IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 October 2024, accepted 14 November 2024, date of publication 20 November 2024, date of current version 4 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3503686

RESEARCH ARTICLE

Collaborative Routing With Multi-Heuristic Algorithm for Enhanced Delivery Efficiency in Truck-Multi Unmanned Aerial Vehicles Systems

RAMDHAN NUGRAHA [1,2], MD MASUDUZZAMAN [1], (Member, IEEE), AND SOO YOUNG SHIN [1], (Senior Member, IEEE)

[1]Department of IT Convergence Engineering, Kumoh National Institute of Technology (KIT), Gumi 39177, South Korea
[2]Faculty of Electrical Engineering, Telkom University, Bandung 40257, Indonesia

Corresponding author: Soo Young Shin (wdragon@kumoh.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Innovative Human Resource Development for Local Intellectualization Support Program, Supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), 50%, under Grant IITP-2024-2020-0-01612; and in part by the MSIT through the Information Technology Research Center (ITRC) Support Program, Supervised by IITP, 50%, under Grant IITP-2024-RS-2024-00437190.

ABSTRACT This paper proposes the integration of Unmanned Aerial Vehicles (UAVs) into conventional delivery trucks as a solution to the challenges caused by the growing requirements of e-commerce and the congestion in urban areas in the logistics sector. The variable neighborhood priority routing problem using truck and multi-UAV (VNPRPTmD) is presented as a multi-heuristic technique aimed at minimizing delivery delays by utilizing both terrestrial and aerial delivery methods. The VNPRPTmD model improves delivery efficiency by utilizing consumer segregation and clustering, routing optimization, and multi-vehicle collaborative routing. The concept enables UAVs to operate with flexibility, departing from and returning to any specified spot, hence eliminating the requirement to return to their original location. This approach minimizes operational expenses and enhances the efficiency of delivering parcels by effectively leveraging the capabilities of trucks and multiple UAVs. The research provides an elaborate mathematical model, evaluates the materials and methods employed, and concludes with findings that clearly establish the effectiveness of this innovative delivery system.

INDEX TERMS Vehicle routing problem, unmanned aerial vehicle, synchronization heuristic algorithm, K-nearest neighbor, variable neighborhood search.

I. INTRODUCTION

The rapid evolution of e-commerce and the growing demand for timely parcel delivery present significant challenges for the logistics industry. Traditional delivery systems struggle to meet these demands due to increasing road congestion and the limitations of existing infrastructure.

Unmanned Aerial Vehicles (UAV) have emerged as a promising solution to these challenges. Companies like DHL [1], Amazon [2], Alibaba [3], UPS [4], Google [5], FedEx [6] are rapidly integrating UAVs into their delivery networks. Unlike traditional ground vehicles, Ackerman et al. [7] mention UAVs can navigate areas lacking proper roads and bypass road traffic, and Joerss et al. [8] promise the UAV can have faster delivery times to consumers. Several delivery companies have also carried out experiments to assess the feasibility of employing UAVs for food delivery. Transitioning from conventional urban traffic to urban air traffic will increase available air traffic capacity [9].

However, UAVs face significant challenges, including limited payload capacities and short battery life, which restrict the size of parcels they can carry and the distance

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

TABLE 1. List of abbreviations.

Abbreviation	Full Name
$ACVRP$	asymmetric capacitated vehicle routing problem
BKS	best-known solution
$FTRPD$	flexible traveling repairman problem with UAVs
GA	genetic algorithm
$GRASP$	greedy randomized adaptive search procedure
HGA	hybrid genetic-sweep algorithm
$K-NN$	K-nearest neighbors
SA	simulated annealing
$TmDTL$	truck-multi UAV team logistics
$TSP-D$	traveling salesman problem with UAVs
UAV	unmanned aerial vehicle
$VNPRPTmD$	variable neighborhood priority routing problem using truck and multi-UAV
VNP	variable neighborhood priority
VRP	vehicle routing problem
$VRPTW$	vehicle routing problem with time windows

they can travel [10], [11]. To overcome these limitations, integrating UAVs with traditional ground vehicles, such as trucks, presents a groundbreaking approach [12]. This hybrid system utilizes trucks for delivering larger parcels and UAVs for smaller or more urgent deliveries, potentially reducing delivery costs significantly [13], [14], [15].

The problem becomes crafting the most efficient delivery routes that accommodate both trucks and UAVs, ensuring they meet payload, battery, timing, and distance requirements. This challenge is known as the Vehicle Routing Problem (VRP) [16]. Solving the VRP presents a challenge because the potential routes multiply exponentially with each added location or vehicle, making it difficult to fetch the optimal route quickly.

The VRP is an NP-hard problem, which means the problem does not have a straightforward solution as it grows, requiring significant computational power and time to find even an approximate answer. To navigate this complexity, several mathematical programming and heuristic techniques have been developed to solve the VRP's unique constraints and optimization goals [17], [18], [19].

Numerous studies [10], [20] have proposed solutions to the VRP by combining trucks with UAVs. Most research in this area has simplified the routing challenge by pairing a single UAV with each truck [17], [18], [19]; while this approach streamlines the routing problem, it fails to fully capture the complexities and demands of real-world logistics operations, prompting the need for more comprehensive investigations and diverse approaches.

In [18], all of the authors suggested the use of a truck both as a depot and battery replacement station for the UAVs. However, they overlooked the potential for trucks to also serve directly in delivering parcels, alongside their support for UAVs. Incorporating parcel delivery by trucks could further enhance the efficiency of this system.

Moshref et al. [24] introduced a model called the Flexible Traveling Repairman Problem with Drones (FTRPD), where a UAV can leave from and return to a moving depot or truck, allowing for dynamic delivery routes. The limitation here is a UAV's single-parcel capacity per flight, which constrains delivery efficiency.

Conversely, Leon-Blanco et al. [25] proposed the Truck-multi-Drone Team Logistics Problem (TmDTL) to allow UAVs to carry multiple parcels in one flight. However, this model requires the truck to wait for all UAVs to return before proceeding, potentially delaying the delivery process. The development of automation, aviation, battery, and artificial intelligence technologies has been rapid, and they are now extensively utilized in the trucks and UAVs inside of this system. Over the past few years, there has been an increase in the carrying capacity of trucks and UAVs. This means that a single truck can now transport more UAVs, and each UAV can do more flying excursions over a longer period of time. As a result, a UAV can visit and service more customers during a single flying trip [26].

Compared to the existing literature summarized in Table 2, our study considers UAV speed and energy consumption. By proposing a heuristic approach, the variable neighborhood priority routing problem using truck and multi-UAV (VNPRPTmD) aims to minimize delays in current delivery systems. This method allows each UAV to depart from the depot and return to any designated location, either a depot or truck, after parcel delivery. This flexibility means UAVs do not have to return to their starting point, enabling more efficient delivery paths.

The proposed method enhances delivery efficiency through a three-phase, multi-heuristic algorithm:

1) **Consumer Segregation and Clustering:** The first step is to sort and group consumer orders using Variable Neighborhood Priority (VNP) and K-Nearest Neighbors (K-NN), streamlining delivery orders.
2) **Path Planning:** Next, employ a heuristic to plan the best routes for a truck and its UAVs, aiming to cut delivery times and boost efficiency.
3) **Multi-vehicle Collaborative Routing:** Lastly, fine-tune the delivery strategy to tackle time constraints, elevating system performance.

The structure of our paper is organized as follows: Section II reviews related works; Section III details the VNPRPTmD model; Section IV provides a mathematical model description; Section V discusses the materials and methods; and Sections VI and VII present the results and conclusions, respectively. Additionally, the abbreviation is listed in Table 1.

II. RELATED WORKS

Murray and Chu [18] introduced the Traveling Salesman Problem with Drones (TSP-D), exploring two variations and focusing on heuristic and mathematical models for efficient routing. They innovated a heuristic based on dividing truck routes into UAV-assisted deliveries after path acquisition.

Agatz et al. [17] furthered this by creating truck and UAV route permutations, suggesting an operation-based formulation. They improved upon Murray et al. [18] work

TABLE 2. Existing literature comparison.

Existing Research	Number of UAV	UAVs parcel delivery	UAV Capacity	UAVs Multiple visits	UAVs Energy Consumption	UAV speed Constrain	UAVs rendezvous
[14]	≥ 1	1	x	x	x	x	Different-location
[18]	1	1	x	x	x	x	-
[19]	1	1	o	x	x	x	-
[20]	≥ 1	1	o	x	x	x	Same-location
[24]	≥ 1	≥ 1	o	o	x	x	Same-location
[25]	≥ 1	≥ 1	o	o	x	x	Same-location
[27]	1	1	x	x	x	x	-
[28]	≥ 1	1	o	x	x	x	Same-location
[29]	≥ 1	≥ 1	o	x	x	x	Different-location
[30]	≥ 1	≥ 1	o	x	x	x	Different-location
Proposed	≥ 1	≥ 1	o	o	o	o	Different-location

by allowing trucks to meet UAVs at the start of their flight, introducing dynamic programming and local search heuristics.

Ha et al. [19] delved into TSP-D delivery costs using the Greedy Randomized Adaptive Search Procedure (GRASP) and local search, showcasing results for scenarios with up to 100 consumer nodes. In parallel, Ponza et al. [31] applied Simulated Annealing (SA) to TSP-D, focusing on multi-vehicle and multi-UAV operations and providing theoretical boundaries and worst-case scenarios. Moreover, Poikonen et al. [32] further expanded on the collaboration between truck and UAV and suggested a theoretical investigation.

Multiple studies [18], [19], [21], [22], [23] considered trucks as mobile depots and battery replacement hubs, removing battery charge time from the equation due to the assumption of instant battery replacement after each flight.

The research conducted by Moshref-Javadi et al. [24] explained the FTRPD model, which was built once a UAV had been launched from a vehicle. This model allows the UAV to rejoin the truck at a location distinct from its initial departure point. This concept is most prevalent in the literature on truck and UAV delivery paradigms and can be used in urban areas where it is difficult for trucks to locate parking spaces [18], [19], [27], [31]. Hence, a UAV can be dispatched and recovered at a different location on the truck route [33].

Leon-Blanco et al. [25] proposed the TmDTL model to tackle multi-UAV and single-truck routing without considering UAV landing, recharging, or take-off times. Instead, it focused on loading and unloading times and UAV speed adjustments based on parcel load, revealing potential time delays due to UAVs departing and returning simultaneously.

Baek et al. [34] emphasized the critical role of UAV energy consumption in dictating delivery speed and capacity. An optimal balance between energy requirements and delivery speed is crucial for operational efficiency. According to studies conducted by Youngmin et al., the flight range of a UAV decreases according to the amount of payload it carries [35].

According to Murray and Raj et al. [20], the speed of the UAV has an effect on its power consumption, which in turn has an effect on its flying range and endurance. For example, when considering the speed of a variable UAV, it is best to select a speed that is lower than the maximum speed in order to reduce the amount of energy [23]. Therefore, the energy consumption of the UAV provides a realistic representation of its lifespan. It could be proportional to the flight distance of the UAV [36] or depends on the weight of the UAV as well as the speeds and other characteristics of the packages [23], [37], [38], [39]. According to the findings of these types of research, optimizing the relationship between the payload, the UAV's endurance, and the battery capacity is one of the most important factors in reducing either the cost or the amount of time required for UAV deliveries [40], [41].

Despite these advancements, existing literature often overlooks the flexibility of UAV departures and returns, along with detailed investigations into speed control and energy consumption during parcel delivery. Our review in Table 2 underscores this gap, indicating the need for further research in these areas.

CONTRIBUTION

This section discusses several technical contributions related to the proposed system that has been developed:

1) Various UAV parameters, including UAV energy consumption and controlling speed, are considered in the routing process.
2) Non-dominated Sorting Genetic Algorithm (NSGA-II) [42] and 2-opt are combined to achieve a better initial route for truck routing with limited capacity.
3) A priority customer system is set up that sorts customers on each route based on the size of the parcels they have ordered, considering the capabilities of UAVs and trucks. Truck delivery is an option for packages that do not meet the requirements for UAV delivery.
4) VNPRTmD routing planning based on a multi-heuristic algorithm is proposed to achieve better performance.

TABLE 3. Notation for set representation and decision variables.

