Universiteit Leiden
The Netherlands

Cluster-centric local search strategies for enhanced multi-objective logistics optimization

Liu, W.; Bäck, T.H.W.; Fan, Y.

Citation

Liu, W., Bäck, T. H. W., & Fan, Y. (2024). Cluster-centric local search strategies for enhanced multi-objective logistics optimization. *2024 Ieee Congress On Evolutionary Computation (Cec)*, 1-8. doi:10.1109/CEC60901.2024.10612182

Version:	Publisher's Version
License:	Licensed under Article 25fa Copyright Act/Law (Amendment Taverne)
Downloaded from:	https://hdl.handle.net/1887/4254662

Note: To cite this publication please use the final published version (if applicable).

Cluster-centric Local Search Strategies for Enhanced Multi-Objective Logistics Optimization

Wei Liu, Thomas Bäck, Yingjie Fan
LIACS, Leiden University, Netherland
w.liu@liacs.leidenuniv.nl, t.h.w.baeck@liacs.leidenuniv.nl, y.fan@liacs.leidenuniv.nl

2024 IEEE Congress on Evolutionary Computation (CEC) | 979-8-3503-0836-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/CEC60901.2024.10612182

***Abstract*—Solving multi-objective vehicle routing problem with time windows (VRPTW) is a challenge in last-mile logistics. While traditional methods have predominantly focused on minimizing travel distances and vehicle numbers, recent innovations acknowledge the need for a more refined balance across various objectives. These advancements incorporate a range of heuristic algorithms, blending hybrid techniques that merge genetic algorithms with local search strategies, and employing advanced methodologies such as particle swarm optimization and multi-objective evolutionary algorithms based on decomposition (MOEA/D). Such developments signify a shift towards more effective solutions for VRPTW.**

In response to these challenges, this paper introduces a novel approach that integrates MOEA/D with K-means clustering for generating efficient solutions. In our approach, we refine the tri-objective VRPTW efficiency by establishing a criterion that prioritizes customer sequences based on their spatial proximity and angular alignment relative to cluster centers and depot, utilizing an integrated metric of distance and cosine angles, which can enhance the local search process and effectively balance the three objectives in VRPTW. Experimental results on Solomon's 56 VRPTW 100-customer dataset instances demonstrate the superiority of our approach.

***Index Terms*—Vehicle routing problem with time windows (VRPTW), K-means clustering, MOEA/D, Local search, Multi-objective optimization.**

I. Introduction

THE Vehicle Routing Problem with Time Windows (VRPTW) [1] stands as a pivotal challenge in logistics, emphasizing the optimization of vehicle routes within specific customer time constraints for maximal satisfaction and cost efficiency. This problem is critical not only in logistics and dispatching but also plays a significant role in diverse areas such as traffic control, supply chain management, power scheduling, and communication networks [2] [3] [4].

In the typical framework of VRPTW, the primary focus is often on minimizing travel distance and vehicle numbers as a single-objective [5] [6] [7]. However, decision-makers in real-world scenarios frequently face multiple conflicting goals that extend beyond these traditional parameters [8]. The inherent challenge lies in the potential conflict among these objectives: optimizing one aspect may lead to compromises in another. Consequently, VRPTW emerges as a complex multi-objective optimization problem (MOP), demanding a nuanced approach to achieve an optimal balance among these varied and often competing objectives.

In light of these complexities, heuristic algorithms have been increasingly employed to provide effective solutions to multi-objective VRPTW problems. A notable example is the use of a hybrid approach combining of genetic algorithms and local search techniques. For instance, Tan's research [9] introduces a hybrid multiobjective evolutionary algorithm (HMOEA) that uses heuristic methods for local exploitation of VRPTW. Gong et al. [10] proposed a Set-based Particle Swarm Optimization algorithm for vrptw (S-PSO-VRPTW), which innovatively adapts the PSO framework for discrete com-

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

binatorial optimization. Zhang et al. [11] presents HMOEA-GL aiming to reduce vehicle numbers and minimize delivery time waste due to early arrivals. Das et al. [12] presents an drone-truck synchronization model for efficient parcel delivery, utilizing a new Collaborative Pareto Ant Colony Optimization algorithm to optimize travel costs and customer service level in terms of timely deliveries. Qi et al. [13] enhances the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [14] with a novel selection operator and three local search methods. Legrand et al [15] presents an enhanced Local Search (LS) algorithm within an MOEA/D for a bi-objective VRPTW, incorporating novel neighborhood exploration strategies and metrics, resulting in improved performance. Lan et al. [16] introduce D-VND algorithm, which integrates MOEA/D with Variable Neighborhood Descent for tri-objective VRPTW.

