A Comparison of Metaheuristic Methods for the Vehicle Routing Problem

Manal El Jaouhari[1], Ghita Bencheikh[2], Ghizlane Bencheikh[3]
Moulay Ismaïl University of Meknes, Morocco[1, 3]
LINEACT, Cesi Engineering School, Assat, France[2]

***Abstract*—The Capacitated Vehicle Routing Problem (CVRP) is a fundamental NP-hard combinatorial optimization problem with important applications in logistics and distribution systems. Although numerous advanced approaches have been proposed in recent years, systematic benchmarking of classical metaheuristic algorithms under a unified experimental framework remains limited. This study evaluates the performance and trade-offs of four well-known metaheuristics: Hill Climbing (HC), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithms (GA). All methods are implemented within the same computational environment and assessed on benchmark CVRP instances, using the CPLEX exact solver as a reference for global optimality. The results indicate that ACO achieves the smallest optimality gaps and often approaches optimal solutions, at the cost of higher computational effort. PSO strikes a favorable balance between solution quality and runtime across the tested instances, whereas HC delivers very fast solutions but degrades as problem complexity increases. GA exhibits higher variability and less competitive performance under the selected parameter settings. Overall, this comparative analysis highlights the strengths and limitations of classical metaheuristics and establishes a reproducible baseline for future research, including hybrid and learning-assisted approaches for scalable vehicle routing optimization.**

Keywords*—*Metaheuristics; Ant Colony Optimization; Hill Climbing; Genetic Algorithm; Particle Swarm Optimization; exact algorithm; Capacitated Vehicle Routing Problem

I. Introduction

The Vehicle Routing Problem (VRP), first introduced by Dantzig and Ramser [3], is one of the most important problems in combinatorial optimization and has many applications in logistics and transportation. One of its best-known variants is the Capacitated Vehicle Routing Problem (CVRP). In this problem, a fleet of vehicles with limited capacity must deliver goods to a set of customers, and the objective is to minimize the total travel cost.

The CVRP is an NP-hard problem, meaning that it becomes very difficult to solve exactly when the number of customers increases. Exact optimization methods [14], [16] can find optimal solutions, but their computational cost grows quickly, which limits their use to small problem instances. For this reason, heuristic and metaheuristic algorithms are widely used to obtain good-quality solutions in a reasonable time, especially for medium- and large-scale CVRP instances.

Over the years, many classical metaheuristics have been applied to solve the CVRP, such as Genetic Algorithms (GA) [7], [8], Ant Colony Optimization (ACO) [4], [5], Particle Swarm Optimization (PSO) [9], [15], and Hill Climbing (HC). In addition, modern state-of-the-art VRP solvers often rely on more advanced strategies, including Adaptive Large Neighborhood Search (ALNS), Variable Neighborhood Search (VNS), Tabu Search, and hybrid metaheuristics that combine different optimization mechanisms to improve performance on large-scale instances [10], [11], [17], [18], [19].

More recently, learning-assisted methods, such as reinforcement learning approaches, have also gained attention as promising directions for solving routing problems. These developments further highlight the importance of establishing reliable baselines through systematic benchmarking of classical metaheuristics [2], [6].

Despite the large number of studies in this area, direct comparisons of classical metaheuristics under the same experimental conditions remain limited. Many works focus on improving a single algorithm or use different benchmarks and implementations, which makes it difficult to clearly understand the relative strengths and weaknesses of these approaches [1], [21].

The main objective of this study is to provide a structured comparison of four widely used metaheuristic algorithms (HC, ACO, PSO, and GA) for solving the CVRP. All algorithms are implemented within the same experimental framework and tested on identical problem instances. Their performance is evaluated in terms of solution quality and computational time, using optimal solutions obtained with CPLEX as reference.

This study is structured as follows: Section II reviews related work on metaheuristics for CVRP. Section III presents the mathematical formulation of the problem. Section IV describes the four metaheuristic algorithms evaluated in this study. Section V discusses the experimental results, and Section VI concludes the study and suggests directions for future research.

