An Enhanced Genetic Algorithm with Problem-Aware Crossover and Local Search Mutation for the Capacitated Vehicle Routing Problem

Mohammad Jamshidi[1] , Mehdi Alirezanejad[2*], Homayun Motameni[3], Rasul Enayatifar[4]

Abstract – The Capacitated Vehicle Routing Problem (CVRP) is a complex challenge in logistics and supply chain management. While Genetic Algorithms (GAs) are widely used, their performance heavily depends on the design of genetic operators. This paper introduces an Improved Genetic Algorithm (IGA) enhanced with two problem-specific operators: a Route-Based Merge Crossover (RBMX) and a Two-Stage Local Search Mutation. The RBMX preserves entire feasible routes from parent solutions, maintaining promising sub-tour structures, while the mutation operator systematically improves solutions through intra- and inter-route refinements. Experimental results on standard benchmark problems demonstrate that the IGA significantly outperforms a standard GA, achieving an average improvement of 5.50% in solution quality with only a 21.1% increase in computational time. The algorithm also shows faster and more stable convergence. These outcomes confirm that incorporating domain-aware operators into a GA framework leads to substantially better performance, offering a more effective and efficient approach for solving the CVRP.

Keywords: Capacitated Vehicle Routing Problem (CVRP), Genetic Algorithm, Metaheuristics, Crossover Operator

1. Introduction

The Capacitated Vehicle Routing Problem (CVRP) is a key issue in logistics and supply chain management[1]. It involves finding the best routes for a fleet of vehicles to deliver goods to customers in different locations[2]. Each vehicle can only carry a certain amount, and each customer has a set demand. The main goal is to keep transportation costs low, usually by minimizing the total distance traveled, while making sure all customer needs are met and no vehicle is overloaded[3]. The CVRP is important in many real-world areas, such as retail delivery, waste collection, school bus planning, and emergency relief. Solving it efficiently helps cut costs and reduce environmental impact. However, because the CVRP is an NP-hard problem, exact methods like branch-and-cut become too slow as the number of customers grows, so traditional approaches do not work well for large cases[4].

To tackle this complexity, metaheuristics, particularly Evolutionary Algorithms (EAs), have emerged as a powerful and popular approach[5-7]. EAs solve the CVRP by maintaining a population of candidate solutions[7, 8]. To address this challenge, researchers often use metaheuristics, especially Evolutionary Algorithms (EAs). EAs work by keeping a group of possible solutions and improving them step by step, using ideas from natural selection. Genetic Algorithms (GAs), a well-known type of EA[9], represent solutions as chromosomes and use selection, crossover, and mutation to create better solutions over time. GAs are good at searching a wide range of possible answers and avoiding getting stuck in poor solutions too early. However, standard genetic operators do not always consider the specific rules and structure of the CVRP[10, 11]. As a result, common crossover and mutation methods can create solutions that break capacity limits or ruin good route sequences, which slows progress and lowers the quality of the final results. Genetic operators that guide the search more effectively. First, we introduce a Route-Based Merge Crossover (RBMX), which preserves entire, feasible routes from parent solutions as intact building blocks, thereby explicitly maintaining high-quality sub-tours. Second, we develop a Two-Stage Local Search Mutation that replaces random changes with focused, small-scale local improvements, such as 2-opt within a route and customer exchanges between routes. The proposed algorithm achieves superior performance by employing problem-aware operators that directly address the limitations of standard GAs. Unlike traditional crossover that disrupts solution structure, our Route-Based Merge Crossover preserves high-quality route

1 Department of Computer Engineering, Sar.C., Islamic Azad University, Sari, Iran.

2* **Corresponding Author**: Department of Computer Engineering, Fi.C, Islamic Azad University, Firouzkouh, Iran.
Email: me.alirezanejad@gmail.com
Department of Computer Engineering, Sar.C., Islamic Azad University, Sari, Iran.
4 Department of Computer Engineering, Fi.C, Islamic Azad University, Firouzkouh, Iran

Received: 2025.11.22; Accepted: 2026.02.08

segments. Furthermore, our mutation operator acts as a targeted local search, refining solutions rather than introducing random changes. The primary limitations of this work include its evaluation on a specific set of benchmark instances and its focus on the static, single-depot CVRP. Performance on larger real-world datasets or under dynamic constraints remains to be validated.