Reflecting on the above discussed algorithms, especially the application of MOEA/D in VRPTW, we find that the combination of MOEA/D with local search algorithms holds significant advantages in addressing multi-objective VRPTW. Therefore, in this paper, we propose integrating MOEA/D with machine learning techniques to further enhance the local search capabilities of the algorithm. Our approach focuses on refining the tri-objective VRPTW by prioritizing customer visit sequences based on spatial proximity and angular alignment relative to cluster centers and the depot. By utilizing a novel integrated metric combining distance and cosine angles, we enhance the local search process, achieving more balanced and efficient solutions to the complex challenges of VRPTW.

Major contributions of this work are as follows:

1) We propose a novel heuristic algorithm that integrates MOEA/D with a Cluster-Based Sort (CBS) mechanism termed MOEA/D-CBS, offering a unique approach in the local search process for solving MO-VRPTW.
2) We establish criteria to reorder customers based on their proximity to and alignment with cluster centers. This method enhances the efficiency of our solution for the tri-objective MO-VRPTW by prioritizing customer sequences through a combination of distance metrics and cosine angle measurements.

The remainder of this paper is structured as follows: Section II introduces a tri-objective VRPTW. Section III delves into details of our innovative algorithm. Section IV presents the experimental results. Finally, in Section V, we draw conclusions and discuss future perspectives for this research.

II. Problem Statement

In this paper, our goal is to develop an innovative algorithm that provides superior solutions for multi-objective VRPTW problems. We use our model for solving the model proposed in [16]. Given a fleet of vehicles with capacity constraints and a set of customers, each with a known demand and a time window within which the delivery should be made, the objective is to plan routes that simultaneously minimize the total travel cost (see (1)), the number of vehicles (see (2)), and the overall waiting time of drivers due to early arrivals (see (3)).

The first objective is to minimize the overall travel cost of all vehicles:

$$\min f_1 = \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} c_{ij} x_{ij}^k \tag{1}$$

where c_{ij} represents the travel cost from customer i to customer j. In this model, c_{ij} refers to the travel distance between customers i and j. The binary decision variables x_{ij}^k have a value of 1 when vehicle k is assigned to the delivery task from i to j.

The number of vehicles is minimized in the second objective:

$$\min f_2 = \sum_{k=1}^{K} \sum_{j=1}^{n} x_{0j}^k \tag{2}$$

where x_{0j}^k is a binary decision variable that equals 1 if vehicle k starts from the depot to service customer j as the first customer, and 0 otherwise. Here, n is the number of customers.

The third objective is to minimize drivers' waiting time:

$$\min f_3 = \sum_{i=1}^{n} w_i \tag{3}$$

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

where w_i denotes the waiting time of a deliverer at customer i's location. Since we use the same model as presented in [16], we omit additional model-related constraints and details.

III. The Propoesed algorithm - MOEA/D-CBS

The MOEA/D-CBS algorithm includes the initialization, solution regeneration, and environmental selection processes. The specific workflow of the algorithm is as follows.

1. Initialization

First, the chromosome representation method and the random insertion method proposed by Tan et al. [9] are used to initialize the population. Then, we use weight vectors to divide the VRPTW problem into M distinct sub-problems. For each subproblem, we calculate the T nearest weight vectors using euclidean distance. Concurrently, the necessary reference point Z^* for Chebyshev distance calculations is computed. The solutions generated are then assigned to respective sub-problems, setting the stage for mutation and search processes within these sub-problems. Then, for each subproblem, we conduct the GA operator (Crossover and Mutation) and local search to regenerate the solutions.

2. Solution regeneration

1. Crossover and Mutation: In our approach, adhering to the methodology delineated in [17], the crossover operation involves exchanging routes in each solution, while the mutation process entails swapping customers within a route in each solution. These processes are designed to optimize routing configurations and introduce variability for each solution, all while ensuring compliance with the time window constraints of the VRPTW.

