International Journal of Natural and Engineering Sciences
E-ISSN: 2146-0086 19(2): 67-80, 2025
Research Article

Proposing a Method for Vehicle Routing with Capacity Constraints Using a Modified Cuckoo Optimization Algorithm

Mahmoud Foroutannaddafi[1*]

[1] *National University of Skills, Department of Civil Engineering, Faculty of Engineering, Tehran, Iran*

*Corresponding author: mforoutannaddafi@tvu.ac.ir

(Received: 22 February 2025 / Accepted: 4 July 2025)

Abstract

In the vehicle routing problem, a fleet of multiple vehicles departs from one or more depots to serve customers located at different geographical points, aiming to minimize the costs associated with this task. This study proposes a novel approach to solve the routing problem. Considering that numerous paths can be envisioned from the origin to the final destination, an intelligent search algorithm is required. For this purpose, an optimization algorithm is employed in this paper, specifically an modified cuckoo optimization algorithm (MCOA). This algorithm is inspired by the lifestyle of a bird called the cuckoo and is capable of solving high-dimensional problems with good convergence speed. Similar to many evolutionary algorithms, the search process begins with an initial population of cuckoos, gradually moving towards the optimal point. In the improved version of the cuckoo algorithm, the egg-laying radius gradually decreases. Simulation results demonstrate the high efficiency of the proposed method compared to existing approaches.

Keywords: Vehicle Routing, Customer Service, Cost Function, Modified Cuckoo Algorithm

Introduction

The vehicle routing problem is a well-known concept in the field of operations research, which has seen significant progress over the past two decades due to extensive efforts [1-2]. This problem involves a set of scenarios where a fleet of multiple vehicles departs from one or more depots to serve customers located at various geographical points, aiming to minimize the associated costs. In these routes, each customer is visited exactly once, and all their demands are fulfilled by a single vehicle. Each vehicle has a limited capacity, and all routes start from a specific point (loading depot). After visiting a sequence of customers, the vehicle returns to the same starting point, where the route ends. These problems are generally known as Vehicle Routing Problems (VRP) or Transportation Planning Problems [3-5]. The Capacitated Vehicle Routing Problem (CVRP) is a common type of routing problem where each vehicle has a limited capacity for carrying goods or passengers. This problem has broad applications across various industries, including:

1. Goods Distribution and Delivery: In supply chains, for delivering goods from warehouses to customers while respecting vehicle capacity and minimizing costs.

2. Waste Collection and Recycling: For optimizing the routes of waste collection or recycling trucks while considering their capacity constraints.

3. Food and Pharmaceutical Distribution: In the food and pharmaceutical industries, where goods need to be transported under specific conditions (e.g., refrigerated), vehicle capacity must be carefully managed.

4. Bus Routing and Public Transportation Systems: To optimize bus routes while considering passenger capacity and travel time.

5. Paratransit Systems: For planning routes of vehicles that transport people with special needs, considering capacity limits and special equipment.

6. Traveling Salesman with Capacity (TSP with Capacity): In sales and marketing, where a vehicle needs to serve a large number of customers with limited capacity.

7. Fuel and Hazardous Materials Distribution: For optimizing the transportation of fuel, chemicals, or other hazardous materials, which have strict capacity and safety requirements.

8. Maintenance and Repair Units: For dispatching repair teams to multiple locations with limited equipment and spare parts in the vehicle.

Specifically, solving the vehicle routing problem determines a set of paths, each executed by a specific vehicle, starting and ending at its respective center. The goal is to meet customer demands, satisfy operational constraints, and minimize overall transportation costs. The network of routes is used for goods transportation and is typically represented as a graph, where the arcs represent the routes. These arcs are divided into direct and indirect categories, based on whether the route is one-way or two-way. Each arc is associated with a cost, usually determined by the path's length or the time it takes to traverse it, which may also depend on the type of vehicle or the time period during which the route is traveled [6-9].

Evolutionary algorithms, such as genetic algorithms, ant colony optimization, and particle swarm optimization, are highly significant in solving the vehicle routing problem with capacity constraints due to their ability to handle complex, multi-objective problems. These algorithms are especially effective in problems with many variables and constraints, where the search space is large. One of the primary advantages of evolutionary algorithms in the vehicle routing problem is their ability to find optimal or near-optimal solutions in a reasonable amount of time. These algorithms, by mimicking natural evolutionary processes like natural selection and mutation, continuously generate different populations of solutions and guide them toward the most optimal outcomes. In the vehicle routing problem with capacity constraints, considering various limitations such as vehicle capacity and customer demands is essential. Evolutionary algorithms are well-suited to handle these constraints as they can simultaneously address multiple objectives and constraints during the search process. These features make them highly suitable for solving complex problems like vehicle routing with capacity constraints. Furthermore, evolutionary algorithms can effectively operate in dynamic and changing environments. For example, in some vehicle routing problems, environmental conditions and customer requirements may constantly change. Evolutionary algorithms can easily adapt to these changes and find optimal solutions for new conditions.