II. Related Work

The CVRP has been widely studied using constructive heuristics, local search methods, population-based metaheuristics, and hybrid matheuristic frameworks. Although significant progress has been achieved, systematic benchmarking of classical metaheuristics under unified experimental conditions remains relatively limited. As a result, their comparative strengths and weaknesses are not always clearly established.

Avdoshin and Beresneva [1] provide one of the few comparative studies based on a common framework, focusing on constructive heuristics [25] such as nearest neighbor, savings, sweep, and insertion methods. Tiwari and Sharma [21] further investigate local search and Tabu Search strategies for last-mile delivery, comparing neighborhood structures under controlled settings. While informative, these works mainly address local search-based approaches and do not cover broader metaheuristic families. Tan and Yeh [20] note that many recent VRP studies rely on different or newly generated datasets, which makes it difficult to compare algorithms under a unified standard. They emphasize the need for publicly available benchmark datasets to improve comparability across studies.

In recent years, research has increasingly shifted toward advanced and state-of-the-art methods. Variable Neighborhood Search (VNS) and its extensions have been successfully applied to several VRP variants with capacity and time constraints [11], [19], [24]. Adaptive Large Neighborhood Search (ALNS) has also emerged as one of the most effective frameworks for large-scale routing problems, with important contributions on operator design and adaptive mechanisms [17],[18],[23]. In parallel, hybrid matheuristics combining metaheuristics with exact optimization components have achieved highly competitive results for the CVRP and related routing problems [13],[14],[16].

Despite these advances, existing studies often rely on different benchmarks, problem variants, or experimental protocols, which makes direct comparison across algorithms difficult. In particular, a controlled evaluation of classical metaheuristics such as Hill Climbing, Ant Colony Optimization, Particle Swarm Optimization, and Genetic Algorithms under identical conditions is still missing.

To address this gap, we implement HC, ACO, PSO, and GA within a single unified experimental framework and benchmark their performance against exact optimal solutions obtained via CPLEX. By providing a reproducible baseline, this work supports future research on hybrid, adaptive, VNS-based, and learning-assisted approaches for scalable vehicle routing optimization.

III. Mathematical Formulation

- Notation

N : Set of nodes, including the depot and customers.

K : Number of vehicles.

d_{ij} : Euclidean distance between node i and node j.

q_j : Demand of customer j.

C : Vehicle capacity.

x_{ijk} : Binary decision variable that equals 1 if vehicle k travels from node i to node j, and 0 otherwise.

u_{ik} : Continuous variable representing the load of vehicle k when leaving node i.

- Objective Function

The objective is to minimize the total distance traveled by all vehicles:

$$Min\ \sum_{k=1}^{K}\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ijk} \quad (1)$$

- Constraints

Capacity Constraints: Ensure that the total demand served by each vehicle does not exceed its capacity:

$$\sum_{j\in N} q_j \sum_{i\in N} x_{ijk} \leq C \ / \ k \in \{1,..,\mathrm{K}\} \quad (2)$$

Visit Constraints: Each customer must be visited exactly once by one vehicle:

$$\sum_{k=1}^{K}\sum_{i\in N} x_{ijk} = 1 \ / \ \forall j \in N \quad (3)$$

Flow Conservation Constraints: Ensure that if a vehicle arrives at a node, it must leave that node:

$$\sum_{j\in N} x_{ijk} = \sum_{j\in N} x_{jik} \ / \ \forall i \in N, \forall k \in \{1,..,\mathrm{K}\} \quad (4)$$