2. Local Search: The Cluster-Based Search (CBS) algorithm is employed for the regeneration of solution allocated for each sub-problem. The detailed process is described in Algorithm 1. Initially, within a solution for each sub-problem, routes are sorted based on their efficacy (line 2). Subsequently, poor-performing routes are selected based on the predetermined parameter 'ReclusterRatio' δ (line 3), followed by the selection of better-performing routes (line 4). Customers from underperforming routes are then placed into the archive for route reselection (line 5). Routes are reconstructed by the proposed methods (line 6). The final step involves merging the newly generated routes with the previously identified elite routes, thereby updating the solution (line 7).

Cluster and Assign Process: Algorithm 2 details the process for reconstructing routes for customers in the archive. Initially, the data in the archive (coordinates and time windows) undergo processing, involving normalization (line 10) and weight allocation (line 11).

Algorithm 1: CBS algorithm

```
  Input:  - Solution: A solution comprising multiple routes
  - δ: Ratio of routes to re-cluster
  Output: Updated solution with re-clustered routes
1 Function ReclusterAndSort(Solution, ReclusterRatio):
2     Sort routes in solution by distance-to-customer ratio;
3     Select routes for re-clustering according to δ;
4     Rretain the rest as elite_routes;
5     Merge customers from selected routes into an archive;
6     Reconstruct routes by clustering customers in archive;
7     Combine elite_routes with reconstructed routes to form new_solution;
8     return new_solution;
```

Normalization: The solution data in the archive is normalized, aligning customer coordinates and time windows to a consistent scale.

Weight Allocation: Next, the dataset is adjusted with the weights assigned to each corresponding sub-problem. This process accentuates crucial data attributes, like spatial coordinates and time windows, underscoring their importance in different

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2: Cluster and Assign

```
   Input:  - Archive: Set of unassigned customers
   - Depot: Central depot location
   Output: new_routes set R
1  Function Cluster_and_assign(archive, depot):
2      while archive is not empty do
3          new_route, archive = cluster_and_sort(depot, archive);
4          if new_route is valid then
5              append new_route to new_routes set R;
6          end
7      end
8      return new_routes Set;
9  Function cluster_and_sort(depot, archive):
10     Normalize the solution in archive;
11     Weight allocation to normalized data;
12     Finding clustering center using K-means algorithm;
13     Sorting customers based on criterion (7);
14     Inserting customers into route and if not inserted, putting infeasible customers into archive;
15     return updated route, archive;
```

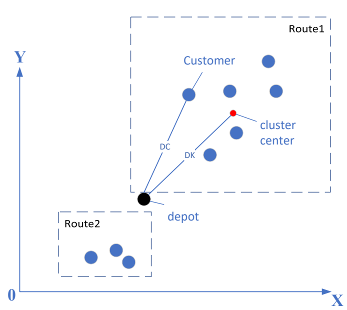

Fig. 1: Illustration of Cluster-Based Sort
Note: Blue circles represent customers, red points represent cluster centers, and the black point represent the depot.

objectives to be optimized, which is vital for establishing unique clustering centers for each subproblem.

After data processing, we use the K-means algorithm to identify clustering centers within the data (line 12). Then we use the following criteria to start the customer selection process (line 13).

Let $\mathbf{C}_i$ be the coordinates of customer i, $\mathbf{D}$ be the coordinates of the depot, and $\mathbf{K}$ be the coordinates of the cluster center.

We use $\mathbf{V}_{DC}$ to indicate the vector from the depot to customer C:

$$\mathbf{V}_{DC} = \mathbf{C} - \mathbf{D} \tag{4}$$

while $\|\mathbf{V}_{DC}\|$ is the Euclidean distance between the depot and customer i. We use $\mathbf{V}_{DK}$ to indicate the vector from the depot to a cluster center $\mathbf{K}$:

$$\mathbf{V}_{DK} = \mathbf{K} - \mathbf{D} \tag{5}$$

while $\|\mathbf{V}_{DK}\|$ is the Euclidean distance between the depot and the cluster center. Let θ be the angle between $\mathbf{V}_{DC}$ and $\mathbf{V}_{DK}$. The cosine similarity of θ is calculated as

$$\cos(\theta) = \frac{\mathbf{V}_{DC} \cdot \mathbf{V}_{DK}}{\|\mathbf{V}_{DC}\| \times \|\mathbf{V}_{DK}\|} \tag{6}$$

where $\mathbf{V}_{DC} \cdot \mathbf{V}_{DK} = (\Delta x_i, \Delta y_i) \cdot (\Delta x_k, \Delta y_k) = \Delta x_i \times \Delta x_k + \Delta y_i \times \Delta y_k$, given that $\mathbf{V}_{DC}$ is $(\Delta x_i, \Delta y_i)$ and $\mathbf{V}_{DK}$ is $(\Delta x_k, \Delta y_k)$.