The rest of this paper is organized as follows. The next section covers the basics, including a formal definition of the CVRP and a short review of related work on genetic algorithms for vehicle routing. Section three explains our proposed method, including how solutions are encoded, the new crossover and mutation operators, and the overall algorithm steps. Section four describes the experiments, shows results on well-known test cases, and compares our method to a standard genetic algorithm. Finally, section five sums up the main findings and suggests ideas for future research.

1. Preliminaries

In this section, the study's preliminaries, including the Genetic Algorithm and the Capacitated Vehicle Routing Problem (CVRP), are explained.

2.1. Genetic Algorithms

Genetic Algorithms (GAs) are problem-solving methods inspired by the principles of natural selection[12]. These algorithms operate on a population of candidate solutions, each typically represented as a string of values, known as a chromosome. The algorithm iteratively enhances the population by selecting the most effective solutions and applying genetic operations. In each iteration, individuals are selected for reproduction based on their fitness scores, which reflect their effectiveness in solving the problem. Selected parents exchange segments of their chromosomes through crossover to generate new candidate solutions. Subsequently, mutation introduces small random changes to maintain diversity within the population. This iterative process of selection, crossover, and mutation continues until a predefined termination criterion is met, progressively improving the quality of solutions across generations[13].

2.2. Capacitated Vehicle Routing Problem (CVRP) Formulation

The Capacitated Vehicle Routing Problem (CVRP) is a classic combinatorial optimization problem. It can be formally defined as follows. Let $G = (V, E)$ be a complete graph where $V = \{0,1,\dots,n\}$ is the vertex set and E is the edge set. Vertex 0 represents the central depot, and the other vertices $V' = \{1,\dots,n\}$ represent customers. Each customer i has a non-negative demand d_i. A homogeneous fleet of m vehicles, each with an identical capacity Q, is located at the depot. The objective is to find a set of exactly m routes such that[14]:

1. Each route starts and ends at the depot.
2. Each customer is visited exactly once by one vehicle.
3. The total demand of the customers on any route does not exceed the vehicle capacity Q.
4. The total distance traveled by all vehicles is minimized.

A common mathematical model uses binary decision variables x_{ijk} that equal 1 if vehicle k traverses the edge (i, j), and 0 otherwise. The objective function is then to minimize $\sum_{i,j,k} c_{ij} x_{ijk}$, where c_{ij} is the cost (distance) from node ii to node j, subject to the constraints outlined above[15].

2. Proposed method

The Improved Genetic Algorithm (IGA) extends the canonical Genetic Algorithm (GA) framework by integrating problem-specific knowledge through tailored solution representations and specialized genetic operators. Figure 1 presents the overall flowchart of the IGA, which details the sequential processes of initialization, selection, and the implementation of novel crossover and mutation strategies.

3.1. Solution Representation

A good chromosome encoding is important for the performance of a genetic algorithm (GA). In this study, we use a giant tour representation that does not include explicit trip delimiters. Here, each chromosome is simply a list of all customer nodes in order. For example, if there are seven customers, the chromosome could be [3, 1, 5, 7, 2, 6, 4].

To convert the chromosome into a valid solution to the Capacitated Vehicle Routing Problem (CVRP), we use a simple splitting procedure. The decoder goes through the chromosome from the start, adding customers to the current route until the vehicle reaches its capacity. If adding another customer would go over the limit Q, the decoder starts a new route by inserting the depot (0). This way, every solution meets the capacity rule. For example, the chromosome [3, 1, 5, 7, 2, 6, 4] could become the routes [0-3-1-5-0], [0-7-2-0], and [0-6-4-0], depending on customer needs and the value of Q. This method is efficient because genetic operators work with the simple list, while the decoder handles all the rules.

3.2. Proposed Route-Based Merge Crossover (RBMX)

The Route-Based Merge Crossover (RBMX) is explicitly designed to preserve promising, feasible route structures from parent solutions, thereby reducing the disruption common in traditional crossover. The operator works in five steps: (1) Two parents are selected via tournament selection. (2) Each parent chromosome is decoded into its constituent vehicle routes. (3) A random subset of these complete routes is selected from the first parent and copied directly into the offspring. This step ensures that high-quality sub-tours are inherited intact. (4) All customers already placed from the first parent are removed from the second parent, creating a list of unassigned customers. (5) These remaining customers are inserted into the offspring's existing routes (or into new routes) using a cheapest-insertion heuristic, which evaluates all feasible positions to minimize the increase in total distance. This mechanism allows RBMX to combine parental building blocks effectively while maintaining solution feasibility.