This paper is structured as follows: In Section 2, a review of existing methods related to the research topic is provided. Section 3 presents a detailed description of the method proposed by the authors, including the models, algorithms, and implementation steps. In Section 4, the simulation results used to evaluate the proposed method are discussed. Finally, Section 5 concludes the paper by summarizing the findings and their significance.

Existing Methods

The vehicle routing problem is often considered a complex challenge for researchers, particularly due to its multi-objective nature and various constraints. It is typically difficult to solve. In reference [1], an improved biological metaheuristic method is proposed to solve the vehicle routing problem with capacity constraints. The main focus of this paper is to find the route with the lowest cost for the vehicle. The distance traveled by the vehicle is determined using an optimization algorithm. The performance of the proposed method is compared with the firefly algorithm, and the results show that the proposed method performs better. In reference [2], a two-stage iterative local search method is proposed to solve the vehicle routing problem with capacity constraints. This method examines the best vehicle path strategy for transporting goods, considering various constraints such as the type of vehicle. Among different metaheuristic methods, iterative local search is a commonly used approach to solve complex problems. The simulation results indicate that the proposed method can effectively solve most of the test cases.

In reference [3], the Fisher and G. Kumar algorithm is introduced to solve the vehicle routing problem. In this proposed method, clusters are formed in such a way that each cluster contains an equal number of nodes. The formation of initial clusters is related to the path optimization problem. Three prominent metaheuristic methods, including genetic algorithms, ant colony optimization, and particle swarm optimization, are considered for path optimization. Simulation results show that the Fisher and G. Kumar algorithm, combined with particle swarm optimization, is an efficient method for solving the vehicle routing problem with capacity constraints. Most of the existing methods for solving the vehicle routing problem with capacity constraints are capable of solving problems with fewer than one hundred nodes, though this process is time-consuming. In reference [4], a new two-stage metaheuristic method is proposed. In the first stage, a set of possible clusters is identified, and in the second stage, the clusters are assigned to vehicles and then to each tour. The ant colony algorithm is used for sorting the clusters.

In reference [5], the artificial bee colony algorithm is used to solve the vehicle routing problem with capacity constraints. The performance of the proposed method is evaluated using several test functions to assess the minimum total distance traveled. Then, its performance is compared with existing methods. Simulation results show that this approach can be effective in transportation problems, especially in minimizing the total distance, transportation costs, and improving service quality. In reference [6], a modified partition clustering method is proposed. The main goal of this method is to divide the entire area into smaller clusters, considering the demand of each vehicle node during the cluster formation. Each cluster is then treated as a separate vehicle routing problem, solved using the ant colony optimization algorithm. Simulation results indicate that the proposed method performs better than Kamins' partition clustering method.

In reference [7], an improved Differential Evolution algorithm with local search is proposed to solve the vehicle routing problem with capacity constraints. The proposed method utilizes three different mutation operators to enhance the algorithm's convergence. Local search can also play a role in improving the solution quality. The performance of the proposed method and existing methods are compared using several test functions. The Ant Colony Optimization algorithm has recently been applied to combinatorial optimization problems. In reference [8], the Ant Colony Optimization algorithm is employed to solve the vehicle routing problem with random demands. To enhance the solution quality, a two-option local search is incorporated into the Ant Colony Optimization algorithm. Simulation results indicate the high efficiency of the proposed method.

In reference [9], a combined genetic algorithm and ant colony optimization is proposed to solve the vehicle routing problem with capacity constraints. The advantages of the genetic algorithm and ant colony optimization are combined with two local search methods, namely the Prim algorithm and a two-option method. The performance of the proposed method is evaluated using eight test functions. The results show that the combined genetic and ant colony optimization algorithm outperforms existing methods. In reference [10], an optimal routing method with minimum cost is proposed, taking into account routing constraints. Traditional methods are not suitable for large-scale problems, and therefore, some researchers have turned to metaheuristic algorithms, which are well-suited for global search in large search spaces. This paper explores various metaheuristic algorithms, including particle swarm optimization, firefly, and cuckoo, and examines the key challenges in solving the vehicle routing problem with capacity constraints.

Proposed Method

Evolutionary algorithms are nature-inspired optimization methods that explore the solution space using heuristic approaches. In complex multi-objective optimization problems, due to the difficulty of finding an optimal solution, population-based evolutionary algorithms are commonly employed. Some well-known methods include genetic algorithm (GA), ant colony optimization, and particle swarm optimization (PSO). Since this study is based on the Cuckoo Search Algorithm, we first introduce this algorithm and then examine its improved version. Finally, the proposed application of the improved Cuckoo Search Algorithm in vehicle routing is presented.

