International Journal of Industrial Engineering & Production Research September 2025, Vol. 36, No. 3: 169-185
DOI: 10.22068/ijiepr.36.3.2116

RESEARCH PAPER

Clustering-Based Intelligent Route Optimization Algorithm for Capacitated Vehicle Routing Problem

Md. Rafsan Islam[1], Md. Azizur Rahman[2,*], Kazi Mohammad Nazib[3], Lasker Ershad Ali[4]

Received 15 September 2024; Revised 16 August 2025; Accepted 08 September 2025;
© Iran University of Science and Technology 2025

ABSTRACT
The Capacitated Vehicle Routing Problem (CVRP) is a significant variant of the vehicle routing problem that incorporates constraints related to customer demand and vehicle capacity. Owing to its extensive applications in logistics and transportation, CVRP has attracted substantial research attention, with numerous algorithms proposed from the perspective of intelligent search. A common solution strategy involves two phases: first, assigning customers to different vehicles to form feasible routes, and second, optimizing these routes. This paper presents a two-phase CVRP solution framework through the clustering concept with intelligent search to improve route planning. In the first phase, a set of clustering methods -fuzzy c-means, k-means, and k-medoids - combined with a nearest neighbor heuristic search, are applied to generate feasible routes for each vehicle. In the second phase, these routes are iteratively optimized using the Simulated Annealing (SA) algorithm. The process yields three distinct solution pathways: fuzzy c-means with SA, k-means with SA, and k-medoids with SA. For performance evaluation, 46 benchmark CVRP datasets from a publicly available library are used. Simulation results demonstrate that k-means with SA performs the best, surpassing the other two approaches and outperforming other clustering-based two-phase state-of-the-art algorithms in terms of solution quality.

KEYWORDS: *Combinatorial optimization; Capacitated vehicle routing problem; Two-phase algorithm; Clustering algorithm; Simulated annealing algorithm.*

1. Introduction

The vehicle routing problem (VRP) is a complex and well-studied combinatorial optimization problem in the fields of operations research and logistics. It was first introduced by Dantzing and Ramser [1]. In this scenario, a fleet of vehicles departs from a fixed location (called a depot), serves a set of customers, and then returns to the depot. The main goal of the problem is to minimize the total cost of transportation, which could include distance traveled, time, fuel consumption, or a combination of these factors. The VRP can be seen as an extension of the Traveling Salesman Problem (TSP), where each vehicle route is akin to a single TSP. Due to significant applications in logistics and transportation, it has emerged as one of the most important and active research topics in the scientific community [2]. In recent years, various constrained versions of VRP have been explored, among which the Capacitated Vehicle Routing Problem (CVRP) is particularly noteworthy.

The CVRP is a specialized version of the VRP that includes constraints related to vehicle capacity and customer demand. In CVRP, a fleet of vehicles with limited capacity departs from a central depot to serve a set of customers, each with a specific demand. After serving the customers, the vehicles return to the central depot. A route becomes invalid if the total demand on that route exceeds the vehicle's capacity. The main objective in solving CVRP is to minimize the total cost of the fleet of vehicles while ensuring that vehicle capacities are not exceeded [3]. The CVRP has extensive applications in logistics, transportation, disaster management, and various supply chain management sectors [4-5]. Besides that, it has been proven to be an NP-hard problem [6]. These broad applications with solution difficulties underscore the importance of finding effective and high-quality solutions to CVRP. This study is motivated by the need to develop better approaches for solving CVRP, given its critical role in optimizing logistical and supply chain operations.

* *Corresponding author: Md. Azizur Rahman*
mdazizur@math.ku.ac.bd

1. *Department of Mathematics, Hamdard University Bangladesh, Gazaria, Munshiganj 1510, Bangladesh*
2. *Mathematics Discipline, Science, Engineering and Technology School, Khulna University, Khulna 9208, Bangladesh*
3. *Mathematics Discipline, Science, Engineering and Technology School, Khulna University, Khulna 9208, Bangladesh*
4. *Mathematics Discipline, Science, Engineering and Technology School, Khulna University, Khulna 9208, Bangladesh*

The solution of CVRP is extremely challenging due to its NP-hard nature [6]. Researchers worldwide have employed various techniques to tackle this problem. Metaheuristic algorithms are among the most widely used approaches [7], yet many fail to exploit problem-specific structures-such as the natural clustering of customer locations-leading to inefficient searches and slow convergence, especially in large-scale cases. Traditional metaheuristics explore the solution space without leveraging clustering to reduce complexity, while clustering-based VRP methods seldom integrate metaheuristics in a way that balances exploration and exploitation. Furthermore, metaheuristics typically start with randomly generated initial solutions, and their effectiveness strongly depends on this starting point; better initial solutions improve the chances of achieving near-optimal results. However, random initialization often yields inconsistent outcomes, creating a gap for methods that can systematically generate high-quality starting solutions.

To address this, we propose a clustering-based two-phase approach for solving the CVRP. In the first phase, a set of clustering algorithms is employed to group customer locations, and a nearest neighbor-based method is then used to construct a structured, high-quality initial feasible solution. In the second phase, a metaheuristic algorithm is adopted to refine this solution to improve efficiency, scalability, and adaptability to real-world scenarios while maintaining competitive solution quality. By integrating clustering with metaheuristics, our approach enhances search performance and achieves near-optimal solutions more consistently. Actually, three clustering techniques-fuzzy c-means, k-means, and k-medoids-are employed along with a nearest neighbor-based approach in the first phase to generate the initial solution. Initially, the clustering methods organize the consumer data points into distinct clusters. Following this, the nearest neighbor-based approach is applied to produce the initial solutions, thus completing the first phase. Specifically, the fuzzy c-means algorithm groups homogeneous data points within the same cluster based on their membership values [8]. The k-means algorithm partitions data points according to their proximity to cluster centers, with initial centers chosen randomly and updated iteratively until convergence [9]. Similarly, the k-medoids method, which is conceptually similar to k-means, randomly selects initial medoids and updates them iteratively until convergence, assigning data points to clusters based on their closeness to these medoids [10].

In the second phase, the Simulated Annealing (SA) algorithm is used to optimize the initial routes generated by the three clustering techniques. The SA, inspired by the metal annealing process, is a metaheuristic method that starts at a predefined temperature. In each iteration, local search operators are applied to generate and improve solutions, while the temperature gradually decreases according to a cooling schedule. This process continues until the system reaches its final temperature. Three local search operators-swap, revision, and insertion-are employed within SA to refine the initial solutions. These operators iteratively adjust the routes, enhancing their quality and efficiency. Integrating each clustering algorithm with SA produces three distinct solution streams: fuzzy c-means+SA, k-means+SA, and k-medoids+SA. Each stream was independently tested on 46 benchmark CVRP datasets. The simulation results show that the k-means+SA stream consistently outperforms the other two proposed streams. Moreover, it significantly surpasses other clustering-based two-phase state-of-the-art algorithms.

This paper is organized as follows: clustering-based two-phase approaches to solve CVRP are reviewed in Section 2. In Section 3, a mathematical discussion of the CVRP problem is presented to provide a clear understanding of the problem under study. In Section 4, our proposed two-phase method is discussed in detail, including a brief overview of the clustering and routing phases, along with the algorithms used and their working architectures. We present the results and discussion, including experimental setups, in Section 5. In this section, we identify the best performer among the three proposed streams for solving the CVRP problem. A performance comparison between our proposed method and the previous clustering-based techniques with a rigorous statistical test is also presented in this Section. Finally, we conclude the paper and offer recommendations for future research in Section 6.

2. Related Work

Various strategies have been proposed and developed for solving the CVRP problem. In this study, we employ a two-phase strategy to address this problem. In the first stage, we generate an initial solution or a feasible solution using a set of

clustering methods. We then improve this feasible solution with the help of a SA metaheuristic. Consequently, it is natural to review previous studies that have employed similar two-phase approaches. This section, therefore, focuses on prior research that has used a two-phase strategy to address the CVRP.

In 2017, Akhand et al. proposed a two-phase approach to solving the CVRP [11]. In the first phase, they used a variant sweep technique to cluster the data and create initial routes. This variant sweep algorithm (VS) considered multiple starting angles to improve clustering results. After clustering, the initial route was optimized in the second phase through several methods: the Producer-scrounger Method (PSM), Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Velocity Tentative Particle Swarm Optimization (VTPSO). Their simulation results demonstrated that clustering with the variant Sweep and route optimization with VTPSO generated better results in all cases. Considering the two-phase method, Peya et al. solved CVRP in 2018 utilizing adaptive sweep (AS) and swarm intelligence-based algorithms [12]. In the initial phase, they clustered the data using an adaptive sweep technique. In the second stage, they optimize the clustered route using swarm intelligence-based algorithms. Actually, the clustered routes are optimized using three swarm intelligence-based algorithms: Ant Colony Optimization (ACO), the Producer-scrounger Method (PSM), and Velocity Tentative Particle Swarm Optimization (VTPSO). Additionally, they incorporated a local search-based Genetic Algorithm (GA) for route optimization. Their findings indicated that the combination of adaptive sweep with VTPSO performed better than other existing methods.