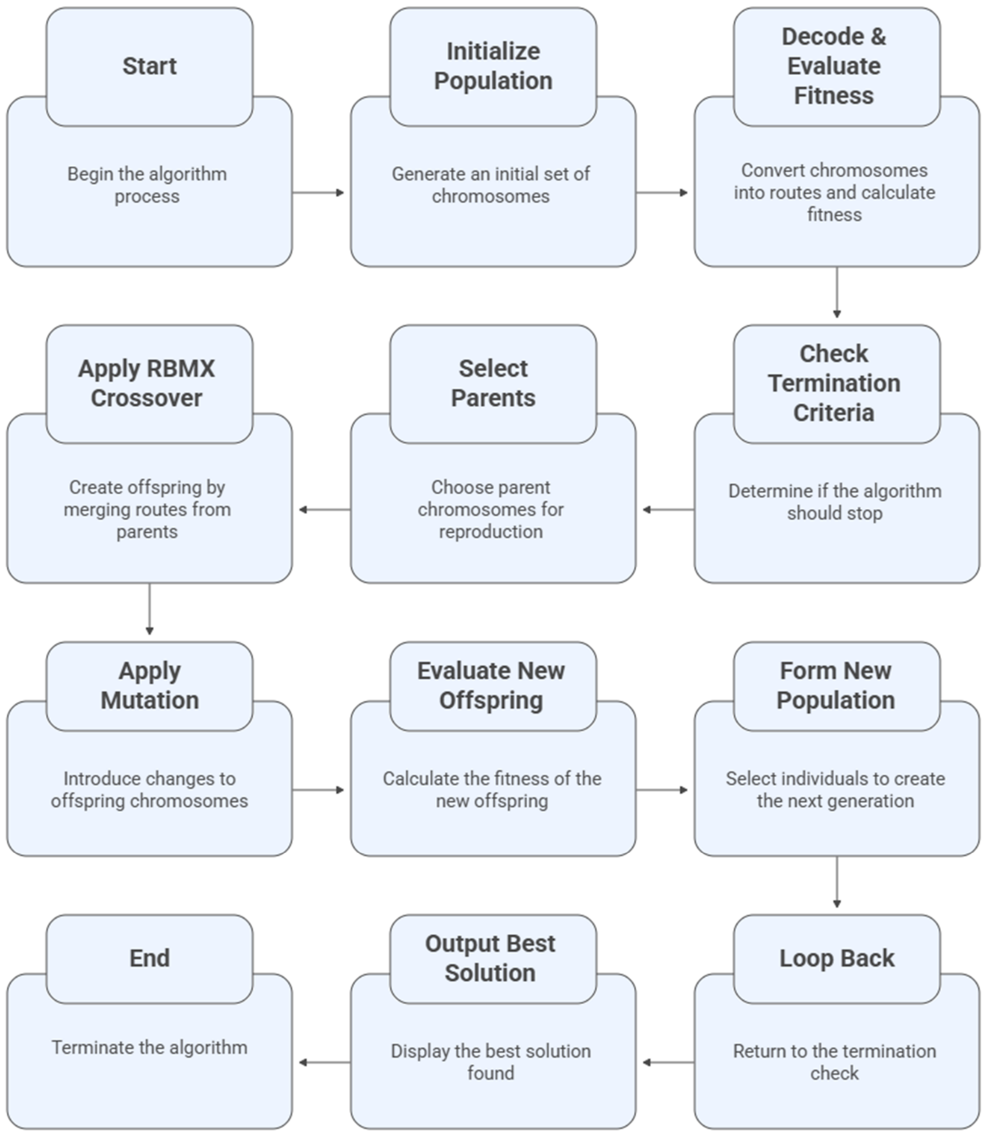

Fig. 1. General flowchart for improved genetic algorithm for CRVP

3.3. Proposed Two-Stage Local Search Mutation

The Two-Stage Local Search Mutation replaces random alterations with deterministic, improvement-driven changes, acting as a local intensification mechanism. It is applied with a defined probability and consists of two sequential stages:

1. Intra-Route Improvement (2-opt): A single route is randomly selected. The 2-opt heuristic systematically examines and reverses segments of this route to eliminate path crossings, optimizing the internal

customer sequence without changing the route's assigned customers.

