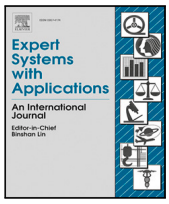
Expert Systems With Applications 269 (2025) 126434

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



A multi-objective approach for the integrated planning of drone and robot assisted truck operations in last-mile delivery

Ali Mokhtari-Moghadam [a,*], Abdellah Salhi [b], Xinan Yang [b], Trung Thanh Nguyen [a], Pourya Pourhejazy [c]

[a] School of Engineering, Liverpool John Moores University, Liverpool, L3 3AF, United Kingdom
[b] School of Mathematics, Statistics and Actuarial Science, University of Essex, Colchester, CO4 3SQ, United Kingdom
[c] Department of Industrial Engineering, UiT The Arctic University of Norway, Narvik 8514, Norway

ARTICLE INFO

Keywords:
Multi-objective optimization problem
Robot
Drone
Last mile delivery
E-commerce
Logistics

ABSTRACT

Supply chains are experiencing a major transition driven by changing customer expectations, environmental concerns, and technological development. Considering the surge in e-commerce since the pandemic, the path forward for affordable, responsive supply chains is autonomous last-mile delivery. Drone and robot technologies complement the last-mile delivery's operational requirements and hence should be incorporated to assist truck deliveries. This study develops a bi-objective optimization framework for the integrated planning of Drone-And-Robot-assisted Truck (DART) delivery operations to minimize total delivery cost and maximize customer satisfaction considering a soft time window. Three models, including DART, drone-, and robot-assisted trucks are compared considering different operational situations. The results show that the DART delivery mode outperforms with an increase in the number of demand points. DART is particularly preferred when there is a moderate combination of high-density and distant demand points in last-mile delivery. Numerical experiments confirmed that the robot-assisted delivery model brings about cost-effectiveness in heavily populated areas. On the other hand, the drone-assisted truck model stands out in situations where there is a small number of demand points with high dispersity.

1. Introduction

The surge in online shopping and e-commerce has accelerated the adoption of new technologies in Last-Mile Delivery (LMD). Autonomous technologies offer solutions to many of the old and emerging logistics challenges. The consumer acceptance of autonomous vehicles, aerial drones, sidewalk and bipedal robots for delivery services is raising (Kim & Hur, 2024; Said, Aeschliman, & Stathopoulos, 2023). In this situation, major logistics service providers are investing heavily in the adoption of these technologies to reduce operational costs and delivery time in LMD (Kitjacharoenchai, Ventresca, Moshref-Javadi, Lee, Tanchoco, & Brunese, 2019). The market for autonomous LMD was valued at USD 8.78 billion in 2020 and is expected to reach a high of 51 billion U.S. dollars in 2028. Statista (2022) has projected an even sharper growth rate late in the 2030s, surpassing 85 billion U.S. dollars.

The traditional delivery system uses trucks, which offer a large load capacity and high endurance in LMD. However, environmental footprint, and growing urbanization with dense urban areas are challenging truck deliveries. Besides, the increasing customer expectation of fast delivery, the challenges of reaching remote areas, particularly during emergencies, the need for replacing face-to-face interactions in special conditions, like that in the pandemic limit the advantages of using trucks. Adopting autonomous technologies in LMD addresses these challenges and has implications for pursuing sustainability in smart cities (Andreas, 2024). Robot-based delivery is quite efficient and eco-friendly for delivering parcels in dense areas but suffers from battery power and load capacity limitations. Besides, robots travel slowly, making them inefficient for long-distance delivery tasks (Jingi & Yang, 2023). Drones, on the other hand, can deliver goods in remote or hard-to-reach areas; drone adoption reduces traffic congestion, and carbon emissions, offering an efficient solution for emergency delivery needs. The downside of using drones is that they usually have limited battery endurance and are incapable of multiple deliveries at a time at the current technology maturity level. Table 1 summarizes The Table 1 summarizes the pros and cons of these delivery modes.