The cuckoo optimization algorithm (COA) is a novel intelligent global search method inspired by the behavior of a bird called the cuckoo [11]. Similar to other evolutionary algorithms, COA begins with an initial population of cuckoos. These cuckoos lay a certain number of eggs in the nests of host birds. Some of these eggs, which more closely resemble the host bird's eggs, have a higher chance of survival and developing into adult cuckoos, while the remaining eggs are identified and eliminated by the host bird. Consequently, the number of successfully hatched eggs indicates the suitability of nests in that region. The greater the number of eggs that hatch and grow in a particular area, the higher the benefit (preference) assigned to that region. Therefore, the location where the highest number of eggs survive will be the parameter that COA aims to optimize. Cuckoos search for the best region to maximize the survival of their eggs. Each cuckoo randomly lays eggs in the nests of host birds within its Egg Laying Radius (ELR). Figure 1 illustrates an example of the egg-laying radius. The ELR relationship for each pigeon is given by equation 1.

$$ELR = \alpha \times \frac{\text{Number of current cuckoo's eggs}}{\text{Total number of eggs}} \times (var_h - var_l) \tag{1}$$

α is used to control the maximum value of ELR, and var_i, var_h represent the lower and upper bounds of the problem variables, respectively.

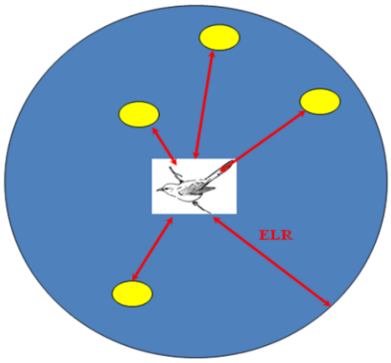

Figure 1. *Egg-laying radius illustration [11].*

After all the cuckoos have laid their eggs, some eggs that are less similar to the host bird's eggs are identified and removed from the nest. Therefore, after each egg-laying phase, a percentage of the eggs with the lowest fitness value are destroyed. The remaining eggs continue to be fed and grow in the host nests. Another interesting feature of the cuckoo chicks is that only one egg can grow in each nest. This is because when cuckoo chicks hatch, they throw out the eggs of the host bird. If the host bird's chicks have hatched earlier, the cuckoo chick, which is three times larger, consumes the most food brought by the host bird and pushes the other chicks aside. As a result, after a few days, the host bird's chicks die from starvation, and only the cuckoo chick survives. Once the cuckoo chicks have grown and matured, they live for a while in their environment and groups. However, as the breeding season approaches, they migrate to better habitats where the chances of egg survival are higher.

After the cuckoo groups are formed in different regions of the general habitat (the problem search space), the group with the best position is chosen as the target for the migration of other cuckoos. When adult cuckoos live in various parts of the environment, identifying which group each cuckoo belongs to becomes a challenging task. To address this issue, cuckoo grouping is performed using the k-means method. The k-means method is a clustering algorithm used to group data into different clusters. Here are the steps for clustering cuckoos using this method:

1. Select the number of clusters (k): First, the number of clusters or groups that the cuckoos should be assigned to must be specified. This value, k, is determined based on the data and the objective of the clustering.

2. Randomly select k initial centroids: k points are randomly selected from the data (in this case, the cuckoos) as the initial cluster centroids. These initial centroids serve as the starting centers for the clusters.

3. Assign data points to the nearest centroid: In this step, each cuckoo is assigned to the group whose centroid is the closest in distance. The distance is usually calculated using Euclidean distance.

4. Calculate new centroids for each group: After assigning the cuckoos to groups, a new centroid for each group is calculated. This new centroid is the mean position of all the cuckoos in that group.

5. Repeat steps until convergence: Steps 3 and 4 are repeated iteratively until the centroids do not change or change very little, indicating that the algorithm has converged.

6. End of the algorithm: When the clusters have stabilized and no significant changes in the centroids occur, the algorithm ends, and the final clustering of the cuckoos is obtained.

This method is well-suited for clustering data and simulating cuckoo migration into different groups. After the cuckoos are clustered, the group with the highest average fitness value is selected as the target group, and the other groups migrate towards it. During migration to the target point, the cuckoos do not travel the entire path to the target location. Instead, they only cover part of the path and have some deviation along the way. This movement is shown in Figure 2.