The customers are sorted based on a criterion that combines both the cosine value and the distance:

$$\text{sort}\left(\text{customers},\ \text{key} = \frac{1}{\frac{\cos(\theta)}{\|\mathbf{V}_{DC}\| + \varepsilon}}\right) \tag{7}$$

where ε represents a very small positive number, used to prevent the denominator from becoming zero. Specifically, customers are sorted in ascending order of the reciprocal of the ratio of the cosine to the distance (plus a small constant). The vectors $\mathbf{V}_{DC}$ and $\mathbf{V}_{DK}$ correspond to DC and DK in Fig. 1, respectively. This sorting approach prioritizes customers who are closely aligned with the direction from the depot to the cluster center and have shorter distances to the depot. In this way, by leveraging cluster centers to consider both

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

directional alignment and geographical proximity, a search for viable solutions can be conducted to reduce the overall travel cost.

Route Construction and Archive Update: Customers are inserted into routes based on the sorting (line 14 of Algorithm 2). Customers who are infeasible assigned to a route are returned to the archive for recluster. The iterative process continues until the archive no longer contains any customers.

3. Environmental selection

In the environmental selection phase, we merge solutions generated from each sub-problem to form the solution set then it combines with the parent population, using NSGA-II [18] to retain superior solutions. The algorithm then cycles between solution regeneration and selection, based on termination criteria, ensuring thorough exploration of the solution space.

IV. Experimental Studies

In this section, we introduce the test datasets and performance metrics used in our computational experiments. Next, we outline the experimental setup employed to assess the algorithms. Finally, we evaluate the tri-objective Inverted Generational Distance (IGD) values [19], comparing the performance of MOEA/D-CBS against the latest multi-objective algorithm, D-VND [16], and a specialized variant of MOEA/D that has been specifically developed for our algorithm.

A. Test Instance and Performance Metrics

1. Solomon Test Instance: We employ Solomon's well-known benchmark [20], consisting of 100 customers, as our test instances. These instances are categorized into C1, C2, R1, R2, RC1, and RC2 based on customer location and time window patterns. R-type and C-type instances have randomly and clustered located customers, respectively. RC-type instances combine both. Tighter time windows and fewer customers characterize R1, C1, and RC1, while R2, C2, and RC2 have extended time windows, allowing more customers per vehicle.

2. IGD Performance Metrics: IGD represents the average minimum distance from the true Pareto Front (PF) to the approximated PF. A superior algorithm results in a smaller IGD value. In equation 8, P^* and P denote the true PF and the approximated PF, respectively. IGD is computed as follows:

$$IGD(P, P^*) = \frac{1}{|P^*|} \sum_{v \in P^*} \text{dist}(v, P) \qquad (8)$$

Here, $\text{dist}(v, P)$ refers to the minimum Euclidean distance from point v to all points in P. Ideally, points in P should be evenly distributed around the true PF. However, since the true PF is unknown in VRPTW, the reference point (P^*) is derived from the best solutions obtained across all algorithms after 20 independent runs.

B. Experiment setting

The MOEA/D-CBS algorithm is tested on the Solomon's instances. Since both MOEA/D-CBS and D-VND are based on the MOEA/D framework and primarily differ in their local search processes, the parameter settings for MOEA/D-CBS have been aligned with those of D-VND for consistent comparison. Especially, the different values of δ represent the number of routes that need to be reordered, and based on empirical values, the value of δ has been adopted 80% in this paper. The proposed algorithm was implemented in Python 3.7 on a PC with an Intel Core i9-14900HX CPU and 16 GB RAM. The experimental settings are as follows,

- Number of sub-problems N: 200,
- Neighborhood size T: 10,
- Maximum number of iterations: 200,
- Crossover rate: 1,
- Mutation rate: 0.1,
- ReclusterRatio: 80%.