There are many studies on implementing drone-assisted (Das, Sewani, Wang, & Tiwari, 2021; He, He, Li, Zhang, & Xiao, 2022; Ramos

* Corresponding author.
E-mail addresses: enramokh@ljmu.ac.uk (A. Mokhtari-Moghadam), as@essex.ac.uk (A. Salhi), xyangk@essex.ac.uk (X. Yang), enrtngu1@ljmu.ac.uk (T.T. Nguyen), pourya.pourhejazy@uit.no (P. Pourhejazy).

https://doi.org/10.1016/j.eswa.2025.126434
Received 11 May 2024; Received in revised form 29 November 2024; Accepted 4 January 2025
Available online 10 January 2025

Table 1
Comparison of Truck, Drone and Robot.

Characteristics	Delivery mode		
	Truck	*Drone*	*Robot*
Load Capacity	many	one	few
Endurance	unlimited	short	short
Travel Speed	medium	high	low
Carbon Emission	high	low	low
Route	along road network	flexible 3D moving	pedestrian walk

& Vigo, 2023; Wang, Pesch, Kress, Fridman, & Boysen, 2022) and robot-assisted truck delivery systems for LMD (Chen, Demir, & Huang, 2021; Heimfarth, Ostermeier, & Hübner, 2022; Jingi & Yang, 2023; Ostermeier, Heimfarth, & Hübner, 2023; Simoni, Kutanoglu, & Claudel, 2020; Yu, Puchinger, & Sun, 2022). For a comprehensive review of the relevant literature, interested readers are referred to the surveys on autonomous LMD using robots (Alverhed, Hellgren, Isaksson, Olsson, Palmqvist, & Flodén, 2024), innovative solutions for LMD using new technologies (Mohammad, Diab, Elomri, & Triki, 2023), and the existing operations research methods in the same context (Boysen, Fedtke, & Schwerdfeger, 2021). Having described the advantages and disadvantages of the truck-, robot-, and drone-based delivery modes, integrating them is of interest to overcome their limitations.

The studies that combine drone and robot technologies to assist truck deliveries are quite limited. In the most relevant study, (Morim, Campuzano, Amorim, Mes, & Lalla-Ruiz, 2024) investigated the drone-assisted vehicle routing problem with robot assistance in the stations. Their approach starts with planning the delivery operations to a single local depot where the logistics operations inside the stations are separately optimized considering different fleets. These operations are interrelated and planning them simultaneously results in more effective and feasible solutions. The present study offers a twofold contribution to address this research gap. First, a new model for integrated planning of Drone-And-Robot-assisted Trucks (DART) for last-mile parcel delivery is introduced. The problem is formulated using an original Mixed-Integer Linear Programming (MILP) formulation for the bi-objective optimization of truck delivery operations. The proposed model uniquely considers customer local depots along with the autonomous delivery modes, which is quite practical. Second, Analyzing the impacts of geographical location and density of demand nodes in the DART delivery operations, DART is compared with the basic robot-assisted trucks and drone-assisted trucks. For this purpose, the Adaptive Multi-Objective Genetic Algorithm (AMOGA) is developed to solve the problem of minimizing total delivery cost and maximizing customer satisfaction, which are conflicting in nature.

The rest of this manuscript is organized into five sections. An analysis of the relevant literature is presented in Section 2. In Section 3, the drone-and-robot-assisted truck operations in LMD is described and a mathematical formulation is proposed to represent the optimization problem. The solution algorithm is explained in Section 4. The computational experiments and results analysis are presented in Section 5, followed by some concluding remarks in Section 6.

2. Literature review

The cluster analysis method is used to objectively analyze the literature and draw the big picture. The Web of Science database is used for data collection; [(robot* OR dron*) AND truck] searched in the title, abstract, and keywords of the published articles. Fig. 1 shows the network visualization of the keywords considering a total of 973 papers, including conference proceedings and research articles. In this figure, the node size shows the number of keyword occurrences, and the proximity between the nodes indicates the keywords' co-occurrences. In this definition, proximity between the nodes and clusters indicates higher simultaneity of the related studies in the literature.

The right cluster (red) is focused on the path planning of autonomous vehicles with robot-assisted operations. The upper-left cluster (green) is mostly about developing metaheuristics for the routing and scheduling of LMD services. The other three clusters in the bottom left are formed around optimizing drone operations for parcel delivery. There is a clear divide between the studies on robot- and drone-assisted truck deliveries. The most relevant methods are reviewed as follows.