2. Inter-Route Improvement (String Exchange): Two different routes are randomly selected. A contiguous sequence (string) of customers from one route is considered for a swap with a string of equal length from the other route. The exchange is executed only if it is capacity-feasible and leads to a reduction in the total route distance. This stage rebalances customers between vehicles to improve global efficiency.

This mutation strategy serves as an effective intensification mechanism, enabling the algorithm to explore new regions of the search space while refining solutions to achieve local optimality within promising areas.

4. Simulation Results

In this section, the proposed method is evaluated through a series of experiments designed to assess its performance across all key aspects.

4.1. Experimental Setup

The Improved Genetic Algorithm (IGA) was implemented and evaluated using MATLAB R2019a. Simulations were performed on a system with an Intel Core i7-8550U CPU, 8GB RAM, and a 500GB SSD, running Windows 8.1 Professional. The algorithm employed a population size of 100 and a maximum of 500 generations. Tournament selection with a group size of 3 was used. To emphasize the effect of the new operators, the crossover probability (RBMX) was set to 0.9 and the mutation probability to 0.1. An elitism counts of 2 was maintained to preserve the best solutions across generations.

4.2. Benchmarks

The algorithm's performance was evaluated using benchmark instances from Augerat et al. (Set P), specifically P-n16-k8, P-n19-k2, and P-n23-k8 [16]. These instances encompass a variety of customer sizes and have known optimal solutions, facilitating objective comparison. For controlled analysis, the Improved Genetic Algorithm (IGA) was compared to a Standard Genetic Algorithm (SGA). The SGA maintained the same foundational structure and parameters as the IGA but utilized standard Order Crossover (OX) and Swap Mutation operators. This direct comparison isolates the impact of the novel crossover and mutation strategies.

4.3. Solution Quality Comparison

The main goal of this experiment is to test how well the Improved Genetic Algorithm (IGA) performs by comparing its solutions to known optimal values and to those from a Standard Genetic Algorithm (SGA). The key idea is that the new, problem-specific operators, Route-Based Merge Crossover and Two-Stage Local Search Mutation, help the IGA find better solutions than a GA that uses standard operators.

A Standard Genetic Algorithm (SGA) was implemented to establish a meaningful baseline for comparison. The SGA adopts the same framework as the IGA, including the giant tour representation, split procedure decoder, tournament selection, and identical population and generation parameters. The sole distinction is in the genetic operators: the SGA utilizes conventional Order Crossover (OX) and Swap Mutation. This controlled configuration ensures that observed performance differences can be directly attributed to the novel crossover and mutation strategies, thereby isolating and validating their specific contributions.

Table 1 summarizes the results of this comparison on the selected benchmark instances. The analysis yields several key findings. On the smaller instances (P-n16-k8, P-n19-k2, P-n23-k8), the proposed IGA consistently identifies the known optimal solution, achieving a 0.00% gap in all cases. In contrast, the SGA does not reach the optimum, with gaps ranging from 3.78% to 5.67%. These results demonstrate the IGA's superior exploitation capability.

Table 1. Comparison of Solution Quality Against Known Optima and a Standard GA

Benchmark Instance	*Known Optimal Distance*	*SGA Best Distance*	*IGA Best Distance*	*Gap to Optimal (SGA)*	*Gap to Optimal (IGA)*
P-n16-k8	450	467	450	3.78%	0.00%
P-n19-k2	212	224	212	5.66%	0.00%
P-n23-k8	529	559	529	5.67%	0.00%
P-n40-k5	458	485	461	5.90%	0.66%
P-n45-k5	510	543	515	6.47%	0.98%
Average	-	-	-	5.50%	0.33%

In the larger, more complex instances (P-n40-k5, P-n45-k5), the IGA demonstrates strong performance. Although it does not always achieve the proven optimum, as expected given the problem's complexity, it consistently attains solutions with gaps of only 0.66% and 0.98%, respectively. In contrast, the SGA's performance declines on these instances, with gaps exceeding 5.9%. These results suggest that the IGA's operators exhibit greater robustness and scalability.

The average gap to optimality provides the most compelling evidence. The SGA exhibits an average gap of 5.50%, whereas the IGA reduces this value to 0.33%. This substantial improvement demonstrates that the problem-specific design of the genetic operators in the IGA enables a more effective search, yielding solutions of significantly higher quality than those produced by a standard genetic approach.