Notation	Description
a. List of set representation	
at_j^u	UAV arrival time at j
at_j^g	UAV arrival time at j
d_{ij}	Distance from i to j
k	set of vehicles $N = \{k_0,\ k_1,\ k_2, \ldots,\ k_m\}$
l_i^u	loading time for the UAV at the depot or meeting location with the truck
l_i^g	loading time for the truck consumer location
N_t	Customer served by truck
N_d	Customer served by UAV
P_{cd}	Number of parcels carried by the UAV
ps	parcel size
pw	parcel weight
P_{cd}	Number of parcels carried by the UAV
q_j	Parcels ordered by customer j
S^u	speed of UAV
S^g	speed of truck
t_i^o	take-off time of the UAV at location i
t_j^l	landing time of the UAV at location j
U	the maximum capacity of UAV
ul_j^g	Unloading time truck at location j
ul_j^u	Unloading time UAV at location j
v_k	Speed of vehicle (truck or UAV)
v_g	Speed of truck
v_u	Speed of UAV
b. List of decision variable	
Notation	***Description***
c_{ij}^u	time required from i to j by UAV
c_{ij}^g	time required from i to j by truck
F_u	Flight range limit of UAVs
kb_{ij}^u	UAV battery consumption from i to j
Z	Represents the total number of trips, calculated as a weighted sum of all individual trips across all routes and all vehicles.
dt_j^u	Departure time of UAV from node j
dt_j^g	Departure time of truck from node j
wt_j^g	Waiting time of truck at node j
x_{ij}^k	Binary variable; 1 if vehicle k trip travels from node i to node j; 0 otherwise.
x_{iji}^k	Binary variable; 1 if vehicle k trip travels from node i to node j and returns to the truck at node i; 0 otherwise
x_{ijn}^k	Binary variable; 1 if vehicle k trip travels from node i to node j and continue from node j trip travel to node n ; 0 otherwise

III. VARIABLE NEIGHBORHOOD PRIORITY ROUTING PROBLEM USING TRUCK AND MULTI-UAV

Trucks play an essential part in our heterogeneous logistics systems. Trucks are necessary to carry large quantities of goods and serve as mobile storage facilities for UAVs. This dual functionality enables the efficient management of larger packages and serves as a foundation for UAV operations in places not readily reachable by road. In addition, trucks enhance the operational range of UAVs by bringing them in proximity to delivery sites, hence minimizing the necessity for UAVs to travel back to a central depot for recharging or reloading. Moreover, trucks play a crucial part in the route planning algorithms by being incorporated in a way that optimizes their direct delivery routes and enhances their ability to assist in deploying and recovering UAVs at strategic locations and ensuring a comprehensive assessment of all technology elements inside our logistic systems. This emphasizes the crucial role of trucks in facilitating UAV operations and maintaining the dependability of the entire supply chain.

VNPRPTmD is a combination of a truck (k_0) and several UAVs (k_1, k_2, .., k_m). The notation and their description are described in Table 3. Each consumer can be served by a truck; however, not every consumer can be accommodated by a UAV due to varying parcel orders. The limited capacity of each UAV to carry parcels further contributes to this constraint. The maximum of parcel dimensions (M_{sd}) and maximum of parcel weight (M_{wd}) represent the initial state. When the UAV leaves the truck to deliver the parcels, the time taken to load the parcels (l_i^u) onto the UAV at the meeting location or depot and the time for unloading parcels (ul_j^u) from the UAV at the consumer target are considered.

After the UAV has left the truck to deliver the parcel to a consumer, the truck simultaneously leaves the location and heads to the next location. A truck can only wait at the rendezvous point for the return of the UAV at the consumer's location. This is a concept prevalent in cities where a truck cannot park randomly because this would disrupt traffic. Additionally, several regions need to permit UAVs to operate within their territory. Regarding the meeting location, the UAV (N_d) will move on to the meeting point at the consumer's location served by the truck (N_t) after it has completed delivering packages to the consumer. In addition, the vehicle (truck/UAV) that arrives at the earliest waits for the others before continuing to the next destination. Fig. 1 represents the above-described scenario. First, while arriving at the meeting point at (N_t), the truck unloads the customer's parcel into the UAV. The UAV can carry up to three small packages in a single flight. In addition, consumers who order more than three small or large parcels can be delivered by truck. After loading the parcel, the UAV moves to the customer (N_d) to deliver the packages. After delivering the parcel to each customer ((N_d), (N_{d_1})....(N_{d_n})), the UAV meets the Truck at the meeting point ((N_{t_1}),(N_{t_2}), ... (N_{t_n})). A delay is considered in this scenario when the truck unloaded the parcel before the arrival of the UAV at the meeting location.

Several important aspects and limitations must be considered to solve these problems:

1) Specific consumers will be served by trucks and UAVs.
2) The designated locations for the takeoff and landing of the UAV.
3) The scheduled times for departing and returning the UAV.
4) The approach does not consider the possibility of vehicle collisions when determining routes.
5) UAV battery replacement will take place at depots or on trucks using battery swapping systems, which can be executed rapidly, eliminating the requirement for time calculations for UAV battery charging [43].
6) Energy consumption is not considered when the UAV waits for the truck's arrival at the meeting location [36].

The problem formulation in this research is modeled as follows:

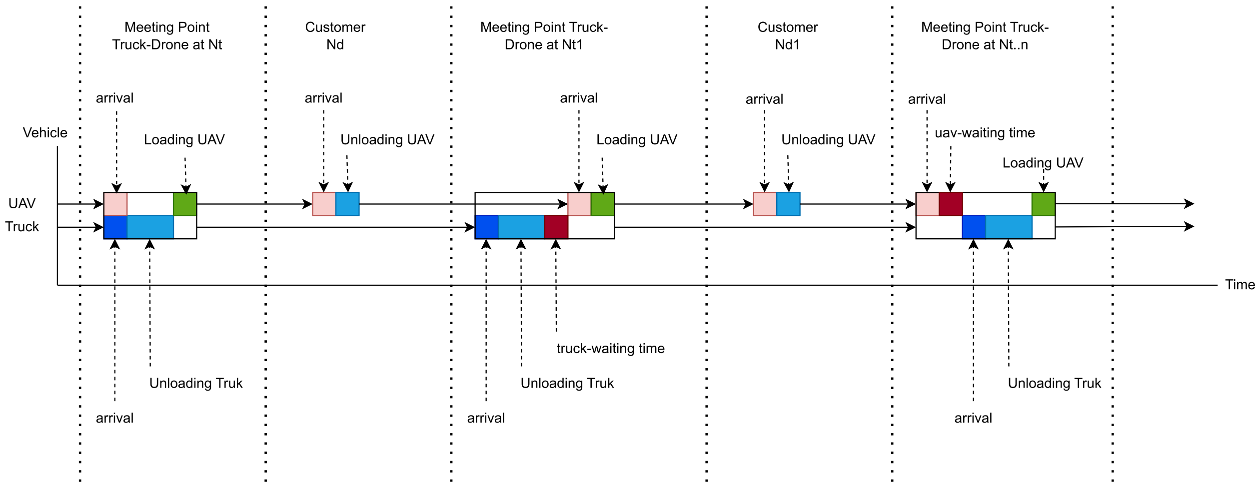

FIGURE 1. An estimation scheme for the arrival of UAV and truck at consumer locations.

1) Minimize travel time

$$\min Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\in\{g,u\}} x_{ij}^{g}\left(\frac{d_{ij}}{v_g}+ul_j^{g}\right) + \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij}\left(l_i^{u}+t_i^{o}+\frac{d_{ij}}{v_u}+ul_j^{u}+t_j^{l}\right) + \sum_{i=1}^{n}\sum_{j=1}^{n} wt_i^{g}$$

2) UAV and truck routing: Either the truck or a UAV must visit each consumer location exactly once.

$$\sum_{j,k} x_{ij}^{k} + \sum_{j} y_{ij} = 1 \quad \forall i$$

3) Capacity constraint for UAV: The total number of parcels carried by each UAV cannot exceed its capacity.

$$\sum_{i,j} P_{cd} x_{ij}^{u} \le U \quad \forall k$$

4) Flight time constraint for UAV: The total flight time for each UAV must not exceed its flight time limit.

$$\sum_{i,j} \frac{d_{ij}}{S^{u}(P_{cd})} x_{ij}^{u} \le F_u \quad \forall k$$

5) Connectivity and launch/retrieve mechanisms: Ensure that if a UAV starts a journey, it must end at either a consumer location or the depot.

$$\sum_{i} x_{i0}^{u} = \sum_{j} x_{0j}^{u} = 1 \quad \forall k$$

6) UAV speed: The speed of a UAV is determined by the total number of products carried by the UAV.
7) Launch and retrieve points: UAVs can be launched from and retrieved at different points by the truck.

IV. MATHEMATIC MODEL

$G\ (N, A)$ is defined as a directed graph, $N = \{v_0, v_1, v_2, \ldots, v_n\}$ is a set of vertices representing n consumers, v_0 represents a depot. $A = \{(v_i, v_j) \mid v_i, v_j \in N, i \neq j\}$ represents a set of edges connecting the vertices, which represents the distance between consumers (i and j are indices for the nodes/consumers $i, j = 0, 1, 2 \ldots, n$). K is the total set of vehicles m with a total capacity of U, and index k is the index for all vehicles $M = \{k_0, k_1, k_2, \ldots, k_m\}$. Additionally, k_0 represents the truck, and the others represent UAVs. The speed of the UAV (S^u) is higher than the speed of the truck (S^g). It is assumed that the UAV's movement is straight toward the targets and does not follow the highway like the truck. The parameters used in this study are q_i and c_{ij}, where q_i is a simple node of i, and c_{ij} is the time required to cover the distance from i to j. The objective of this study is to minimize travel time by ensuring the parcel is sent to the consumer as soon as possible.

The variable x_{ij}^{k} is a binary variable to the decision-making process in the model. It is a binary variable that represents whether there is a direct trip from node i to node j by vehicle k (truck or UAV). Specifically, x_{ij}^{k} is set to 1 if vehicle k travels directly from node i to node j; otherwise, it is set to 0, indicating no such trip is made by vehicle k. This binary representation simplifies the problem's constraints and objectives by clearly defining each vehicle's routes, thus facilitating the calculation of costs, distances, or other route-related metrics within optimization frameworks, particularly in transportation or logistics network models.

$$x_{ij}^{k} = \begin{cases} 1, & \textit{if there is a trip from } i \textit{ to } j \\ & \textit{using vehicle } k. \\ 0, & \textit{else}. \end{cases} \tag{1}$$

The total number of trips Z is obtained by adding the total number of trips on each route.

$$Z = \sum\nolimits_{k=1}^{m}\sum\nolimits_{i=0}^{n}\sum\nolimits_{j=0}^{n} c_{ij}x_{ij}^{k} \tag{2}$$

Each consumer can only be visited once. If the consumer is being served, leave the consumer using one of the routes.

$$\sum_{k=1}^{m}\sum_{i=0}^{n} x_{ij}^{k} = 1, \forall j \in \{1, 2, \ldots, n\} \tag{3.a}$$

$$\sum_{k=1}^{m}\sum_{j=0}^{n} x_{ij}^{k} = 1, \forall i \in \{1, 2, \ldots, n\} \tag{3.b}$$

If a vehicle visits consumer $x_{ij}^{k} = 1$, and leaves the consumer after the visit, then,

$$\sum_{i=0}^{n} x_{ir}^{k} - \sum_{j=0}^{n} x_{rj}^{k} = 0, \quad \forall r \in \{1, 2, \ldots, n\}, k \in \{1, 2, \ldots, m\} \tag{4}$$

The maximum parcel capacity that the UAV can deliver (U) along a route in one flight is calculated by aggregating the orders of every consumer (q_j) along the route.

$$\sum_{j=0}^{m} q_j \left(\sum_{i=0}^{n} x_{ij}^{k}\right) \leq U, \quad \forall k \in \{1, 2, \ldots, m\} \tag{5}$$

The total UAV travel time for each route Z must be less than the maximum set time (T_{max}), and the total UAV travel time is obtained by summing the travel times required to visit each consumer using a route.

Each vehicle travel route begins and ends at a depot. If each route starts at the depot, there will be vehicle trips on each route leaving the depot. If each vehicle travel route ends at the depot, then a trip back to the depot will occur on each route.

$$\sum_{j=1}^{n} x_{0j}^{k} \leq 1, \quad \forall k \in \{1, 2, \ldots, m\} \tag{6}$$

$$\sum_{i=1}^{n} x_{i0}^{k} \leq 1, \quad \forall k \in \{1, 2, \ldots, m\} \tag{7}$$

The Euclidean distance is used to calculate the distance to every location, where the distance from i to j is equal to the distance from j to i. The Euclidean distance calculates between two points x and y in an n-dimensional space. Here, x_i and y_i are the coordinates of points x and y in the i^{th} dimension, respectively. The equation calculates the distance by summing the squares of the differences between corresponding coordinates of the two points across all n dimensions and then taking the square root of this sum. By applying the following formula, a straight line can be calculated from the query point to the point being measured:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{8}$$

The time taken by the vehicle to travel from i to j (c_{ij}) is the distance from i to j divided by the speed of the vehicle.

$$\sum_{i \in N} c_{ij}^{u} = \frac{d_{ij}}{S^u}, \forall j \in N \tag{9.a}$$

$$\sum_{i \in N} c_{ij}^{g} = \frac{d_{ij}}{S^g}, \forall j \in N \tag{9.b}$$

For each node j in the set N (which represents all customer nodes or locations), the total traffic or resources entering node j (from other nodes i) is exactly equal to the total traffic or resources leaving node j to other nodes i.