From the studies on robot-assisted trucks in LMD, (Boysen, Schwerdfeger, & Weidinger, 2018) stated that the integrated planning of truck and robot delivery operations with a single objective is an NP-hard optimization problem, and developed a general heuristic to minimize the weighted number of late deliveries. In one of the first attempts to implement the integrated trucks and robots system for industry-scale delivery problems, Simoni et al. (2020) developed a set improvement neighborhood search algorithm to minimize total costs. Chen et al. (2021) developed an adaptive large neighborhood search heuristic to solve a vehicle routing problem for delivery robots. Aiming to minimize the summation of all routes' duration, they used the dispatch-wait collect policy, by which all robots are retrieved at the same locations where they are dispatched and served only one customer. Despite the simplicity element, the approach increased the overall route duration. Heimfarth et al. (2022) introduced a novel concept for mixed truck and robot delivery where robots are employed in depots. They developed a general variable neighborhood search to minimize the total costs including costs of truck time, truck distance, robot travel time, and delayed deliveries. Ostermeier et al. (2023) studied the multi-vehicle robot-assisted truck delivery problem with a robot depot aiming to minimize the total costs. Their numerical experiments showed that using an integrated multi-vehicle routing and robot scheduling approach can reduce the overall costs of transportation by up to a quarter compared with the sequential cluster-first-route-second approach. The most relevant study, Jingi and Yang (2023) investigated the robot-assisted truck delivery problem, considering the customer's local depot, aiming to minimize the truck travel time. They developed a mixed integer mathematical model and solved small-scale instances using an exact method and a heuristic algorithm.

Compared with robots, drones can travel faster and to areas where other means of transportation cannot. However, the drone's limited capacity results in that only one customer can be served at each departure time. Chang and Lee (2018) developed a nonlinear programming model to facilitate wider drone-delivery areas along a shorter truck route. Their approach consisted of clustering the delivery locations into areas within drone delivery ranges; finding the optimal route for the depot and the centers in the clusters; and adjusting the centers of the clusters for wider drone-delivery coverage. From other studies on drone-assisted trucks in LMD, Agatz, Bouman, and Schmidt (2018) developed an integer programming model based on a traveling salesman problem for planning drone-assisted truck deliveries. They tested several fast route-first, cluster-second heuristics based on local search and dynamic programming to solve the problem. Bouman, Agatz, and Schmidt (2018) proposed an exact solution method based on dynamic programming to solve the integrated truck-drone delivery problem. Tu, Dat, and Dung (2018) developed an adaptive large neighborhood search heuristic to optimize the problem of combining a single truck with multiple drones for LMD operations. Kitjacharoenchai et al. (2019) extended the traveling salesman problem for drone-assisted truck delivery; they employed commercial software to solve the problem for small instances and developed an adaptive insertion heuristic to address the larger instances, minimizing the travel route time of both trucks and drones. The collaborative truck and drone routing using Non-dominated Sorting Genetic Algorithm II (Das et al., 2021; Liu, Yan, Pu, Wang, & Kaisar, 2021) and the polynomial-time approximation algorithm of for a special case of transporting drones launched from a flying warehouse (Wang et al., 2022) are other seminal examples. There also are general optimization models for planning delivery operations