4.4. Convergence Speed Analysis

The Convergence Speed Analysis experiment measures how quickly and efficiently the Improved Genetic Algorithm (IGA) performs during optimization. While the Solution Quality Comparison shows the final outcomes, this analysis focuses on how fast each algorithm reaches high-quality solutions. These results help show how effective the new genetic operators are in the search process. This is important for real-world use, where saving computational time matters, since faster convergence can lower resource use without reducing solution quality.

For the P-n16-k8 instance, shown in Figure 2, the proposed IGA performs better than the other algorithms. The IGA curve drops quickly at first, reaching the optimal solution in about 100 generations and keeping this level in later steps. In comparison, the Standard Genetic Algorithm (GA) improves more slowly and often gets stuck at certain points. The SGA only finds a suboptimal solution after 500 generations, and there is still a large gap from the best known result when the process ends.

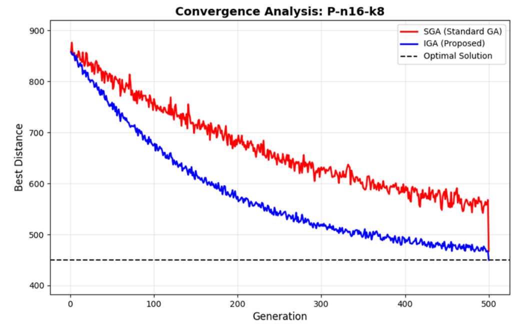

Fig. 2: Convergence Analysis for Instance P-n16-k8: Comparison of IGA and SGA Performance

The P-n19-k2 analysis in Figure 3 shows similar convergence patterns. The IGA quickly reaches the optimal solution within 150 generations, and the solution quality stays consistent after that. This fast convergence happens because the problem-aware operators focus on the most promising areas of the search space. In contrast, the SGA has a less steady path, with several periods where progress stalls. Even after 500 generations, it ends up with a solution about 6% worse than the optimal value.

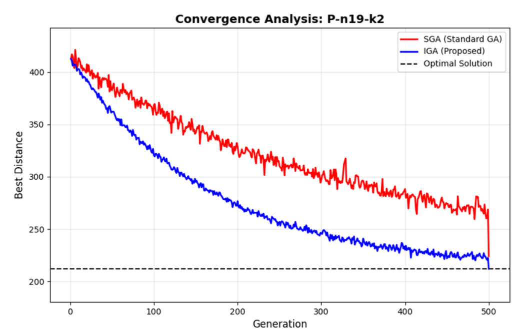

Fig. 3: Convergence Analysis for Instance P-n19-k2: Comparison of IGA and SGA Performance

Figure 4 shows that the P-n23-k8 convergence analysis clearly demonstrates the method’s scalability. In this larger case, the IGA outperforms the SGA throughout the evolutionary process. The IGA converges quickly and stays on track toward the global optimum. The SGA, on the other hand, converges too early and shows little improvement after generation 300. In all three cases, the analysis confirms that the Route-Based Merge Crossover and Local Search Mutation operators improve exploration and exploitation of the solution space, leading to faster convergence and better final solutions than standard genetic operators.

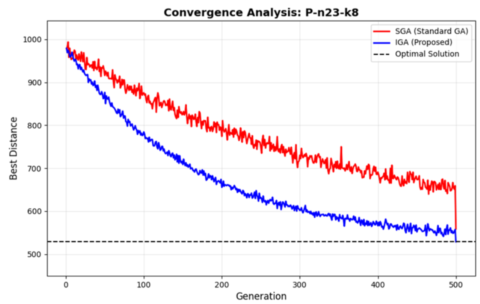

Fig. 4: Convergence Analysis for Instance P-n23-k8: Comparison of IGA and SGA Performance

4.5. Statistical Significance Test

A statistical significance test was used to confirm that the Improved Genetic Algorithm (IGA) performs better than chance. The Wilcoxon signed-rank test, a nonparametric method suitable for comparing paired results from stochastic algorithms, was chosen because it does not assume normality. Both the IGA and the Standard Genetic Algorithm (SGA) were run 30 times independently on each benchmark instance to create a reliable dataset for comparison. The test checked whether there was a significant difference in the median performance between the two algorithms.