Depot Constraints: Each vehicle must start and end at the depot:

$$\sum_{j\in N} x_{0jk} = 1 \ / \ \forall k \in \{1,..,\mathrm{K}\} \quad (5)$$

$$\sum_{j\in N} x_{j0k} = 1 \ / \ \forall k \in \{1,..,\mathrm{K}\} \quad (6)$$

Subtour Elimination (MTZ Constraints): To prevent subtours (cycles that do not include the depot), we adopt the classical Miller–Tucker–Zemlin (MTZ) formulation [12]:

$$u_{jk} \geq u_{ik} + q_j - C\,.\left(1 - x_{ijk}\right) \ / \ \forall i \in N, \forall j \in N\backslash\{0\}, \mathrm{i} \neq \mathrm{j}, \forall k \in \{1,..,\mathrm{K}\} \quad (7)$$

The load variables are bounded as follows:

$$q_j \leq u_{ik} \leq C \ / \ \forall i \in N, \forall k \in K \quad (8)$$

These constraints eliminate subtours by ensuring that the vehicle load increases consistently along each route, thereby forcing all visited customers to be connected to the depot.

IV. Metaheuristic Algorithms

Metaheuristic algorithms are generally classified into two categories: single solution-based algorithms and population-based algorithms [1]. In this study, we include algorithms from both categories: HC as a single solution-based metaheuristic, and ACO, PSO, and GA as population-based metaheuristics. These methods are selected as representative classical approaches, providing a reproducible baseline for comparing solution quality and computational efficiency on the Capacitated Vehicle Routing Problem (CVRP) [22]. In the following subsections, we briefly describe each algorithm and its adaptation to the CVRP.

A. Hill Climbing

Hill Climbing is a local search optimization algorithm that starts from an initial solution and iteratively improves it by moving to a neighboring solution with a better objective value. At each step, the algorithm selects the best available improvement in the local neighborhood and continues the search until no further improvement is possible. The process stops when a local optimum is reached.

The application of HC to the CVRP is summarized in Algorithm 1:

Algorithm 1: HC Algorithm applied to the CVRP

```
x0 : Initial solution
x : Current solution
x* : Best solution
x ← x0
x* ← x0
While True do
|  x ← argmin_{xi neighbor of x} f(xi)
|  if (f(x) < f(x*)) then
|  |    x* ← x
|  else
|  |    Break
|  End
End
```

Solutions are represented as permutations of customer indices, with vehicle routes separated by depot markers. The neighborhood operator generates new solutions through 2-opt exchanges within routes and relocation moves between routes, ensuring feasibility by checking capacity constraints before accepting modifications.

B. Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic inspired by the behavior of ants in nature. Artificial ants construct their solutions by selecting components based on two values: the pheromone trails $(\tau_{ij})_{i,j\in N}$ and heuristic information $(\eta_{ij})_{i,j\in N}$, in the CVRP case, it's the inverse of the distance between customers. The pheromone trails are updated over time to reinforce good solutions and reduce the influence of poor ones.

The application of ACO to the CVRP is summarized by Algorithm 2:

Algorithm 2: ACO Algorithm applied to the CVRP

```
Initialize the pheromone trail by a small value : τij ← τ0
For (a Maximal number of iterations) do
|  For (each ant k) do
|  |  Initialize the current vehicle v by 1 : v ← 1
|  |  Initialize the vehicle's route by depot 0
|  |  While (The set of unserved clients is not empty) do
|  |  |  Select a client j according to :
|  |  |      P_ij = (η_ij^α . τ_ij^β) / (Σ_l η_il^α . τ_il^β)
|  |  |  If ( vehicle capacity is not exeed) then
|  |  |  |  Insert j into the route
|  |  |  else
|  |  |  |  return to depot and v ← v + 1
|  |  |  End
|  |  |  Remove j from the unserved clients set
|  |  End
|  End
|  Calculate the Best Solution
|  Update the pheromone
End
End
```

$$P_{ij} = \frac{\eta_{ij}^{\alpha}.\tau_{ij}^{\beta}}{\sum_l \eta_{il}^{\alpha}.\tau_{il}^{\beta}}$$

Each ant constructs a solution by sequentially selecting customers using the probabilistic transition rule. Feasibility is maintained by tracking cumulative vehicle load and returning to the depot when adding the next customer would violate capacity constraints.