A large area of operations research involves routing and scheduling. Some of these routing problems include scheduling. Due to the high cost of fuel, cost-efficient routes often also reduce CO_2 emissions. Routing problems for which an assignment of demands occurs are called routing and scheduling problems. Routing is selecting a path by which an object will travel, and scheduling is the assignment of tasks or commodities to objects, often including time [44].

$$x_{iji}^{k} = \begin{cases} 1, & \textit{trip from } i \textit{ to } j \textit{ and back to } i \\ & \textit{using vehicle } k. \\ 0, & \textit{else}. \end{cases} \tag{10.a}$$

$$x_{iji}^{k} = d_{ij}^{k} + d_{ji}^{k} \tag{10.b}$$

This variable (x_{iji}^{k}) indicates whether a round trip is made from node i to node j and then returns back to node i. It is used to model routing scenarios where a vehicle or agent returns to the starting point after visiting another location.

$$x_{ijn}^{k} = \begin{cases} 1, & \textit{trip from } i \textit{ to } j, \textit{ and } j \textit{ to } n, \\ & \textit{using vehicle } k. \\ 0, & \textit{else}. \end{cases} \tag{11.a}$$

$$x_{ijn}^{k} = d_{ij}^{k} + d_{jn}^{k} \tag{11.b}$$

This variable (x_{ijn}^{k}) is used when modeling paths that include sequential trips, where a vehicle k trip starts at node i, goes to node j and continues to node n. It is particularly useful in scenarios like multi-stop delivery routes or sequential task assignments.

$$\sum_{i=1}^{n} c_{ij}^{k} = \sum_{j=1}^{n} x_{ij}^{k} \quad \forall k \tag{12}$$

Ensure that every truck leaving node i arrives at node j. This equation represents the equality of two summations, one over index i and the other over index j, for each k. The notation $\forall k$ indicates that this equality holds for all k. The vehicle that arrives at the rendezvous location will first wait for the arrival of the other vehicles.

$$wt_j^g \geq at_j^u - \left(at_j^g + ul_j^g\right) \tag{13}$$

Waiting time constraint if the UAV is delayed at the meeting point. The formula calculates truck waiting time at j (wt_j^g) by taking the difference between the UAV's arrival time (at_j^u), the truck's arrival time (at_j^g) at location j, and truck unloading at j (ul_j^g).

$$dt_j^g > at_j^u + l_j^u + ul_j^g \tag{14}$$

$$dt_i^g = \max(l_i^g + at_i^g, at_i^u, ((m_i \times (l_i^u + t_i^o)) + ((1 - m_i) \times t_i^0))) \tag{15}$$

Given in Eq. 14 is the synchronization of truck departure time at location j (dt_j^g) with UAV arrival, without concern about UAV activity. Eq. 15 is the synchronization of truck departure time at location j with UAV arrival and concern about UAV activity at the meeting point location. If $m_i = 1$ (UAV performs loading and takeoff again), the truck must wait until the UAV has finished loading and has taken off. If $m_i = 0$ (UAV does not perform loading), the truck must wait for the UAV to land on the truck if the UAV does not perform loading.

1) $l_i^g + at_i^g$: This is the earliest time needed for the truck to complete its loading and be ready to depart without considering interactions with the UAV.
2) at_i^u: This is the arrival time of the UAV at location i. The truck must ensure that the UAV has arrived before departing if they need to interact further (such as transferring parcels or equipment).
3) $(m_i \times (l_i^u + t_i^o))$: If location i is a meeting point ($m_i = 1$), the truck must wait for the UAV to complete loading (l_i^u) and take-off (t_i^o) before departing.
4) $((1 - m_i) \times t_i^l)$: If location i is not a meeting point ($m_i = 0$), then the relevant time is the UAV landing time (t_i^l). The truck must wait for the UAV to land if the location is used as a rendezvous point where the UAV meets back with the truck for further joint travel or cargo transfer.

$$dt_i^j = l_i^j + at_i^j \tag{16}$$

According to Eq. 16 is a concern if the truck is not at the meeting point location; the truck only needs to complete its own loading activities and then can depart from the consumer location.

Owing to changes in the speed, weight, and dimensions of the UAV, the number of parcels carried is affected by the battery power consumption (kb) of the UAV [34]. Eq. 17 presents the energy consumption of the drone when using arc (i, j) with a payload, where the payload is adjusted according to the remaining customer demand Q_j. The energy consumption is $EC_{ij}[W]$, where the symbols and parameters are defined in Amine et al. [43].

$$EC_{ij}[W] = \left(\sum\nolimits_{r \in R} m(Q_j) + F + m\right)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varepsilon\omega}} \tag{17}$$

V. METHODOLOGIES

This section will examine the VNPRPTmD method in greater depth. The previous section covered the method's components. In Section V-A, the task is to create routes to serve all customers, assuming the use of only trucks. Each truck has a capacity limit for delivering parcels to customers. Once the routes are established, the next step involves separating the customers into two groups: consumers who will be served by the truck (N_t) and consumers who will be prioritized by the UAVs (N_d). After separation, in Section V-B, a specific path will be created to serve customers who will be served by trucks only. In Section V-C, the clustering of UAV customers will be conducted in relation to the truck customer routes. Section V-D involves the merging process of truck customers with UAV customers. It is ensured that the routes created by combining trucks and UAVs do not exceed the truck's capacity for a single trip.

A. CONSUMER SEGMENTATION

Even though all consumers can be served by truck, it is possible to accelerate delivery when the truck and UAVs cooperate together. Therefore, the consumers are divided into two groups: consumers who will be served by the truck (N_t) and consumers who will be prioritized by the UAVs (N_d). For example, the number of parcels for each consumer is selected as shown in Table 4. The process of separating consumers can be observed in Alg. 1.

Algorithm 1 Segmentation N_d and N_t

$N \leftarrow$ Input Data consumer
$ps \leftarrow$ Size of Parcel for N_i. $i \in N$
$pw \leftarrow$ Weight of Parcel for N_i. $i \in N$
$ps^u_{max} \leftarrow$ Max Size of Parcel for UAV
$pw^u_{max} \leftarrow$ Max Weight of Parcel for UAV
repeat
  for $(i = 0, i < n, i++)$ **do**
    if $(ps \leq ps^u_{max})$ and $(pw \leq pw^u_{max})$ **then**
      consumer N_d
    else
      consumer N_t
    end if
  end for
until $N = 0$

The algorithm presented aims to classify consumer data into two groups based on parcel size and weight. The consumer input data is represented by N, with each consumer N_i having a parcel size (ps) and a parcel weight (pw). There are also two given maximum weight thresholds: maximum parcel size (ps^u_{max}) and maximum parcel weight (pw^u_{max}) for UAVs. The Alg. 1 proceeds as follows:

1) Initialization starts by setting all consumers in the dataset N.
2) The algorithm repeatedly checks each consumer from $i = 0$ to n_i, where n_i is the total number of consumers.
3) For each consumer N_i, the algorithm checks if the ps is less than or equal to ps^u_{max} and if the pw is less than or equal to pw^u_{max}.
4) If both conditions are met, consumer N_i is considered suitable for inclusion in group N_d. Otherwise, they are placed in group N_t.
5) This process continues until consumer data remains in N.

B. ROUTING TRUCK

The following consumers have been categorized using Alg. 1. The first task is to create a parcel delivery route for Consumer N_t and Consumer N_d. This section introduces a novel approach that integrates the NSGA-II algorithm with the 2-Opt algorithm to obtain the optimal routing results for consumer N_t and Consumer N_d.

TABLE 4. Example: List of consumers and their parcel sizes.

No. Consumer	Total Parcel	No. Consumer	Total Parcel	No. Consumer	Total Parcel	No. Consumer	Total Parcel
1	5 (Big)	6	5 (Big)	11	2	16	1
2	2	7	1	12	1	17	2
3	5 (Big)	8	2	13	5 (Big)	18	1
4	1	9	1	14	2	19	5 (Big)
5	2	10	5 (Big)	15	5 (Big)	20	2

The NSGA-II algorithm, introduced by Deb et al. [42], is widely employed for solving multi-objective optimization problems. NSGA-II is designed to identify Pareto-optimal solutions in multi-objective optimization problems. A comprehensive survey by Ma et al. [45] highlights the continuous improvements and applications of NSGA-II in various fields, noting its competitive performance compared to other evolutionary algorithms. These solutions are characterized by the fact that improving one objective would require compromising at least one of the other objectives. It recognizes that no one solution can achieve the optimum performance for all objectives.

Algorithm 2 NSGA-II With 2-Opt

function NSGAIIWith2Opt(P)
  Initialize population P with feasible random solutions
  Evaluate each individual in P
  while termination condition not met **do**
    Divide P into Pareto fronts F using non-dominated sorting
    Calculate crowding distance $CD(i)$ for each individual i
    $P' \leftarrow$ Select parents based on rank and CD
    $P'' \leftarrow$ Apply crossover and mutation to P'
    for all $x \in P''$ **do**
      Validate and adjust x to ensure each route complies with capacity
    end for
    Evaluate each $x \in P''$
    for all $x \in P''$ **do** ▷ Improvement with 2-Opt
      $x \leftarrow$ Alg.3(x)
    end for
    $P \leftarrow$ Merge and sort P and P'' by F and CD
    Select the top N individuals from P for the next generation
  end while
  return the best solution from F_1 with the highest CD
end function

The goal of Alg. 2 is to find the smallest total travel cost, which is given by $\sum_{i\in V}\sum_{j\in V, j\neq i} c_{ij}x_{ij}$, while keeping the vehicle capacity limit at $\sum_{i\in V}\sum_{j\in V} q_i x_{ij} \leq Q, \forall k \in K$. Each individual in the population is evaluated based on objective functions pertinent to Routing vehicles with limited capacity, such as the total distance traveled and the number of vehicles used. This evaluation helps rank individuals based on their performance. Each solution in the offspring undergoes an improvement process using the 2-Opt heuristic. This local search technique optimizes the routing paths by iteratively reversing segments of routes to decrease the total travel distance or improve other objectives without increasing the solution's complexity. The population for the next generation is formed by merging the current parents and offspring and then selecting the top-performing individuals based on their Pareto ranking and crowding distance. This elitist approach ensures that the best solutions are retained, preventing regression in solution quality. The algorithm repeats the main loop until a specified termination condition is met, which might be a maximum number of generations or a convergence criterion where no significant improvements are observed. At the end of the evolutionary process, the best solution from the first Pareto front with the highest crowding distance is selected and returned. This solution is considered the optimal trade-off among the objectives and offers a diverse and balanced approach to solving the problem of routing vehicles with limited capacity.

Following this, the 2-Opt algorithm, a well-established method for refining routes, is employed as a local search strategy to enhance the shaken path further. If the newly obtained path from the 2-Opt process has a lower cost function value, indicating an improved route, it replaces the current solution. This loop continues, incrementing an iteration counter each time until no further improvements are made or a pre-defined maximum number of iterations is reached. The algorithm halts when the population of paths converges on an optimal or near-optimal solution. The 2-Opt algorithm, detailed separately, takes a given route as input and searches for a shorter path by reversing segments of the tour between two nodes if it results in a cost reduction. This improvement check runs in a loop until no further gains are achieved, ensuring that the function returns the most optimized tour possible within the local search space. It is possible to find a good solution to the path planning problem by combining the 2-Opt algorithm with the NSGA-II algorithm. This is possible because NSGA-II algorithm can get out of local optima, and 2-Opt is efficient at fine-tuning routes.

Alg. 2 provides and outlines the implementation of the NSGA-II algorithm, augmented with the 2-Opt technique at Alg. 3, to address the routing problem. Initially, an initial population P is initialized with feasible random solutions, and

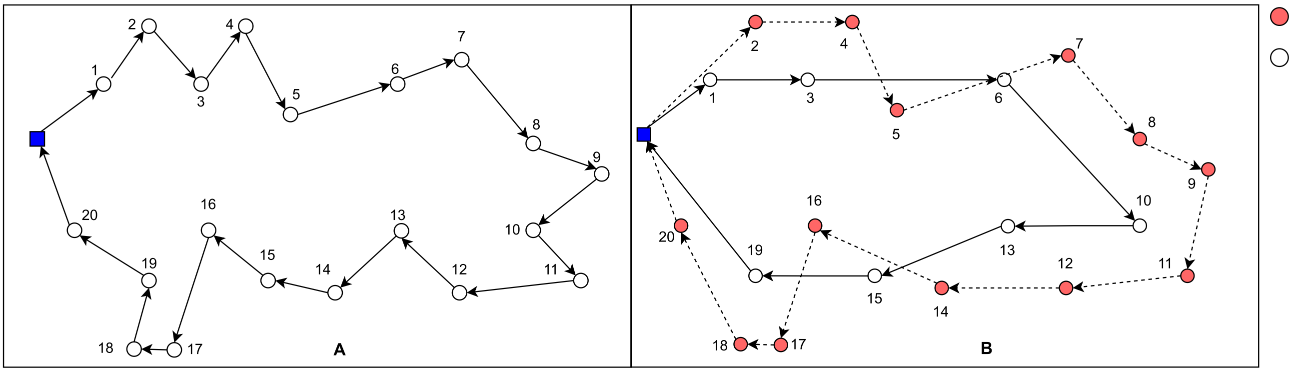

FIGURE 2. **(A) Routing consumer by one truck. (B) routing consumers for (N_d) and (N_t).**

Algorithm 3 2-Opt Algorithm

```
function 2-Opt(TourS)
    S' ← S
    improvement ← true
    while improvement do
        improvement ← false
        for i ← 1 to length(S') − 3 do
            for j ← i + 2 to length(S') − 1 do
                if cost(S'[i] to S'[i + 1]) + cost(S'[j] to
                    S'[j + 1]) > cost(S'[i] to S'[j]) +
                    cost(S'[i + 1] to S'[j + 1]) then
                        Reverse the tour segment between
                            S'[i + 1] and S'[j]
                    improvement ← true
                end if
            end for
        end for
    end while
    return S'
end function
```

each individual in this population is evaluated. Subsequently, the algorithm partitions the population into various Pareto fronts through non-dominated sorting and calculates the crowding distance for each individual. Based on these rankings and crowding distances, individuals are selected for the reproduction process, where crossover and mutation are applied to generate new offspring P''. Each route within this new offspring is then validated and adjusted to ensure compliance with vehicle capacity constraints. Following this, the offspring are evaluated, and each solution is enhanced using the 2-Opt technique. The enhanced population is then remerged and sorted based on the Pareto fronts and crowding distances, and the best individuals are selected for the next generation. This process is repeated until termination conditions are met, with the best solution derived from the first Pareto front with the highest crowding distance as the final output of the algorithm.