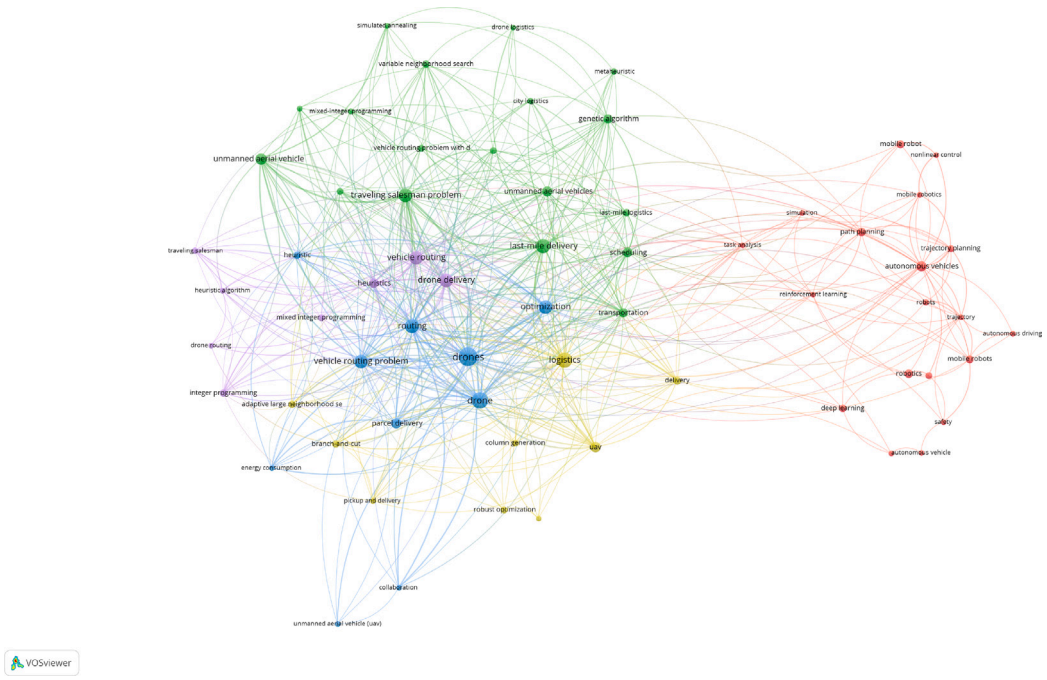


Fig. 1. Cluster analysis of the keywords in drone/robot-assisted truck deliveries.

using either drones or robots to assist truck deliveries (Kloster, Moeini, Vigo, & Wendt, 2023).

From the limited existing works that simultaneously consider drone and robot technologies for studying LMD, Lemardelé, Estrada, Pagès, and Bachofner (2021) explored the strategies to combine these technologies considering the practical needs and characteristics of LMDs. Figliozzi (2020) estimated CO2 emissions of drone-, sidewalk autonomous delivery robots, and road autonomous delivery robots, comparing them with electric-, and conventional internal combustion engine vans-based LMD. In the most relevant study, Morim et al. (2024) investigated the drone-assisted vehicle routing problem with robot stations and developed a metaheuristic, the General Variable Neighborhood Search (GVNS) algorithm, to solve the routing problem. This study tackles the two problems separately, where the stations and a main depot are to be served using two different fleets. Besides, they are merely focused on optimizing costs, which is not the case in real-world practices. Table 2 summarizes the most relevant studies, comparing their optimization approach with the present study's development. Except for the study of Morim et al. (2024), the rest considered either drone- or robot-assisted truck deliveries. To the authors' best knowledge, there are no multi-objective optimization approaches for optimizing the DART delivery operations.

3. Mathematical formulation

This section introduces the MOVRP_DR problem and the related assumptions. The mathematical model developed by Jingi and Yang (2023) is extended to integrate drones for covering remote areas. However, to assimilate today's real world concern, we designed the objective functions as minimizing overall cost of delivery and maximizing the customer satisfaction, which the later objective is inspired from Luo, Wu, Ji, Wang, and Suganthan (2022) with some modification for this specific problem.

3.1. Mixed integer programming formulation of MOVRP_DR

The MOVRP_DR is defined on a directed graph $G = (V, A)$, where V is the set of n nodes representing customers with p local depots and one main depot and A is the set of arcs. Given a fleet of K identical delivery vans, each equipped with subset of D drones and R self-driving robots, aiming to serve N customers by delivery of their small to medium parcels. There are P local depots from which customers can select those points that suit them for collecting their orders. Trucks which are loaded with drones, robots and customer orders starts from the distribution center, which in some real cases is far from the residential area. Drones in comparison could serve areas with some road restrictions that are not accessible by trucks and robots to travel. However, due to technology limitations, they usually can only deliver light-weight packages and rarely more than one package. On the other hands, although a robot can deliver multiple packages, it is often restricted by battery limit and low-speed traveling. In this research work, customers who require door delivery could be served by either truck, drone or robot, however, local depots with positive demands are only visited by truck. Drones and robots are launched or collected from trucks at suitable customer or local depot nodes. Trucks return to the distribution center with the collection of all drones and robots after visiting the assigned nodes (customers and local depots).