Wang et al. developed a two-stage solution approach for the large-scale CVRP problem in the year of 2018 [13]. To effectively solve the large-scale CVRP, they established a three-class and three-tier model, employing a two-phase strategy. To generate an initial or feasible route in the first stage, k-means clustering is utilized, whereas the initial route is optimized in the second phase through an improved Ant Colony Optimization (ACO) algorithm. The integration of k-means clustering with the improved ACO results in a solution approach that generates higher-quality solutions while consuming less computational time and space complexity. To solve the real-world CVRP data, Comert et al. proposed a cluster-first, first-route, second strategy in the year 2018 [14]. Their approach involved two distinct phases: clustering and routing. In the clustering phase, they utilized three different clustering techniques-k-means, k-medoids, and random clustering-taking into account vehicle capacity constraints. Subsequently, in the routing phase, they employed a branch and bound algorithm to optimize the routes generated from the clustering phase. They implemented their framework on a case study in a supermarket chain. Through simulation, they found that k-medoids outperformed the other clustering techniques in delivering good quality solutions tailored to the specific real-world scenario they studied.

In 2022, Mahmood proposed a two-stage approach for the solution of CVRP [15]. In the first stage, he employed a sweep algorithm for accomplishing clustering tasks. The initial routes generated from this clustering were then optimized using the Meerkat Clan Algorithm (MCA). In comparison to other combinations, this one yielded a better outcome. Considering a two-phase approach, Akhand et al. solved the CVRP in 2017 using adaptive sweep and VTPSO [16]. For clustering purposes, a heuristic-based adaptive sweep technique is utilized in the first phase. On the other hand, VTPSO is employed to optimize the feasible route obtained from the clustering phase. This combination of algorithms yielded better outcomes compared to other standard existing algorithms.

In 2019, Peya et al. established a clustering-based two-phase approach for solving CVRP [17]. In the first phase, they employed three algorithms for clustering the data: Distance-based Sweep Nearest (DSN), Sweep (SW) algorithm, and Sweep Nearest (SN) algorithm. In the routing (optimization) phase, they optimized the clustered routes using three different intelligent algorithms: Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). They concluded from simulation results that the DSN outperformed SN and SW in most datasets, and DSN combined with PSO provided better results than others for CVRP. Recently, Yavsan et al. developed a real-time Android application that addresses the CVRP [18]. They employed a clustering-based approach to solve the CVRP, beginning with data clustering using a fuzzy-c-means (FCM) method. In the second phase, they optimized the initial routes with an Ant Colony Optimization (ACO) algorithm. Their proposed

method was evaluated on 10 benchmark CVRP datasets as well as two real-world datasets. Simulation results showed that the combination of fuzzy-c-means with ACO demonstrated superior outcomes compared to other methods on both benchmark and real-world datasets.

Hamta et al. studied the open vehicle routing problem while considering uncertainties in vehicle demand and costs [19]. To address these uncertainties, they developed a robust optimization model using the method proposed by Bertsimas and Sim. In parallel, a deterministic open vehicle routing model was also formulated. Both models were solved using GAMS software. Simulation results demonstrated that the robust model out-performed the deterministic one. Meanwhile, Abbal et al. investigated the Multi-Level Multi-Capacitated Facility Location Problem (ML-MCLP), which generalizes the Capacitated P-Median Problem [20]. Their primary objective was to minimize assessment costs while accounting for consumer demands. To solve this problem, they employed an exact Benders Decomposition approach enhanced with modern acceleration techniques aimed at improving convergence.

Yong et al. investigated the use of an adaptive ant colony optimization (AACO) technique to solve the CVRP problem [21]. First, the AACO was used to generate different sub-routes. Then, a Genetic Algorithm (GA) was applied to optimize the combined route and produce approximate solutions. The Clarke and Wright algorithm was subsequently employed to verify the feasibility of these approximate solutions. To further improve the performance of the feasible solutions, the 2-opt and sweep algorithms were used. Fitzpatrick et al. developed a machine-learning-based heuristic approach for solving large-scale CVRP instances [22]. They divided the problem into smaller subproblems and designed a machine-learning-based heuristic to solve each subproblem individually. The subproblem solutions were then combined using an Integer Linear Programming (ILP) formulation to produce the final solution while ensuring capacity constraints were met.

Muriyatmoko et al. conducted a comparative study to examine the effectiveness of heuristic and metaheuristic algorithms in solving a real-world CVRP [23]. The real-world case involved faculty transportation between the male campus in Ponorogo and the female campus in Mantingan Ngawi at Universitas Darussalam Gontor. For the simulations, they used Google OR-Tools with Python, implemented on Google Colab. Their findings showed that metaheuristic approaches-specifically Christofides Guided Local Search and Christofides Simulated Annealing-outperformed heuristic methods in finding better routes. Ekayanti and Efendi addressed the CVRP using a combination of the sweep algorithm and the nearest neighbor approach [24]. The sweep algorithm was first applied to cluster cities into groups based on their polar angles. Then, the nearest neighbor algorithm was used within each cluster to optimize the routes. The Haversine distance formula was employed for distance calculations in the nearest neighbor process. Their results demonstrated that the combined sweep and nearest neighbor approach performed effectively.

3. Mathematical Description of CVRP

The Capacitated Vehicle Routing Problem (CVRP) is a well-known combinatorial optimization problem in logistics and transportation. The goal of this problem is to find the optimal routes for a fleet of vehicles that minimize the total travel cost while ensuring that each customer is visited exactly once, the vehicle capacities are not exceeded, and all routes start and end at the depot. Mathematically, the CVRP consists of a collection of uniform vehicles $v \in V$, each serving a specific area represented on a graph G=(N,E), where N is the union of two sets-D and C, i.e., $N=D \cup C$. Here, D is the single depot represented at node 0, and C is the set of n customers, with $i \in C$. E represents an arc joining nodes(i,j). The traveling distance from node i to node j is d_{ij}, where the distances are symmetric, i.e., $d_{ij}=d_{ji}$. If a vehicle uses the arc (i,j), then the decision variable $x_{ijv}=1$; otherwise, $x_{ijv}=0$. The objective function of the CVRP can be defined as follows [25]:

$$\text{Minimize } \sum_{i\in N}\sum_{j\in N}\sum_{v\in N} d_{ij}x_{ijv} \quad (1)$$

Subject to the following constraints-

$$\sum_{j\in N}\sum_{v\in V} x_{ijv} = 1 \qquad \forall i \in C \quad (2)$$

$$\sum_{i\in N}\sum_{v\in V} x_{ijv} = 1 \qquad \forall j \in C \quad (3)$$

$$\sum_{j\in N}\sum_{j\in N} m_i x_{ijv} \le Q \qquad \forall v \in V \quad (4)$$

$$\sum_{i\in N} x_{ijv} - \sum_{i\in N} x_{ijv} = 0 \qquad \forall j \in C,\ v \in V \quad (5)$$

$$\sum_{j\in C} x_{0jv} \le 1 \qquad \forall v \in V \quad (6)$$

$$\sum_{i\in C} x_{i0v} \le 1 \qquad \forall v \in V \quad (7)$$

$$\sum_{i\in P}\sum_{j\in P} x_{0jv} \le |P| - 1 \qquad \forall P \subseteq C,\ |P| \ge 2,\ v \in V \quad (8)$$

$$x_{ijv} \in \{0,1\} \qquad \forall i,\ j \in N,\ v \in V \quad (9)$$

Eq. (1) represents the objective function of the problem, which minimizes the cost associated

with the set of routes covered by vehicles $v \in V$. The constraints in Eq. (2) and Eq. (3) ensure that each customer is visited exactly once by a vehicle and that each customer is left exactly once by a vehicle. The constraint Eq. (4) ensures that the load of any vehicle does not exceed its capacity. The constraints Eq. (5) balance the inflow and outflow of vehicles. The constraints Eq. (6) and Eq. (7) confirm that each vehicle starts and ends at the depot. Equation (8) demonstrates the sub-tour breaking, and Eq. (9) describes the domain of decision variables.

4. Methodology

Functionally, the proposed algorithm addresses the CVRP problem through a structured two-phase approach, first constructing feasible solutions and then iteratively improving them. In the first phase, the algorithm generates candidate routes for each vehicle using three distinct clustering techniques: fuzzy c-means, k-means, and k-medoids, each combined with a modified nearest neighbor search to sequentially select customers while respecting vehicle capacity constraints. In each case, the modified nearest neighbor heuristic constructs initial routes by iteratively selecting the nearest feasible customer, resulting in three unique sets of initial solutions that reflect different spatial and clustering perspectives. In the second phase, these initial solutions are further optimized using a Simulated Annealing (SA) algorithm, which iteratively refines routes by exploring the solution space and occasionally accepting worse solutions to escape local optima. Each phase is discussed in detail in the following sub-sections, and the complete workflow is illustrated in Fig. 1.