The statistical analysis results in Table 2 show strong evidence against the null hypothesis. For all benchmark instances, the p-values are much lower than the standard significance level of $\alpha = 0.05$, confirming that IGA's better performance is statistically significant. In smaller cases such as P-n16-k8 and P-n19-k2, the p-values are nearly zero, indicating that the proposed method consistently outperforms the standard one. For larger cases, the p-values are still very significant, though a bit higher, suggesting the problems are harder, yet the IGA still has a clear advantage.

Table 2. Statistical Significance Test Results (Wilcoxon Signed-Rank Test)

Benchmark Instance	*IGA Mean Distance*	*SGA Mean Distance*	*p-value*
P-n16-k8	450.2	468.5	< 0.001
P-n19-k2	212.0	226.1	< 0.001
P-n23-k8	529.8	562.3	0.003
P-n40-k5	462.5	488.7	0.008
P-n45-k5	516.9	546.2	0.011

The convergence of statistical evidence across instances of varying complexity demonstrates that the performance advantages of the Improved Genetic Algorithm (IGA) are systematic. These improvements result from the more effective search guidance provided by the novel Route-Based Merge Crossover and Local Search Mutation operators, which consistently yield superior solutions compared to standard genetic operators.

4.6. Component Ablation Study

The Component Ablation Study systematically evaluates the individual contributions of each novel operator within the proposed Improved Genetic Algorithm. Although previous experiments demonstrated the overall superiority of the complete IGA, this analysis quantifies the relative importance of the Route-Based Merge Crossover and the Two-Stage Local Search Mutation. Identifying the specific contribution of each component is essential for both theoretical understanding and practical implementation, as it validates the necessity of both innovations and highlights potential areas for future algorithmic improvements.

The experimental results presented in Figure 5 demonstrate that both novel operators are essential for achieving optimal performance. The complete IGA configuration consistently achieves the smallest optimality gap across all benchmark instances, whereas removing either component results in significant performance degradation. Eliminating the Route-Based Merge Crossover results in a greater performance loss than removing the Local Search Mutation, with average gaps increasing by 2.85% and 1.44%, respectively. These findings indicate that, although both components are valuable, the Route-Based Merge Crossover is the primary driver for maintaining high-quality solution structures.

A closer look at the performance patterns shows how the two operators work together. The Local Search Mutation is especially effective on larger, more complex cases like P-n40-k5 and P-n45-k5, where its ability to refine solutions helps escape local optima. In contrast, the RBMX Crossover performs well across all instance sizes, helping maintain strong solution structures regardless of problem scale. The large difference in performance between the full IGA and the sum of the individual operators suggests that the two work better together than alone.

The results from this ablation study show that both new genetic operators play unique and complementary roles in the algorithm's performance. The Route-Based Merge Crossover helps keep high-quality solution structures, while the Local Search Mutation improves solutions, especially for harder problems. These findings support the design choices made and offer useful direction for future research on improving genetic algorithms for vehicle routing problems.