C. Genetic Algorithm

Genetic Algorithms (GA) are population-based metaheuristics inspired by Darwin's theory of evolution. A set of solutions (population) evolves over a given number of iterations by applying genetic operators such as selection, crossover, and mutation. The quality of a solution (individual), called fitness, depends on the objective function of this solution. At each iteration, several individuals are selected according to their fitness in order to create new ones by applying two operators: Crossover and Mutation. At the end of the iteration, a phase of replacement is applied to select the individuals that pass to the next iteration. Algorithm 3 presents the application of GA to the CVRP.

Algorithm 3: GA applied to the CVRP

```
Generate the initial population
For (a maximal number of iterations) do
|  Evaluate individuals
|  Select a subset of individuals for crossover
|  Apply crossover operator to selected individuals taken two by two.
|  Select a subset of individuals for mutation
|  Apply mutation operator to selected individuals.
|  Replace the current population by the new one
|  Calculate the best solution
End
```

Individuals are encoded as permutations of customers. The crossover operator preserves relative customer order from parents, while mutation randomly swaps two customers. Infeasible solutions are repaired by redistributing customers among vehicles to satisfy capacity constraints.

D. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based metaheuristic inspired by the collective behavior of social organisms such as bird flocks or fish schools. In PSO, each particle represents a candidate solution and moves through the search space by updating its position based on its own best solution and the best solution found by the swarm. The application of PSO to the CVRP is presented in Algorithm 4.

Algorithm 4: PSO applied to the CVRP

```
Initialize particles with random positions and velocities
Calculate the best solution x*
For (a maximal number of iterations) do
|   For (each particle p) do
|   |   If (f(p) < f(x*)) then
|   |   |   x* ← p
|   |   End
|   p ← Update personal best position
|   End
|   For (each particle i)
|   |   Update velocity for particle i
|   |   Update position for particle i
|   End
End
```

Particle positions are represented as priority-based encodings, where real-valued priorities determine customer visit order. Positions are decoded into feasible routes using a greedy insertion procedure that respects vehicle capacity constraints.

The detailed parameter settings for each algorithm, including population size, iteration limits, and coefficients, are reported in Section V. All algorithms are executed multiple times to account for stochastic behavior, and results are averaged over these runs to ensure statistical reliability.

V. Experiment Results

To evaluate the performance of the four metaheuristic algorithms, we implemented them in Python and applied them to randomly generated CVRP instances. The number of clients in each instance varies from 5 to 15, while the number of vehicles ranges from 2 to 5. Exact solutions for these instances were obtained using the CPLEX solver in Python, providing a reference for solution quality.

The CVRP instances used in this study are deliberately small (5–15 customers) so that they can be directly compared with exact optimal solutions obtained using CPLEX. This creates a controlled testing environment where the solution quality of each metaheuristic can be evaluated in a reliable way. Although these instances are not representative of large real-world logistics problems, they provide a useful baseline for understanding and comparing the behavior of classical metaheuristic algorithms before considering larger instances in future work.

All computations were performed on a laptop with an Intel® Core™ i5-8300H processor (2.30 GHz) and 8 GB of RAM. Each metaheuristic was executed multiple times to account for its stochastic nature, and the results reported are averaged over these runs. Detailed parameter settings for each algorithm are presented in the following subsection.

A. Parameter Settings of Metaheuristic Algorithms

The parameter values for all four metaheuristic algorithms (HC, ACO, PSO, and GA) were chosen based on preliminary tests on a small set of instances. Table I shows the settings used for all methods.