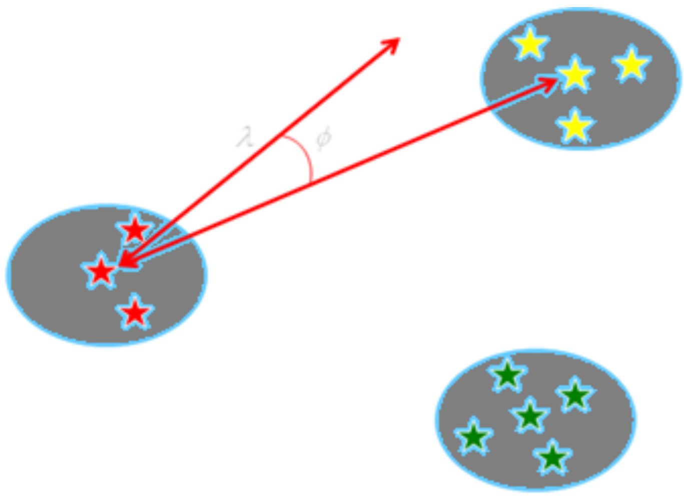

Figure 2. *Cuckoo movement pattern [11].*

As shown in Figure 2, each cuckoo only travels λ% of the total path towards the ideal target and has a deviation of φ radians. These two parameters help the cuckoos explore a larger environment. λ is a random number between 0 and 1, and φ is a number between -π/6 and π/6. When all the cuckoos have migrated towards the target and their new resting points are determined, each cuckoo lays a number of eggs, and based on the number of eggs, an ELR is assigned to it. The egg-laying process then begins. Finally, it should be noted that a number like Nmax controls and limits the maximum number of cuckoos that can live in a given environment.

The main steps of the cuckoo algorithm are as follows:

1-Determine the initial random positions for each cuckoo

2-Assign a specific number of eggs to each cuckoo

3- Define a specific radius for egg-laying for each cuckoo

4- Cuckoos lay eggs within the designated egg-laying radius

5- Host birds identify and remove some of the eggs

6- The eggs grow and hatch into chicks

7- Evaluate the dwelling locations of the newly hatched chicks using the cost function

8- Limit the number of cuckoos according to the maximum permissible number of cuckoos that can inhabit an environment

9- Cluster the cuckoos using the fuzzy c-means method and identify the group with the best dwelling location

10- Cuckoos migrate towards the group with the best dwelling location

11- If the stopping condition is met, the algorithm terminates; otherwise, it returns to step 2.

In [12], a modified version of the cuckoo algorithm (MCOA) is presented. The main objective of this method is the gradual reduction of the ELR. In the standard cuckoo optimization algorithm, this change can be achieved by gradually decreasing the coefficient in equation (1). The proposed method has been implemented on several engineering problems, and the results indicate that it outperforms the original cuckoo algorithm in terms of efficiency.

In the proposed MCOA, specific alterations were made to enhance the balance between exploration (searching new areas of the solution space) and exploitation (refining the current best solutions), which is particularly crucial in complex, high-dimensional problems such as the Vehicle Routing Problem (VRP). Key modifications include:

1. Dynamic Egg-Laying Radius (ELR):
2. Unlike the standard COA where the egg-laying radius may remain fixed or follow a simplistic rule, in the MCOA, the ELR gradually decreases over time. This adaptive mechanism allows the algorithm to explore widely in early stages (high diversity) and converge finely in later stages (local intensification). Such a mechanism enhances the algorithm's capacity to avoid premature convergence and improves the solution's quality.
3. Problem-Specific Encoding and Decoding:
4. For vehicle routing, the individuals (i.e., cuckoos) are encoded as feasible route sequences, ensuring that the solutions remain valid throughout the search. This significantly reduces the search space and focuses computation on promising regions.
5. Fitness-Guided Replacement Strategy:
6. Instead of randomly replacing host nests with new solutions, the proposed MCOA uses a fitness-guided elitist selection, preserving the best routes found while still allowing diversity in less-fit individuals.

These modifications collectively improve the convergence speed, solution accuracy, and robustness of the algorithm. Particularly for VRP, where the solution space is large and highly combinatorial, the gradual shift from exploration to exploitation ensures that the MCOA does not get trapped in local minima and can identify high-quality routes effectively.

As the problem size increases—in terms of the number of customers, vehicles, and constraints—MCOA continues to perform efficiently. This is due to its dynamic adjustment mechanism, where the ELR decreases gradually, enabling a natural balance between exploration (global search) and exploitation (local refinement). Early in the search, a wider ELR promotes exploration, while in later stages, a narrower ELR encourages convergence.

While MCOA does not mathematically guarantee convergence to the global optimum, several mechanisms are included to avoid local optima:

- Periodic randomization of part of the population to maintain diversity.
- Fitness-based elitism to retain high-quality solutions across generations.

These strategies enhance the robustness and reliability of the algorithm in solving high-dimensional, complex optimization problems like VRP.

Simulation Results

The proposed algorithm and the existing method have been implemented using MATLAB software from MathWorks. The processor used is an Intel(R) core^{TM} i5-4460 CPU @ 3.2 GHz with 16 GB of RAM. The parameters of the improved cuckoo algorithm are as follows:

- The initial number of cuckoos is 10.
- The minimum number of eggs for each cuckoo is 5.
- The maximum number of eggs for each cuckoo is 8.
- The maximum number of iterations is 500.
- The number of clusters is 3.
- The movement coefficient is 2.
- The maximum number of cuckoos that can live at the same time is 60.
- The egg-laying radius decreases from 20 to 5.
- The population variance, which indicates the end of the algorithm, is 10^-13.
- These parameters are chosen to create a balance between the speed and accuracy of the algorithm.