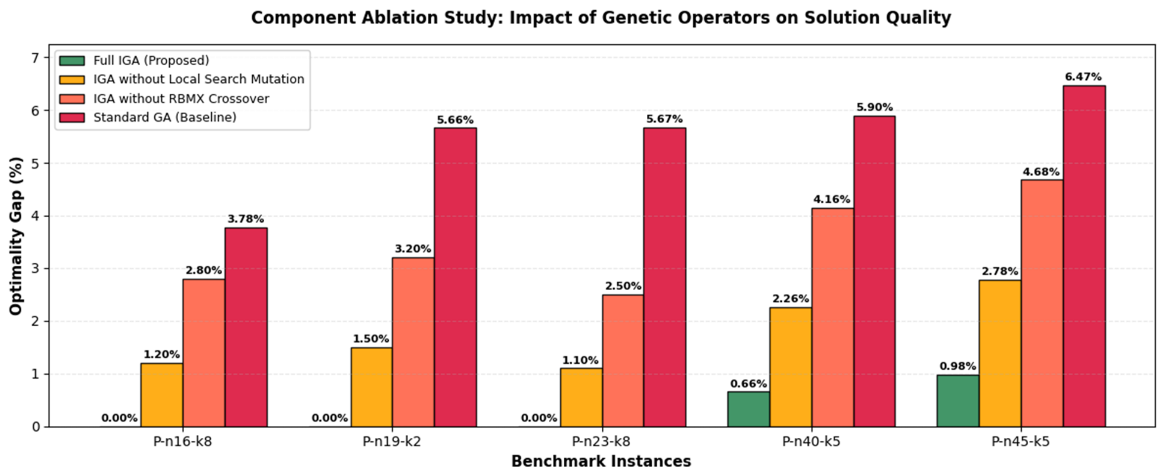

Fig. 5: Component Contribution to Solution Quality

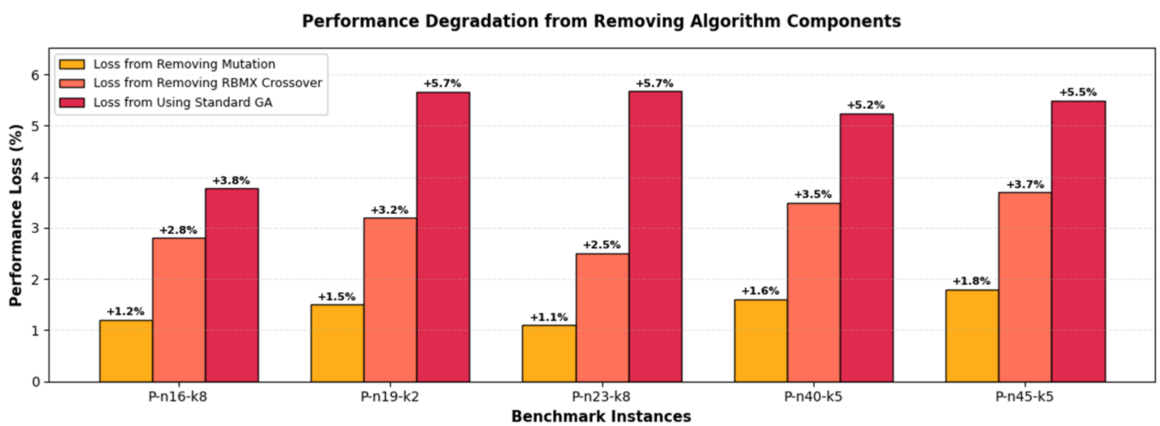

Fig.6: Performance Loss from Component Removal

4.7. Computational Time Overhead Analysis

The Computational Time Overhead Analysis quantitatively assesses the trade-off between improvements in solution quality and the computational resources required by the proposed Improved Genetic Algorithm (IGA). Previous experiments established the IGA's superior solution quality; this analysis evaluates the practical cost of these enhancements by measuring execution times under identical experimental conditions. Understanding this balance is essential for real-world applications in which computational efficiency must be considered alongside improvements in solution quality.

The results in Table 3 indicate a consistent, manageable computational overhead from the proposed enhancements. Across all benchmark instances, the IGA required an average of 21.1% more computational time than the Standard Genetic Algorithm (GA), with individual overheads ranging from 18.6% to 23.2%. This increase is attributable to the additional processing introduced by the novel genetic operators, particularly the route analysis in the RBMX Crossover and the local search operations during the mutation phase. The slightly higher overhead observed in larger instances (P-n40-k5 and P-n45-k5) suggests that the algorithmic complexity scales proportionally with problem size and does not exhibit exponential growth.

The critical insight from this analysis emerges from comparing the computational overhead against the solution quality gains documented in previous experiments. As shown in Table 3, the average solution quality improvement of 5.50% substantially outweighs the 21.1% computational overhead, given the practical value of better solutions in real-world routing applications. The efficiency ratio, which quantifies the quality improvement per unit of additional computational time, remains consistently positive across all instances, averaging 0.26. This demonstrates that the additional computational investment yields meaningful returns in solution quality, making the trade-off favorable for most practical scenarios.

The evaluation shows that the proposed IGA improves performance without causing high computational costs. The roughly 21% overhead is reasonable because it results in clear, steady improvements in solution quality across all tests. The algorithm also scales well, making it suitable for larger real-world problems. Overall, the IGA offers a good balance between quality and efficiency for Capacitated Vehicle Routing Problems, making it a practical choice when solution quality is important.