TABLE I. PARAMETER SETTINGS FOR METAHEURISTIC ALGORITHMS

Algorithm	Parameter	Value
HC	Number of iterations	1000
ACO	Number of ants	100
	Pheromone importance (α)	1
	Heuristic importance (β)	3
	Pheromone deposit factor	70
GA	Population size	150
	Number of generations	500
	Mutation rate	0.1
PSO	Number of particles	50
	Number of iterations	200
	Inertia weight (w)	0.4
	Cognitive coefficient (c_1)	1.5
	Social coefficient (c_2)	1.7

B. Computational Results

Table II presents the computational results obtained for the CVRP using the four metaheuristic algorithms and compares them with the exact optimal solutions provided by the CPLEX solver.

For each problem instance, the table reports the optimal value found by CPLEX as well as the results obtained by each metaheuristic method. The "Optimal Value" column corresponds to the exact solution returned by CPLEX, while the "CPU" column indicates the computational time (in seconds) required to obtain this solution. Table II shows computational results.

Since metaheuristic algorithms are stochastic, each experiment was repeated 10 independent times for every instance. For each algorithm, the solution value reported corresponds to the best result obtained over these 10 executions.

For the metaheuristic algorithms, three performance indicators are presented:

- BD (Best Distance): the best solution value obtained over the 10 runs.
- GAP (%): the relative difference between BD and the optimal solution provided by CPLEX. It is computed as follows:

$$GAP = \frac{BD - Optimal\ value}{Optimal\ value} \times 100 \quad (8)$$

- CPU: the execution time required by each algorithm.

The problem instances are grouped according to their size and the number of vehicles considered in each case.

TABLE II. COMPUTATIONAL RESULTS

Instances	CPLEX		ACO			HC			GA			PSO		
	Optimal Value	CPU	BD	GAP	CPU	BD	GAP	CPU	BD	GAP	CPU	BD	GAP	CPU
A-n5-k2	737,3459	0,03	737,3459	0%	0,6423	737,3459	0%	0,0015	737,3459	0%	3,4149	737,3459	0%	0,2787
B-n10-k3	1488,97	0,92	1488,9701	0%	5,4658	1590,455	7%	0,0266	1747,5083	17%	7,6644	1698,1143	14%	0,9505
C-n10-k3	1043,24	0,05	1043,2374	0%	1,9157	1184,322	14%	0,0165	1284,0174	23%	8,3398	1076,0501	3%	0,4345
D-n10-k3	1345,6564	0,09	1345,6565	0%	2,1337	1410,9831	5%	0,013	1618,81817	20%	19,3497	1407,0059	5%	0,3069
E-n10-k3	1344,8953	0,28	1344,8954	0%	1,4901	1344,8954	0%	0,0207	1931,4923	44%	18,35	1561,5705	16%	0,3965
F-n10-k3	1205,1432	0,05	1205,1432	0%	1,9442	1205,1432	0%	0,0202	1578,3889	31%	17,0545	1312,8234	9%	0,3379
G-n10-k3	1055,0557	0,12	1119,7954	6%	1,9904	1219,6917	16%	0,0085	1216,5738	15%	19,7098	1226,3541	16%	0,3473
H-n10-k3	1164,7701	0,05	1164,7702	0%	2,3681	1164,7702	0%	0,0368	1511,8412	30%	7,8897	1506,6458	29%	0,3494
X-n15-k5	1292,7975	0,11	1292,7975	0%	4,5217	1292,7975	0%	0,1402	2459,6503	90%	13,9031	1814,1539	40%	0,2681
A-n15-k5	1812,6965	0,14	1903,1864	5%	4,6785	2255,2393	24%	0,0693	2396,3247	32%	11,7859	2292,7553	26%	2,5267
B-n15-k5	1432,8015	0,34	1432,8015	0%	4,4889	1549,2988	8%	0,0847	2592,7037	81%	11,7976	1863,9569	30%	0,2843
C-n15-k5	1403,0113	0,5	1403,0114	0%	3,4049	1952,2029	39%	0,0747	2582,2474	84%	12,075	1852,3453	32%	0,2852
D-n15-k5	1437,4749	0,06	1437,4749	0%	4,2683	1731,5309	20%	0,1034	2590,5014	80%	12,1401	1742,3011	21%	0,2902
E-n15-k5	1469,6608	0,22	1469,6609	0%	3,2084	1473,7939	0%	0,1027	2663,3091	81%	15,9196	1862,7521	27%	0,2625
F-n15-k5	1677,3105	0,05	1677,3106	0%	4,4511	1695,7807	1%	0,0993	2543,7429	52%	12,1651	2165,1462	29%	0,4388
G-n15-k5	1575,154	0,53	1593,1939	1%	4,5538	1978,5961	26%	0,0461	2134,9409	36%	12,4126	2212,5714	40%	0,4285
H-n15-k5	1401,1991	0,08	1411,4728	1%	4,4402	1403,0068	0%	0,106	2216,4903	58%	10,8925	2218,7381	58%	0,4055
I-n15-k5	1673,7506	0,23	1673,7506	0%	4,5742	1789,8502	7%	0,0939	2503,7557	50%	13,0543	2705,0827	62%	0,486