The algorithm maintained high performance across different settings, confirming the stability of the chosen configuration.

In particular:

- A moderate initial population ensures sufficient diversity without excessive computational cost.
- The dynamic egg-laying radius helps transition from global exploration in early iterations to local exploitation near convergence.
- The stopping criterion based on variance ensures the algorithm halts when meaningful improvement ceases.

For practical applications, we recommend:

1. Conducting parameter tuning on small representative datasets.
2. Considering adaptive strategies for certain parameters, such as dynamically adjusting the radius or movement coefficient.
3. Avoiding very large populations or iteration counts unless problem complexity justifies them.

Problem Modeling

The Traveling Salesman Problem (TSP) is one of the well-known and classical problems in optimization and graph theory. The main goal of this problem is to find the shortest possible route for a salesman who must visit a specific set of cities and return to the starting city at the end [13-17]. Here is a more detailed explanation of TSP:

1. Problem Definition

In TSP, a salesman starts from one city, visits all the other cities, and eventually returns to the starting city. Each city should be visited only once, and the goal is to minimize the total distance traveled.

2. Input and Output

- Input: A set of cities and the distances or costs between them (usually represented as a distance or cost matrix).
- Output: The shortest route (tour) that visits all the cities and returns to the starting city.

3. The TSP has applications in various fields, including:

- Delivery logistics: Route planning for delivering goods to customers.
- Manufacturing and production: Optimizing the sequence of machines in production lines.
- Network management: Optimizing transport network routes.

- Data analysis: Solving clustering problems.

Complexity

TSP is an NP-hard problem, meaning that there is no efficient algorithm to solve all its instances. While for a small number of cities, exact algorithms like combinatorial methods or linear programming can find the optimal solution, for a large number of cities, approximation and metaheuristic algorithms are needed. Many studies and algorithms for TSP have been published in recent years, and they are continuously improving. Despite the challenges, solving TSP can help optimize and save time and costs in various projects. Ultimately, TSP is considered a central problem in graph theory and optimization, and research in this area remains an active field in mathematics and computer science.

A sample of solving the TSP problem related to Figure 3, which represents the locations of the cities, using the proposed method is shown in Figure 4, which illustrates the solutions found by the MCOA. The optimization process is displayed in Figure 5. A sample of solving the MTSP related to Figure 6, which shows the problem setup or city locations, using the proposed method is illustrated in Figure 7. This figure demonstrates the solutions obtained by applying the suggested approach to the MTSP. Furthermore, the optimization process, which outlines the steps taken to improve the solutions and achieve the best possible route, is displayed in Figure 8.