C. Tri-objective IGD Values Comparison

In our comparative analysis, we evaluate three algorithms in addressing the tri-objective VRPTW: MOEA/D-CBS, D-VND, and MOEA/D. The algorithms analyzed include MOEA/D without local search, D-VND with VND local search, and CBS (MOEA/D-CBS), which uses a clustering-based

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

TABLE I

IGD Performance Comparison of MOEA/D-CBS, D-VND, and MOEA/D in Tri-Objective VRPTW

INSTANCES	MOEA/D-CBS mean(std dev)	D-VND mean(std dev)	MOEA/D mean(std dev)
R101	**1.62E+1(6.50E+01)**	1.67E+1(2.74E+0)≈	2.04E+1(2.31E+0)-
R102	**2.04E+1(4.55E+0)**	2.49E+1(1.30E+0)-	2.81E+1(1.13E+0)-
R103	8.87E+0(2.74E+0)	**1.24E+0(4.09E+0)+**	2.20E+1(1.05E+0)-
R104	**2.40E+0(5.30E-01)**	3.69E+1(2.37E+0)-	6.11E+0(2.00E+02)-
R105	**2.84E+0(3.77E+0)**	3.34E+0(4.45E+0)-	7.68E+0(3.03E+0)-
R106	**7.50E+0(2.11E+0)**	8.83E+0(1.14E+0)-	8.83E+0(4.32E+0)-
R107	**2.31E+0(1.65E+0)**	2.91E+0(2.98E+0)-	4.59E+0(3.72E+0)-
R108	**1.35E+1(2.25E+0)**	3.19E+1(2.49E+0)-	3.93E+1(4.32E+0)-
R109	**1.82E+1(2.00E+0)**	2.31E+1(4.42E+0)-	5.47E+1(4.63E+0)-
R110	**2.31E+1(3.84E+0)**	4.00E+1(4.91E+0)-	6.46E+1(5.70E+01)-
R111	**3.05E+0(7.40E+01)**	6.54E+0(4.61E+0)-	7.03E+0(2.70E+01)-
R112	**1.84E+0(1.71E+0)**	6.55E+0(3.62E+0)-	2.55E+1(1.94E+0)-
R201	**2.87E+0(2.87E+0)**	3.15E+0(2.65E+0)-	3.53E+1(1.50E-01)-
R202	**6.37E+0(3.51E+0)**	8.74E+0(4.37E+0)-	9.76E+1(2.29E+0)-
R203	**1.12E+0(2.08E+0)**	3.51E+1(1.24E+0)-	5.57E+1(2.99E+0)-
R204	**1.51E+0(1.06E+0)**	1.55E+0(3.91E+0)≈	4.58E+1(4.06E+0)-
R205	**4.35E+0(2.77E+0)**	8.75E+0(5.60E-01)-	4.83E+1(3.39E+0)-
R206	**2.36E+0(2.80E-01)**	8.88E+0(3.31E+0)-	4.29E+1(3.82E+0)-
R207	**1.92E+0(1.41E+0)**	1.99E+0(7.20E-01)≈	2.77E+1(4.37E+0)-
R208	3.03E+0(3.73E+0)	**1.26E+0(1.56E+0)+**	3.49E+1(3.86E+0)-
R209	3.50E+0(4.75E+0)	**3.18E+0(1.77E+0)+**	3.75E+1(2.70E-01)-
R210	**2.19E+0(4.69E+0)**	4.83E+1(3.56E+0)-	3.89E+1(8.00E+01)-
R211	**3.55E+0(4.65E+0)**	6.29E+1(4.30E+0)-	6.47E+1(4.31E+0)-
C101	2.50E+0(3.36E+0)	**1.50E+0(2.88E+0)+**	1.59E+1(4.80E+0)-
