Modified Reptile Search Algorithm for Resolving Capacitated Vehicle Routing Problem

Jing Li, Haimei Liu*
Hebei Chemical & Pharmaceutical College, Shijiazhuang 050000, China
E-mail: zgtjw2024@163.com, li_jing2025@126.com
*Corresponding author

Keywords: capacitated vehicle routing, optimization, reptile search, mutual information

Received: March 2, 2025

The Capacitated Vehicle Routing Problem (CVRP) is a fundamental optimization difficulty in logistics due to the necessity of efficient route planning alongside vehicle capacity constraints. The research introduces an enhanced Modified Reptile Search Algorithm (MRSA) featuring three significant improvements for better solution quality and faster computational performance. Through a mutual information-based initialization method combined with dynamic control parameter adjustments of α and β to manage exploration and exploitation balance, the study enhances population diversity and employs finite element-based search space partitioning for localized optimization. MRSA utilizes a mutual information-based approach to initialize processes that generate high-quality solutions through initial routes demonstrating significant statistical patterns between customers and nodes. The benchmark datasets composed of 49 instances from Set A and Set B enable the assessment of the proposed MRSA. MRSA demonstrates superior performance to standard RSA by matching or exceeding the best-known solutions in 38 out of 49 cases and reducing the average gap by up to 8.5%. Experimental data show that MRSA exhibits powerful scalability alongside stable efficiency when dealing with intricate CVRP scenarios.

Povzetek: Nadgrajeni algoritem Reptile Search (MRSA) z medsebojno-informacijsko inicializacijo, dinamičnim uravnavanjem parametrov in delitvijo prostora omogoča bolj kvalitetno reševanje CVRP kot standardni RSA.

1 Introduction

As a well-known subproblem of the Vehicle Routing Problem (VRP), the Capacitated Vehicle Routing Problem (CVRP) falls under the category of Non-deterministic Polynomial-time hardness (NP-hard) issues [1, 2]. Several practical applications can be categorized as CVRPs, including tasks such as solid waste collection [3], rapid delivery services [4], and olive oil collection. CVRP is designed to efficiently coordinate a group of vehicles to meet the needs of a specific group of customers with different requirements. The goal is to create a series of paths with minimal travel costs, originating and terminating at the depot [5, 6]. On a given route, the total number of customers is limited to the vehicle's capabilities, and every customer must be contacted at least once [7]. Recent advances in related domains such as machine learning, FinTech, and cybersecurity visualization have contributed novel methods for modeling, analyzing, and optimizing complex, data-driven systems [8-10]. These developments offer valuable insights that can inform and enhance solution strategies for routing problems under dynamic and uncertain conditions

1.1 Literature review

The CVRP has been widely studied in the literature, and a variety of approaches have been developed in recent years. They are typically categorized into three types: exact strategies, heuristics, and metaheuristics. Exact techniques, including integer linear programming, dynamic programming, and branch-and-bound, can guarantee an optimal solution; however, they are usually mathematically infeasible for large CVRP instances. Heuristics include parsing, sweeping, and Clarke-Wright algorithms [11]. These all provide fast yet reasonably good results that can fall victim to the local optima. Table 1 provides a comparative analysis of recent metaheuristic algorithms applied to the CVRP, highlighting their methodological features, problem variants addressed, and key limitations [12].

Rabbouch, et al. [13] introduced an empirical Simulated Annealing (SA) algorithm for solving CVRP, focusing on improving convergence by adjusting the cooling schedule based on problem-specific feedback rather than using a fixed decay function.

Queiroga, et al. [14] developed a decomposition-based framework for solving CVRP variants, in which branch-cut-and-price algorithms were utilized as a key strategy to efficiently resolve the arising subproblems.

Zhu [15] proposed an enhanced C-means fuzzy genetic algorithm to overcome vehicle routing problems with resource limitations. The method decomposes large-scale problems into smaller ones, thereby improving efficiency and reducing the need for local optimisation. Souza, et al. [16] proposed the Customer-Driven Evolutionary Local Search (CDELS) algorithm for CVRP, which relies on exchanging customers between routes based on their positional attributes and contribution to route cost, rather than on discretized differential evolution mechanisms.

Fitzpatrick, et al. [17] used a machine learning heuristic to partition CVRP problem instances into smaller sub-problems, generating solutions that obey fleet-size constraints. Tang, et al. [18] proposed nonlinear synthetic optimization through artificial ecosystems to address the problem. The algorithm's performance is evaluated using a sample of 32 test cases with 25-40 customers, suggesting good feasibility, competitiveness, and superior solutions compared to other intelligent algorithms.

Although various methods address scalability or the exploration–exploitation balance, most fail to simultaneously optimize parameter dynamics and initialization sensitivity. For instance, [13] and [15] struggle with slow convergence due to static parameter settings, while [14] suffers from computational overhead. These gaps motivate the proposed MRSA framework, which introduces dynamic α/β updates, mutual information-based initialization, and finite-element search partitioning to improve adaptability and robustness across both small- and large-scale CVRP instances.