In the following Alg. 2, it is crucial to examine two more factors: crossover and mutation. Crossover, also known as recombination, is a fundamental mechanism in NSGA-II, where two-parent solutions are combined to generate new offspring, thus introducing necessary variability into the genetic pool. The Partially Mapped Crossover (PMX) approach, outlined by Ting et al. [46], accelerates the convergence of these algorithms by utilizing partial sequences inherited from each parent. This method operates by selecting two random points within the parent genomes, with the segment between these points exchanged between the parents to ensure offspring inherit a combination of traits. To preserve genetic diversity and prevent duplication, areas outside these points are filled with original parental genes, substituting any duplicates that emerge during the crossover. This approach effectively integrates beneficial traits from both parents, enhancing solution quality while maintaining essential characteristics.

Following the crossover phase, mutation mechanisms are employed to further diversify the solution space, as exemplified in research by Kuo et al [47]. The swap mutation technique is applied, where two positions within a chromosome are randomly selected and their contents swapped. The insertion mutation also plays a role, involving the relocation of one gene to a different position on the chromosome. Additionally, the inversion mutation reverses the sequence between two selected points. This study utilizes a randomized selection among these mutation methods with equal likelihood. Following these mutations, the viability of the offspring is evaluated by recalculating the objective function values, thereby enhancing the algorithm's ability to address complex optimization challenges.

As shown in Fig. 2(A), the routing problem states that all consumers are only served by the truck. Fig 2(B) shows the routing division between the consumers only served by the truck and others that are served by the UAVs. Fig. 2(A) and Fig. 2(B) show the result of path planning based on the Alg. 2. A more detailed explanation of Fig. 2(B) is provided in Fig. 3 where a truck route (r_t) starts at the depot and goes from consumer Gt_1 (no 1) until Gt_n and finally returns to the depot.

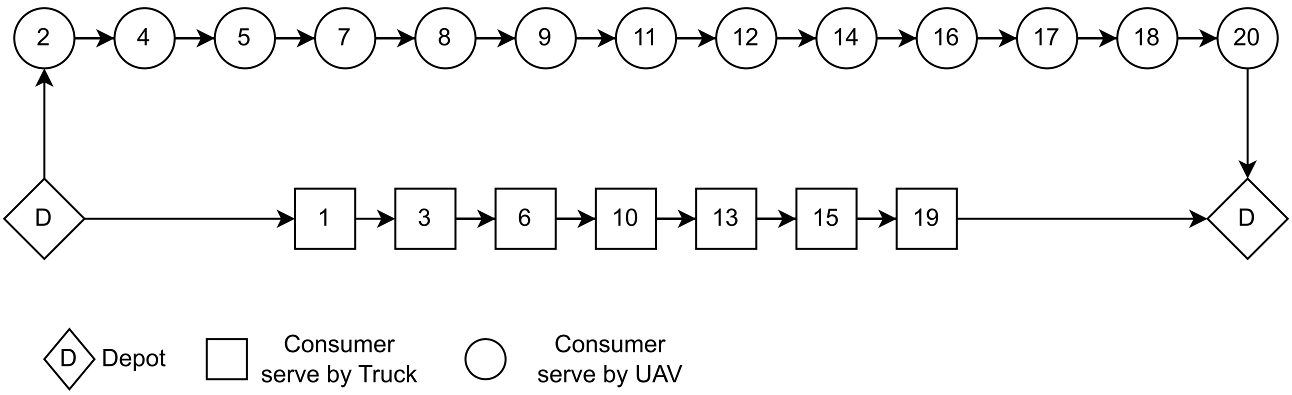

FIGURE 3. Pathway segmentation for consumer clustering according to characteristics purchased.

The routing UAV (r_d) starts from the depot and then continues to the consumer Gd_1 (no 2) until Gd_n before finally returning to the depot.

C. K-NEAREST NEIGHBORS (K-NN) ALGORITHM

In contrast to the segregation in Section V-A, which tries to separate consumers into two clusters, the clustering in this section is intended to enable the merging of consumer N_d with the existing path of the truck. It is essential to streamline the process of developing collaborative routing between trucks and UAVs. The initial stages entail identifying the takeoff location of the UAV for delivering parcels to customers, as well as determining the return location of the UAV after delivering parcels from the consumers.

This section introduces the K-NN algorithm, which is implemented to calculate the members of each cluster based on the consumers N_d and r_t. The K-NN algorithm is a simple and effective algorithm for clustering and classification in machine learning. It is a nonparametric method applied to classify data points based on the class of their K-nearest neighbors. The basic idea of K-NN is to assign a data point to the class that is most commonly represented among its nearest neighbors. The value of K determines the number of nearest neighbors used to make the assignment, and it is a parameter that can be adjusted to control the complexity of the model.

The K-NN can be used for clustering in a group of similar data points. The algorithm assigns each data point to a cluster of its K-nearest neighbors, which can be performed using various distance measures such as the Euclidean distance or cosine similarity. The K-NN is a simple and effective algorithm; however, it can be computationally expensive for large data sets because it needs to search for the K-NN of each data point. Additionally, the choice of k can have a substantial effect on the algorithm's performance, making the determination of the ideal value of k difficulties.

Despite its limitations, the K-NN is widely used for clustering and classification owing to its simplicity, flexibility, and ability to handle noisy and incomplete data. It is often used as a baseline method for comparison with other sophisticated

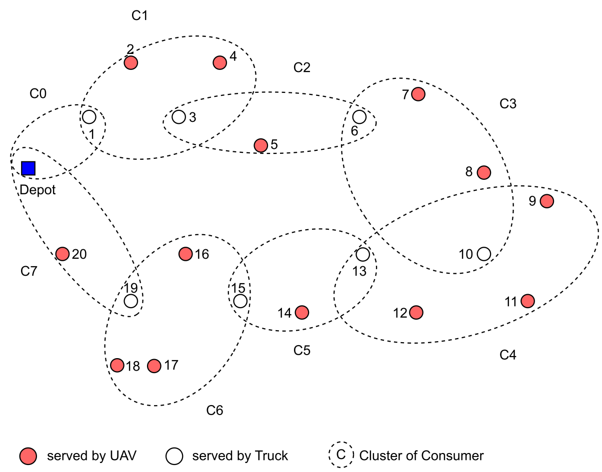

FIGURE 4. Consumer clustering using K-NN.

algorithms and in combination with other methods to improve performance and robustness.

Algorithm 4 K-NN Clustering

for ($i = 0, i<n-1, i++$) **do** ▷ n = total element in Gt
  $C_i \leftarrow Gt_i$ and Gt_{i+1} ▷ C = list of clusters with their members
end for
for ($i = 0, i<m, i++$) **do** ▷ m = total element in N_d
  for ($j = 0, j<k, j++$) **do** ▷ k = total element in C
    $d_1 \leftarrow$ euclidean Distance(N_d^i, C_j^0)
    $d_2 \leftarrow$ euclidean Distance(N_d^i, C_j^1)
    $W_j \leftarrow (d_1 + d_2)/2$ %%
    $W_j \leftarrow w$
  end for
  $idx \leftarrow W.index(min(W))$
  $C_{idx}.append(N_d\hat{\ }i)$
end for

Over the stage of cluster determination, several modifications are performed according to the K-NN previously

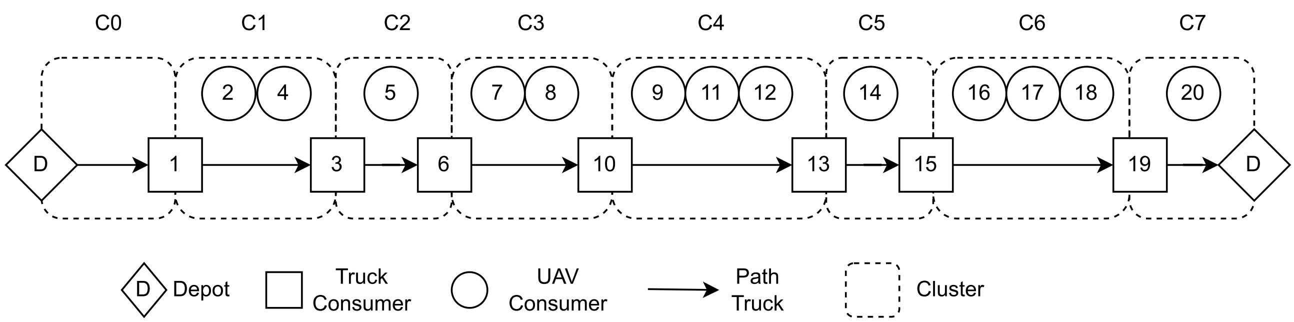

FIGURE 5. **Diagram K-NN clustering.**

implemented by Mazin et al. [48]. And some of these steps are shown in Alg. 4. In the first step of cluster creation, a cluster member from (N_t) is created whose routing has been formed. Then the cluster members going to i (C_i) and entering Gt_i and Gt_{i+1} are determined. Up to this stage, each cluster contained two members from N_t. Subsequently, the members of the cluster are defined as C_j^0 and C_j^1, and new cluster members are added from N_d by calculating the distance between each N_d and the members of each cluster using Eq. 8. The calculation of the distance from N_d to each cluster is shown in Alg. 4, where each N_d is a member of a cluster with the shortest distance from it. This clustering can be seen in Fig. 4, and the clustering results are illustrated in more detail in Fig. 5.

For instance, cluster 1 (*C1*) starts with Gt_1, which is the 1^{st} consumer, and ends with Gt_2, which is the 3^{rd} consumer. In addition, C1 has N_{d1} and N_{d2} as members, where N_{d1} is the 2^{nd} consumer, and N_{d2} is the 4^{th} consumer. The sequence of consumer numbers is listed in Table 4.

D. NEIGHBORHOOD ROUTING STRUCTURES

An explanation of the technique used to generate the routing path of the truck and the UAVs while sending parcels is outlined in this section. The location where the UAVs have loaded the parcel from the truck or depot is initialized as l_{dt}, and the point where the UAVs return to the truck after delivering parcels is initialized as m_{dt}.

Each UAV has loaded the parcels (l_i^u) from the truck or depot within t seconds before traveling to the intended consumer and unloading the parcels (ul_i^u) at the first consumer's location in cluster i (C_i). After unloading, the UAV flies to the next target and repeats the procedure until its cargo capacity is empty. When the delivery is complete, the UAV flies to the meeting point at the next consumer location (m_{dt}). When the UAV is about to make a delivery, it will take into account the maximum number of packages it can carry, its battery capacity, and the distance of each consumer. It is assumed that when a UAV carries more parcels, it consumes more energy to deliver them. The number of parcels the UAV can carry in this situation affects its speed, with a high number of parcels slowing it down and lengthening the travel time.

Algorithm 5 Synchronize Routing

input $\leftarrow R_t, R_d, c_j, C, U$
input $\leftarrow Max_{speed}d, Max_{speed}t, Max_{flighttime}d$
$r \leftarrow R_d$
set c_j as r_0
set the starting point of c_j in C as r_0
delete r_n
set r_{n+1} with the reconvene point of r_n in C
setisfied $\leftarrow$ *False*
while !*setisfied* **do**
  if total distance of route $r <= Max_{flighttime}$^d **then**
    if total parcels of route $r <= U$ **then**
      setisfied $\leftarrow$ *True*
      $R_s.append(r)$
    else
      delete r_n and r_{n-1} from r
      set $r_n + 1$ with the reconvene point of r_n in C
    end if
  else
    delete r_n and r_{n-1} from r
    set $r_n + 1$ with the reconvene point of r_n in C
  end if
end while

For the synchronized routing algorithm, the output of Alg. 5 represents the routing results for parcel delivery using truck and UAVs collaboration.

The routing results of Alg. 5 can be seen in Fig. 6, and more details are shown in Fig. 7. It can be seen that consumers number 12, 17, and 18 are not served in the delivery of parcels. This is owing to the influence of the algorithm that limits the number of parcels that can be carried by a UAV, the maximum distance traveled by the UAVs, and the impossibility of UAVs departing from one location and returning to the same location. To solve this problem, a VNPRPTmD algorithm is proposed in Alg. 6.

After routing, if any consumer parcels are not delivered by the UAVs (nr_d), N_{dn} is transformed into N_{tn}. However, not all N_d are transformed into N_{tn}; thus, a transfer scheme is performed by checking the location of N_d. This is done

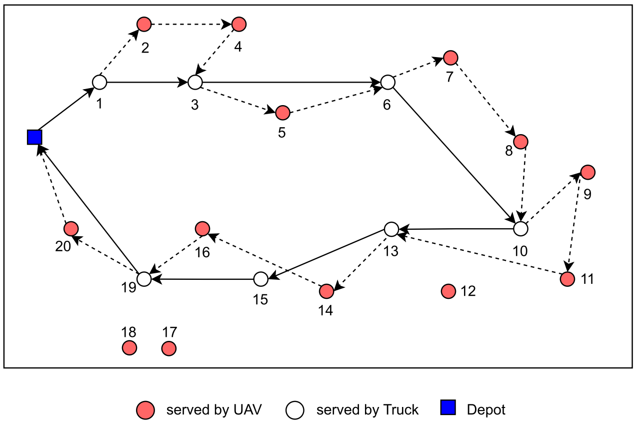

FIGURE 6. Initial routing truck and multi UAV for the first step.