C102	**2.40E+0(2.56E+0)**	2.78E+1(3.90E+0)-	3.52E+1(4.24E+0)-
C103	**2.34E+0(1.90E-01)**	4.38E+0(8.00E+01)-	6.08E+0(3.90E-01)-
C104	**1.88E+0(1.39E+0)**	3.72E+0(4.47E+0)-	9.36E+0(2.52E+0)-
C105	**3.40E+0(1.01E+0)**	5.36E+1(5.10E+01)-	3.78E+1(2.98E+0)-
C106	**2.94E+0(4.30E-01)**	4.49E+0(2.69E+0)-	6.52E+0(3.05E+0)-
C107	**2.89E+0(2.01E+0)**	3.61E+1(4.77E+0)-	5.80E+1(1.24E+0)-
C108	**3.62E+0(7.80E-01)**	8.25E+0(2.42E+0)-	3.81E+1(2.88E+0)-
C109	**1.73E+0(4.60E-01)**	2.93E+1(2.60E+0)-	3.21E+1(3.90E-01)-
C201	**1.02E+0(1.56E+0)**	1.60E+1(8.00E-01)-	3.92E+1(3.60E+0)-
C202	**3.95E+0(1.00E+01)**	7.22E+1(4.26E+0)-	7.82E+1(4.45E+0)-
C203	**1.27E+0(6.10E+01)**	1.80E+1(2.03E+0)-	2.60E+1(8.70E-01)-
C204	**2.15E+0(3.47E+0)**	2.49E+1(1.60E+01)-	3.22E+1(4.24E+0)-
C205	**1.09E+0(4.49E+0)**	1.52E+1(4.29E+0)-	2.26E+1(3.65E+0)-
C206	9.95E+0(4.24E+0)	**1.36E+0(4.10E+0)+**	1.15E+1(8.80E+01)-
C207	8.22E+0(4.18E+0)	**3.60E+0(3.36E+0)+**	1.24E+1(4.51E+0)-
C208	**1.38E+0(1.56E+0)**	2.36E+0(1.48E+0)-	1.46E+1(4.08E+0)-
RC101	**1.78E+1(7.60E+01)**	2.54E+1(2.48E+0)-	2.97E+1(2.78E+0)-
RC102	**2.73E+1(2.84E+0)**	3.62E+1(6.20E+01)-	4.42E+1(3.41E+0)-
RC103	**1.79E+1(1.56E+0)**	4.32E+1(7.50E+01)-	5.53E+1(4.70E+0)-
RC104	**3.21E+0(3.02E+0)**	5.12E+0(2.10E+0)-	6.07E+1(2.48E+0)-
RC105	**6.56E+0(1.66E+0)**	9.03E+0(2.40E+0)-	9.89E+0(2.37E+0)-
RC106	**2.97E+1(1.08E+0)**	7.43E+1(7.00E-02)-	6.57E+1(8.50E-01)-
RC107	**1.26E+1(1.99E+0)**	1.68E+1(1.73E+0)-	2.14E+1(3.67E+0)-
RC108	**1.36E+1(4.37E+0)**	2.19E+1(2.95E+0)-	4.75E+1(2.53E+0)-
RC201	**3.82E+1(2.79E+0)**	3.82E+1(4.16E+0)≈	3.82E+1(4.10E+0)≈
RC202	**3.30E+1(2.05E+0)**	3.30E+1(1.85E+0)≈	3.30E+1(4.57E+0)≈
RC203	**1.75E+1(2.67E+0)**	2.05E+1(4.47E+0)-	8.75E+1(2.68E+0)-
RC204	**3.25E+1(2.30E+0)**	4.81E+1(4.06E+0)-	6.54E+1(4.73E+0)-
RC205	**2.86E+1(1.25E+0)**	3.09E+1(4.46E+0)≈	3.50E+1(4.12E+0)-
RC206	**2.91E+1(4.35E+0)**	6.23E+1(4.18E+0)-	3.81E+1(3.68E+0)-
RC207	**3.97E+1(4.60E-01)**	6.48E+1(3.21E+0)-	6.97E+1(5.00E+0)-
RC208	**2.10E+1(1.39E+0)**	2.87E+1(3.59E+0)-	2.87E+1(4.00E+0)-
+/-/≈		**6/44/6**	**0/54/2**