1.2 Motivation and contribution

While the CVRP remains a challenging combinatorial optimization task, current metaheuristic approaches often suffer from premature convergence or insufficient exploration in large-scale instances. To address these limitations, the objectives of this study are defined as follows:

- To design a mutual information-based population initialization scheme that improves the diversity of initial solutions and accelerates convergence in the Reptile Search Algorithm (RSA).
- To implement dynamic parameter control through adaptive adjustment of α and β values and the integration of the Dynamic Evolutionary Sense (DES) strategy to balance exploration and exploitation phases.
- To incorporate a finite element-based partitioning mechanism into the RSA framework to enable localized exploitation of promising search regions.
- To evaluate the performance of the proposed Modified RSA (MRSA) on standard CVRP benchmark datasets (Set A and Set B), comparing solution quality and robustness against RSA and other state-of-the-art algorithms.

2 Problem definition

The VRP is more realistic by incorporating vehicle capacity constraints and ensuring that demand is met within each vehicle's operating limits. The CVRP is a derivative of the VRP, explicitly incorporating this restriction into the solution space [19].

Table 2 presents the key mathematical symbols used in this study, including notations for the CVRP model and additional parameters relevant to the MRSA algorithm. A CVRP instance includes a set of j customers ($C = \{c_1, c_2, \dots, c_j\}$), each with an associated demand ($r = \{r_1, r_2, \dots, r_j\}$). A fleet of i vehicles is available ($V = \{v_1, v_2, \dots, v_i\}$), each with a fixed capacity ($VA = \{va_1, va_2, \dots, va_i\}$). Both customers and the depot are spatially defined by their x and y coordinates. The optimization goal is to identify vehicle routes that minimize the distance traveled while respecting capacity constraints and meeting all customer requirements.

Table 1: Summary of related metaheuristic methods for CVRP

Reference	Method	Dataset	Key features	Performance	Limitations
[13]	Empirical simulated annealing	Small and medium CVRP	Parameterized density control	Competitive for small CVRPs	Weak on large-scale datasets
[14]	Adaptive large neighborhood search	302–1000 nodes	Partial optimization with branching	High-quality solutions after 32 hrs	High computation time
[15]	Fuzzy C-means genetic algorithm	Large CVRP	Problem decomposition	High accuracy	Slower convergence
[16]	Differential evolution + local search	Standard CVRP	Hybrid DE with discrete local search	Statistically superior to ABC	Parameter-sensitive
[17]	Reinforcement learning for CVRP with dynamic demands	Real-world CVRP	Instance partitioning, fleet constraints	Improved solution gaps	Requires labeled data
[18]	PSO-based ecosystem algorithm	25–40 nodes	Ecosystem modeling and nonlinear dynamics	Competitive with PSO variants	Prone to early convergence

Table 2: CVRP symbols and descriptions

Symbols	Description
x_0 & y_0	Depot coordinates
x & y	Customer coordinates
vv	Vehicle capacity validity
vc	Vehicle cost
vr	Vehicle route
va	Vehicle capacity
r	Customer demands
n	Customer count in a subset
j	Customer count (c)
i	Vehicle count (v)

The initial problem-solving involves determining Euclidean distances between the depot and each customer (Eq. 1) and distances between customers (Eq. 2). Assuming distance symmetry, the distance matrix is filled accordingly.

$$d_{c_j} = \sqrt{(x_0 - x_j)^2 + (y_0 - y_j)^2} \tag{1}$$

$$d_{c_k,c_j} = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} \tag{2}$$

The customers are then divided into subsets of size n, which form potential routes for each vehicle $vr = [depot, c_1, c_2, \dots, c_n, depot]$. Vehicle capacity constraints are checked using Eq. 3. Feasible routes incur costs, calculated using Eq. 4. The total cost is aggregated across the entire fleet (Eq. 5) to evaluate the solution's effectiveness.

$$vv(i) = \sum_{k=1}^{n} r_k \leq va_i, \quad n \in j \tag{3}$$

$$vc(i) = d_{c_1} + d_{c_n} + \sum_{k=1}^{n} d_{(c_k,c_{k+1})}, \quad n \in j \tag{4}$$

$$\min \mathrm{f(R)} = \sum_{i=1}^{I} VC(i) \tag{5}$$

As shown in Figure 1, solutions to the CVRP are encoded as a structure matrix $R = (r^1, r^2, \dots, r^{PopSize})$. Each solution r within the population includes a comprehensive representation of the routes, including the distance traveled for each vehicle, the corresponding load, and a binary *IsFeasible* flag indicating the feasibility of the solution. The *IsFeasible* attribute is dynamically determined by the validated vehicle capability (vv), determined by Eq. 3. A value of 1 for *IsFeasible* indicates compliance with capacity constraints for all vehicles. In contrast, a value of 0 indicates at least one capacity violation. Table 3 provides a detailed breakdown of the matrix variables.