The important assumptions of the proposed model are summarized below:

1. It is assumed that there are homogeneous K trucks, D drones and R robots, each of which with limited order capacity of Q_k, Q_d and Q_r, which are starting and finishing at the distribution center or main depot. In addition, it is assumed that there are P local depots equipped with certain capacity of Q_p to service local customer orders.
2. There are N customers to be served, each with known location, preferred delivery time window, order size q_i and service time o_i. Also, the delivery location either at customer door or local depot is known in advance.
3. Each customer who selects door delivery is served either by a truck, a drone or a robot carried on a truck.
4. The drop-off and pick-up locations of drones and robots launched from trucks can be different to allow the trucks, drones and robots to serve customers simultaneously, rather than have one vehicle wait while the others are servicing.

Table 2
The shortlist of the relevant studies and methods.

Reference	Drone-assisted	Robot-assisted	Synch.	Customer Pick-up point	No. of Obj.	Objective(s)	Proposed algorithm
Chen et al. (2022)	✓	×	×	×	Single	Number of customers served	Deep Q-learning
Cheng Chen et al. (2021)	×	✓	✓	×	Single	Minimize travel time	Adaptive Large Neighborhood Search
Heimfarth et al. (2022)	×	✓k	✓	×	Single	Minimize total costs	General Variable Neighborhood Search
Agatz and Bouman (2018)	✓	×	✓	×	Single	Minimize travel time	Dynamic programming
Bouman and Agatz (2018)	✓	×	✓	×	Single	Minimize travel time	Exact solution approach based on DP
Chang et al.(2018)	✓	×	✓	×	Single	Minimize travel time	Non-linear programming
Tu et al. (2018)	✓	×	✓	×	Single	Minimize travel time	an adaptive large neighborhood search heuristic
Kitjacharoenchai et al.(2019)	✓	×	✓	×	Single	Minimize travel time	Exact solution approach
Kitjacharoenchai et al.(2019)	✓k	×	✓	×	Single	Minimize travel time	Adaptive Insertion Heuristic
Simoni et al. (2020)	×	✓	✓	×	Single	Minimize travel time	Adaptive Insertion Heuristic
Ostermeier et al.(2022)	×	✓	✓	×	Single	Minimize total costs	Set Improvement Neighborhood Search
Boysen et al.(2018)	×	✓	✓	×	Single	Minimizes the weighted number of late deliveries	A heuristic solution approach
Jingi and Yang (2023)	×	✓	✓	✓	Single	Minimize truck travel time	Exact and heuristic solution approaches
Liu et al. (2021)	×	✓	✓	×	Multi	Minimize economic costs and environmental impacts while maximizing customer satisfaction	ϵ-constraint and hybrid artificial immune algorithm
Das et al. (2021)	✓	×	✓	×	Multi	Minimize travel costs and maximize customer service level	Pareto Ant Colony Optimization algorithm
Lue et al. (2022)	✓	×	✓	×	Multi	Minimize total delivery costs and maximize customer satisfaction	MOGA+Local Search
Morim et al. (2024)	✓	✓	✓	×	Single	Minimize makespan and operational cost distinctly	GVNS
This study	✓	✓	✓	✓	Multi	Minimize total delivery costs and maximize total customer satisfaction	AMOGA

5. Drones and robots carried by trucks are identical, however robot storage is separated into compartments, which allows for multiple deliveries with respect to their battery charge limit. Drones allow single delivery.
6. Multiple drones and robots can be launched and retrieved at the same customer node.
7. The drones and robots must be picked up by the truck from which they are dropped.
8. The location of local depots, their coverage areas and service capacity are known.

Variables and Notations

Sets:

$C = \{1, \dots, c\}$ - set of all customer nodes.

C^p - Subset of customers select local depot for delivery, with $C^p \subseteq C$.

C^o - Subset of customers require door delivery, with $C^o \subseteq C$.

$P = \{1, \dots, p\}$ - set of local depot nodes.

$DC = \{1, n+p+2\}$ - distribution centre.

$N = (C \cup P \cup DC)$ - all nodes.

$N' = N \setminus (C^p \cup \{1\})$ Subset of customers and local depots require delivery.

$K = \{1, \dots, k\}$ - set of trucks.

$D = \{1, \dots, d\}$ - set of drones.

$R = \{1, \dots, r\}$ - set of robots.

Parameters:

q_i - order size of customer node, $i \in C$.

o_i^k - service time of truck k at node i, $i \in (C^o \cup P)$.

o_i^d - service time of drone d at node i, $i \in (C^o \cup P)$.

o_i^r - service time of robot r at node i, $i \in (C^o \cup P)$.

e_i - early desired time window of customer i.

l_i - late desired time window of customer i.