Algorithm 6 VNPRPTmD

$R_t \leftarrow PathRouting(N_t)$
$R_d \leftarrow PathRouting(N_d)$
$C \leftarrow K-NNClustering(R_t, N_d)$
$c \leftarrow SynchronizeRouting$
$D \leftarrow UAVScheduling(d)$
$P \leftarrow$ UAV-on-truck positions
for $(i = 0, i < n, i++)$ **do** ▷ n = total element in C
  if C_i has N_d that needs to be served **then**
    $c \leftarrow N_d \in C_i$
    for $(j = 0, j < m, j++)$ **do** ▷ m = total member in c
      if there is a UAV available **then**
        $R_s.append(RoutingAlgorithm())$ ▷ R_s = list of UAV route
        if c_j cannot be served **then**
          $N_t.append(c_j)$
          $MainAlgorithm(N_t, N_d)$
        end if
      else
        $N_t.append(c_j)$
        $MainAlgorithm(N_t, N_d)$
      end if
    end for
    if there is an unused UAV **then**
      $P \leftarrow moveUAVsWithTruck(P)$
    end if
  else
    $P \leftarrow moveUAVsWithTruck(P)$
  end if
end for

sequentially based on the order of the clusters, where the consumer is transformed as the nr_d, which is closest to tc_1^n in that cluster. After one nr_d has been transformed into N_{tn}, then routing for every N_t is regenerated using a heuristic algorithm.

Algorithm 7 UAV Schedulling

repeat
  for $UAV > 0$ **do**
    $r \leftarrow$ Select trip from list
    $u \leftarrow$ Select the earliest available UAV
    $u \leftarrow$ Assign trip r
    Remove r from List
  end for
until List = 0

TABLE 5. List of parameters.

a. Simulation parameters		
Parameter	***Value***	***Condition***
UAV Speed	10-20 m/s	Based on payload
Truck Speed	10 m/s	Static
Consumer	20 - 90 location	Based on dataset
Consumer Parcel	2, 2.5 and 4 kg	Small parcel
Consumer Parcel	5 kg	Big parcel
l_i^u	0,01 s	Static
ul_i^g	0,01 s	Static
ul_j^u	0,01 s	Static
t_i^o	10-20 m/s	Based on payload height 3 meter
t_j^l	10-20 m/s	Based on payload height 3 meter
b. Parameters for tuning NSGA-II		
Parameter	***Value***	
Population size	500	
Number of generations	1000	
Crossover probability	0.9	
Crossover distribution index	10	
Mutation distribution index	20	

After the N_d is reclustered, the UAV and truck routing is regenerated. Afterward, nr_d is rechecked. This process continues until nr_d is no longer observed. For the algorithm itself, the results based on Alg. 6 are shown in Fig. 8, and more details are presented in Fig. 9 and Fig. 10 where it can be seen that all consumer deliveries can be served by using the truck and UAVs. It can also be seen how to select the UAVs that will be used to deliver the parcel in Alg. 7.

VI. RESULTS ANALYSIS

A. EXPERIMENTAL DESCRIPTION

This section presents the results of the experimental tests, which were conducted using the previously described VNPRPTmD algorithm in Section V-D. In order to assess the efficacy of our system, experiments were carried out using diverse sample sizes derived from existing research. The experiments were conducted on a Windows-based desktop computer with an Intel Core i7 2.8 GHz processor and 16 GB RAM. The implementation used Python code.

Operational parameters used in NSGA-II are listed in Table 5-b. The table illustrates key parameters utilized in tuning the NSGA-II. It includes a population size of 500, which indicates the number of candidate solutions evaluated in each generation. There are 1000 generations through which the algorithm iterates, refining solutions at each step. A high

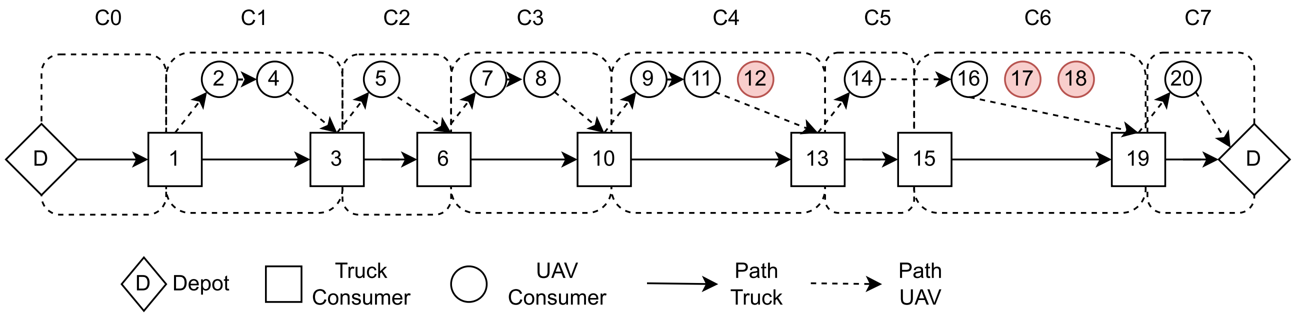

FIGURE 7. Diagram of initial routing truck and multi UAV for the first step.

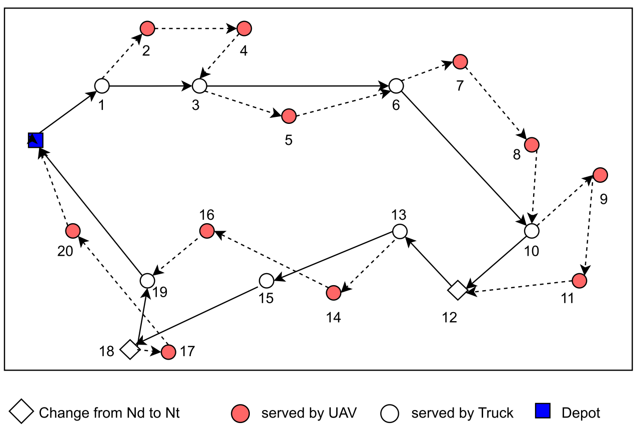

FIGURE 8. Routing truck and multi UAV after re-clustering.

crossover probability of 0.9 ensures that there is a significant blending of genetic material between pairs of solutions, aiming to explore a diverse solution space effectively. The crossover distribution index set at 10 and the mutation distribution index set at 20 dictate the granularity and range of the genetic variations introduced during crossover and mutation events, respectively. These parameters collectively fine-tune the algorithm's performance to balance exploration and exploitation, aiming to find a diverse set of optimal solutions efficiently.

Owing to changes in the speed, weight, and dimensions of the UAV, the number of parcels carried is affected by the battery power consumption (*kb*) of the UAV [34], [43]. The testing parameters can be seen in Table 5-a. The UAV's maximum speed is 20 meters per second without additional weight, while its maximum carrying capacity is 4 kg. The velocity of the UAV decreases proportionally with the load it carries, resulting in a speed of merely 10 meters per second when carrying the maximum load. This relationship is formulated in the mathematical model in Eq. 18.

$$S_{ij}^{u} = (-2.5 * payload) + 25 \tag{18}$$

In our simulation, Eq. 18 shows the relationship between UAV speed and payload (the load carried by the UAV). A constant of -2.5 indicates the impact of each kilogram of payload on the speed reduction. Thus, for every kilogram of load added, the UAV's speed decreases by 2.5 meters per second. The number 20 is the maximum speed of the UAV without a payload, which is 20 meters per second. For example, if the payload = 0 kg, then the UAV's speed = 20 m/s. If the payload = 4 kg, then the UAV's speed = (IV)-2.5 × 4) + 20 = 10 m/s. Eq. 18 shows that the UAV's speed decreases linearly as the load increases.

B. PERFORMANCE OF SYSTEM MODEL

The first test is conducted to evaluate the combination of NSGA-II [42] and 2-opt to serve all consumers for the truck routing with limited capacity in consideration. The finest outcomes are obtained when an improved algorithm is implemented. The computational results of benchmark problems were obtained by running the coded algorithm on a computer.

As shown in Fig. 11, the test results of the TSP algorithm using one truck. The behavior of the agent-based algorithm has been tested when applied to a set of different-size instances from the literature [49] by modifying the payload for each consumer. Meanwhile, the location of the consumers remains the same. The authors considered only the UAV to serve the consumer, whereas our proposed model uses UAVs and a truck to serve the consumer. In addition, to determine the number of parcels ordered by consumers, they are categorized as small and large-sized parcels. A small-sized parcel implies that a truck or a UAV can perform delivery, while a large-sized parcel implies that a truck can only deliver the ordered parcel. Testing comparisons are made with SA literature [50] and Genetic Algorithm (GA) literature [51]. Fig. 11 depicts a schematic representation of the experimental outcomes related to the transportation of parcels only by utilizing a truck. Our proposed system shows greater time efficiency compared to both GA and SA. The dataset has an average value of 90-A-1 to 90-A-10, indicating the presence of ten distinct datasets comprising 90 consumers in each. The geographical distribution of consumers and the composition of their ordered parcels vary across each dataset. The obtained results for this proposed model indicate a time of 169.98 seconds. In contrast, the time taken by the GA

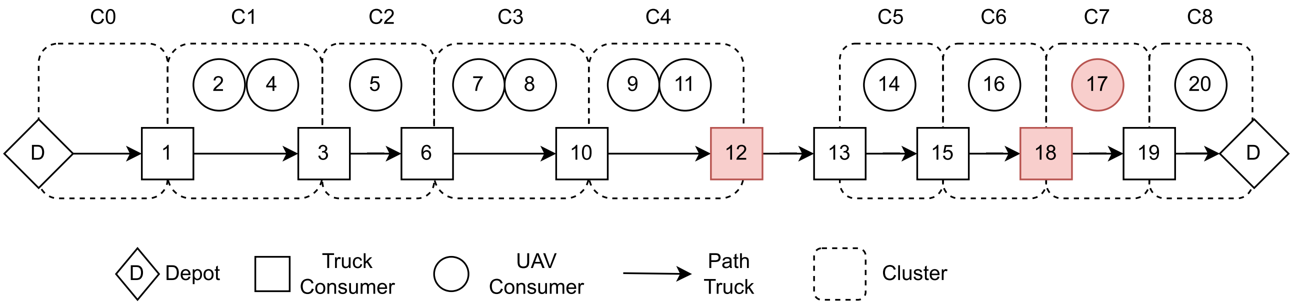

FIGURE 9. Diagram after reclustering.

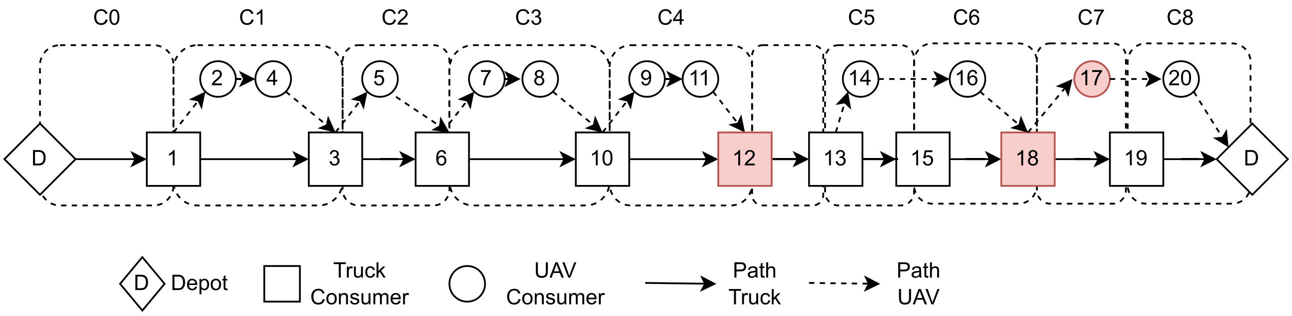

FIGURE 10. After re-routing.

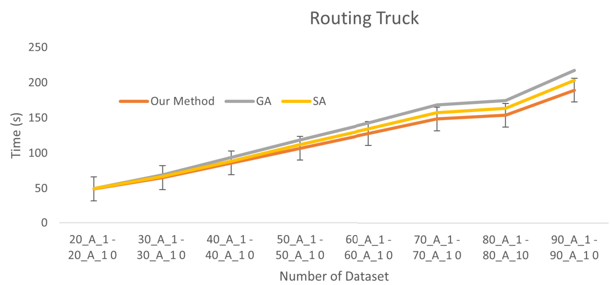

FIGURE 11. The average time traveled by truck.

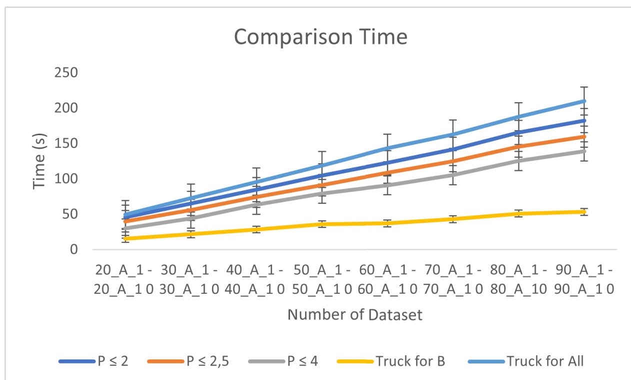

FIGURE 12. The time difference between a truck and a multi-UAV with a variety of maximum payloads.

and SA approaches are 195.23 seconds and 182.45 seconds, respectively.