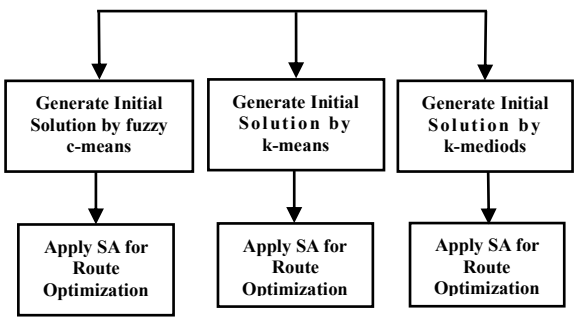


Fig.1. The complete workflow of the Proposed Approach

4.1. Phase-1: Feasible solution generation

In this phase, we generate a feasible route or solution for each vehicle in the CVRP. Actually, we employ three different clustering techniques: fuzzy c-means, k-means, and k-medoids to develop the initial solutions. Initially, data is clustered using these techniques, ensuring that the number of clusters matches the number of vehicles. After clustering, all cluster centers are utilized, and a nearest-neighbor-based approach is applied. The distance of each data point from each cluster center is calculated, and based on the shortest distance and the capacity of each vehicle, data points are assigned to different vehicle paths. This process defines the initial routes for each vehicle. Once the initial routes for all vehicles are generated, they are merged into a single route. This merged route serves as the initial solution for the Simulated Annealing (SA) Algorithm. Detailed descriptions of the clustering techniques and the nearest-neighbor-based approach are provided in the following sub-sections. In addition, the flowchart and pseudocode for generating initial solutions are offered in Fig. 2 and in Algorithm 1, respectively.

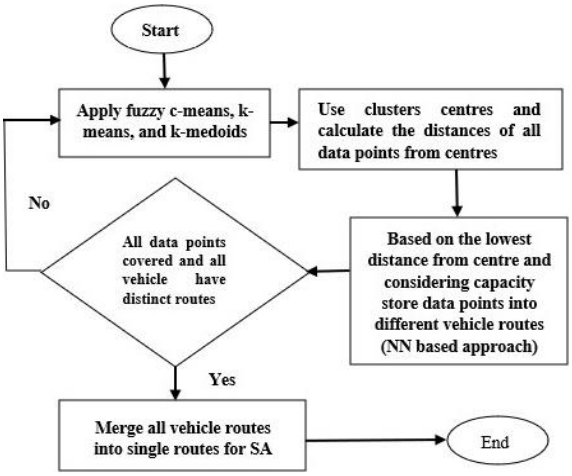


Fig.2. Flowchart for generating an initial or feasible route for each vehicle

4.1.1. Nearest neighbour approach

To generate a feasible route using clustering algorithms, a Nearest Neighbor (NN) algorithm is adopted in the process. Actually, we modify the traditional NN algorithm by incorporating a capacity constraint along with the minimum distance criterion. This modified NN algorithm identifies the nearest unvisited vertex while taking vehicle capacity into account. This approach can be described step-by-step as follows:

Step 1: Use the cluster centers obtained from clustering as the initial points.

Step 2: Find the nearest unvisited vertex b from

the centers.
Step 3: Store b in the vehicle routes.
Step 4: Check the capacity of vehicles.
Step 5: Repeat steps 2-4 until all data points are covered and each vehicle has a distinct route.

4.1.2. Fuzzy C-means

The Fuzzy C-Means (FCM) algorithm is a probabilistic clustering technique in which each data point has a specific probability, known as the membership value, of belonging to a cluster. These membership values are used to assign data points to various clusters. This method extends the k-means clustering algorithm, with a significant difference in how it assigns data points to clusters. The objective function of FCM can be mathematically defined as follows [26]:

$$\text{Minimize } Z = \sum_{i\in 1}^{N}\sum_{j\in 1}^{Y}\mu_{ij}^{m}\left|\left|x_i - c_j\right|\right|^2 \quad (10)$$

where N represents the number of data points, Y stands for the number of cluster centers, x_i is the ith data point, c_j is the jth cluster centers, μ_{ij} represents the membership matrix of ith data from j^{th} cluster, m is the fuzziness parameter, and $\left|\left|x_i - c_j\right|\right|$ is the Euclidean distance between x_i and c_j. To implement the FCM algorithm, the following steps are followed:
Step 1: The initialization of the membership matrix is $U^0 = \mu_{ij}$.
Step 2: At any stage k, calculate the centers $C^k = c_j$ with the membership matrix U^k. Actually, the centers can be found using the following formula:

$$c_j = \frac{\sum_{i\in 1}^{N}\mu_{ij}^{m}x_i}{\sum_{i\in 1}^{N}\mu_{ij}^{m}} \quad (11)$$

This step computes the new position of each cluster center based on the current membership values.
Step 3: Update the membership matrix of U^{k+1} using the following formula:

$$\mu_{ij} = \frac{1}{\sum_{k\in 1}^{Y}\left(\frac{x_i - c_j}{x_i - c_k}\right)^{\frac{2}{m-1}}} \quad (12)$$

This ensures that the closer a data point is to a cluster center, the higher its membership value in that cluster.
Step 4: Check the convergence for a small number $\in$ using the following inequality:

$$\left|U^{k+1} - U^k\right| < \in \quad (13)$$

If Step 4 is satisfied, the process should be stopped. Otherwise, the next iteration should begin at Step 2.

4.1.3. K-means

The k-means is a straightforward clustering algorithm that partitions data into groups based on the distances between each observation and the cluster centers. Initially, it randomly selects the cluster centers. After each iteration, the cluster centers are updated, and the distances between the cluster centers and each observation are recalculated. This process continues until the objective function is satisfied. For a collection of observations $X_i (i = 1,2,3, \dots, n)$, the primary goal is to divide the observations into k sets $C = \{C_1, C_2, \dots C_k\}$. The objective function of k-means can be mathematically defined as follows [27]:

$$\text{Minimize } J = \sum_{i=1}^{k}\sum_{x\in C_i}\|x - \bar{x}_i\|^2 \quad (14)$$

where k is the number of clusters, C_i is the set of data points in cluster i, x is a data point in a cluster i, $\bar{x_i}$ is the center of cluster i, and $\left|\left|x - \bar{x_i}\right|\right|^2$ is the squared Euclidean distance between data point x and the center $\bar{x_i}$. The working procedure of the k-means algorithm can be described stepwise as follows:
Step 1: Initialize k cluster centers randomly.
Step 2: Assign each observation to the nearest cluster center.
Step 3: Update the cluster centers using the following formula:

$$\bar{x_i} = \frac{1}{|C_i|}\sum_{x\in C_i} X \quad (15)$$

where C_i is the set of data points assigned to a cluster i.
Step 4: Calculate the value of the objective function as defined in Eq. (14).
Step 5: Repeat steps 2-4 until the value of the objective function is minimized.

4.1.4. K-medoids

k-medoids is a clustering algorithm similar to k-means but with distinct advantages, especially in managing noise and outliers. Unlike k-means, k-medoids does not calculate cluster centers. Instead, it selects representative data points, or medoids, from the total observations to serve as centers. The remaining process is the same as k-means. Here's an overview of the k-medoids algorithm [28]:
Step 1: Select k objects randomly from the dataset as the initial medoids.
Step 2: Calculate the distances between each observation and the cluster medoids.
Step 3: Assign each observation to the cluster with the closest medoid.

Step 4: Update the cluster medoids by selecting the object within each cluster that minimizes the total dissimilarity to all other objects in the cluster.
Step 5: Repeat steps 2-4 until the medoids remain the same in two consecutive iterations.

4.2. Phase-2: Feasible solution optimization

In this phase, the feasible route generated in the first stage is refined to achieve a better overall solution using the meta-heuristic algorithm Simulated Annealing (SA). The feasible route for each vehicle is taken as the initial solution, and SA iteratively explores neighboring solutions by making controlled modifications. Improvements are accepted directly, while less favorable changes may still be accepted with a probability that decreases over time according to a cooling schedule, allowing the search to escape local optima. This process continues until the algorithm converges, producing an optimized route that minimizes the defined objective, such as cost, distance, or travel time. The optimization process of the SA algorithm is briefly described below.