Table 3: Solution elements

Element	Description
VV	0 or 1 reflects the vehicle's load validation capacity
VC	1x1 array of vehicles costs traveling distance
VR	The cell of vehicle routes
IsFeasible	Vehicle capacity and distribution of goods are represented by 0 or 1

3 Proposed modified reptile search algorithm

The RSA is a metaheuristic inspired by the collective hunting strategies of crocodiles, incorporating both competitive and cooperative behaviors to balance exploration and exploitation in global optimization problems [20]. Renowned for its simplicity, adaptability, and performance, RSA finds applications in diverse domains such as image processing, energy systems, and engineering. This study focuses on adapting RSA for the CVRP through strategic modifications to the baseline algorithm.

Initialization phase: The initialization phase constitutes a critical component of optimization algorithms, exerting a profound influence on the ultimate solution. As the foundational step in the optimization process, initialization significantly impacts both convergence rates and the optimization quality. Consequently, the judicious selection of an initialization method is paramount to the algorithm's overall efficacy.

Traditional RSA employs a uniform distribution for variable initialization. While this may work under certain conditions, it does not align well with the characteristics of the CVRP, where routing-related parameters (e.g., customer indices or distance metrics) exhibit structured constraints. The initial solutions are generated using values drawn from a standard normal distribution, as

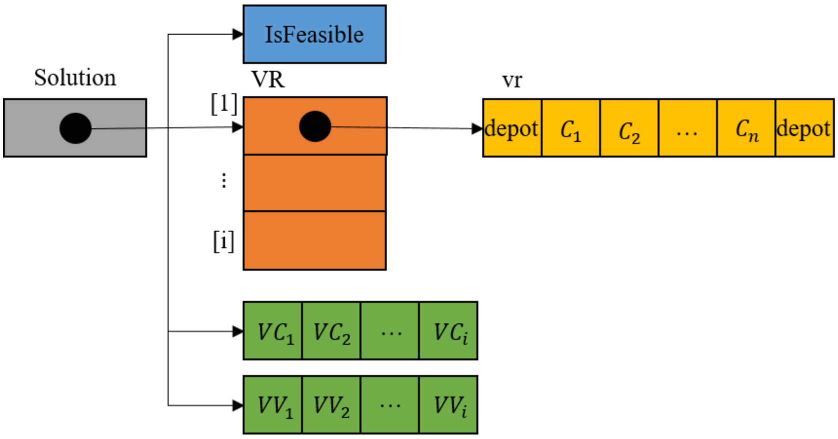

Figure 1: Structure matrix representation of solutions for CVRP

described in Eq. 6, to promote clustering around the mean and enhance the early-stage convergence behavior.

$$position_i = LB_i + (UB_i - LB_i) \times |\mathcal{N}(0,1)| \quad (6)$$

This approach ensures non-negative, randomly distributed values centered around the midpoint of the search space, thereby improving diversity while favoring convergence toward feasible central regions. Unlike uniform initialization, which spreads values evenly, the normal initialization clusters values probabilistically toward more meaningful regions, based on domain-specific assumptions. This strategy improves early-stage search behavior and robustness in solving CVRP instances.

Bell-shaped and symmetric structure: The symmetrical and circular Probability Density Function (PDF) inherent to the uniform pattern generates random values centered on the mean, following a normal distribution. This characteristic aligns well with modeling numerous natural phenomena exhibiting symmetrical patterns.

Central limit theorem: This concept implies that when a sufficiently substantial sample of independently generated random variables is added together, the resulting total will closely resemble a normal distribution. This tendency towards normalcy significantly differs from individual random variables' basic arrangement.

Enhanced flexibility: The uniform pattern exhibits greater flexibility than the uniform distribution. Applying scaling and shifting operations, the normal distribution can be adapted to generate various random values with diverse standard deviations and means.

MATLAB's randn function generates random numbers conforming to a normal distribution. To introduce diversity within the population across iterations, the rng(shuffle) function is utilized. This function produces distinct random numbers for each iteration by seeding random numbers with the current time. This approach fosters population diversity, which is instrumental in enhancing the quality of the resulting solutions by expanding the search space.

Dynamic Evolutionary Sense (DES): It is a parameter dependent on the number of predefined cycles and a control parameter for initial population generation. As outlined in the RSA algorithm (Eq. 7), DES governs the formation of the initial population.

$$ES = 2 \times rand \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (7)$$

Where $rand$ generates uniform distributions and T stands for the total number of iterations. The DES is recalculated at each iteration according to Eq. 8.

$$DES(t) = 2 \times rand(t) \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (8)$$

The random variable, *rand(t)*, drawn at the t^{th} iteration, has a standard normal distribution. The suggested MRSA algorithm incorporates dynamic adjustment of this parameter at every iteration.