Presented in Fig. 12 is a comparison of the travel time for parcel delivery between VNPRPTmD and truck-only systems. VNPRPTmD itself is divided into three categories based on the maximum payload for the UAV: $P_{uav} \leq 2\ kg$ (maximum UAV payload of less than or equal to 1 kg), $P_{uav} \leq 2.5\ kg$ (maximum UAV payload of less than or equal to 2 kg), and $P_{uav} \leq 4\ kg$ (maximum UAV payload of less than or equal to 3 kg). Each dataset has a different number of consumers, and each consumer has different orders. For example, for the dataset 90-A-1 to 90-A-10, there are 90 consumers, with 10 models consisting of 90 consumers each. Each consumer orders a different parcel. Each consumer can only order one parcel weighing 2 kg, 2.5 kg, 4 kg, or 5 kg. Setting the maximum payload for the UAV at $P_{uav} \leq 4kg$ implies that the UAV can only deliver one 4 kg or two parcels weighing 2 kg each. The 5 kg category falls into the heavy category (B), which only the truck can service. Fig. 12 demonstrates that the collaboration of the UAV and truck takes less time to deliver the parcel compared with a single truck delivery scenario. Moreover, when a UAV carries a large payload, the proposed system's efficiency declines a little bit, but it is still better than the only truck-based parcel delivery scenario.

As illustrated in Fig. 13, an overview is presented of a comparative analysis focusing on the time required for parcel delivery through a collaborative system combining UAVs and trucks. The depicted scenario assumes a UAV speed of 20 m/s

TABLE 6. Number of UAV delay at meeting point.

S_{UAV} (m/s)	Speed Truck 10 m/s			Speed Truck 5 m/s		
	$P_{uav} \leq 2kg$	$P_{uav} \leq 2.5kg$	$P_{uav} \leq 4kg$	$P_{uav} \leq 2kg$	$P_{uav} \leq 2.5kg$	$P_{uav} \leq 4kg$
5	54	68	69	47	70	74
10	54	68	69	47	70	74
15	50	64	67	47	70	74
20	43	61	63	47	70	74

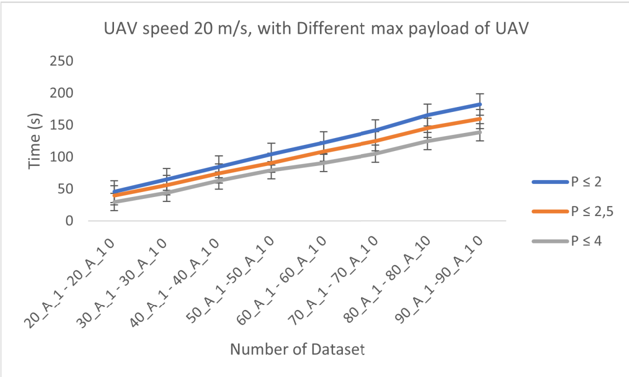

FIGURE 13. UAV flight duration with speed 20 meters per second with varying payloads.

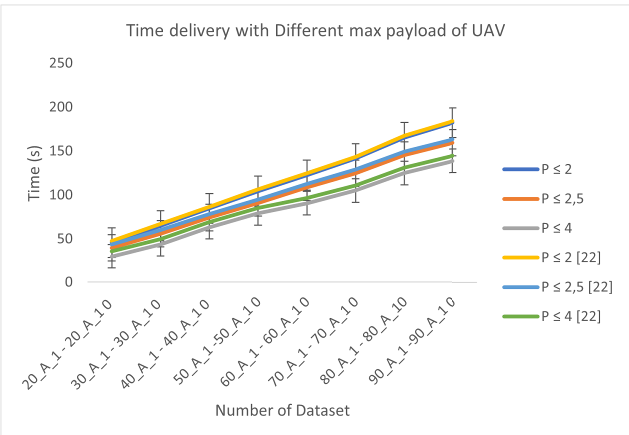

FIGURE 14. A comparison of delivery times for various maximum payload capacities.

and a truck speed of 10 m/s. The findings indicate that there is a positive correlation between the load capacity of the UAV and the speed of delivery.

Presented in Fig. 14, parcel delivery times are compared across various maximum load sizes that the UAV can carry. Additionally, the TmDTL model is written according to the algorithm architecture in the literature [25] for comparison with the model presented. The result shows that VNPRPTmD has better values than TmDTL. This is because the UAV in TmDTL can not deliver the parcel to another consumer in a different cluster, but our proposed model can efficiently solve this drawback and deliver the parcel. As seen in datasets 90-A-1 to 90-A-10, the TmDTL value for a maximum UAV load of 4 kg is 144.67 seconds, while the VNPRPTmD value is 138.67 seconds.

The UAV's expected arrival time at the designated rendezvous point with the truck is detailed in Table 6. A different dataset comprising 20-A-1 to 90-A-10 was utilized for this test. UAV speeds were set to 5 m/s, 10 m/s, 15 m/s, and 20 m/s, while the truck speed was maintained at 10 m/s and 20 m/s for the initial test. The pace of the UAV remains consistent despite variations in the payloads. It is evident from Table 6 that the number of delays in the UAV's arrival at the rendezvous point decreases with increasing speed.

In addition, a series of tests were performed to evaluate the efficacy of our proposed algorithm compared to existing ones, employing various testing situations that differed from the previous scenario. Each consumer is categorized as type one, two, and three, and large-sized parcels. The numbers 1, 2, and 3 represent the number of parcels ordered by the consumer, which means that delivery can be performed by truck or UAV. A large-sized parcel implies that the parcel ordered by the consumer is large and can only be delivered by truck. In addition, the TmDTL algorithm is written according to the algorithm architecture in the literature [25] and the Adaptive Tabu Search Algorithm (ATSA) is written according to the architecture in the literature [52] models are compared, and each algorithm is given the same data for each experimental test. In our simulation, the speed of the UAV is two times faster than the speed of a truck. The experimental test that is related to the time it takes the UAV to deliver a parcel is shown in Table 7. Total UAV is the number of UAVs used in performing the parcel delivery, and the consumer parcel is the number of parcels delivered to each consumer. For example, if the consumer parcel is 1, it means that 70% of all consumers have ordered parcel type 1, and 30% of them have ordered a large-size parcel.

When multiple UAVs are utilized, the travel time and distance of each UAV varies according to its route. Based on Table 7, which represents the average flight time for the UAV to deliver parcels, it can be seen that the VNPRPTmD algorithm has the fastest result compared to other algorithms.

When all consumers order type-2 and type-3 parcels, the routing result is the same because the maximum number of parcels that a UAV can carry during a flight is three. However, the time required for delivery varies because the UAV speed is affected by the number of parcels.

Table 8 as a continuation of Table 7 and only describes the truck's travel time for each delivery. The first column of Table 8 shows the total use of UAVs. The next column is the

TABLE 7. The average time (in seconds) traveled with UAV. (Dataset: 20-A-1 to 20-A-10).

Total UAV	No Consumer Parcel	VNPRPTmD			ATSA			TmDTL		
		uav 1	uav 2	uav 3	uav 1	uav 2	uav 3	uav 1	uav 2	uav 3
1	1	20.256	-	-	28.719	-	-	25.118	-	-
1	2	23.295	-	-	29.736	-	-	23.299	-	-
1	3	24.478	-	-	31.091	-	-	24.488	-	-
2	1	12.601	10.021	-	16.580	12.138	-	12.601	10.021	-
2	2	17.947	15.243	-	17.149	12.586	-	17.947	16.143	-
2	3	18.738	16.083	-	17.907	13.184	-	18.738	17.083	-
3	1	9.186	7.092	5.388	14.283	98.40	4.596	9.886	6.216	6.578
3	2	14.173	15.741	11.064	14.780	10.217	4.738	13.213	16.771	11.214
3	3	14.736	16.365	11.707	15.444	10.720	4.928	13.646	16.776	12.395

TABLE 8. The average time (in seconds) traveled by truck. (Dataset: 20-A-1 to 20-A-10).

T UAV	CP	VNPRPTmD	TmDTL	ATSA
1	1	36.207	38.571	60.731
	2	35.872	39.322	61.748
	3	35.872	39.925	63.104
2	1	31.613	35.524	53.467
	2	34.366	36.492	54.218
	3	35.396	37.723	55.219
3	1	30.281	30.987	49.067
	2	32.201	34.215	49.704
	3	33.212	35.421	50.552

total number of payloads that can be allowed by UAV (*CP*). The travel time for the truck is shorter with an increasing number of UAVs, as listed in Table 8. The time taken by the truck is the sum of travel time, loading time of the parcels into the UAV, unloading the parcels at the consumer location, and the waiting time for the UAV if the truck arrives first at the meeting location.

In Table 8, it can be observed that the truck's travel time in the VNPRPTmD algorithm is faster than that in the others because of the UAV's capability to deliver multiple parcels in one flight by maximizing its travel time, distance, and capacity. The data in Table 8 show that the delivery time will be faster because the number of UAVs used to assist the truck is increased. Trucks only serve consumers who order large packages and consumers whose distance to their location exceeds the maximum range of the UAV. As shown in Table 8 in the condition when every consumer orders one parcel, the time needed for a Truck to send a parcel assisted by one UAV is 36.207 seconds, assisted by two UAVs takes 31.613 seconds and 30.281 seconds when assisted by three UAVs. Therefore, travel time is affected when the number of UAVs is varied.

Table 9 displays a set of attempted benchmarks: the type A instances from Augerat (1995). The first field in the name refers to the problem type, where *A* refers to the Asymmetric Capacitated Vehicle Routing Problem (ACVRP) instances. The problem instances were downloaded from the Capacitated Vehicle Routing Problem Library [54]. The first column of Table 9 shows the name of each problem. The next three columns are the most important characteristics, namely the number of nodes (n), the capacity of the vehicles (Q), and the number of vehicles (K). Columns 5–7 display the BKS's best, worst, and average. Columns 8–10, respectively, show the best, the worst, and the average from 10 tests for each dataset. Columns 11–13, respectively, show the quality of the solutions produced and measure the efficiency of the proposed algorithm. The quality is given in terms of the percentage relative deviation from the best-known solution, calculated using the formula:

$$Best = 100 \times \left(\frac{{}^{2}\text{Best} - {}^{1}\text{Best}}{{}^{1}\text{Best}}\right) \tag{19.a}$$

$$Worst = 100 \times \left(\frac{{}^{2}\text{Worst} - {}^{1}\text{Worst}}{{}^{1}\text{Worst}}\right) \tag{19.b}$$

$$Average = 100 \times \left(\frac{{}^{2}\text{Average} - {}^{1}\text{Average}}{{}^{1}\text{Average}}\right) \tag{19.c}$$

The formulas presented are used to measure a proposed solution's performance deviation compared to the BKS in optimization problems. Specifically, the formulas calculate the percentage deviation for the best, worst, and average results obtained by the proposed method relative to the BKS. The Best indicates how much the best result of the proposed method deviates from the known optimal result, the Worst measures the deviation of the worst result, and the Average captures the overall deviation of the average performance. These metrics are useful in determining whether a new optimization approach performs better, worse, or similarly to existing benchmark solutions, with positive values indicating inferior performance and negative or zero values suggesting comparable or superior performance. Based on Table 9 showing the percentage deviation (deviation (%)) for the best, worst, and average results of the proposed solution compared to the BKS, the following conclusions can be drawn: Best: Most of the values in the "Best" column are close to zero or negative, indicating that the best results of the proposed method are generally very close to, or in some cases even better than, the BKS for most problems. Negative values suggest that the proposed solution has outperformed

TABLE 9. Computational results for the problem set A.