Algorithm.1. Pseudocode for generating an initial or feasible route for each vehicle

```
While all data points are not covered and all vehicles have distinct routes
  Apply k-medoids, fuzzy c-means, and k-means and store cluster centres
    For each cluster centre
      For each data point
        Determine each data point's distance from the cluster centre
      End
        Sort the index numbers of calculated distances into ascending order
        Store the sorted indices into $D$
          For each index in $D$
            Store the data points of $D$ into vehicle routes till capacity holds
          End
    End
End
```

4.2.1. Simulated annealing algorithm

Simulated Annealing (SA) is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. It's particularly effective for solving combinatorial optimization problems like the Traveling Salesman Problem (TSP), the Capacitated Vehicle Routing Problem (CVRP), etc. The SA algorithm, described by Kirkpatrick et al. [29], begins by generating an initial solution randomly. In this context, a feasible route generated by Phase-1 serves as the initial solution, denoted as S_1. The fitness value of S_1 is calculated based on the overall transportation cost, typically computed using Euclidean distance. Next, initial parameters are set: the starting temperature T_0, the end temperature T_{end}, and the cooling rate q. A new solution, S_2, is produced from S_1 using local search operators—Insertion, Reversion, and Swap-selected randomly. The fitness value of S_2 is then calculated. If S_2 has a better fitness value (lower total transportation cost) than S_1, S_2 replaces S_1 for the next iteration. If not, a random value r and a probability $p = e^{-dc/T}$, where dc is the increase in fitness and T The current temperature are computed. If p is greater than or equal to r, S_2 is accepted; otherwise, S_1 remains. The temperature is then updated at each iteration using the formula $T = T_0 * q$, and the process repeats until T_0 falls to or below T_{end}.
The path representation of the SA algorithm for the CVRP problem is a crucial aspect of finding an optimal solution. To represent the CVRP path, we consider all feasible vehicle routes. By combining these routes, we generate the initial solution for SA. The representation of these routes helps in evaluating the cost and making necessary adjustments during the SA process. To decode each vehicle route after implementing SA, we use some dummy variables. These variables assist in translating the SA results into actual vehicle paths. To better understand the representation and the overall process, visual aids are used. For instance, a dummy representation of the path is provided in Fig. 3, while a flowchart outlining the complete SA process for CVRP is provided in Fig. 4.

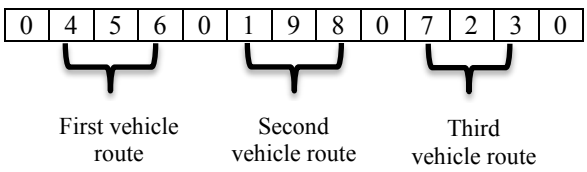


Fig.3. Path representation of the Simulated Annealing (SA) algorithm for CVRP

To generate new solutions in the SA algorithm, three local search approaches-swap, reversion, and insertion-are applied. Actually, these local search approaches are selected randomly at each iteration of SA.

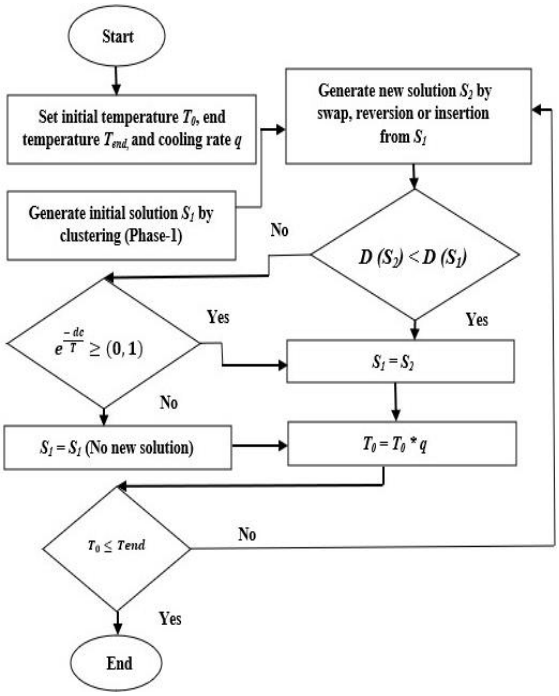


Fig.4. Flowchart of Simulated Annealing (SA) algorithm for the CVRP

The swap operator alternates the vertex positions. It selects two vertices and exchanges their positions. In this instance, vertex index numbers are selected at random. Therefore, swap in our case will randomly select two vertices and change their positions. Fig. 5 shows the implementation of the swap operation. The reversion operator reverses the sub-routes between any two randomly chosen vertices. Once selected, the sub-routes connecting these vertices are reversed, creating a new route. Fig. 6 shows the new solution generation using the reversion operator. On the other hand, insertion is the process of adding a vertex to the routes. A random vertex is selected from the routes, removed from its current position, and then reinserted at a new, randomly chosen position.

Before Swap												
0	4	5	6	0	1	9	8	0	7	2	3	0
After Swap												
0	4	7	6	0	1	9	8	0	5	2	3	0

Fig.5. The new solution generation of the SA through swap operator

Before Reversion												
0	4	5	6	0	1	9	8	0	7	2	3	0
After Reversion												
0	4	5	6	0	3	2	7	0	8	9	1	0

Fig.6. The new solution generation of the SA through reversion operator

Fig. 7 demonstrates the working process of the insertion operator. For a better understanding of the complete procedure of the SA algorithm, Algorithm. 2 provides a pseudocode for the process.

Before Insertion												
0	4	5	6	0	1	9	8	0	7	2	3	0
After Insertion												
0	4	5	6	0	3	1	9	8	0	7	2	0

Fig.7. The new solution generation of the SA through the insertion operator

Algorithm.2. Pseudocode of the Simulated Annealing (SA) algorithm for CVRP

1. Setting initial temperature (T_0), End Temperature (T_{end}), Cooling Rate (q)
2. Create the initial solution (S_1) by imposing Phase-1
3. While $T_0 \leq T_{end}$
4. For the number of inner iterations
5. S_2= create a new solution from S_1 by swap, reversion, or insertion
6. If distance$(S_2) <$ distance(S_1)
7. $S_1 = S_2$
8. Else If $e^{-dC/T} \geq (0,1)$
9. $S_1 = S_2$
10. Else
11. $S_1 = S_1$ (no new route generation)
12. End
13. End
14. $T_0 = q * T_0$
15. End While
16. return (S_1)

5. Simulation Results and Analysis

This section presents the simulation results and analysis for the proposed approaches, evaluated using 46 benchmark CVRP datasets from publicly available sources [30], comprising 21 from the A set, 11 from the B set, 10 from the P set, and 4 from the E set. In each dataset, the numerical value in its name denotes the number of customer nodes and vehicles. Detailed dataset descriptions are available on the provider's website [30]. Three methods-fuzzy c-means with SA, k-means with SA, and k-medoids with SA-were tested independently on all datasets. Simulations were executed in MATLAB R2013a on a desktop with an Intel(R) Core(TM) i5-4590 CPU @ 3.30 GHz and 8 GB RAM. Parameters were tuned through trial and error, and the optimal configurations for each method are summarized in Table 1.

Tab.1. Optimized parameter settings for the proposed streams

Parameter name	Optimized value
Fuzzy parameter	2.0
Initial temperate	1000
End temperature	0.0001
Cooling rate	0.99
Inner iteration	100
Total iteration	20

5.1. Results analysis

This subsection presents the simulation results from our experiments, summarized in Tables 2–5. In these tables, the "Data Sets" column represents the dataset name, while the "BKS" column indicates the best-known solution reported by the data library. For each dataset, the best (minimum) results, average results, and average computational times from 20 iterations of each approach-k-medoids+SA, fuzzy c-means+SA, and k-means+SA are presented in the third, fourth, and fifth columns, respectively. The bolded values highlight the best performer among the three approaches.

The simulation results on benchmark B datasets are demonstrated in Table 2. From the table, it is evident that both k-means and fuzzy c-means with SA achieve a comparable number of wins in terms of both average and minimum results. It can also be visible that k-means with SA consistently produce better results on larger datasets, excelling in both minimum and average outcomes. On the other hand, while k-medoids with SA achieve superior average results on the B-n57-k9 dataset, its overall performance on the B datasets is inferior to that of the other two methods. The simulation results from our proposed approaches on the benchmark P datasets are summarized in Table 3. The table shows that k-means combined with the SA algorithm outperforms other methods in both average and minimum outcomes, particularly excelling on larger datasets. In contrast, while fuzzy c-means and k-medoids with SA demonstrate satisfactory performance on average results, they exhibit certain limitations when it comes to achieving minimum results.

Table 4 shows the simulation results from the experiment on the benchmark E datasets. It can be observed from the table that k-means combined with the SA algorithm significantly dominates the other two methods in both average and minimum outcomes, particularly demonstrating strong success on larger data sets.