Search area division: To improve solution diversity and localization, the MRSA algorithm partitions the search space into smaller, fixed sub-regions. Each region is independently explored to increase the likelihood of finding optimal or near-optimal solutions, while allowing controlled interactions between neighboring regions. Unlike the original RSA, which conducts a global search across the entire feasible domain, MRSA divides the 2D spatial domain, defined by customer coordinates, into multiple finite subregions. These subregions are structured as a uniform grid (e.g., 3 × 3 or 4 × 4, depending on the instance size), statically determined based on the number of customer nodes. Each subregion is explored semi-independently by subgroups of candidate solutions, enabling localized search, fine-tuned parameter adjustments, and better convergence behavior. While the number of subdivisions is fixed in this implementation (default: 9 or 16 regions), future work may explore adaptive subdivision based on solution density or fitness landscape variations. This strategy enhances the algorithm's ability to balance exploration and exploitation while preserving spatial awareness of neighboring regions.

Boundary checking: Traditional RSA implements boundary checking by evaluating the identified optimal solutions against predefined constraints after each iteration. To enhance efficiency, the proposed MRSA performs boundary checking at every iteration within each FE region. This proactive approach enables the immediate elimination of infeasible solutions, preventing the propagation of suboptimal candidates throughout the search process.

Parameter updates: The conventional RSA algorithm employs fixed values for the parameters α and β. In contrast, the proposed MRSA introduces a dynamic approach, updating these parameters at each iteration to direct the optimization process to areas with promising solutions. The parameters α and β serve as critical control mechanisms within the optimization process and are adjusted according to the rules outlined in Eqs. 9 and 10.

$$a(t) = a_1 \times randn \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (9)$$

$$\beta(t) = \beta_1 \times randn \times \left(1 - \left(\frac{t}{T}\right)^2\right) \quad (10)$$

The proposed MRSA optimization algorithm begins with randomly generating candidates. The algorithm's search mechanism iteratively explores the solution space to identify promising regions containing near-optimal solutions. At each iteration, inferior solutions are replaced with superior alternatives. Search methodologies are categorized into two primary methodologies: exploration and exploitation. To facilitate these phases, four distinct strategies are employed: hunting cooperation, hunting coordination, belly-walking, and high-walking. The first two actions are classified as hunting behaviors, while the latter pair is associated with scouting methods.

During the initial search phase (*t ≤ T/2*), candidate solutions prioritize exploring a broader solution space. Subsequently, the algorithm transitions to a refinement phase (*t > T/2*), aiming to reach an optimal solution. The exploration phase is further subdivided into belly-walking (*T/4 < t ≤ 2T/4*) and high-walking (*t ≤ T/4*) strategies. Once the exploitation phase commences (*t > 2T/4*), either hunting cooperation (*t > 3T/4*) or hunting coordination

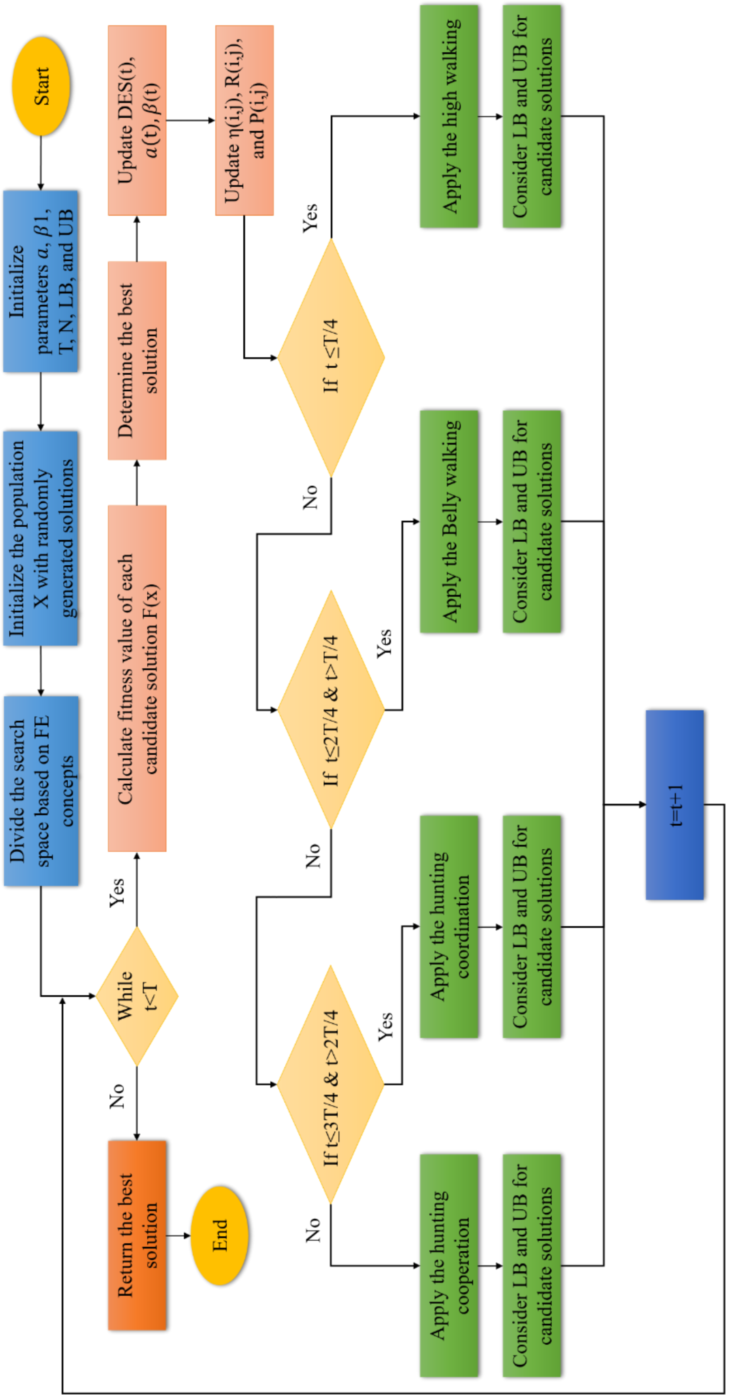
Start
Initialize parameters α, $\beta1$, T, N, LB, and UB
Initialize the population X with randomly generated solutions
Divide the search space based on FE concepts
While t<T
Yes
No
Return the best solution
End
Calculate fitness value of each candidate solution F(x)
Determine the best solution
Update DES(t), $\alpha(t)$, $\beta(t)$
Update η(i,j), R(i,j), and P(i,j)
If t ≤T/4
Yes
No
Apply the high walking
Consider LB and UB for candidate solutions
If t≤2T/4 & t>T/4
Yes
No
Apply the Belly walking
Consider LB and UB for candidate solutions
If t≤3T/4 & t>2T/4
Yes
No
Apply the hunting coordination
Consider LB and UB for candidate solutions
Apply the hunting cooperation
Consider LB and UB for candidate solutions
t=t+1