Problem	n	Q	K	BKS			Proposed Model			Deviation (%)		
				[1]Best	[1]Worst	[1]Average	[2]Best	[2]Worst	[2]Average	Best	Worst	Average
A-n32-k5	32	100	5	784	784	784	784	786	785	0.000	0.255	0.128
A-n33-k5	33	100	5	661	661	661	661	661	661	0.000	0.000	0.000
A-n33-k6	33	100	6	742	742.9	742.9	742	742	742	0.000	-0.121	-0.121
A-n34-k5	34	100	5	778	782	782	778	780	779	0.000	-0.256	-0.384
A-n36-k5	36	100	5	799	808.8	808.8	799	802	801	0.000	-0.841	-0.964
A-n37-k5	37	100	5	669	673.2	673.2	669	669	669	0.000	-0.624	-0.624
A-n37-k6	37	100	6	949	949.7	949.7	949	949	949	0.000	-0.074	-0.074
A-n38-k5	38	100	5	730	731.2	731.2	730	730	730	0.000	-0.164	-0.164
A-n39-k5	39	100	5	822	823.5	823.5	822	824	823	0.000	0.061	-0.061
A-n39-k6	39	100	6	831	832.5	832.5	831	832	832	0.000	-0.060	-0.060
A-n44-k6	44	100	6	937	938.5	938.5	937	938	938	0.000	-0.053	-0.053
A-n45-k6	45	100	6	944	949.1	949.1	944	947	946	0.000	-0.221	-0.327
A-n45-k7	45	100	7	1148	1164	1163.5	1146	1150	1149	-0.174	-1.160	-1.246
A-n46-k7	46	100	7	914	920.7	920.7	914	918	917	0.000	-0.293	-0.402
A-n48-k7	48	100	7	1073	1095	1095.4	1073	1090	1086	0.000	-0.493	-0.858
A-n53-k7	53	100	7	1010	1018	1017.8	1010	1015	1013	0.000	-0.275	-0.472
A-n54-k7	54	100	7	1172	1185	1184.5	1172	1186	1184	0.000	0.127	-0.042
A-n55-k9	55	100	9	1073	1084	1083.6	1073	1080	1078	0.000	-0.332	-0.517
A-n60-k9	60	100	9	1354	1374	1373.7	1354	1370	1365	0.000	-0.269	-0.633
A-n61-k9	61	100	9	1037	1044	1043.7	1037	1043	1041	0.000	-0.067	-0.259
A-n62-k8	62	100	8	1299	1325	1324.7	1288	1299	1295	-0.847	-1.940	-2.242
A-n63-k9	63	100	9	1621	1634	1633.7	1616	1625	1625	-0.308	-0.533	-0.533
A-n63-k10	63	100	10	1319	1327	1327.3	1320	1330	1326	0.076	0.203	-0.098
A-n64-k9	64	100	9	1418	1439	1439.1	1410	1420	1415	-0.564	-1.327	-1.675
A-n65-k9	65	100	9	1177	1182	1182.1	1180	1182	1181	0.255	-0.008	-0.093
A-n69-k9	69	100	9	1169	1181	1181	1162	1174	1172	-0.599	-0.593	-0.762
A-n80-k10	80	100	10	1804	1834	1833.5	1763	1768	1765	-2.273	-3.572	-3.736

[1] Best-known solution (BKS) results provided by Awad et al. [53].
[2] Proposed model.

the BKS in some instances (e.g., problem A-n80-k10 with a -2.273% deviation). Worst: The "Worst" column shows a wider range of deviation. In several instances, the proposed method's worst performance has deviated positively from the BKS, implying worse performance (e.g., A-n38-k5 with 1.215% deviation). However, there are also cases where the worst result was better than the BKS, with negative deviations. Average: The "Average" column shows that, for many problems, the average performance of the proposed solution is close to the BKS, with several deviations being less than 1% (e.g., problem A-n32-k5 with 0.128%). However, some problems display more significant positive deviations, indicating that the average performance is worse than BKS in certain cases (e.g., problem A-n80-k10 with a 3.736% deviation). In summary, the proposed solution generally performs well compared to BKS, achieving near-optimal or better best-case results. While its performance varies more significantly in worst-case and average-case scenarios, the overall performance is reassuring. Some problems exhibit notable performance improvements, while others show degradation, particularly in the worst and average results.

C. BENCHMARK INSTANCES

A benchmark dataset derived from VRP-D [43] is utilized in this section to assess the efficacy of our solutions more efficiently. A new collection of instances has been produced in light of the fact that VRP-D-MC is a novel contribution to the field. These instances are derived from the benchmark instances developed by Pugliese et al. [36] for the UAV delivery problem, which are altered iterations of the instances originally proposed by Solomon et al. [55] for the widely recognized vehicle routing problem with time windows (VRPTW) benchmark. C1, C2, R1, R2, RC1, and RC2 are the six categories that comprise these samples. 50 to 100 customers or requests are included in each dataset. Category C1 and Category C2 contain every customer location. Consumers may specify a weight range of 1 kg to 100 kg for each shipment. Drone delivery is a prevalent method of transporting small-sized packages by Poikonen et al. [23]. Concepts, such as generation, were employed to capture the constraints associated with UAV delivery, as [36] suggested.

The algorithm was applied to instances of VRP-D by David et al. [29]. Table 10 summarizes [29] findings for small and medium-large instances, which range from 100 to 200 consumers. In contrast, our algorithm's performance is evaluated in relation to that of the Hybrid Genetic-sweep Algorithm (HGA) approach proposed by Euchi et al. [30], which is regarded as the cutting-edge algorithm in the field of VRP-D solutions for the instances of the Sacramento data set [29]. The objective in such cases is to reduce the overall travel time required to convey the package.

The findings from Table 10 in our study of the VNPRPTmD algorithm, executed over 1000 iterations, reveal that utilizing a single UAV yields results closely comparable to those obtained using the HGA algorithm, as reflected by

TABLE 10. Comparison with the BKS on medium and large instances.

Inst.	HGA [30]		VNPRPTmD 1 UAV				VNPRPTmD 2 UAV			
	HGA[1]	CPU (s) [1]	1 P/UAV[2]	CPU (s)	2 P/UAV[3]	CPU (s)	1 P/UAV[2]	CPU (s)	2 P/UAV[3]	CPU (s)
100.10.1	6.36332	3.306	6.365	2.000	5.863	1.991	5.7633	1.986	5.4633	1.9855
100.10.3	8.11945	3.32	8.121	2.060	7.619	2.062	7.5195	2.061	7.2195	2.0565
100.20.1	12.541	3.332	12.556	2.644	12.041	2.664	11.941	2.66	11.641	2.6585
100.20.2	14.0118	6.339	14.012	4.189	13.512	4.186	13.4118	4.184	13.1118	4.1805
100.30.1	23.5686	6.356	23.588	5.77	22.088	5.76	21.9882	5.755	21.6882	5.7545
100.30.2	22.0396	7.375	23.040	6.133	21.54	6.126	21.4396	6.12	21.1396	6.1205
100.40.1	29.1397	9.427	29.940	7.610	28.64	7.604	28.5397	7.599	28.2397	7.5985
100.40.2	30.361	10.446	31.361	8.129	29.861	8.129	29.761	8.13	29.461	8.1235
150.10.1	8.5865	0.495	8.790	1.0015	8.29	1.001	8.1903	1.001	7.8903	1.000
150.10.2	8.14615	0.994	8.259	0.9	7.759	0.899	7.6591	0.889	7.3591	0.7965
150.20.1	17.0099	2.323	17.319	1.972	16.819	1.862	16.7194	1.859	16.4194	1.8565
150.20.2	16.1837	2.612	16.634	2.141	16.134	2.133	16.0341	2.128	15.7341	2.1275
150.30.1	25.8524	3.489	25.985	2.611	25.485	2.608	25.3854	2.607	25.0854	2.6025
150.30.2	26.137	3.569	26.296	2.901	25.706	2.898	25.6055	2.895	25.3055	2.8925
150.40.1	29.4683	6.477	34.012	4.226	29.512	4.211	29.4121	4.202	29.1121	4.2055
150.40.2	35.2343	6.997	36.562	5.956	36.062	5.956	35.9616	5.956	35.6616	5.9505
200.10.1	10.0956	10.446	10.995	12.60	9.595	12.6	9.4945	12.596	9.1945	12.5945
200.10.2	10.2855	10.597	10.423	12.666	9.923	12.758	9.8226	12.66	9.5226	12.7525
200.20.1	21.2185	15.755	21.715	15.090	20.715	15.087	20.6151	15.082	20.3151	15.0815
200.20.2	21.0193	17.884	21.988	15.265	20.958	15.245	20.8585	15.24	20.5585	15.2395
200.30.1	30.073	19.699	31.256	19.001	29.86	18.553	29.7602	18.549	29.4602	18.5475
200.30.2	32.0915	19.835	32.813	19.900	32.313	19.889	32.2128	19.881	31.9128	19.8835
200.40.1	41.2954	25.568	42.498	21.624	40.998	22.604	40.898	22.599	40.598	22.5985
200.40.2	43.0717	26.182	44.250	24.500	42.75	24.033	42.6502	24.03	42.3502	24.0275

[1] HGA algorithm provided by [30].Programmed in C++ and executed on Intel Core i5-2450 M with 2.50 GHz and 4 GB of RAM.
[2] 1 P/UAV = The maximum number of parcels a UAV can carry per delivery is 1.
[3] 2 P/UAV = The maximum number of parcels a UAV can carry per delivery is 2.

the data in the columns for HGA and VNPRPTmD with one UAV. When the number of UAVs is increased from one to two, there is a noticeable improvement in the VNPRPTmD algorithm's performance. This enhancement is demonstrated by the reduced CPU time required and the better correlation with the HGA results.

In the simulation, the UAV model is distinguished from the BKS model by the capabilities of the UAVs. While the BKS model limits each UAV to delivering only one parcel per flight, the enhanced UAV functionality allows for the delivery of multiple parcels in a single trip. This improvement leads to a more efficient routing strategy, especially beneficial in areas with dense consumer clusters and modest parcel demands. The same dataset was used to facilitate a direct comparison with the BKS benchmark, maintaining the consumer coordinates to ensure that performance improvements are attributed to the methodology rather than geographic differences. However, the demand parameters of the dataset were adapted to accommodate the increased capabilities of the UAVs. Specifically, UAVs are configured to service consumers whose demands do not exceed 3.0 kg, while higher demands are handled by trucks, which are better equipped for larger loads. These strategic modifications in demand management ensure that the dataset aligns with the UAV model's capabilities without compromising the fundamental challenges of the original BKS problem. By only modifying the demand parameters and not the locations or the essential nature of the deliveries, the integrity of the BKS benchmark is upheld while demonstrating the efficiency of the enhanced UAV delivery model.

TABLE 11. Details of the solutions from a proposed algorithm. 1 Truck and 2 UAV with max payload UAV 10 kg.

Inst	VRP Solution		Proposed			
	TD (m)	TT (s)	UC	TC	TT (s)	CPU Time (minute)
C101	690,58	271	49	51	135,9	1.12
C201	732,8	275,2	50	50	150,5	1.24
R102	833,9	285,3	40	60	132,2	0.88
RC101	797,6	281,7	40	60	140,2	1.21

Regarding CPU time efficiency, adding more UAVs reduces the time needed for data processing, suggesting that deploying multiple UAVs can lead to more efficient operations. Furthermore, the VNPRPTmD algorithm, when used with two UAVs, displays better scalability and enhanced performance in larger scenarios, particularly when the delivery demands involve two packages per consumer. This situation presents more complex optimization challenges and highlights the advantages of utilizing two UAVs to manage increased delivery demands.

According to Table 11, the maximum payload for UAVs is 10 kg, as indicated. Additionally, Table 12 states that the maximum payload for UAVs is 20 kg. The first column of Table 11 and 12 shows the name of each problem. The next four columns are the most important characteristics, namely the truck delivery (*TD*), the total travel time for VRP solution

TABLE 12. Details of the solutions from a proposed algorithm. 1 truck and 2 UAV with max payload UAV 20 kg.

Inst	VRP Solution		Proposed			
	TD (m)	TT (s)	UC	TC	TT (s)	CPU Time (minute)
C101	690,58	271	60	40	110,9	1.08
C201	732,8	275,2	65	35	108,4	1.14
R102	833,9	285,3	67	33	126,4	1.06
RC101	797,6	281,7	65	35	125,7	1.43

(TT), the consumer served by UAV (UC), the consumer served by truck (TC), the total travel time for our proposed method truck multi UAV (TT), and CPU time. The velocity of the truck is 10 meters per second, the duration of unloading at each customer is 1 second, and the maximum velocity of the UAV is 20 meters per second. The data clearly demonstrates the advantages of integrating UAVs into delivery systems. The use of UAVs reduces the total delivery time, significantly decreases road traffic, and increases customer service efficiency. This dual delivery mode (UAVs and trucks) allows for more flexible, efficient, and faster delivery services, particularly useful in urban areas or regions where road traffic can significantly delay traditional delivery methods. Although the CPU time is slightly increased in some cases, it remains competitive, indicating that the algorithm can be implemented without significant computational overhead. Overall, the proposed algorithm outperforms traditional VRP solutions, highlighting its potential to optimize logistics and delivery systems effectively.

VII. CONCLUSION AND FUTURE WORK

In this study, the VNPRPTmD algorithm was developed to improve the efficiency and effectiveness of parcel delivery. A unique collaborative model was proposed to solve the problem of parcel delivery through the collaboration of multiple UAVs and a truck. A heuristic model was synchronized to determine the best location of the UAV for the initial delivery and the meeting point. Moreover, In our model, the Combination NSGA-II algorithm [42] along with the 2-opt algorithm has been successfully utilized to efficiently solve the CVRP problems. This approach has been proven to be superior to other methods in terms of solution quality, particularly when considering truck routing with limited capacity (Tabel 9). A VNP algorithm was utilized to facilitate the classification of each consumer based on the packages they ordered. Furthermore, heuristics with a VNP algorithm were utilized to generate separate routes for trucks and UAVs before integrating routes for all agents. In addition, the K-NN algorithm was utilized to cluster each consumer based on their location. Each cluster has to include two connected (N_t) or one (N_t) with a connected depot based on the routing results generated by the heuristic with the VNP algorithm. Finally, the experimental results have shown that the proposed model performed better than other existing literature for time estimation (Tabel 10); out of the 24 datasets tested, 8 datasets showed that the quality of our algorithm was better. Moreover, this suggested approach has eliminated the conventional problem where waiting time increases proportionately as more UAVs arrive at a rendezvous point.

