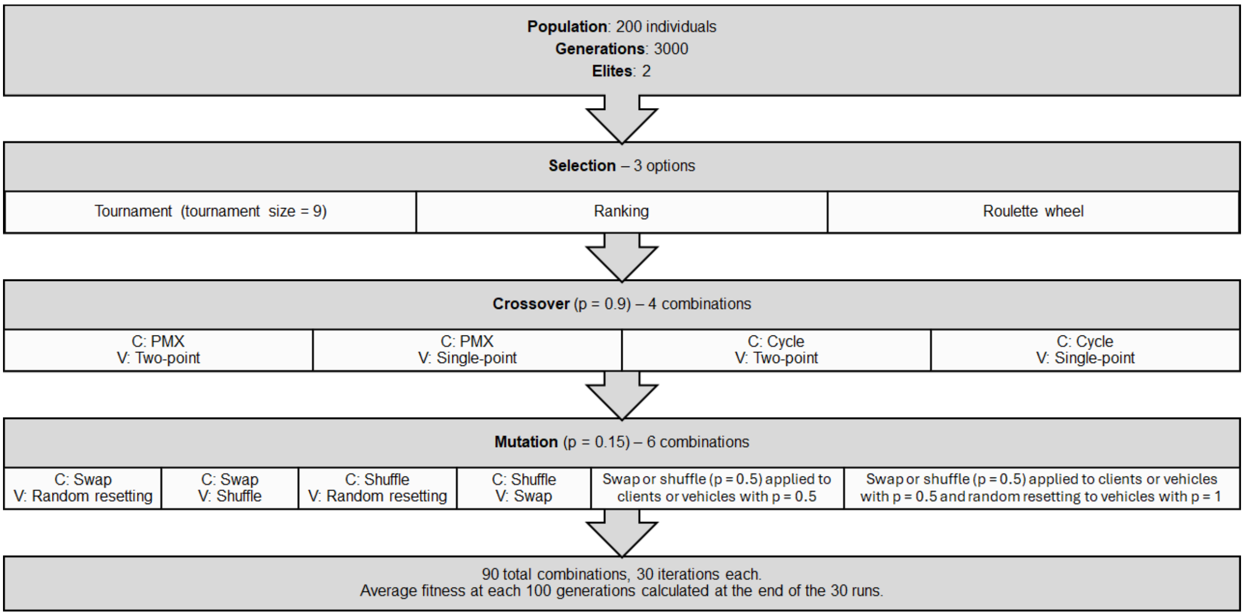

Fig 3. Schema of the strategy adopted to solve the problem. For crossover, C represents clients and V vehicles.

3. Results and discussion

The best configuration obtained for this model used roulette wheel as selection algorithm, cycle crossover and swap mutation for the clients list, and single-point crossover and random resetting mutation for the vehicles list. The average fitness value obtained with these parameters was 787.79, which is a very close value to the global optimum, 784. Both solution routes can be seen in fig 4.

The best 20 combinations (in terms of average fitness) are detailed in Annex 1. We can observe that for this dataset, roulette wheel selection consistently outperformed tournament and rank selection, likely due to its ability to favour higher fitness individuals while maintaining diversity. Tournament selection was robust but less consistent, and rank selection was less effective overall. Cycle crossover for clients combined with single-point crossover for vehicles produced the best results, preserving essential gene sequences and creating feasible solutions. Swap mutation for clients and random resetting for vehicles enhanced local exploration and diversity, preventing premature convergence.