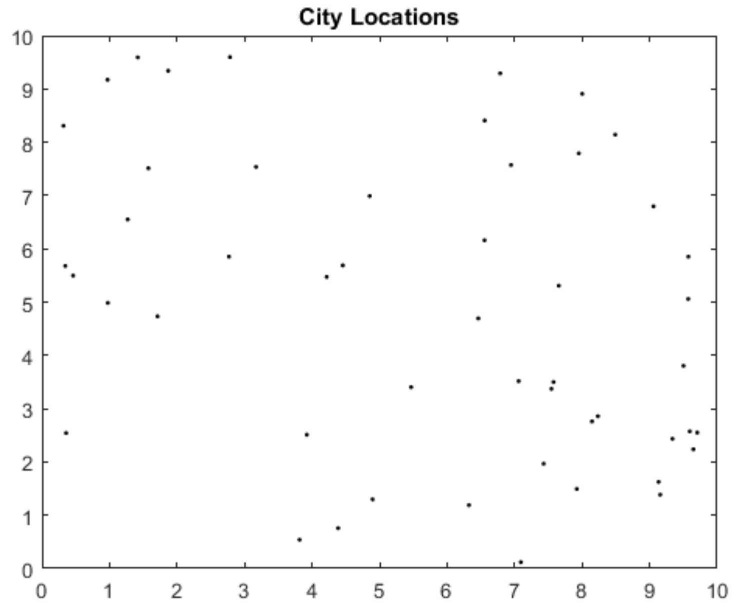

Figure 3. *The locations of the cities in TSP.*

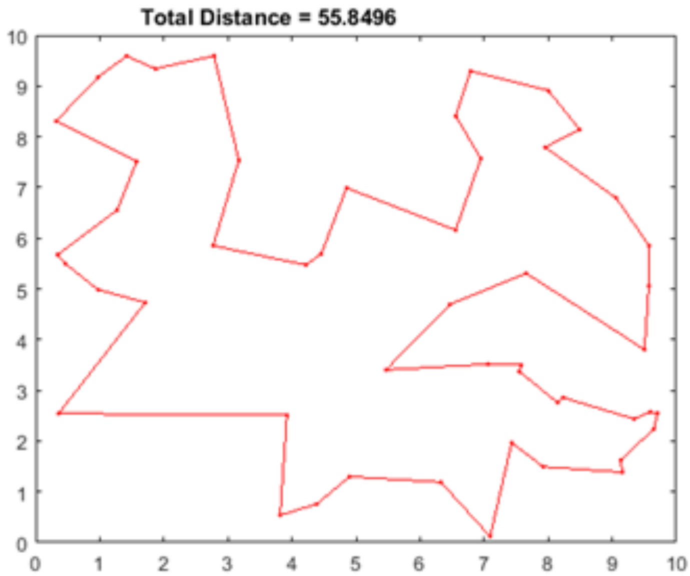

Figure 4. *The proposed solution for the TSP.*

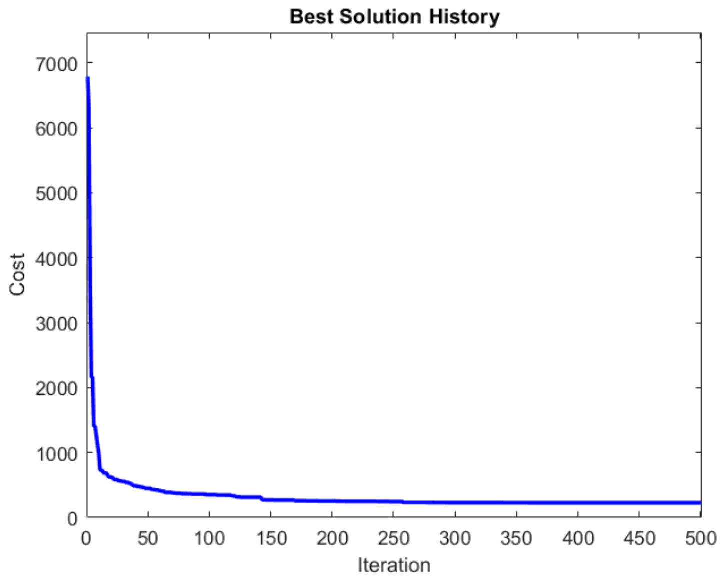

Figure 5. *The optimization process*

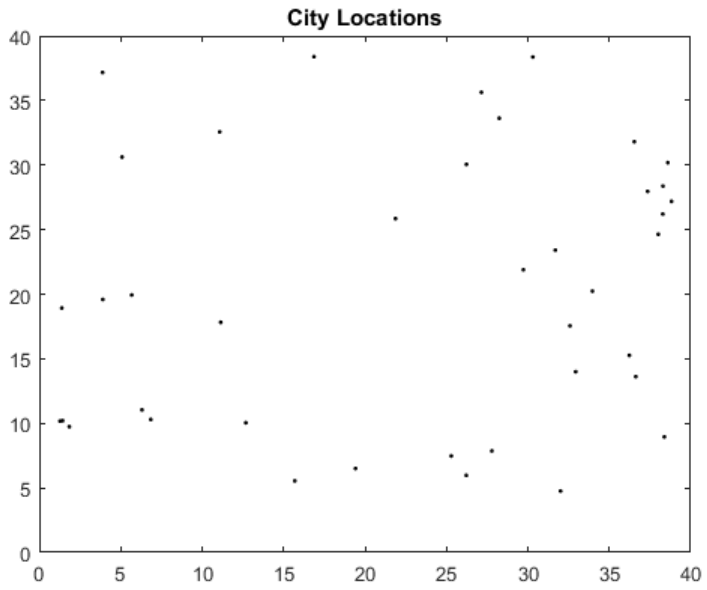

Figure 6. *The locations of the cities in MTSP.*

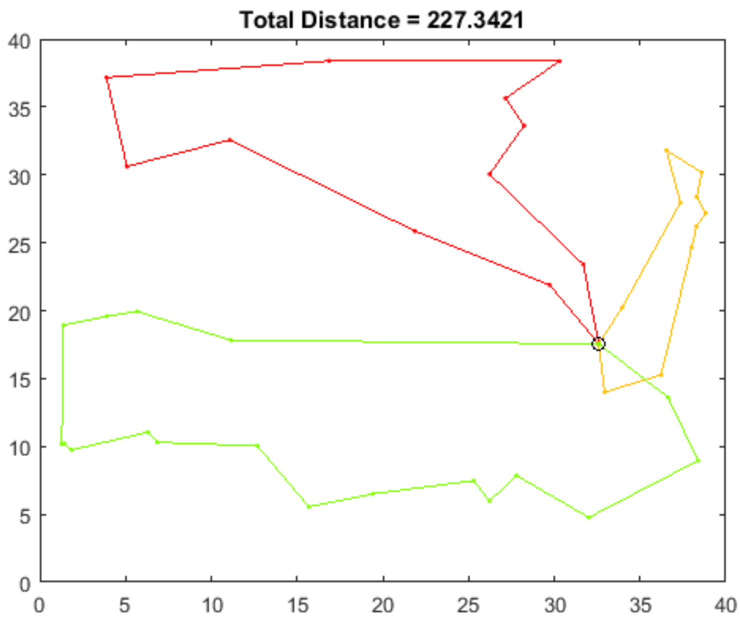

Figure 7. *The proposed solution for the MTSP.*

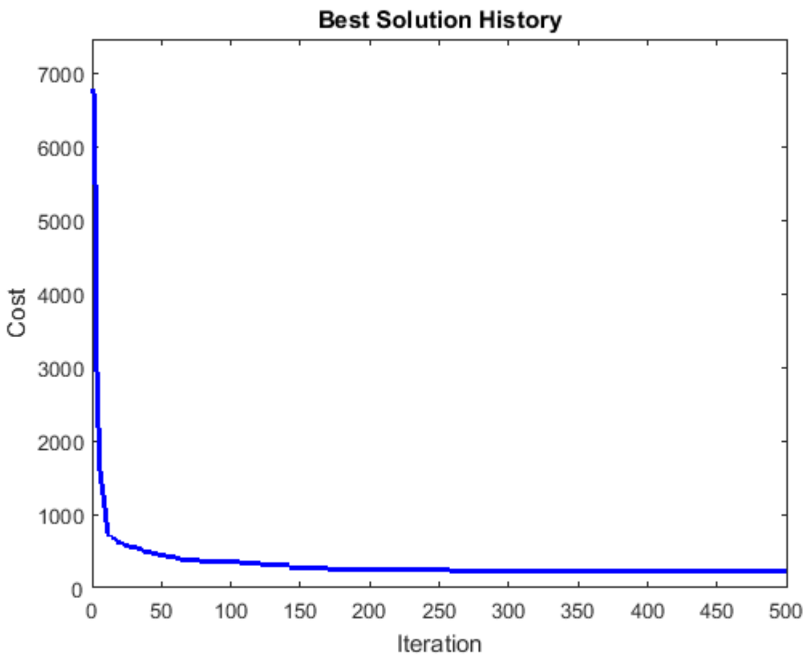

Figure 8. *The optimization process.*

In the following, to conduct further experiments, data from the Christophides dataset has been utilized [13-14]. This dataset consists of 14 different problems, labeled C1 through C14. In these problems, the number of customers ranges from 50 to 199, with their locations represented in Cartesian coordinates. The distances between customers are calculated using the Euclidean distance metric. Detailed information about the dataset and its characteristics is provided in Table 1. The results of the comparison are presented in Table 2. Based on the findings, the proposed method shows better performance across all scenarios compared to other methods. These results highlight the effectiveness and advantages of the proposed approach in solving the Christophides dataset problems.

***Table* 1.** Weight and atomic percentages of Barium ferrite from EDS analysis

Problem	Number of customers	Number of vehicles	Capacity	Maximum length of each tour	Service time
C1	50	5	160	∞	-
C2	75	10	140	∞	-
C3	100	8	200	∞	-
C4	150	12	200	∞	-
C5	199	17	200	∞	-
C6	50	6	160	200	10

C7	75	11	140	160	10
C8	100	9	200	230	10
C9	150	14	200	200	10
C10	199	18	200	200	10
C11	120	7	200	∞	-
C12	100	10	200	∞	-
C13	120	11	200	720	50
C14	100	11	200	1040	90

Table 2. Results of Comparing the Proposed Method with Existing Methods