E_i - early tolerable time window of customer i.

L_i - late tolerable time window of customer i.

$d_{i,j}$ - distance between node i and node j $\forall i, j \in N$.

Q_p - maximum capacity that can be served by the local depot.

s_r - capacity occupation of a robot.

s_d - capacity occupation of a drone.

RB - maximum robot battery limit, $r \in R$.

DB - maximum drone battery limit, $d \in D$.

Q_k - capacity of truck.

Q_d - capacity of drone.

Q_r - capacity of robot.

v_k - speed of truck.

v_d - speed of drone.

v_r - speed of robot.

h - setup time of robot which include time to fill the robot with orders, setup its traveling routes and pick up location.

ι - setup time of drone which include time to fill the drone with orders, setup its traveling routes and pick up location.

vc_k - variable cost of truck per time unit.

vc_d - variable cost of drone per time unit.

vc_r - variable cost of robot per time unit.

fc_p - fixed cost of using depot.

ρ_p - selected local depot p.

$$\lambda_{i,j} = \begin{cases} 1, & \text{if customer } i \text{ is covered by depot } j,\ i \in C, j \in P \\ 0, & \text{otherwise} \end{cases}$$

$$\zeta_{i,j} = \begin{cases} 1, & \text{if node } i \text{ is served by depot } j,\ i \in C, j \in P \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables

$$x_{i,j,k} = \begin{cases} 1, & \text{if truck } k \text{ travel } (i,j),\ i \neq j \in N \setminus C^o, k \in K \\ 0, & \text{otherwise} \end{cases}$$

$$z_{i,j,d} = \begin{cases} 1, & \text{if drone } d \text{ travel } (i,j),\ i \neq j \in C^o, d \in D \\ 0, & \text{otherwise} \end{cases}$$

$$y_{i,j,r} = \begin{cases} 1, & \text{if robot } r \text{ travel } (i,j),\ i \neq j \in C^o, r \in R \\ 0, & \text{otherwise} \end{cases}$$

$$\theta_{k,d} = \begin{cases} 1, & \text{if drone } d \text{ is placed on truck } k,\ k \in K, d \in D \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_{k,r} = \begin{cases} 1, & \text{if robot } r \text{ is placed on truck } k,\ k \in K, r \in R \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{i,k,d} = \begin{cases} 1, & \text{if drone } d \text{ is launched from truck } k \text{ at node } i, \\ & k \in K, d \in D, i \in C^o \cup P \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_{i,k,r} = \begin{cases} 1, & \text{if robot } r \text{ is launched from truck } k \text{ at node } i, \\ & k \in K, r \in R, i \in C^o \cup P \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_{i,k,d} = \begin{cases} 1, & \text{if drone } d \text{ is collected by truck } k \text{ at node } i, \\ & k \in K, d \in D, i \in C^o \cup P \\ 0, & \text{otherwise} \end{cases}$$

$$\eta_{i,k,r} = \begin{cases} 1, & \text{if robot } r \text{ is collected by truck } k \text{ at node } i, \\ & k \in K, r \in R, i \in C^o \cup P \\ 0, & \text{otherwise} \end{cases}$$

$\mathrm{ar}_{k,r}$ = the truck capacity occupied by all orders that are to be delivered by robot r, if robot r is to be dispatched from truck k, $\forall k \in \mathcal{K}, \forall r \in R$.

$\mathrm{ad}_{k,d}$ = the truck capacity occupied by all orders that are to be delivered by drone d, if drone d is to be dispatched from truck k, $\forall k \in \mathcal{K}, \forall d \in D$.

$\mathrm{b}_{j,k}$ = the truck capacity occupied by all orders that are to be served by depot j, if depot j is to be visited by truck k, $\forall k \in \mathcal{K}, \forall j \in P$.

$t_{i,k}$ = the visiting time at node i by truck k, $\forall i \in C^o \cup P, \forall k \in K$.

$v_{i,d}$ = the visiting time at node i by drone d, $\forall i \in C^o \cup P, \forall d \in D$.