According to these results, we note that for small instances with 5 customers, all metaheuristics reached the optimal solution. With small differences in execution time, HC achieves the solution almost instantly, while ACO and PSO require more computational time. This indicates that for very small problem sizes, simple local search methods are highly efficient.

For instances with 10 customers, we observe differences between the algorithms in terms of solution value and computational time. ACO consistently provides high-quality solutions, often reaching the optimal solution or maintaining a very small gap. However, this accuracy comes at the cost of higher computational times compared to HC and PSO. Hill Climbing remains extremely fast, but its solution quality degrades as the problem becomes more complex, with gaps reaching up to 44% in some instances. Genetic Algorithm (GA) shows moderate performance, producing acceptable solutions but with relatively long execution times. PSO generally achieves a good compromise between solution quality and computational efficiency, maintaining low gaps in most cases while keeping execution times reasonable.

When the problem size increases to 15 customers, the differences between the algorithms become clearer. ACO continues to provide the most accurate solutions, often reaching the optimal value or showing very small gaps. However, its execution time increases significantly. HC remains the fastest algorithm, but its solution quality decreases with very large gaps that can exceed 80%, which limits its use for larger instances. GA shows weak performance in both solution quality and execution time, with large gaps and slow convergence. PSO exhibits more stable behavior, achieving better solutions than HC and GA while keeping reasonable execution times, although it does not always reach the optimal solution.

To better visualize the differences between the metaheuristics in terms of CPU time, Fig. 1 presents the execution time required by each algorithm for all instances.

As discussed previously, we observe that the Genetic Algorithm is the most time-consuming metaheuristic, requiring significantly more CPU time than the other methods. While Hill Climbing (HC) is the fastest algorithm, obtaining solutions almost instantly. ACO needs moderate computational times, generally higher than HC and PSO but lower than GA. PSO demonstrates a balanced performance, maintaining reasonable execution times across all instances.

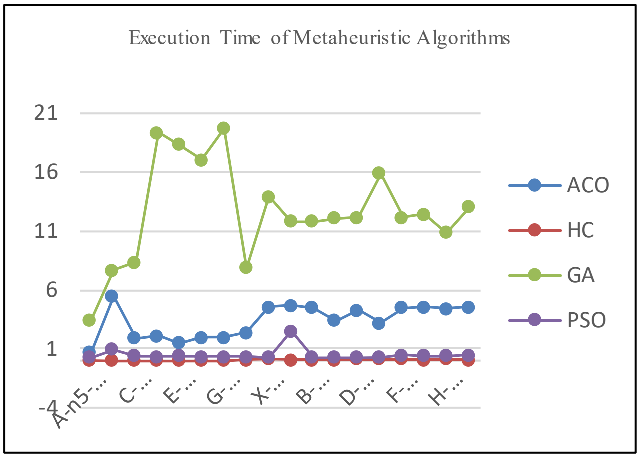

Fig. 1. Execution time of metaheuristic algorithms.