Tab.2. Simulation results on B datasets by the proposed three streams

Data Sets	BKS	k-mediods + SA			fuzzy c-means + SA			k-means + SA		
		Avg.	Min	AVT/s	Avg.	Min	AVT/s	Avg.	Min	AVT/s
B-n31-k5	672	**685.60**	678.71	27.30	688.90	676.09	37.05	686.43	676.09	25.85
B-n34-k5	788	815.72	790.25	29.13	812.83	792.11	37.37	**812.57**	**790.06**	37.37
B-n38-k6	805	828.83	818.81	50.43	825.45	810.81	48.00	**824.49**	**808.70**	41.87
B-n39-k5	549	596.89	**553.67**	31.68	608.27	563.56	47.18	**584.37**	561.81	47.83
B-n43-k6	742	790.90	751.80	34.83	**788.68**	**750.72**	54.45	792.74	762.29	52.13
B-n45-k5	751	881.80	793.75	35.41	**878.17**	798.87	51.96	898.97	**791.63**	54.83
B-n50-k7	741	801.58	751.53	36.98	795.13	**745.75**	39.73	**792.52**	746.24	57.11
B-n52-k7	747	835.48	761.31	60.56	**827.78**	763.42	60.28	833.86	**756.24**	62.21
B-n56-k7	707	781.64	730.35	48.51	**781.57**	739.57	62.08	782.23	**719.84**	66.99
B-n57-k9	1598	**1702.20**	1638.60	55.20	1723.60	1648.56	43.89	1703.14	**1611.17**	64.32

Tab.3. Simulation results on P datasets by the proposed three streams

Data Sets	BKS	k-mediods + SA			fuzzy c-means + SA			k-means + SA		
		Avg.	Min	AVT/s	Avg.	Min	AVT/s	Avg.	Min	AVT/s
P-n16-k8	450	451.46	451.34	15.07	451.46	451.34	14.52	**451.37**	451.34	20.23
P-n21-k2	211	224.46	212.71	19.76	224.46	212.71	26.78	**221.08**	212.71	19.48
P-n22-k2	216	**232.08**	217.85	20.64	235.08	217.85	19.50	238.86	217.85	20.64
P-n40-k5	458	511.95	479.57	34.70	**494.45**	461.73	32.77	497.68	461.73	34.44
P-n45-k5	510	579.62	532.08	39.75	**567.87**	519.72	36.09	573.94	**513.42**	34.86
P-n50-k7	554	**608.14**	579.49	39.71	613.40	**570.90**	40.60	613.26	570.92	40.60
P-n55-k7	568	631.24	587.95	43.04	626.62	590.99	62.86	**615.14**	**587.93**	62.20
P-n55-k8	588	**634.73**	**602.34**	42.76	640.29	620.64	45.34	640.38	606.10	69.86
P-n60-k10	744	850.67	797.46	48.18	**835.22**	796.84	67.53	842.01	**786.63**	71.03
P-n65-k10	792	895.39	852.14	48.78	896.80	846.13	74.14	**884.64**	**817.81**	76.88

Tab.4. Simulation results on E datasets by the proposed three streams

Data Sets	BKS	k-mediods + SA			fuzzy c-means + SA			k-means + SA		
		Avg.	Min	AVT/s	Avg.	Min	AVT/s	Avg.	Min	AVT/s
E-n22-k4	375	400.73	375.28	19.59	**389.35**	375.28	21.96	391.76	375.28	19.56
E-n23-k3	569	591.79	568.56	22.67	601.27	568.56	21.39	**587.53**	568.56	20.81
E-n33-k4	835	895.46	843.70	27.19	884.74	845.15	28.45	**881.11**	**841.35**	29.01
E-n101-k14	1077	131731	1257.12	107.79	1324.97	1233.88	107.43	**1317.03**	**1227.66**	75.39

It can also be observed that k-medoids with SA shows poor performance on E datasets, while fuzzy c-means with SA achieves satisfactory performance on average results. The simulation results for the three proposed approaches on the benchmark A datasets are presented in Table 5. In this case, k-means combined with the SA algorithm outperforms the others in both average and minimum outcomes, particularly on larger datasets. The k-medoids with SA demonstrates acceptable performance on average results, while fuzzy c-means with SA also achieves satisfactory outcomes.

5.2. Results comparison

The analysis in the previous subsection indicated that each proposed approach demonstrates superior performance on certain datasets while exhibiting comparatively lower performance on others. In this subsection, we first present a comparative evaluation of the three proposed approaches, followed by a comparison with other clustering-based two-phase methods reported for the CVRP.

5.2.1. Comparison of the proposed three streams

In this subsection, we assess the relative performance of the three proposed approaches using several indicators: overall average across all datasets, dataset-wise wins/draws, success rate, and convergence rate. The first indicator-comprising the average of average results and the average of minimum results for the A, B, P, and E datasets calculated for each approach and presented in Table 6. Bold values denote the best performance among the three approaches. As shown, the k-means algorithm with the SA approach achieves the lowest averages for both metrics across all dataset categories, indicating that k-means with SA outperforms the other two approaches according to this criterion. Actually, each proposed method shows varying performance depending on the dataset criteria, making it challenging to identify the best approach.

Tab.5. Simulation results on A datasets by the proposed three streams

Data Sets	BKS	k-mediods + SA			fuzzy c-means + SA			k-means + SA		
		Avg.	Min	AVT/s	Avg.	Min	AVT/s	Avg.	Min	AVT/s
A-n32-k5	784	**842.39**	**788.74**	37.89	845.31	797.45	28.58	851.45	797.45	39.21
A-n33-k5	661	699.89	662.26	40.19	699.06	662.26	28.25	**696.19**	662.26	40.13
A-n33-k6	742	**771.01**	743.004	27.85	774.10	743.004	26.86	776.90	743.004	26.86
A-n34-k5	778	822.32	786.80	40.13	**813.46**	**780.94**	40.03	830.88	786.80	29.96
A-n36-k5	799	**854.92**	**810.001**	42.90	870.43	818.40	28.75	856.30	815.57	44.33
A-n37-k5	699	727.70	682.41	43.62	725.64	689.19	32.08	**725.21**	**672.47**	44.07
A-n37-k6	949	1019.81	976.95	30.32	1010.58	**952.08**	30.19	**1009.48**	956.81	33.83
A-n38-k5	730	859.37	765.29	31.99	847.11	**744.32**	47.67	**836.84**	768.96	31.56
A-n39-k5	822	908.95	848.59	37.97	**897.29**	**831.50**	44.85	902.75	838.78	32.86
A-n39-k6	831	898.82	853.25	32.10	**893.92**	**836.25**	45.75	911.27	843.63	30.17
A-n44-k6	937	1061.39	979.26	36.86	1044.23	975.10	33.13	**1038.33**	**964.43**	35.72
A-n45-k7	1146	1246.93	1193.80	37.33	1234.23	1181.78	91.93	**1233.69**	**1178.66**	36.99
A-n46-k7	914	1001.00	957.38	36.71	1002.31	936.57	53.83	**981.93**	**922.59**	98.99
A-n53-k7	1010	1212.63	1142.54	42.14	1179.58	**1101.27**	74.45	**1177.63**	1102.18	39.77
A-n54-k7	1167	1351.72	1271.15	40.66	1326.35	1231.61	62.51	**1319.74**	**1224.38**	42.00
A-n55-k9	1073	**1191.32**	**1095.89**	43.89	1199.99	1125.72	44.66	1214.92	1119.18	69.36
A-n60-k9	1354	1520.86	1413.27	48.56	**1504.96**	1432.63	68.22	1513.28	**1393.18**	69.38
A-n62-k8	1288	1497.30	1410.70	47.55	1478.65	1397.24	63.76	**1465.94**	**1393.11**	49.86
A-n63-k10	1314	**1475.27**	1403.92	68.25	1480.61	1408.04	69.31	1481.16	**1390.63**	49.77
A-n64-k9	1401	**1581.40**	1517.82	71.45	1592.09	**1517.71**	72.52	1595.82	1525.48	72.52
A-n80-k10	1763	**2073.41**	2008.63	89.87	2088.34	1958.03	83.79	2079.02	**1947.03**	83.79

Therefore, considering the total number of wins or draws across all datasets can help in selecting the most effective method. Table 7 presents the number of wins and draws for both the average and minimum results across all 46 datasets. The bold values in the table indicate the method that performed the best across most datasets. According to Table 7, the k-means algorithm with the SA approach has the highest number of wins for both average and minimum results on each A, B, P, and E dataset. This suggests that k-means with SA produces the best results in the majority of datasets compared to the other two approaches.

The next performance evaluation metric is the Success Rate (SR), calculated using Eq. (16). The Success Rate represents the win percentage of an algorithm, indicating how often it outperforms others across a given set of data. The algorithm with the highest success rate wins more frequently, reflecting superior performance. Table 8 presents the success rate of each approach on the A, B, P, and E datasets. As shown in Table 8, it can be concluded that the k-means algorithm combined with the SA algorithm achieves the highest success rate across all datasets in both average and minimum result scenarios.

$$\text{Success Rate (SR)} = \frac{\text{Number of win}}{\text{Total number of data sets}} \times 100\% \tag{16}$$

The convergence rate is another critical factor in evaluating an algorithm's performance. The convergence plots for each method on the (a) A-n37-k5, (b) B-n31-k5, (c) P-n21-k2, and (d) E-n22-k4 datasets is analyzed and depicted in Fig. 8. As shown in the figures, it can be seen that in all four subplots, the total cost decreases over the number of iterations, reflecting the algorithms' optimization process. Notably, k-means combined with SA consistently achieves lower total cost values more quickly than the other methods, indicating faster and more effective convergence. For example, in the A-n37-k5 and B-n31-k5 datasets, k-means+SA reaches near-minimal cost significantly earlier than both fuzzy c-means+SA and k-medoids+SA. Similarly, for the P-n21-k2 and E-n22-k4 datasets, k-means+SA demonstrates smoother and more rapid cost reduction, highlighting its superiority in efficiently finding optimal or near-optimal solutions.