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

sorting search process. Especially, all algorithms utilize NSGA-II's environmental selection process for consistency. Table I displays the mean and standard deviation of the IGD for each instance over 20 independent runs. Wilcoxon rank-sum tests [21] at a 5% significance level were conducted to identify significant differences between MOEA/D-CBS and the other algorithms. The symbols "+", "-", and "≈" in the table's notation indicate whether the results of an algorithm are significantly better, worse, or similar to MOEA/D-CBS, respectively. Bold results denote the best mean values.

Lower IGD values indicate better performance in terms of diversity and convergence of solutions. In Table I, the MOEA/D-CBS algorithm outperforms MOEA/D in 54 instances, with a notably stronger performance in the R2 and C1 datasets, underscoring the effectiveness of our CBS local search approach compared to MOEA/D that do not incorporate local search mechanisms. Furthermore, our algorithm outperforms the VND local search in 44 our of 56 instances and performs equally well in six instances. Overall, upon comparison of the number of best IGD values, MOEA/D-CBS algorithm exhibits a 96.43% improvement over MOEA/D and 78.57% over D-VND, confirming its robust performance and enhanced efficiency in tri-objective VRPTW optimization.

V. Conclusion

In this paper, we introduced an innovative approach that amalgamates the MOEA/D with a K-means clustering based local search to address a tri-objective VRPTW. Our method uniquely leverages customer coordinates and time windows, utilizing clustering techniques to reorder the customers and thereby enhancing vehicle routing efficiency.

Major contributions of this study include the development of a novel heuristic algorithm that combines a cluster-based mechanism with MOEA/D, offering a distinctive strategy in local search processes for the tri-objective VRPTW. We also established criteria for reordering customers based on proximity to cluster centers and alignment, thus improving the efficiency of solutions. Experiments validate the algorithm's effectiveness in tri-objective VRPTW scenarios.

For future research, we intend to expand our methodology to encompass a broader range of complex VRPTW scenarios. A key aspect of this expansion will involve a focused effort on optimizing the time complexity of our algorithm. Achieving a harmonious balance between computational efficiency and the demands of processing larger datasets is imperative, especially in the context of practical, real-world logistics applications.

References

[1] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, part i: Route construction and local search algorithms," *Transportation Science*, vol. 39, no. 1, p. 104–118, feb 2005.

[2] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[3] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias, "Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification," *Operational Research*, vol. 22, no. 3, pp. 2033–2062, 2022.

[4] S.-Y. Tan and W.-C. Yeh, "The vehicle routing problem: State-of-the-art classification and review," *Applied Sciences*, vol. 11, no. 21, p. 10295, 2021.

[5] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[6] K. Tang, S. Liu, P. Yang, and X. Yao, "Few-shots parallel algorithm portfolio construction via co-evolution," 2020.

[7] S. Liu, K. Tang, and X. Yao, "Memetic search for vehicle routing with simultaneous pickup-delivery and time windows," *Swarm and Evolutionary Computation*, no. 5, p. 100927, 2021.

[8] B. Qu, Y. Zhu, Y. Jiao, M. Wu, P. Suganthan, and J. Liang, "A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems," *Swarm and Evolutionary Computation*, vol. 38, pp. 1–11, 2018.

[9] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *computational optimization and applications*, vol. 34, pp. 115–151, 2006.

[10] Y.-J. Gong, J. Zhang, O. Liu, R.-Z. Huang, H. S.-H. Chung, and Y.-H. Shi, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 254–267, 2012.

[11] W. Zhang, D. Yang, G. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for vrptw," *Expert Systems with Applications*, vol. 145, p. 113151, 2020.

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.

[12] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5772–5782, 2021.

[13] Y. Qi, Z. Hou, H. Li, J. Huang, and X. Li, "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 62, no. C, p. 61–77, oct 2015. [Online]. Available: https://doi.org/10.1016/j.cor.2015.04.009

[14] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] C. Legrand, D. Cattaruzza, L. Jourdan, and M.-E. Kessaci, "Improving neighborhood exploration into moea/d framework to solve a bi-objective routing problem," *International Transactions in Operational Research*, vol. n/a, no. n/a.

[16] Y.-L. Lan, F. Liu, W. W. Ng, J. Zhang, and M. Gui, "Decomposition based multi-objective variable neighborhood descent algorithm for logistics dispatching," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 826–839, 2020.

[17] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, pp. 17–30, 2006.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[19] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 892–899.

[20] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations research*, vol. 35, no. 2, pp. 254–265, 1987.

[21] R. G. D. Steel and J. H. Torrie, *Principles and Procedures of Statistics: A Biometrical Approach*, 2nd ed. McGraw-Hill Kogakusha, Ltd., 1980.

Authorized licensed use limited to: Universiteit Leiden. Downloaded on January 31,2025 at 13:09:15 UTC from IEEE Xplore. Restrictions apply.