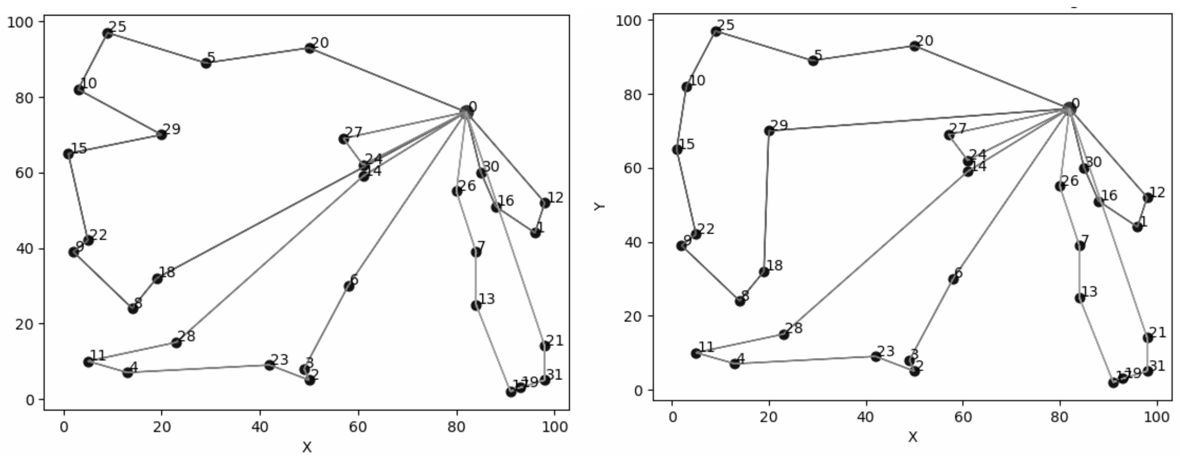

Fig 4. Routes for the best solution obtained (left) and global optimum (right). It is possible to observe that only 1 out of the 5 routes is different between both solutions.

To statistically assess the results, the top 5 combinations (first 5 models described in Annex 1) were considered. After initial observation of the progression of the average fitness during the generations, Models 1 and 2 are clearly separated from the others (fig 5). Configurations with roulette wheel selection, cycle crossover, and single-point crossover consistently showed superior results.