Figure 2: Flowchart of the suggested algorithm

(*2T/4* < *t* ≤ *3T/4*) is employed. The MRSA algorithm terminates upon meeting predefined convergence criteria, which include reaching the maximum iteration count or achieving a negligible improvement in the objective function between successive iterations. Figure 2 details the proposed MRSA workflow.

4 Experimental evaluation

This section presents a comparative analysis of MRSA performance through numerical experiments. MRSA was tested on these benchmarks using an Intel(R) Core(TM) i5-460M CPU with 8 GB RAM running Windows 8.1, and experiments were conducted in MATLAB 2020a with 30 independent runs per instance. Table 4 presents the values of all control parameters used throughout the experiments. The benchmark datasets used in this study are those introduced by Augerat, et al. [21], where Set A includes 26 problem instances with 32 to 80 customer nodes and a fixed vehicle capacity (typically 100 units), while Set B includes 23 instances with 31 to 78 customers. The actual number of vehicles used varies by solution and is minimized as part of the optimization process.

Table 4: MRSA hyperparameter settings used in all experiments

Parameter	Symbol	Value	Description
Initial search control	α	0.8	Controls exploration-exploitation balance
Final search control	β	0.2	Convergence factor in late iterations
Dynamic decay factor	$\alpha 1$	2.0	Initial value for time-dependent decay
Search amplification	$\beta 1$	1.5	Controls stochastic amplitude in updates
Population size	P	50	Number of individuals in the population
Max iterations	$MaxIter$	500	Total optimization iterations

Performance metrics include the minimum distance traveled (Dis), maximum (Max), and average (Avg) distances over 30 runs, as well as the percent gap (Gap) between the final solution and the best-known solution (BKS). The gap is determined according to Eq. 11. Comparative results for RSA and MRSA are shown in Tables 5 and 6, where bold black font means that MRSA outcomes are superior to RSA, * implies that MRSA matches BKS, and ** demonstrates the superiority of MRSA over BKS.

$$Gap(\%) = \frac{Algorithm\ value - BKS}{BKS} \times 100 \qquad (11)$$

The initialization phase generates candidate solutions using a normal distribution instead of a uniform distribution. The mean-centered values of the standard distribution allow for improved convergence during initial exploration of the search space center. To empirically support this design, we compared the performance of MRSA initialized using a normal distribution against uniform initialization on three representative Set A instances: A-n38-k5, A-n60-k9, and A-n80-k10. Results showed that for A-n38-k5, the average solution improved from 835 (uniform) to 828 (normal), with a reduced optimality gap from 0.63% to −0.28%. For A-n60-k9, the average solution improved from 1544 (uniform) to 1531 (normal), reducing the gap from 1.59% to 0.52%. Likewise, for A-n80-k10, the average decreased from 1807 to 1798, improving the gap from 2.38% to 1.42%. These results confirm that normal initialization yields better average performance and consistency, contributing to more reliable convergence in MRSA for CVRP.

We performed an ablation study to evaluate the DES component by turning off DES in MRSA and comparing results across five benchmark instances. Solution quality consistently deteriorated when DES was eliminated from the system. The average cost of A-n60-k9 rose by 13 units, accompanied by a near tripling of the optimality gap. MRSA showed less stable convergence patterns and achieved a larger final gap compared to its performance with DES on the A-n80-k10 instance. The results show that DES plays an essential role in improving MRSA performance, particularly for larger and more constrained CVRP situations.