Fig. 2 presents the performance gaps of the four metaheuristic algorithms relative to the optimal solutions obtained by CPLEX.

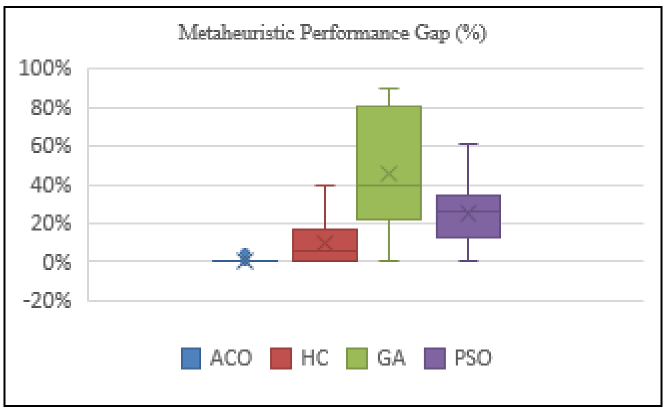

Fig. 2. Metaheuristic performance gap (%).

As shown in Fig. 2, ACO consistently achieves the smallest gaps, with median values near zero and little variation across all instances, confirming its high solution quality. HC performs well on small instances with very low gaps, but its gaps increase and become more variable as the problem size grows, showing that it is less effective for larger instances. GA has large gaps and wide variation, indicating inconsistent performance and slower convergence. PSO achieves moderate gaps with relatively stable performance, offering a good balance between solution quality and reliability.

VI. Conclusion and Future Directions

This study evaluated the performance of four metaheuristics (ACO, PSO, HC, and GA) for solving the CVRP using randomly generated instances. The algorithms were implemented under a unified framework and benchmarked against optimal solutions obtained via CPLEX.

The results reveal distinct performance trade-offs among the methods, consistent with the "no free lunch" theorem: no single metaheuristic dominates across all problem characteristics. ACO consistently achieved the highest solution quality, frequently matching or closely approximating optimal values, though at the cost of substantially longer computation times. PSO demonstrated the most balanced performance, maintaining reasonable solution quality while keeping execution times low, making it well-suited for time-sensitive applications. HC excelled in computational efficiency, generating solutions almost instantaneously, but exhibited significant performance degradation as problem complexity increased, with optimality gaps exceeding 80% on the largest instances tested. GA showed the weakest overall performance in our experiments, requiring the longest execution times while producing solutions with large and inconsistent optimality gaps. This latter result warrants further investigation, as it may reflect parameter sensitivity or implementation-specific issues rather than fundamental algorithmic limitations.

The current study has important limitations. Our analysis focused on small-scale instances (5-15 customers) to enable rigorous validation against exact solutions. Additionally, the relatively poor GA performance diverges from typical findings in the literature and requires further examination.

Future research will proceed along two complementary directions. First, we will extend this comparative analysis to larger problem instances (50-200 customers) and incorporate additional modern metaheuristics, including ALNS, VNS, and ILS, to better understand algorithmic scalability and behavior under more challenging conditions. Second, given the complementary strengths observed across methods, we will investigate hybrid and adaptive approaches. Specifically, we plan to develop a multi-agent framework that dynamically selects or combines these metaheuristics based on problem characteristics. This approach aims to leverage the solution quality of ACO, the efficiency of PSO, and the speed of HC within an integrated system that adapts to instance-specific features, potentially offering more robust performance across diverse CVRP scenarios.

References

[1] Avdoshin, S. M., & Beresneva, E. N. (2019). Constructive heuristics for Capacitated Vehicle Routing Problem: a comparative study. Proceedings of ISP RAS, 31(3), 145–156.

[2] Bencheikh, G. (2024). Metaheuristics and machine learning convergence: A comprehensive survey and future prospects. In Metaheuristic and Machine Learning Optimization Strategies for Complex Systems (pp. 276–322). Springer.