Tab.6. Average of average results and the average of minimum results on A, B, P, and E datasets by the proposed three streams

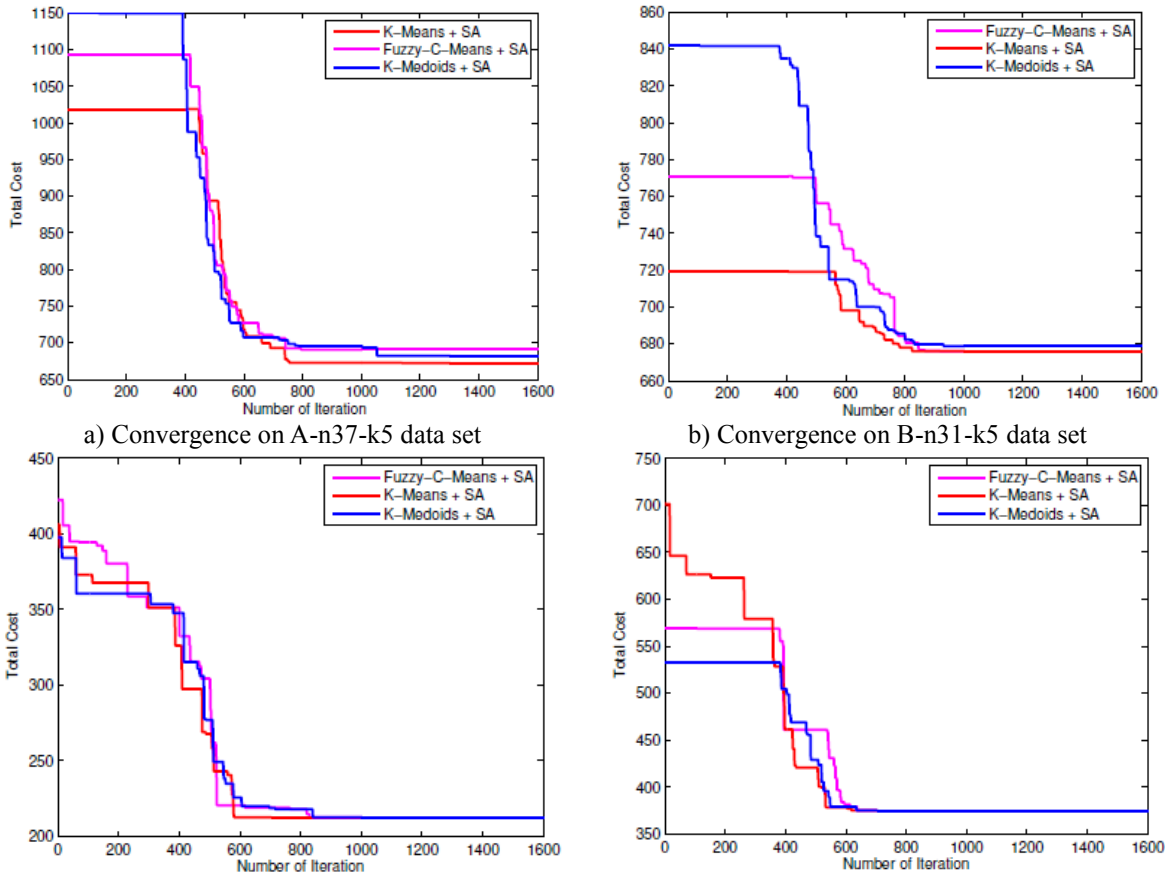
Proposed streams	B Datasets		A Datasets		P Datasets		E Datasets	
	Avg. of Avg.	Avg. of Min.	Avg. of Avg.	Avg. of Min.	Avg. of Avg.	Avg. of Min.	Avg. of Avg.	Avg. of Min.
k-mediods+SA	917.999	872.866	1124.686	1062.46	561.974	531.293	801.32	761.17
fuzzy c-means+SA	918.569	874.529	1119.44	1053.385	558.565	528.885	800.08	755.72
k-means+SA	**916.506**	**868.623**	**1118.987**	**1049.837**	**557.836**	522.644	**794.36**	**753.21**

Tab.7. Number of wins and draws of average results and minimum results on A, B, P, and E datasets by the proposed three streams

Proposed streams	B Datasets				A Datasets				P Datasets				E Datasets			
	Avg.		Min		Avg.		Min		Avg.		Min		Avg.		Min	
	Draw	Win	Draw	Win	Draw	Win	Draw	Win	Draw	Win	Draw	Win	Draw	Win	Draw	Win
k-mediods + SA	0	2	0	1	0	7	2	3	0	3	3	1	0	0	2	0
FCM + SA	0	4	1	3	0	4	2	7	0	3	4	1	0	1	2	0
k-means + SA	0	**5**	1	**6**	0	**10**	2	**9**	0	**4**	4	**4**	0	**3**	2	**2**

Tab.8. Success rate of proposed three streams on A, B, P, and E datasets

Proposed streams	B Datasets		A Datasets		P Datasets		E Datasets	
	Avg.	Min	Avg.	Min	Avg.	Min	Avg.	Min
	SR	**SR**	**SR**	**SR**	**SR**	**SR**	**SR**	**SR**
k-mediods + SA	18.18%	9.09%	33.33%	14.29%	30%	10%	0%	0%
fuzzy c-means+SA	36.36%	27.27%	19.45%	33.33%	30%	10%	25%	0%
k-means + SA	**45.45%**	**54.54%**	**47.62%**	**42.86%**	**40%**	**40%**	**75%**	**50%**



a) Convergence on A-n37-k5 data set

b) Convergence on B-n31-k5 data set

c) Convergence on P-n21-k2 data set

c) Convergence on E-n22-k4 data set

Fig.8. Convergence rate comparison of the proposed three approaches on different datasets

Overall, these results suggest that integrating SA with k-means provides the most effective convergence behavior among the compared algorithms. Therefore, based on the above-discussed performance evaluation metrics, it can be decided that the k-means with the SA algorithm is the most effective solution stream for the CVRP among the proposed three streams.

5.2.2. Comparison with other clustering-based algorithms

In this section, we compare our proposed best-stream approach, k-means+SA, with other benchmark clustering-based two-phase methods for CVRP problems. Specifically, the simulation results of our k-means+SA method are compared with those reported in the reference literature. The performance comparison of our k-means+SA method with others is presented in Tables 9 to 11. In these tables, the bolded values indicate the best-performing algorithms. Additionally, the average route length across all data sets for each algorithm is calculated and shown at the bottom of the tables, along with the success rate for each algorithm.

Table 9 demonstrates a comparison of our proposed k-means+SA algorithm with five other standard algorithms: Adaptive Sweep (AS)+VTPSO [16], Variant Sweep (VS)+VTPSO [11], Sweep+MCA [15], DSN+PSO [17], and Adaptive Sweep (AS)+VTPSO [12], on benchmark A data sets. According to Table 9, it can be seen that the proposed k-means+SA outperforms the other five standard algorithms on nearly all A datasets, with the exception of four: A-n34-k5, A-n45-k7, A-n53-k7, and A-n80-k10. Besides that, the average over all 21 datasets of our proposed k-means+SA is 1049.83, which is smaller than the 1124.952 of AS+VTPSO, 1129.380 of VS+VTPSO, 1134.667 of Sweep+MCA, 1141.85 of DSN+PSO, and 1124.95 of AS+VTPSO. In addition, the success rate of our proposed k-means+SA is 80.95%, which is significantly better than the 9.52% of AS+VTPSO, 9.52% of VS+VTPSO, 14.29% of Sweep+MCA, 4.76% of DSN+PSO, and 9.52% of AS+VTPSO. Therefore, it can be said that our proposed k-means+SA is a better solution approach compared to the five other approaches on the datasets.

Table 10 shows a comparison of our proposed

k-means+SA algorithm with three other standard algorithms: Adaptive Sweep (AS)+VTPSO [16], Adaptive Sweep (AS)+VTPSO [12], and Variant Sweep (VS)+VTPSO [11], on benchmark P datasets. As shown in Table 10, it can be observed that the proposed k-means+SA outperforms the other three standard algorithms on nearly all P datasets, with the exception of three: P-n21-k2, P-n22-k2, and P-n55-k8. Besides that, the average over all 10 datasets of our proposed k-means+SA is 522.644, which is lower than the 543.6 of AS+VTPSO, 543.6 of AS+VTPSO, and 540.1 of VS+VTPSO. In addition, the success rate of our proposed k-means+SA stands at 70%, significantly higher than the 30% of AS+VTPSO, 30% of AS+VTPSO, and 30% of VS+VTPSO. Therefore, it can be concluded that the k-means+SA algorithm is a superior solution approach compared to the three other approaches on P datasets.