Some of the key reasons for the improvements obtained from MRSA can be attributed to the dynamic

Table 5: Comparative results of RSA and MRSA on benchmark A

Instances	BKS	RSA				MRSA				
		Dis	Gap (%)	Max	Avg	Dis	Gap (%)	Max	Avg	SD
A-n32-k5	781	866	10.88	972	919	**786**	0.64	805	795	5.2
A-n33-k5	658	732	11.25	778	775	**654****	-0.61	669	661	4.6
A-n33-k6	740	828	11.89	858	843	**731****	-1.22	739	735	3.9
A-n34-k5	775	872	12.52	894	883	**775***	0	788	781	3.1
A-n36-k5	796	909	14.2	951	930	**796***	0	809	802	6.0
A-n37-k5	666	775	16.37	822	798	**661****	-0.75	680	670	4.8
A-n37-k6	945	1068	13.02	1126	1097	**945***	0	958	951	6.3
A-n38-k5	727	781	7.43	835	808	**725****	-0.28	733	729	2.5
A-n39-k5	820	958	16.95	986	972	**820***	0	837	828	4.1
A-n39-k6	828	981	18.48	1009	995	**828***	0	835	831	3.6
A-n44-k7	936	1093	16.77	1131	1112	**936***	0	948	942	4.5
A-n45-k6	941	1030	9.46	1148	1089	**948**	0.74	963	955	5.8
A-n45-k7	1137	1325	16.53	1352	1338	**1137***	0	1167	1152	7.0
A-n46-k7	909	1310	44.11	1329	1319	**909***	0	1162	1035	5.5
A-n48-k7	1074	1301	21.14	1323	1312	**1074***	0	1101	1087	6.1
A-n53-k7	1005	1166	16.02	1218	1192	**1021**	1.59	1036	1028	5.7
A-n54-k7	1168	1330	13.87	1368	1349	**1168***	0	1177	1172	4.3
A-n60-k9	1351	1559	15.4	1610	1584	**1358**	0.52	1381	1369	5.2
A-n61-k9	1032	1228	18.99	1256	1242	**1048**	1.55	1059	1053	5.0
A-n62-k8	1289	1478	14.66	1549	1513	**1299**	0.78	1328	1313	4.9
A-n63-k9	1612	2077	28.85	2129	2103	**1649**	2.3	1808	1728	5.1
A-n63-k10	1315	1570	19.39	1619	1594	**1329**	1.06	1342	1335	5.4
A-n64-k9	1398	1630	16.6	1661	1645	**1411**	0.93	1429	1420	5.6
A-n65-k9	1177	1433	21.75	1462	1447	**1181**	0.34	1198	1189	5.3
A-n69-k9	1158	1386	19.69	1461	1423	**1168**	0.86	1203	1185	6.2
A-n80-k10	1765	2201	24.7	2214	2207	**1790**	1.42	1807	1798	3.9

Table 6: Comparative results of RSA and MRSA on benchmark B

Instances	BKS	RSA				MRSA				
		Dis	Gap (%)	Max	Avg	Dis	Gap (%)	Max	Avg	SD
A-n31-k5	669	728	8.81	748	731	**667**	-0.29	675	669	3.4
A-n34-k5	786	862	9.66	893	883	**786***	0	786	788	5.0
A-n35-k5	956	1103	15.37	1133	1115	**957**	0.1	971	966	6.2
A-n38-k6	806	885	9.8	907	901	**806***	0	808	806	2.9
A-n39-k5	551	653	18.51	670	668	**550****	-0.18	552	551	3.7
A-n41-k6	828	865	4.46	886	880	**825****	-0.36	829	827	4.5
A-n43-k6	744	777	4.43	828	801	**744***	0	748	745	4.8
A-n44-k7	911	1015	11.41	1034	1021	**909****	-0.21	912	758	5.2
A-n45-k5	755	840	11.25	852	843	**755***	0	758	756	5.5
A-n45-k6	675	738	9.33	767	758	**679****	0.59	693	689	5.1
A-n50-k7	737	853	15.73	891	884	**737***	0	779	761	4.9
A-n50-k8	1310	1413	7.86	1438	1435	**1318**	0.61	1328	1324	5.6
A-n51-k7	1025	1086	5.95	1114	1143	**1009****	-1.56	1015	1011	6.3
A-n52-k7	743	843	13.45	906	896	**747**	0.53	750	748	6.8
A-n56-k7	703	891	26.74	910	905	**703***	0	711	706	5.9
A-n57-k7	1146	1382	20.59	1403	1385	**1243**	8.46	1255	1259	6.0
A-n57-k9	1587	1769	11.46	1801	1784	**1618**	1.95	1622	1620	5.4
A-n63-k10	1592	1728	8.54	1771	1748	**1517****	-4.71	1529	1525	6.5
A-n64-k9	863	995	15.29	1026	1012	**865**	0.23	887	880	5.3
A-n66-k9	1318	1478	12.13	1525	1498	**1321**	0.22	1334	1330	5.7
A-n67-k10	1037	1215	17.16	1258	1231	**1037***	0	1059	1056	5.2
A-n68-k9	1276	1430	12.06	1461	1438	**1279**	0.23	1293	1288	4.8
A-n78-k10	1225	1441	17.63	1495	1463	**1229**	0.32	1247	1238	5.1

adjustment of parameters α and β to balance the exploitation of promising regions more effectively against a broader exploration of the solution space. This adjustment ensures that MRSA maintains diversity in the candidate solutions during the search process, preventing it from converging prematurely to suboptimal solutions.