$\tau_{i,r}$ = the visiting time at node i by robot r, $\forall i \in C^o \cup P, \forall r \in R$.

w_i^r = waiting time of robot r at node i to be collected by truck, $\forall i \in C^o \cup P$.

w_i^d = waiting time of drone d at node i to be collected by truck, $\forall i \in C^o \cup P$.

Customer satisfaction is defined as the fulfillment of expectations, significantly influenced by the timeliness of vehicle arrivals. It is posited that the satisfaction level, $ctw_i(t)$, achieves a maximum of 1 when a vehicle arrives within the desired time frame $[e_i, l_i]$, decreasing linearly to 0 for arrivals outside this window.

$$\mathrm{ctw}_i(t) = \begin{cases} \frac{t-l_i}{L_i-l_i}, & l_i < t \le L_i, \\ 1, & e_i \le t \le l_i, \\ \frac{t-E_i}{e_i-E_i}, & E_i \le t < e_i, \\ 0, & \text{otherwise.} \end{cases}$$

A MILP formulation of the proposed MOVRP_DR is presented as follows based on the notations and variables:

$$\min\ TRC = vc_k\ x \sum_k \left(t_{c+p+2,k} - t_{1,k} \right) + \sum_i \sum_j \sum_d vc_d\ \frac{d_{i,j}}{v_d}\ z_{i,j,d} + \sum_i \sum_j \sum_r vc_r\ \frac{d_{i,j}}{v_r}\ y_{i,j,r} + \sum_{i,r} vc_r\ w_i^r + \sum_{i,d} vc_d\ w_i^d + \sum_p fc_p\ \rho_p \tag{1}$$

$$\max\ TCS = \sum_{i,j,k} x_{i,j,k} ctw_j(t_{j,k}) + \sum_{i,j,d} z_{i,j,d} ctw_j(v_{j,d}) + \sum_{i,j,r} y_{i,j,r} ctw_j(\tau_{j,r}) \tag{2}$$

$$\sum_{j \in N'} x_{1,j,k} = 1\ , \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in N'} x_{i,c+p+2,k} = 1\ , \qquad \forall k \in K \tag{4}$$

$$\sum_{j \in N' \cup \{1\}} x_{i,j,k} = \sum_{j \in N' \cup \{1\}} x_{j,i,k}\ , \qquad \forall i \in N', k \in K \tag{5}$$

$$\sum_{j \in N'} y_{i,j,r} + \sum_{k \in K} \eta_{i,k,r} = \sum_{j \in N'} y_{j,i,r} + \sum_{k \in K} \gamma_{i,k,r}\ , \qquad \forall i \in N', r \in R \tag{6}$$

$$2\ \gamma_{i,k,r} \le \delta_{k,r} + \sum_{j \in N' \cup \{1\}} x_{j,i,k}\ , \qquad \forall k \in K, r \in R, i \in N' \tag{7}$$

$$2\ \eta_{i,k,r} \le \delta_{k,r} + \sum_{j \in N' \cup \{1\}} x_{j,i,k}\ , \qquad \forall k \in K, r \in R, i \in N' \tag{8}$$

$$\sum_{k \in K} \delta_{k,r} \le 1\ , \qquad \forall r \in R \tag{9}$$

$$\sum_{j \in N'} z_{i,j,d} + \sum_{k \in K} \alpha_{i,k,r} = \sum_{j \in N'} z_{j,i,d} + \sum_{k \in K} \beta_{i,k,r}\ , \qquad \forall i \in N', d \in D \tag{10}$$

$$2\ \beta_{i,k,d} \le \theta_{k,d} + \sum_{j \in N' \cup \{1\}} x_{j,i,k}\ , \qquad \forall k \in K, d \in D, i \in N' \tag{11}$$

$$2\ \alpha_{i,k,d} \le \theta_{k,d} + \sum_{j \in N' \cup \{1\}} x_{j,i,k}\ , \qquad \forall k \in K, d \in D, i \in N' \tag{12}$$

$$\sum_{k \in K} \theta_{k,d} \le 1\ , \qquad \forall d \in D \tag{13}$$

$$\sum_{i \in N' \cup \{1\}} \sum_{k \in K} x_{i,j,k} \ge \frac{1}{M} \sum_{i \in C} q_i\ \zeta_{i,j}\ , \qquad \forall j \in P \tag{14}$$