Performance comparison between our proposed k-means+SA and FCM+Improved ACO [18] on the A and E datasets is shown in Table 11. As presented in Table 11, it can be seen that the proposed k-means+SA significantly dominates FCM+Improved ACO on all considered data sets except for A-n38-k5. Overall, the average route length of our proposed k-means+SA is 715.3, which is shorter than the 720.29 achieved by FCM+Improved ACO. Moreover, the success rate of our proposed k-means+SA stands at 85.71%, considerably higher than the 14.29% success rate of FCM+Improved ACO. Therefore, it can be pointed out that the k-means+SA algorithm is a more effective approach than FCM+Improved ACO on the A and E datasets. Based on the discussion and analysis above, our proposed k-means+SA algorithm consistently delivers superior results in terms of data-specific performance, average route length, and success rate. Therefore, k-means+SA can be considered a more effective solution approach for the CVRP problem compared to other state-of-the-art clustering-based two-phase algorithms. To further explore the differences between the performance of our algorithm and the comparable methods, a detailed statistical analysis is provided in the next subsection.

5.3. Statistical analysis

To validate the superior performance of our proposed k-means+SA method, we conducted a pair-wise hypothesis test known as the "Wilcoxon Signed-Rank Test" [31]. In this test, the null hypothesis (H_0) and the alternative hypothesis (H_1) are defined as follows:

Tab.9. Performance comparison of our proposed k-means+SA with AS+VTPSO, VS+VTPSO, Sweep+MCA, DSN+PSO, AS+VTPSO on A datasets

Data Sets	BKS	AS+VTPSO [16]	VS+VTPSO [11]	Sweep+MCA [15]	DSN+PSO [17]	AS+VTPSO [12]	Proposed k-means + SA
A-n32-k5	784	882	882	882	870	882	**797.45**
A-n33-k5	661	698	698	690	720	698	**662.26**
A-n33-k6	742	751	751	751	867	751	**743.004**
A-n34-k5	778	**785**	**785**	**770**	859	**785**	786.80
A-n36-k5	799	881	881	879	887	881	**815.57**
A-n37-k5	669	754	739	739	895	754	**672.47**
A-n37-k6	949	1112	1097	998	1077	1112	**956.81**
A-n38-k5	730	813	813	813	856	813	**768.96**
A-n39-k5	822	877	877	860	960	877	**838.78**
A-n39-k6	831	972	969	969	893	972	**843.63**
A-n44-k6	937	1056	1056	1012	1037	1056	**964.43**
A-n45-k7	1146	1305	1343	**1043**	1264	1305	1178.66
A-n46-k7	914	975	990	987	1087	975	**922.59**
A-n53-k7	1010	**1090**	**1090**	**1091**	1293	**1090**	1102.18
A-n54-k7	1167	1361	1361	1278	1248	1361	**1224.38**
A-n55-k9	1073	1190	1201	1900	1225	1190	**1119.18**
A-n60-k9	1354	1503	1503	1510	1482	1503	**1393.18**
A-n62-k8	1288	1408	1501	1498	1403	1408	**1393.11**
A-n63-k10	1314	1477	1446	1461	1512	1477	**1390.63**
A-n64-k9	1401	1598	1598	1577	1654	1598	**1525.48**
A-n80-k10	1763	2136	2136	2120	**1890**	2136	1947.03
Average		1124.95	1129.380	1134.667	1141.857	1124.952	**1049.837**
Success Rate (SR)		9.52%	9.52%	14.29%	4.76%	9.52%	**80.95%**

Tab.10. Performance comparison of our proposed k-means+SA with AS+VTPSO [16], AS+VTPSO [12], VS+VTPSO [11] on P datasets

Data Sets	BKS	AS+VTPSO [16]	AS+VTPSO [12]	VS+VTPSO [11]	Proposed k-means + SA
P-n16-k8	450	549	549	553	**451.34**
P-n21-k2	211	**211**	**211**	**211**	212.71
P-n22-k2	216	**216**	**216**	**216**	217.85
P-n40-k5	458	483	483	474	**461.73**
P-n45-k5	510	524	524	523	**513.42**
P-n50-k7	554	583	583	579	**570.92**
P-n55-k7	568	595	595	593	**587.93**
P-n55-k8	588	**586**	**586**	**585**	606.10
P-n60-k10	744	830	830	830	**786.63**
P-n65-k10	792	859	859	837	**817.81**
Average		543.6	543.6	540.1	**522.644**
Success Rate (SR)		30%	30%	30%	**70%**

Tab.11. Performance comparison between our proposed k-means+SA and FCM+Improved ACO [18] on A and E datasets

Data Sets	BKS	FCM+Improved ACO [18]	Proposed k-means + SA
A-n32-k5	784	810	**797.45**
A-n36-k5	799	825	**815.57**
A-n38-k5	730	**746**	768.96
A-n39-k6	831	849	**843.63**
En-22-k4	375	385	**375.28**
En-23-k3	569	576	**568.56**
En-33-k4	835	851	**837.93**
Average		720.29	**715.34**
Success Rate (SR)		14.29%	**85.71%**

H_0: The data are from the same distribution;
H_1: The data are from different distributions.

The results of this hypothesis test are presented in Table 12. The table clearly shows that the P-values for comparisons between Proposed k-means+SA and AS+VTPSO, Sweep+MCA, AS+VTPSO, VS+VTPSO, and DSN+PSO are all smaller than the significance level alpha. This indicates that in these five cases, the null hypothesis is rejected, suggesting significant differences between our proposed method and the compared methods. This implies that our results are significantly superior in these cases. However, for the comparison between proposed k-means+SA and FCM+Improved ACO, the p-value is greater than the significance level alpha, indicating that the null hypothesis cannot be rejected. This suggests that there are no significant differences between our results and those of FCM+Improved ACO, despite our proposed method having a better success rate and average performance. Overall, the statistical analysis confirms that the performance of our proposed k-means+SA method is significantly better than that of other standard clustering-based two-phase algorithms. Since the proposed k-means+SA algorithm demonstrates superior performance in terms of average route length and overall success rate, managers can leverage this approach to achieve more efficient vehicle routing decisions. Shorter and optimized routes directly translate into lower fuel consumption, reduced transportation costs, and better fleet utilization.

Tab.12. Statistical test results

Methods	Z	P-Value	Alpha
Proposed k-means+SA vs AS+VTPSO [16]	-4.3896	0.000011	0.05
Proposed k-means vs Sweep+MCA [15]	-3.1803	0.0015	0.05
Proposed k-means+SA vs AS+VTPSO [12]	-4.3896	0.000011	0.05
Proposed k-means+SA vs VS+VTPSO [11]	-4.3700	0.000012	0.05
Proposed k-means+SA vs DSN+PSO [17]	-3.8755	0.0001	0.05
Proposed k-means+SA vs FCM+Improved ACO [18]	-	0.2969	0.05

Moreover, the higher success rate of the algorithm ensures greater reliability in planning, which can improve service quality and customer satisfaction. From a strategic perspective, adopting such a robust optimization technique allows managers to handle complex distribution networks more effectively, maintain competitiveness, and support sustainable operations by minimizing unnecessary mileage and emissions.

6. Conclusion

We have established a two-phase approach to address the CVRP by integrating clustering techniques with an intelligent search method. The first phase generates an initial or feasible solution, while the second phase focuses on optimizing this solution. To generate the initial solution, we employ three clustering algorithms: fuzzy c-means, k-means, and k-medoids, combined with a nearest neighbor-based approach. In the second phase, we use Simulated Annealing (SA) as an intelligent search technique to optimize the initial solution. This leads to three solution variants: fuzzy c-means with SA, k-means with SA, and k-medoids with SA. Simulation results on benchmark CVRP datasets indicate that k-means with SA performs the best among our three proposed methods. Moreover, when compared to other approaches, our k-means with SA method significantly outperforms state-of-the-art clustering-based two-phase approaches. While the proposed method demonstrates strong performance, the study is limited to benchmark CVRP datasets and does not account for additional real-world constraints such as time windows, heterogeneous fleets, or dynamic demands. Furthermore, only three clustering algorithms and one intelligent search technique were tested, which may limit generalizability to other VRP variants or larger-scale problems. In the future, we plan to extend this two-phase framework to more complex VRP variants, such as the VRP with time windows, split deliveries, or stochastic demands. Incorporating other intelligent optimization methods-particularly swarm intelligence-based algorithms combined with clustering strategies may yield further improvements. Additionally, testing the approach on large-scale, real-world datasets and analyzing runtime performance will enhance its practical applicability.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The publicly available CVRP data sets were used and analyzed in this paper, which can be found in reference [30].