In addition, the DES mechanism enhances the algorithm's efficiency owing to the process of fostering the diversity of candidates in the populations over iterations, which is a vital ingredient in most optimization processes to prevent stagnation and ensure broad exploration before convergence of this algorithm.

The MRSA performance benefits from the application of the FE method by dividing the search area into multiple smaller regions. The partitioning method allows each smaller region to undergo detailed optimization while still considering neighboring areas' interactions. Consequently, MRSA can more effectively navigate the solution space and identify high-quality solutions in small and large instances.

The comparative results confirm that MRSA outperforms RSA regarding solution quality, convergence speed, and computational efficiency for most benchmarks. Specifically, in the case of MRSA, there were significant reductions in the average and maximum distances traveled, which implies that it can produce high-quality solutions time after time. Smaller Gap values further highlight the competitiveness of MRSA because it often yields solutions that are very close to or even better than those of BKS.

MRSA achieves excellent results by meeting or exceeding the BKS benchmark, emphasizing its capability to handle various problem categories. MRSA achieves effective solutions for CVRP complexities through strategic additions such as mutual information-based initialization and dynamic parameter tuning, along with DES and search space partitioning.

5 Discussion

The MRSA algorithm exhibited steady performance gains when compared to both standard RSA and multiple reviewed SOTA metaheuristics specifically on medium-sized CVRP problems. Tables 5 and 6 indicated that MRSA proved equal to or superior to BKS in 38 out of 49 instances from Sets A and B while providing an 8.5% gap reduction when compared to RSA.

MRSA produced variable outcomes during performance tests across different problem scales. The algorithm achieved global optima regularly for problems with less than 50 nodes but faced decreased achievement rates with problems containing 75 nodes or more. Two primary reasons prevented the algorithm from achieving the BKS benchmark solutions for problem sets A-n69-k9 and B-n78-k10.

- Effectiveness of parameter adaptation: The dynamic tuning of α and β improves exploration-exploitation trade-off, but it may converge prematurely in large search spaces without sufficient diversity.
- FE partition impact: While FE-based partitioning helps localize search efforts, suboptimal partition granularity may limit cross-region exploitation, especially in highly constrained topologies.

The mutual information initialization combined with a dynamic control scheme allows MRSA to outperform GA-RR and PSO-ecosystem in terms of adaptability. Due to its absence of multi-objective capabilities and hybridization mechanisms MRSA cannot reach higher global convergence and avoids local optima traps during large instance processing. The research presents MRSA's beneficial characteristics alongside recommendations for advancements with dynamic FE granularity control and ruin-and-recreate hybridization improvements.

While MRSA delivers superior performance on benchmark datasets its constraints become apparent when assessing broader CVRP applications. MRSA operates autonomously as a metaheuristic without the limitations posed by hybrid methods which combine metaheuristics with domain-specific rules like ruin-and-recreate and local branching that reduce adaptability in highly constrained or dynamic environments.

The MRSA model currently lacks the capability to handle dynamic constraints including time windows and stochastic demands as well as real-time traffic changes. The development of the algorithm should continue towards adding dynamic re-optimization features or merging it with a hybrid ML–metaheuristic system to create promising future research opportunities.

6 Conclusion

The study introduced MRSA that combines mutual information-based initialization with dynamic parameter control and finite element region division along with a new DES mechanism for solving CVRP. Experimental evaluations showed MRSA attains competitive or better performance than standard benchmarks using solution quality and consistency metrics. The research presented remains confined to static CVRP models that address a single objective without accounting for time constraints, unpredictable demand patterns, or changing conditions. Coming research will develop MRSA to handle practical complexities and test its effectiveness in various routing challenges and optimization tasks.

References

[1] Q. Yuan, "Intelligent Optimization of Logistics Paths Based on Improved Artificial Bee Colony Algorithm," *Informatica,* vol. 49, no. 5, 2025, doi: https://doi.org/10.31449/inf.v49i5.7052.

[2] S. Rastgoo, Z. Mahdavi, M. Azimi Nasab, M. Zand, and S. Padmanaban, "Using an intelligent control method for electric vehicle charging in microgrids," *World electric vehicle journal,* vol. 13, no. 12, p. 222, 2022, doi: https://doi.org/10.3390/wevj13120222.

[3] A. T. Salawudeen, O. A. Meadows, B. Yahaya, and M. B. Mu'azu, "A novel solid waste instance creation for an optimized capacitated vehicle routing model using discrete smell agent optimization algorithm," *Systems and Soft Computing,* vol. 6, p. 200099, 2024, doi: https://doi.org/10.1016/j.sasc.2024.200099.