In future research, a significant area of focus will be the integration of collision avoidance mechanisms into VRP. Currently, our models optimize routes and schedules without explicitly considering the potential for vehicular collisions, which could impact the safety and reliability of the transport solutions. Incorporating dynamic collision avoidance strategies will enhance the routing algorithms' safety aspects and reflect more realistic operational scenarios. We aim to develop and implement sophisticated algorithms that can predict potential conflicts between vehicles, and this advancement will involve the use of real-time traffic data, machine learning techniques to predict traffic patterns, and advanced simulation tools to test the efficacy of collision avoidance algorithms under various conditions. Moreover, we will explore the integration of communication technologies that allow vehicles to share their positions and velocities, enabling a cooperative collision avoidance system that enhances the overall efficiency and safety of the logistic networks.

Future studies will also explore the integration of charging stations for UAV batteries into the operational framework. This inclusion has the potential to significantly extend the range and operational duration of UAVs, offering further improvements in logistical efficiency and cost-effectiveness.

By addressing these challenges, our future work will contribute to the theoretical aspects of VRP and significantly improve its practical applications, making it more applicable to complex and dynamic transportation environments. Enhance the study by exploring alternative distance metrics like Manhattan or Network Distance for better real-world route optimization. Integrating real-time traffic data could improve routing algorithms' dynamic adaptability. Additionally, expanding applications to diverse logistics networks and employing machine learning for predictive analytics would be valuable. Investigating sustainability impacts and regulatory challenges and utilizing cross-disciplinary approaches would further advance logistics and transportation research.

REFERENCES

[1] D. Group. (2019). *DHL Launches Its First Regular Fully-Automated and Intelligent Urban Drone Delivery Service*. Accessed: Nov. 18, 2024. [Online]. Available: https://group.dhl.com/de/presse/pressemitteilungen/2019/dhl-express-startet-drohnen-lieferservice-in-china.html

[2] L. Johnson. (2017). *9 Things You Need To Know About the Amazon Prime Air Drone Delivery Service*. Accessed: Nov. 18, 2024. [Online]. Available: https://www.digitalspy.com/tech/a820748/amazon-prime-air-drone-delivery-service/

[3] J. Lin. (2021). *Meet China's Growing Fleet of Automated Delivery Drones*. Accessed: Nov. 18, 2024. [Online]. Available: https://www.popsci.com/china-drone-deliveries/

[4] UPS. (2021). *UPS Operates First Ever US Drone COVID-19 Vaccine Delivery*. Accessed: Nov. 18, 2024. [Online]. Available: https://about.ups.com/ca/en/our-stories/innovation-driven/drone-covid-vaccine-deliveries.html

[5] J. Daleo. (2024). *Google's Wing Introduces New Delivery Drone With Double the Payload*. Accessed: Nov. 18, 2024. [Online]. Available: https://www.flyingmag.com/googles-wing-introduces-new-delivery-drone-with-double-the-payload/

[6] T. N. Memphis. (2023). *FedEx Plans To Test Autonomous Drone Cargo Delivery With Elroy Air*. Accessed: Nov. 18, 2024. [Online]. Available: https://newsroom.fedex.com/newsroom/global/elroyair

[7] E. Ackerman and E. Strickland. (2018). *Zipline Expands Its Medical Delivery Drones Across East Africa*. Accessed: Nov. 18, 2024. [Online]. Available: https://spectrum.ieee.org/zipline-expands-its-medical-delivery-drones-across-east-africa

[8] M. Joerss, J. Schroder, F. Neuhaus, C. Klink, and F. Mann. (2016). *Parcel Delivery: The Future of Last Mile*. Accessed: Nov. 18, 2024. [Online]. Available: https://bdkep.de/files/bdkep-dateien/pdf/2016_the_future_of_last_mile.pdf

[9] J. Gao, L. Zhen, G. Laporte, and X. He, "Scheduling trucks and drones for cooperative deliveries," *Transp. Res. E, Logistics Transp. Rev.*, vol. 178, Oct. 2023, Art. no. 103267.

[10] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105004.

[11] M. Sajid, H. Mittal, S. Pare, and M. Prasad, "Routing and scheduling optimization for UAV assisted delivery system: A hybrid approach," *Appl. Soft Comput.*, vol. 126, Sep. 2022, Art. no. 109225.

[12] B. Madani and M. Ndiaye, "Hybrid truck-drone delivery systems: A systematic literature review," *IEEE Access*, vol. 10, pp. 92854–92878, 2022.

[13] W. Rash. (2017). *UPS Tests Delivery Trucks Equipped With Drones As Efficiency Booster*. Accessed: Nov. 18, 2024. [Online]. Available: https://www.eweek.com/mobile/ups-tests-delivery-trucks-equipped-with-drones-as-efficiency-booster/

[14] M. R. Salama and S. Srinivas, "Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites," *Transp. Res. E, Logistics Transp. Rev.*, vol. 164, Aug. 2022, Art. no. 102788.

[15] G. Wu, N. Mao, Q. Luo, B. Xu, J. Shi, and P. N. Suganthan, "Collaborative truck-drone routing for contactless parcel delivery during the epidemic," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25077–25091, Dec. 2022.

[16] R. Macedo, C. Alves, J. M. V. de Carvalho, F. Clautiaux, and S. Hanafi, "Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model," *Eur. J. Oper. Res.*, vol. 214, no. 3, pp. 536–545, Nov. 2011.

[17] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, pp. 965–981, Apr. 2018.

[18] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 86–109, May 2015.

[19] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the min-cost traveling salesman problem with drone," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 597–621, Jan. 2018.

[20] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 368–398, Jan. 2020.

[21] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, no. 4, pp. 965–981, Aug. 2018.

[22] X. Wang, B. Golden, and E. Wasil, "A Steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem," *Comput. Oper. Res.*, vol. 101, pp. 200–219, Jan. 2019.

[23] S. Poikonen and B. Golden, "Multi-visit drone routing problem," *Comput. Oper. Res.*, vol. 113, Jan. 2020, Art. no. 104802.

[24] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A comparative analysis of synchronized truck-and-drone delivery models," *Comput. Ind. Eng.*, vol. 162, Dec. 2021, Art. no. 107648.

[25] J. M. Leon-Blanco, P. L. Gonzalez-R, J. L. Andrade-Pineda, D. Canca, and M. Calle, "A multi-agent approach to the truck multi-drone routing problem," *Exp. Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116604.

[26] J. Gao, L. Zhen, and S. Wang, "Multi-trucks-and-drones cooperative pickup and delivery problem," *Transp. Res. C, Emerg. Technol.*, vol. 157, Dec. 2023, Art. no. 104407.

[27] E. Es Yurek and H. C. Ozmutlu, "A decomposition-based iterative optimization algorithm for traveling salesman problem with drone," *Transp. Res. C, Emerg. Technol.*, vol. 91, pp. 249–262, Jun. 2018.

[28] Y. S. Chang and H. J. Lee, "Optimal delivery routing with wider drone-delivery areas along a shorter truck-route," *Exp. Syst. Appl.*, vol. 104, pp. 307–317, Aug. 2018.

[29] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 289–315, May 2019.

[30] J. Euchi and A. Sadok, "Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones," *Phys. Commun.*, vol. 44, Feb. 2021, Art. no. 101236.

[31] A. Ponza, "Optimization of drone-assisted parcel delivery," Ph.D. thesis. Università degli Studi di Padova, Italia, Apr. 2016.

[32] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, Aug. 2017.

[33] M. Moshref-Javadi, A. Hemmati, and M. Winkenbach, "A truck and drones model for last-mile delivery: A mathematical model and heuristic approach," *Appl. Math. Model.*, vol. 80, pp. 290–318, Apr. 2020.

[34] D. Baek, Y. Chen, A. Bocca, A. Macii, E. Macii, and M. Poncino, "Battery-aware energy model of drone delivery tasks," in *Proc. Int. Symp. Low Power Electron. Design*, Jun. 2018, pp. 1–6.

[35] Y. Choi and P. Schonfeld, "Optimization of multi-package drone deliveries considering battery capacity," in *Proc. 96th Annu. Meeting Transp. Res. Board*, Jan. 2017, pp. 8–12.

[36] L. D. P. Pugliese, F. Guerriero, and G. Macrina, "Using drones for parcels delivery process," *Proc. Manuf.*, vol. 42, pp. 488–497, Jan. 2020.

[37] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, "Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10483–10495, Dec. 2019.

[38] Y. Liu, Z. Liu, J. Shi, G. Wu, and W. Pedrycz, "Two-echelon routing problem for parcel delivery by cooperated truck and drone," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 12, pp. 7450–7465, Dec. 2021.

[39] Z. Luo, M. Poon, Z. Zhang, Z. Liu, and A. Lim, "The multi-visit traveling salesman problem with multi-drones," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103172.

[40] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[41] S. Meng, X. Guo, D. Li, and G. Liu, "The multi-visit drone routing problem for pickup and delivery services," *Transp. Res. E, Logistics Transp. Rev.*, vol. 169, Jan. 2023, Art. no. 102990.

[42] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[43] M. Amine Masmoudi, S. Mancini, R. Baldacci, and Y.-H. Kuo, "Vehicle routing problems with drones equipped with multi-package payload compartments," *Transp. Res. E, Logistics Transp. Rev.*, vol. 164, Aug. 2022, Art. no. 102757.

[44] G. K. B. Reinhardt, "Routing and scheduling problems," Ph.D. thesis, Dept. Manage. Eng., Tech. Univ. Denmark, 2011.

[45] H. Ma, Y. Zhang, S. Sun, T. Liu, and Y. Shan, "A comprehensive survey on NSGA-II for multi-objective optimization and applications," *Artif. Intell. Rev.*, vol. 56, no. 12, pp. 15217–15270, Dec. 2023.

[46] C.-K. Ting, C.-H. Su, and C.-N. Lee, "Multi-parent extension of partially mapped crossover for combinatorial optimization problems," *Exp. Syst. Appl.*, vol. 37, no. 3, pp. 1879–1886, Mar. 2010.

[47] R. J. Kuo, E. Edbert, F. E. Zulvia, and S.-H. Lu, "Applying NSGA-II to vehicle routing problem with drones considering makespan and carbon emission," *Exp. Syst. Appl.*, vol. 221, Jul. 2023, Art. no. 119777.

[48] M. A. Mohammed, M. K. A. Ghani, R. I. Hamed, S. A. Mostafa, D. A. Ibrahim, H. K. Jameel, and A. H. Alallah, "Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution," *J. Comput. Sci.*, vol. 21, pp. 232–240, Jul. 2017.

[49] N. Cabrera, J.-F. Cordeau, and J. E. Mendoza, "Solving the park-and-loop routing problem by branch-price-and-cut," *Transp. Res. C, Emerg. Technol.*, vol. 157, Dec. 2023, Art. no. 104369.

[50] H. Harmanani, D. Azar, N. Helal, and W. Keirouz, "A simulated annealing algorithm for the capacitated vehicle routing problem," in *Proc. CATA*, Jan. 2011, pp. 96–101.

[51] A. Kadar, M. Shahjalal, M. Faisal, and M. Iqbal, "Solving the vehicle routing problem using genetic algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, pp. 126–131, Aug. 2011.

[52] M. Moshref-Javadi, S. Lee, and M. Winkenbach, "Design and evaluation of a multi-trip delivery model with truck and drones," *Transp. Res. E, Logistics Transp. Rev.*, vol. 136, Apr. 2020, Art. no. 101887.

[53] H. Awad and R. Elshaer, "An effective genetic algorithm for capacitated vehicle routing problem," in *Proc. Int. Conf. Ind. Eng. Operations Manage.*, Mar. 2018, pp. 374–384.

[54] CVRPLIB. (2024). *Capacitated Vehicle Routing Problem Library*. Accessed: May 1, 2024. [Online]. Available: http://vrp.galgos.inf.puc-rio.br/index.php/en/

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

RAMDHAN NUGRAHA received the B.S. degree from the Department of Electrical Engineering, Universitas Pendidikan Indonesia (UPI), Indonesia, and the M.S. degree from the School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Indonesia. He is currently pursuing the Ph.D. degree with the Kumoh National Institute of Technology, Gumi, South Korea. He was a Lecturer with the Department of Electrical Engineering, Faculty of Electrical Engineering, Telkom University, Indonesia, for ten years. His research interests include robotics, unmanned aircraft systems, artificial intelligence, machine learning, and their application in autonomous vehicles.

MD MASUDUZZAMAN (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from American International University–Bangladesh (AIUB) and the Ph.D. degree in IT convergence engineering from the Wireless and Emerging Network System Laboratory (WENS Lab), Kumoh National Institute of Technology (KIT), Gumi, South Korea. He is currently a Postdoctoral Fellow at KIT. Then, he was a Lecturer at AIUB for four years. His major research interests include blockchain, cryptography, deep learning, edge computing, the Internet of Things (IoT), network security, and unmanned vehicles (UxV).

SOO YOUNG SHIN (Senior Member, IEEE) received the Ph.D. degree in electrical engineering and computer science from Seoul National University, in 2006. In 2007, he was a Postdoctoral Researcher with the University of Washington, Seattle, WA, USA, in 2007. From 2007 to 2010, he was with the WiMAX Design Laboratory, Samsung Electronics, Suwon, South Korea. In 2010, he joined as a full-time Professor with the School of Electronics, Kumoh National Institute of Technology (KIT), Gumi, South Korea. In addition, he was a Visiting Scholar with The University of British Columbia, Canada, in 2017. He is currently a Professor at KIT and the Founder of the Wireless and Emerging Network System Laboratory (WENS Lab). His research interests include 5G/6G wireless communications and networks, signal processing, the Internet of Things, mixed reality, and UAV applications.