Instance	**GA (Berger and Barkaoui, 2003)**		**PSO (Ai, Kachitvichyanukul, 2009)**		**Proposed**	
	Fitness	**Time**	**Fitness**	**Time**	**Fitness**	**Time**
C1	524.81	213	524.81	-	520.7	45
C2	849.77	765	844.42	-	831.22	159
C3	840.72	1148	829.40	-	822.14	230
C4	1055.85	2475	1048.89	-	1024.08	505
C5	1378.73	3999	1323.89	-	1290.11	799
C6	560.29	217	555.43	-	550.31	46.1
C7	914.13	786	917.68	-	908.62	157
C8	872.82	1134	867.01	-	845.84	236.8
C9	1193.05	2258	1181.14	-	1152.24	442.7
C10	1483.06	3687	1428.46	-	1396.56	750

C11	1060.24	1633	1051.87	-	1040.11	235
C12	877.8	1160	819.56	-	819.52	241
C13	1562.25	1694	1546.20	-	1540.75	332
C14	872.34	1197	866.37	-	864.27	249

Conclusions

In this paper, a novel method for vehicle routing based on an improved cuckoo algorithm is proposed. The proposed method optimizes the routing process using the enhanced cuckoo algorithm. This algorithm is a new and efficient approach suitable for solving high-dimensional problems and has been shown to outperform traditional algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). The algorithm begins the search process with a population of cuckoos and, through several iterations, moves towards the optimal solution. In each iteration, the search mechanisms are intelligently adjusted to guide the algorithm toward the most optimal solution. The simulations were carried out using MATLAB software to evaluate the performance of the algorithm under various conditions. The results obtained from the simulations demonstrate the superior performance of the proposed method when compared to existing methods. Additionally, these results indicate the algorithm's ability to handle large and complex problems where traditional algorithms may encounter issues such as getting trapped in local optima.