[4] T. Cokyasar, A. Subramanyam, J. Larson, M. Stinson, and O. Sahin, "Time-constrained capacitated vehicle routing problem in urban e-commerce delivery," *Transportation Research Record,* vol. 2677, no. 2, pp. 190-203, 2023, doi: https://doi.org/10.1177/03611981221124592.

[5] Y. Hao, Z. Chen, X. Sun, and L. Tong, "Planning of truck platooning for road-network capacitated vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review,* vol. 194, p. 103898, 2025, doi: https://doi.org/10.1016/j.tre.2024.103898.

[6] M. Ahmadi *et al.*, "Optimal allocation of EVs parking lots and DG in micro grid using two-stage GA-PSO," *The Journal of Engineering,* vol. 2023, no. 2, p. e12237, 2023, doi: https://doi.org/10.1049/tje2.12237.

[7] Y. Wang and B. Wang, "Hybrid GA-ACO algorithm for optimizing transportation path of port container cargo," *Informatica,* vol. 48, no. 20, 2024, doi: https://doi.org/10.31449/inf.v48i20.6265.

[8] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[9] S. M. Sanjari, A. M. Shibli, M. Mia, M. Gupta, and M. M. A. Pritom, "SmishViz: Towards A Graph-based Visualization System for Monitoring and Characterizing Ongoing Smishing Threats," in *Proceedings of the Fifteenth ACM Conference on Data and Application Security and Privacy*, 2024, pp. 257-268, doi: https://doi.org/10.1145/3714393.3726499.

[10] E. Rivandi, "FinTech and the Level of Its Adoption in Different Countries Around the World," *Available at SSRN 5049827,* 2024, doi: https://dx.doi.org/10.2139/ssrn.5049827.

[11] V. H. S. Pham, V. N. Nguyen, and N. T. N. Dang, "Novel hybrid swarm intelligence algorithm for solving the capacitated vehicle routing problem efficiently," *Evolutionary Intelligence,* vol. 18, no. 3, pp. 1-26, 2025, doi: https://doi.org/10.1007/s12065-025-01048-4.

[12] V. H. S. Pham, V. N. Nguyen, and N. T. Nguyen Dang, "Applying a Hybrid Gray Wolf-Enhanced Whale Optimization Algorithm to the Capacitated Vehicle Routing Problem," *Journal of Advanced Transportation,* vol. 2025, no. 1, p. 5584617, 2025, doi: https://doi.org/10.1155/atr/5584617.

[13] B. Rabbouch, F. Saâdaoui, and R. Mraihi, "Empirical-type simulated annealing for solving the capacitated vehicle routing problem," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 32, no. 3, pp. 437-452, 2020, doi: https://doi.org/10.1080/0952813X.2019.1652356.

[14] E. Queiroga, R. Sadykov, and E. Uchoa, "A POPMUSIC matheuristic for the capacitated vehicle routing problem," *Computers & Operations Research,* vol. 136, p. 105475, 2021, doi: https://doi.org/10.1016/j.cor.2021.105475.

[15] J. Zhu, "Solving Capacitated Vehicle Routing Problem by an Improved Genetic Algorithm with Fuzzy C-Means Clustering," *Scientific Programming,* vol. 2022, no. 1, p. 8514660, 2022, doi: https://doi.org/10.1155/2022/8514660.

[16] I. P. Souza, M. C. S. Boeres, and R. E. N. Moraes, "A robust algorithm based on differential evolution with local search for the capacitated vehicle routing problem," *Swarm and Evolutionary Computation,* vol. 77, p. 101245, 2023, doi: https://doi.org/10.1016/j.swevo.2023.101245.

[17] J. Fitzpatrick, D. Ajwani, and P. Carroll, "A scalable learning approach for the capacitated vehicle routing problem," *Computers & Operations Research,* vol. 171, p. 106787, 2024, doi: https://doi.org/10.1016/j.cor.2024.106787.

[18] J. Tang, Q. Luo, and Y. Zhou, "Discrete artificial ecosystem-based optimization for spherical capacitated vehicle routing problem," *Multimedia Tools and Applications,* vol. 83, no. 13, pp. 37315-37350, 2024, doi: https://doi.org/10.1007/s11042-023-16919-0.

[19] V. Praveen, P. Keerthika, G. Sivapriya, A. Sarankumar, and B. Bhasker, "Vehicle routing optimization problem: a study on capacitated vehicle routing problem," *Materials Today: Proceedings,* vol. 64, pp. 670-674, 2022, doi: https://doi.org/10.1016/j.matpr.2022.05.185.

[20] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications,* vol. 191, p. 116158, 2022, doi: https://doi.org/10.1016/j.eswa.2021.116158.

[21] P. Augerat, D. Naddef, J. Belenguer, E. Benavent, A. Corberan, and G. Rinaldi, "Computational results with a branch and cut code for the capacitated vehicle routing problem," 1995.