[3] Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management Science, 6(1), 80–91.

[4] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics—Part B, 26(1), 29–41.

[5] Dorigo, M., & Stützle, T. (2019). Ant colony optimization: Overview and recent advances. In M. Gendreau & J.-Y. Potvin (Eds.), Handbook of Metaheuristics (3rd ed., pp. 311–351). Springer.

[6] El Jaouhari, M., Bencheikh, G., & Bencheikh, Gh. (2025). Metaheuristic and reinforcement learning techniques for solving the vehicle routing problem: A literature review. Journal of Traffic and Transportation Engineering (English Edition). In press.

[7] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

[8] Kala, R. (2024). An introduction to evolutionary computation. In R. Kala (Ed.), Autonomous Mobile Robots (pp. 715–759). Academic Press.

[9] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 1942–1948.

[10] Lubnina, E., & Kochetov, Y. (2024). VNS algorithm for the periodic pick-up and delivery helicopter routing problem in oil and gas offshore. Yugoslav Journal of Operations Research, 34(3), 407–421.

[11] Matijević, L., Ilin, V., Davidović, T., Jakšić-Krüger, T., & Pardalos, P. M. (2024). General VNS for asymmetric vehicle routing problem with time and capacity constraints. Computers & Operations Research, 167, 106630.

[12] Miller C.; Tucker A.; Zemlin R. (1960) Integer programming formulations and travelling salesman problems. J. ACM 7: 326-329.

[13] Xue, N., Bai, R., Qu, R., & Aickelin, U. (2021). A hybrid pricing and cutting approach for the multi-shift full truckload vehicle routing problem. European Journal of Operational Research, 292(2), 500–514

[14] Pecin, D., Pessoa, A., Poggi, M., & Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. Mathematical Programming Computation, 9(1), 61–100.

[15] Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. Swarm Intelligence, 1(1), 33–57.

[16] Sadykov, R., Uchoa, E., & Pessoa, A. (2021). A bucket graph-based labeling algorithm with application to vehicle routing. Transportation Science, 55(1), 4–28.

[17] Shi, J., Mao, H., Zhou, Z., & Zheng, L. (2023). Adaptive large neighborhood search algorithm for the unmanned aerial vehicle routing problem with recharging. Applied Soft Computing, 147, 110831.

[18] Shi, Y., Liu, W., & Zhou, Y. (2023). An adaptive large neighborhood search based approach for the vehicle routing problem with zone-based pricing. Engineering Applications of Artificial Intelligence, 124, 106506.

[19] Tadaros, M., Sifaleras, A., & Migdalas, A. (2024). A variable neighborhood search approach for solving a real-world hierarchical multi-echelon vehicle routing problem involving HCT vehicles. Computers & Operations Research, 165, 106594.

[20] Tan, S. Y., & Yeh, W. C. (2021). The vehicle routing problem: State-of-the-art classification and review. Applied Sciences, 11(21), 10295.

[21] Tiwari, K. V., & Sharma, S. K. (2023). An optimization model for vehicle routing problem in last-mile delivery. Expert Systems with Applications, 222, 119789.

[22] Toth, P., & Vigo, D. (Eds.). (2014). Vehicle Routing: Problems, Methods, and Applications (2nd ed.). SIAM.

[23] Voigt, S. (2025). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. European Journal of Operational Research, 322(2), 357–375.

[24] Woller, D., Kozák, V., Kulich, M., & Přeučil, L. (2025). Variable Neighborhood Search for the Electric Vehicle Routing Problem. arXiv preprint.

[25] Lima, S., Santos, R., De Araujo, S. A., & Schimit, P. (2016). Combining genetic algorithm with constructive and refinement heuristics for solving the capacitated vehicle routing problem. In IFIP International Conference on Advances in Production Management Systems (APMS) (pp. 113–121).