References

[1]. Dantzig, G. B., & Ramser, J. H., "The truck dispatching problem", Management science, Vol. 6, No. 1, (1959), pp. 80-91. https://doi.org/10.1287/mnsc.6.1.80.

[2]. Sar, K., & Ghadimi, P., "A systematic literature review of the vehicle routing problem in reverse logistics operations", Computers & Industrial Engineering, Vol. 177, (2023), pp. 109011. https://doi.org/10.1016/j.cie.2023.109011.

[3]. Sharma, S. K., Routroy, S., & Yadav, U., "Vehicle routing problem: recent literature review of its variants", International Journal of Operational Research, Vol. 33, No. 1, (2018), pp. 1-31. https://doi.org/10.1504/IJOR.2018.094229.

[4]. Dubois, F., Renaud-Goud, P., & Stolf, P., "Capacitated vehicle routing problem under deadlines: An application to flooding crisis", IEEE Access, Vol. 10, (2022), pp. 45629-45642. https://doi.org/10.1109/ACCESS.2022.3170446.

[5]. Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y., "Development of a fuel consumption optimization model for the capacitated vehicle routing problem", Computers & Operations Research, Vol. 39, No. 7, (2012), pp. 1419-1431. https://doi.org/10.1016/j.cor.2011.08.013.

[6]. Ralphs, T. K., Kopman, L., Pulleyblank, W. R., & Trotter, L. E., "On the capacitated vehicle routing problem", Mathematical Programming, Vol. 94, (2003), pp. 343-359. https://doi.org/10.1007/s10107-002-0323-0.

[7]. Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P., "Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification", Operational Research, Vol. 22, (2020), pp. 2033-2062. https://doi.org/10.1007/s12351-020-00600-7.

[8]. Dunn, J. C., "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics, Vol. 3, No. 3, (1973), pp. 32–57. https://doi.org/10.1080/01969727308546046.

[9]. Hartigan, J. A., & Wong, M. A., "A K-means clustering alrorithm", Appl. Stat, Vol. 28, (1979), pp. 126-130. https://cir.nii.ac.jp/crid/1572543025507658752.

[10]. Kaufman, L., & Rousseeuw, P. J., "Finding groups in data: an introduction to cluster analysis", John Wiley & Sons, 1990. https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801.

[11]. Akhand, M. A. H., Peya, Z. J., Sultana, T., & Rahman, M. H., "Solving capacitated vehicle routing problem using variant sweep and swarm intelligence", Journal of Applied Science and Engineering, Vol. 20, No. 4, (2017), pp. 511-524. https://doi.org/10.6180/jase.2017.20.4.13.

[12]. Peya, Z. J., Akhand, M. A. H., & Murase, K., "Capacitated vehicle routing problem solving through adaptive sweep-based clustering plus swarm intelligence based route optimization", Oriental Journal of Computer Science and Technology, Vol. 11, No. 2, (2018), pp. 88-102. http://dx.doi.org/10.13005/ojcst11.02.04.

[13]. Wang, X., & Zhu, H., "Two-stage Algorithm for Capacitated Vehicle Routing Problem", Journal of Engineering Science & Technology Review, Vol. 11, No. 2, (2018), pp. 111-120. https://api.semanticscholar.org/CorpusID:201763239.

[14]. Comert, S. E., Yazgan, H. R., Kır, S., & Yener, F., "A cluster first-route second approach for a capacitated vehicle routing problem: a case study", International Journal of Procurement Management, Vol. 11, No. 4, (2018), pp. 399-419. https://doi.org/10.1504/IJPM.2018.092766.

[15]. Mahmood, N., "Solving Capacitated Vehicle Routing Problem Using Meerkat Clan Algorithm", The International Arab Journal of Information Technology (IAJIT), Vol. 19, No. 4, (2022), pp. 689-694. https://doi:10.34028/iajit/19/4/14.

[16]. Akhand, M. A. H., Zahrul, J. P., & Murase, K., "Capacitated vehicle routing problem solving using adaptive sweep and velocity tentative PSO", International Journal of Advanced Computer Science and Applications, Vol. 8, No. 12, pp. 288-295. https://doi.org/10.14569/IJACSA.2017.081237.

[17]. Peya, Z. J., Akhand, M. A. H., Sultana, T., & Rahman, M. H., "Distance-based sweep nearest algorithm to solve capacitated vehicle routing problem", International Journal of Advanced Computer Science and Applications, Vol. 10, No. 10, (2019), pp. 259-264. https://doi.org/10.14569/IJACSA.2019.0101036.

[18]. YAVŞAN, E., & Ilhan, I., "An Ant Colony Optimization Based Real-time Mobile Application for the Capacitated Vehicle Routing Problem", Hittite Journal of Science and Engineering, Vol. 9, No. 4, (2022), pp. 263-273. https://doi.org/10.17350/HJSE19030000279.

[19]. Hamta, N., & Rabiee, S. "Open Vehicle Routing Problem with Robust Optimization Approach", International Journal of Industrial Engineering & Production Research (IJIEPR), Vol. 32, No. 3, (2021), pp. 1-20. http://ijiepr.iust.ac.ir/article-1-1015-en.html.

[20]. Abbal, K., El Amrani, M., & Benadada, Y. "An Accelerated Benders Decomposition Approach for the Multi-Level Multi-Capacitated Facility Location Problem", International Journal of Industrial Engineering & Production Research (IJIEPR), Vol. 36, No. 1, (2025), pp. 33-44. http://ijiepr.iust.ac.ir/article-1-2196-en.html.

[21]. Young, GU., & Di, LIU. "Adaptive Hybrid Ant Colony Optimization for Capacitated Vehicle Routing Problem", Journal of Northeastern University (Natural Science), Vol. 44, No. 12, (2024), pp. 1686-1695. https://10.12068/j.issn.1005-3026.2023.12.003.

[22]. Fitzpatrick, J., Ajwani, D., & Carroll, P. "A scalable learning approach for the capacitated vehicle routing problem", Computers & Operations Research, Vol. 171, (2024), pp. 106787. https://doi.org/10.1016/j.cor.2024.106787.

[23]. Muriyatmoko, D., Djunaidy, A., & Muklason, A. "Heuristics and metaheuristics for solving capacitated vehicle routing problem: An algorithm comparison", Procedia Computer Science, Vol. 234, (2024), pp. 494-501. https://doi.org/10.1016/j.procs.2024.03.032.

[24]. Ekayanti, E., & Efendi, I. B. "Capacitated Vehicle Routing Problem (CVRP) with Sweep and Nearest Neighbor Algorithm",

Sinergi International Journal of Logistics, Vol. 2, No. 1, (2024), pp. 17-29. https://doi.org/10.61194/sijl.v2i1.187.

[25]. Adewumi, A. O., & Adeleke, O. J., "A survey of recent advances in vehicle routing problems", International Journal of System Assurance Engineering and Management, Vol. 9, (2018), pp. 155-172. https://link.springer.com/article/10.1007/s13198-016-0493-4.

[26]. Demircan, S., & Kahramanli, H., "Application of fuzzy C-means clustering algorithm to spectral features for emotion classification from speech", Neural Computing and Applications, Vol. 29, (2018), pp. 59-66. https://link.springer.com/article/10.1007/s00521-016-2712-y.

[27]. Chang, W., Ji, X., Liu, Y., Xiao, Y., Chen, B., Liu, H., & Zhou, S., "Analysis of university students' behavior based on a fusion K-means clustering algorithm", Applied Sciences, Vol. 10, No. 18, (2020), pp. 6566. https://doi.org/10.3390/app10186566.

[28]. Reynolds, A. P., Richards, G., de la Iglesia, B., & Rayward-Smith, V. J., "Clustering rules: a comparison of partitioning and hierarchical clustering algorithms", Journal of Mathematical Modelling and Algorithms, Vol. 5, (2006), pp. 475-504. https://link.springer.com/article/10.1007/s10852-005-9022-1.

[29]. Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P., "Optimization by simulated annealing", Science, Vol. 220, No. 4598, pp. 671-680. https://www.science.org/doi/abs/10.1126/science.220.4598.671.

[30]. https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old. Accessed on August 13, 2024.

[31]. İlhan, İ., & Gökmen, G., "A list-based simulated annealing algorithm with crossover operator for the traveling salesman problem", Neural Computing and Applications, Vol. 34, (2022), pp. 7627-7652. https://link.springer.com/article/10.1007/s00521-021-06883-x.

Follow this article at the following site:

Md. Rafsan Islam, Md. Azizur Rahman, Kazi Mohammad Nazib, Lasker Ershad Ali. "Clustering-Based Intelligent Route Optimization Algorithm for Capacitated Vehicle Routing Problem" IJIEPR 2025; 36 (3): 169-185
URL: http://ijiepr.iust.ac.ir/article-1-2116-en.html