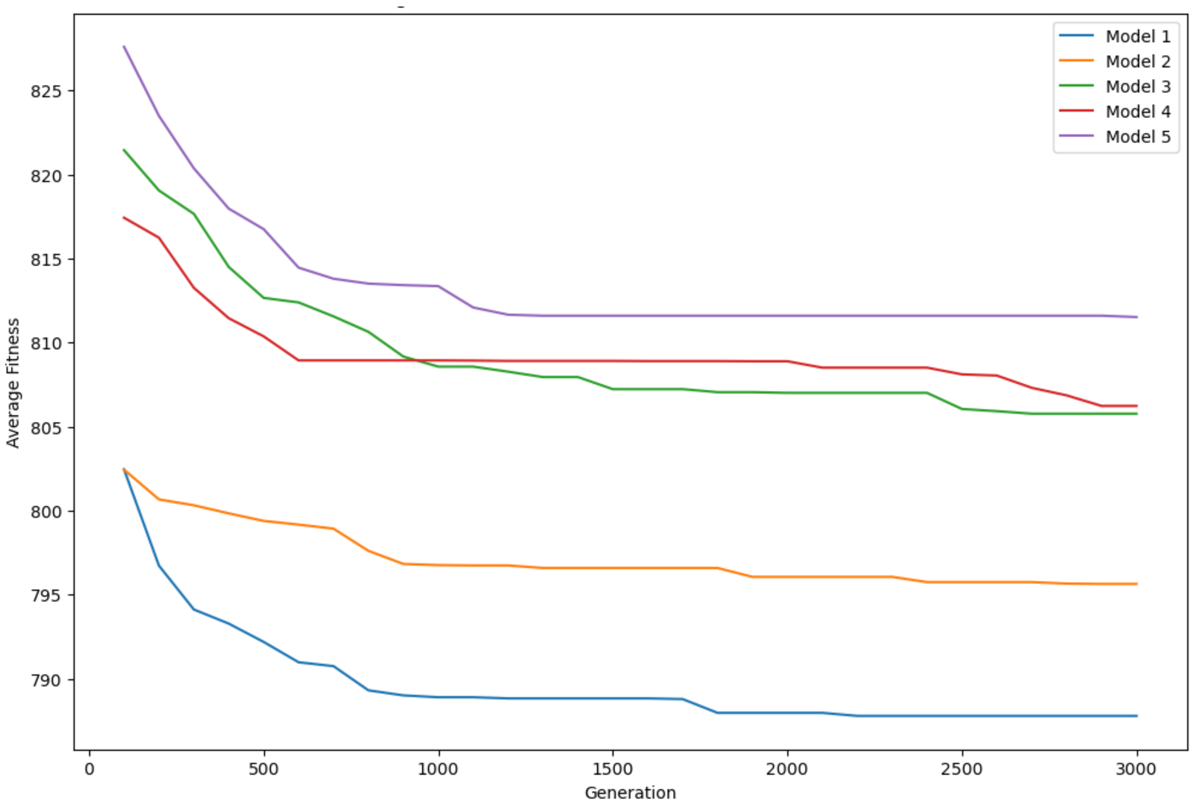

Fig 5. Average fitness over generations for the best 5 combinations of parameters.

These assessments are supported by the distribution of the fitness values for the different models (fig 6). These provide a deeper understanding of the models' robustness and behaviour.

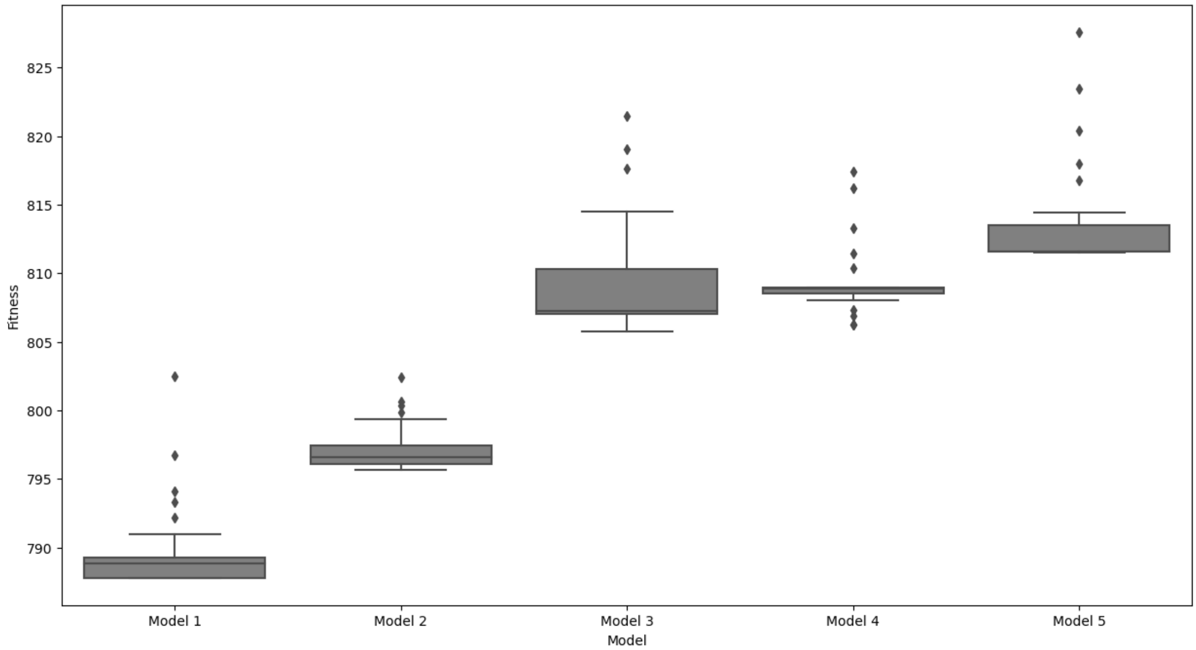

Fig 6. Fitness values boxplots for the top 5 combinations of parameters.