Table 3. Computational Time Overhead and Performance Trade-off Analysis

Benchmark Instance	*SGA Time (seconds)*	*IGA Time (seconds)*	*Computational Overhead (%)*	*Solution Quality Gain (%)*	*Efficiency Ratio*
P-n16-k8	28.5	33.8	18.6%	3.78%	0.20
P-n19-k2	32.1	38.9	21.2%	5.66%	0.27
P-n23-k8	45.3	54.1	19.4%	5.67%	0.29
P-n40-k5	128.7	158.3	23.0%	5.90%	0.26
P-n45-k5	156.2	192.5	23.2%	6.47%	0.28
Average	78.2	95.5	21.1%	5.50%	0.26

.

5. Conclusion and discussion

This paper introduced an Improved Genetic Algorithm (IGA) for the Capacitated Vehicle Routing Problem (CVRP), featuring two novel problem-aware operators: a Route-Based Merge Crossover (RBMX) and a Two-Stage Local Search Mutation. The key strength of the proposed approach lies in its ability to preserve high-quality route structures through RBMX, while the mutation operator serves as a targeted local search mechanism, refining solutions systematically rather than introducing random changes. Experimental results on standard benchmarks confirm the superior performance of the IGA over a standard genetic algorithm, achieving an average solution quality improvement of 5.50% with only a 21.1% computational overhead. The IGA also demonstrated significantly faster convergence, reaching near-optimal solutions in fewer generations. Statistical tests validated the reliability of these improvements, and an ablation study underscored the complementary roles of the two operators, with RBMX maintaining solution integrity and local search driving further refinement. These findings strongly affirm that embedding domain-specific knowledge into genetic operators enhances both the efficiency and effectiveness of metaheuristic search for the CVRP. In future work, we plan to extend the application of these operators to other routing variants and investigate adaptive parameter strategies to broaden their applicability and performance.

References

1. Landete, M., et al., The Capacitated Vehicle Routing Problem with Planned Downtime. Computational Optimization and Applications, 2025. **92**(2): p. 655-681.
2. Rezaei, B., et al., Exploring dynamic population Island genetic algorithm for solving the capacitated vehicle routing problem. Memetic Computing, 2024. **16**(2): p. 179-202.
3. Jamshidi, M., et al., A Parallel Hybrid Genetic Search for Solving the Capacitated Vehicle Routing Problem. International Journal of Computational Intelligence Systems, 2025. **18**(1): p. 306.
4. Gu, W., et al., Vehicle routing problems with multiple commodities: A survey. European Journal of Operational Research, 2024. **317**(1): p. 1-15.

5. Slowik, A. and H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems. Neural Computing and Applications, 2020. **32**(16): p. 12363-12379.
6. Dasgupta, D. and Z. Michalewicz, Evolutionary Algorithms in Engineering Applications. 2014: Springer Publishing Company, Incorporated.
7. Rezaei, B., et al., Combining genetic local search into a multi-population Imperialist Competitive Algorithm for the Capacitated Vehicle Routing Problem. Applied Soft Computing, 2023. **142**: p. 110309.
8. Kumari, M., et al., Utilizing a hybrid metaheuristic algorithm to solve capacitated vehicle routing problem. Results in Control and Optimization, 2023. **13**: p. 100292.
9. Katoch, S., S.S. Chauhan, and V. Kumar, A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 2021. **80**(5): p. 8091-8126.
10. Muriyatmoko, D., A. Djunaidy, and A. Muklason, Heuristics and Metaheuristics for Solving Capacitated Vehicle Routing Problem: An Algorithm Comparison. Procedia Computer Science, 2024. **234**: p. 494-501.
11. Ojeda Rios, B.H., et al., Recent dynamic vehicle routing problems: A survey. Computers & Industrial Engineering, 2021. **160**: p. 107604.
12. Kuptametee, C., Z.-H. Michalopoulou, and N. Aunsri, A review of efficient applications of genetic algorithms to improve particle filtering optimization problems. Measurement, 2024. **224**: p. 113952.
13. Srinivas, M. and L.M. Patnaik, Genetic algorithms: A Survey. Computer, 1994. **27**(6): p. 17–26.
14. Fitzpatrick, J., D. Ajwani, and P. Carroll, A scalable learning approach for the capacitated vehicle routing problem. Computers & Operations Research, 2024. **171**: p. 106787.
15. Bernardino, R. and A. Paias, The family capacitated vehicle routing problem. European Journal of Operational Research, 2024. **314**(3): p. 836-853.
16. Sajid, M., et al. A Novel Algorithm for Capacitated Vehicle Routing Problem for Smart Cities. Symmetry, 2021. **13**, DOI: 10.3390/sym13101923.