Future Research Directions

1. Hybridization with Other Algorithms: Combining MCOA with other optimization techniques such as genetic algorithms, ant colony optimization, or particle swarm optimization could help overcome the limitations of getting trapped in local optima and improve convergence in larger-scale problems.
2. Adaptive Parameter Control: Research into adaptive mechanisms for parameter tuning could make the algorithm more robust. Dynamic adjustment of parameters, such as the egg-laying radius and movement coefficient, based on the progression of the search, could help in more efficiently balancing exploration and exploitation.
3. Extension to Dynamic and Time-Varying VRPs: Future work could explore extending the MCOA to handle dynamic VRP scenarios. This could involve incorporating real-time changes in customer demands, time windows, and other dynamic constraints, making the algorithm more applicable to real-world logistics problems.
4. Parallelization and Scalability: Investigating parallel processing techniques for MCOA could help improve its scalability, especially for large-scale VRPs. Distributed computing or GPU-based optimization could reduce computation time and enable the algorithm to handle larger problem instances more effectively.
5. Incorporating Multi-Objective Optimization: Extending MCOA to handle multiple conflicting objectives, such as minimizing cost while also considering customer satisfaction or environmental impact, could make the algorithm more versatile and applicable to more complex real-world scenarios.

References

[1] Yesodha R, Amudha T. (2019). An improved firefly algorithm for capacitated vehicle routing optimization. Proceedings of the Conference on Optimization and Computing. 1–7.

[2] Yeh CC, Liu DY, Liao YK. (2016). Two-stage iterated local search for solving capacitated vehicle routing problems. Proceedings of the International Conference on Industrial Engineering. 45.

[3] Rahman MMH. (2017). A variant fisher and Jaikumar algorithm to solve capacitated vehicle routing problem. Proceedings of the International Conference on Computational Intelligence. 710–716.

[4] Zhang J. (2017). An efficient density-based clustering algorithm for the capacitated vehicle routing problem. Proceedings of the International Conference on Data Mining and Optimization. 465.

[5] Mingprasert S, Masuchun R. (2017). Applied artificial bee colony algorithm for multiple capacitated vehicle routing problem: Case study of the plastic packaging industry. Proceedings of the International Conference on Industrial Engineering and Applications. 270–273.

[6] Carwalo T, Thankappan J, Patil V. (2017). Capacitated vehicle routing problem. Proceedings of the National Conference on Engineering Optimization. 17–21.

[7] Song L, Dong Y. (2018). An improved differential evolution algorithm with local search for capacitated vehicle routing problem. Proceedings of the International Conference on Intelligent Systems. 801–806.

[8] Janjarassuk U. (2016). An ant colony optimization method for the capacitated vehicle routing problem with stochastic demands. Proceedings of the International Conference on Computer Science and Engineering. 1–5.

[9] Kuo RJ. (2017). Hybrid genetic ant colony optimization algorithm for capacitated vehicle routing problem with fuzzy demand: A case study on garbage collection system. Proceedings of the International Conference on Industrial Engineering and Engineering Management. 244–248.

[10] Shanmugasundaram G, Thilagavathi N, Ramya S, Kanimozhi K. (2019). An investigation of meta heuristic algorithms applied on capacitated vehicle routing problem. Proceedings of the International Conference on Advanced Computing. 1–6.

[11] Rajabioun R. (2011). Cuckoo optimization algorithm. Applied Soft Computing. 11(8): 5508–5518.

[12] Kahramanli HA. (2012). Modified cuckoo optimization algorithm for engineering optimization. International Journal of Future Computer and Communication. 1(2): 199–201.

[13] Berger J, Barkaoui M. (2003). A hybrid genetic algorithm for the capacitated vehicle routing problem. Lecture Notes in Computer Science. 646–656.

[14] Ai TJ, Kachitvichyanukul V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research. 36(5): 1693–1702.

[15] Tong S, Qu H, Xue J. (2023). K-DSA for the multiple traveling salesman problem. Journal of Systems Engineering and Electronics. 34(6): 1614–1625.

[16] Adamo T, Ghiani G, Greco P, Guerriero E. (2023). Learned upper bounds for the time-dependent travelling salesman problem. IEEE Access. 11: 2001–2011.

[17] Nand R, Chaudhary K, Sharma B. (2024). Single depot multiple travelling salesman problem solved with preference-based stepping ahead firefly algorithm. IEEE Access. 12: 26655–26666.