For these models, statistical analysis was performed. Firstly, the Kruskal-Wallis test was used to check for significant differences among the multiple groups. This test was significant (p-value = 2.9265 e-26) at the 1% significance level. Therefore, pairwise comparisons were made using the Wilcoxon rank-sum test, with Bonferroni correction to adjust for multiple comparisons. Results for this test are described in table 1. It is possible to observe that there are statistically significant differences between the distributions of all models' fitness values at the 1% significance level (except between Model 3 and Model 4).

Table 1. Pairwise comparison p-values for Wilcoxon rank-sum test.

	Model 2	**Model 3**	**Model 4**	**Model 5**
Model 1	2.6166 e-08	2.6825 e-10	2.6672 e-10	1.6200 e-10
Model 2	-	2.8287 e-10	2.8127 e-10	1.7124 e-10
Model 3	-	-	0.4490	6.0040 e-05
Model 4	-	-	-	1.4776 e-07

These results suggest that the models have different performances, with Model 1 having a better mean fitness value, indicating superior performance in this problem. This is an expected result, considering how similar the solution obtained was to the global optimum.

4. Conclusion

The results demonstrate the effectiveness of genetic algorithms (GA) in solving the Capacitated Vehicle Routing Problem (CVRP) for the A-n32-k5 instance. By evaluating various configurations of selection, crossover, and mutation operators, we identified combinations that closely approximated the global optimum. This study highlights the efficacy of GAs in solving CVRP, offering insights into optimal genetic operator configurations for improved routing and logistics efficiency.

For CVRP instances like A-n32-k5, using roulette wheel selection with cycle and single-point crossovers provides efficient near-optimal solutions. Future research could explore adaptive mechanisms for operator probabilities and hybrid approaches with other optimization techniques, such as heuristic methods to generate the initial population, for example. Applying this methodology to larger CVRP instances would further validate its scalability.

5. References

1. Dantzig, George Bernard; Ramser, John Hubert (October 1959). "The Truck Dispatching Problem". Management Science. 6 (1): 80–91. doi:10.1287/mnsc.6.1.80.
2. Augerat, P. (1995). Approche polyédrale du problème de tournées de véhicules. PhD Thesis, Université de Montréal, Montreal, Canada.
3. Vanneschi, Leonardo & Silva, Sara. (2023). Lectures on Intelligent Systems. 10.1007/978-3-031-17922-8.

Annex 1

Table 2. Best 20 configurations obtained, based on the average fitness at the end of the 3000 generations. Model number follows rank order.

Model	Selection	Crossover clients	Crossover vehicles	Mutation client	Mutation vehicles	Average fitness
1	Roulette	Cycle	Single point	Swap	Random resetting	787.79
2	Tournament	Cycle	Single point	Random swap or shuffle + random resetting		795.64
3	Roulette	PMX	Two-point	Shuffle	Swap	805.77
4	Roulette	Cycle	Two-point	Swap	Random resetting	806.23
5	Roulette	Cycle	Single point	Swap	Shuffle	811.52
6	Tournament	Cycle	Two-point	Swap	Random resetting	815.16
7	Roulette	Cycle	Two-point	Random swap or shuffle		834.35
8	Roulette	PMX	Single point	Swap	Shuffle	835.1
9	Ranking	Cycle	Single point	Swap	Random resetting	835.7
10	Tournament	Cycle	Single point	Swap	Random resetting	836.76
11	Roulette	Cycle	Single point	Random swap or shuffle + random resetting		837.58
12	Roulette	PMX	Two-point	Random swap or shuffle + random resetting		838.54
13	Roulette	PMX	Single point	Swap	Random resetting	840.28
14	Roulette	PMX	Single point	Random swap or shuffle + random resetting		840.55
15	Tournament	PMX	Two-point	Swap	Random resetting	841.08
16	Roulette	PMX	Two-point	Random swap or shuffle		841.67
17	Roulette	Cycle	Two-point	Swap	Shuffle	849.65
18	Roulette	PMX	Two-point	Swap	Random resetting	851.57
19	Roulette	Cycle	Two-point	Random swap or shuffle + random resetting		851.83
20	Tournament	Cycle	Two-point	Random swap or shuffle + random resetting		853.66