$$\sum_{j \in N' \cup \{1\}} \sum_{k \in K} x_{i,j,k} + \sum_{j \in N'} \sum_{d \in D} z_{i,j,d} + \sum_{j \in N'} \sum_{r \in R} y_{i,j,r} + \sum_{j \in P} \zeta_{i,j} = 1\ , \qquad \forall i \in C \tag{15}$$

$$\zeta_{i,j} \le \lambda_{i,j}\ , \qquad \forall i \in C, \forall j \in P \tag{16}$$

$$\sum_{i \in C} q_i\ \zeta_{i,j} \le Q_j\ , \qquad \forall j \in P \tag{17}$$

$$\sum_{i \in N'} \sum_{j \in N'} \frac{d_{i,j}}{v_r}\ y_{i,j,r} \le RB\ , \qquad \forall r \in R \tag{18}$$

$$\sum_{i \in N'} \sum_{j \in N'} \frac{d_{i,j}}{v_d}\ z_{i,j,d} \le DB\ , \qquad \forall d \in D \tag{19}$$

$$\sum_{i \in C^o} q_i \left(\sum_{j \in N'} y_{i,j,r} - \sum_{k \in K} \gamma_{i,k,r} \right) \le Q_r\ , \qquad \forall r \in R \tag{20}$$

$$\sum_{i \in C^o} q_i \left(\sum_{j \in N'} z_{i,j,d} - \sum_{k \in K} \beta_{i,k,d} \right) \le Q_d\ , \qquad \forall d \in D \tag{21}$$

$$\sum_{i \in C} \sum_{j \in N'} q_i\ x_{i,j,k} + \sum_{r \in R} ar_{k,r} + \sum_{r \in R} s_r\ \delta_{k,r} + \sum_{d \in D} ad_{k,d} + \sum_{d \in D} s_d\ \theta_{k,d} + \sum_{j \in P} b_{k,j} \le Q_k\ , \qquad \forall k \in K \tag{22}$$

Node sequencing vector	13	7	8	12	3	6	4	2	5	11
Truck-Drone-Robot assignment vector	1	2	3	1	1	1	1	1	4	1
Drone-Robot Drop-Collection cell array	2	[]	[]	4	3	3	4	2	[]	[]

Fig. 2. Chromosome of sample instance.

$$\mathrm{ar}_{k,r} \geq \sum_{i \in C^o} q_i \left(\sum_{j \in N'} y_{i,j,r} - \gamma_{i,k,r} \right) - M\left(1 - \delta_{k,r}\right), \quad \forall k \in K, \forall r \in R \tag{23}$$

$$\mathrm{ad}_{k,d} \geq \sum_{i \in C^o} q_i \left(\sum_{j \in N'} z_{i,j,d} - \beta_{i,k,d} \right) - M\left(1 - \theta_{k,d}\right), \quad \forall k \in K, \forall d \in D \tag{24}$$

$$\mathrm{b}_{k,j} \geq \sum_{i \in C^p} q_i \zeta_{i,j} - M\left(1 - \sum_{j \in N} x_{i,j,k}\right), \quad \forall k \in K, \forall j \in P \tag{25}$$

$$t_{j,k} \geq t_{i,k} + o_i^k + \frac{d_{i,j}}{v_k} + h \sum_{r \in R} \gamma_{i,k,r} + \iota \sum_{d \in D} \beta_{i,k,d} - M\left(1 - x_{i,j,k}\right), \quad i \neq j \forall i, j \in N', \forall k \in K \tag{26}$$

$$v_{j,d} \geq v_{i,d} + o_i^d(1 - \beta_{i,k,d}) + \frac{d_{i,j}}{v_d} - M\left(1 - z_{i,j,d}\right), \quad i \neq j \forall i, j \in C^o \cup P, \forall d \in D, \forall k \in K \tag{27}$$

$$\tau_{j,r} \geq \tau_{i,r} + o_i^r(1 - \gamma_{i,k,r}) + \frac{d_{i,j}}{v_r} - M\left(1 - y_{i,j,r}\right), \quad i \neq j \forall i, j \in C^o \cup P, \forall r \in R, \forall k \in K \tag{28}$$

$$t_{i,k} \geq v_{i,d} - M(1 - \alpha_{i,k,d}), \quad \forall i \in C^o \cup P, \forall d \in D, \forall k \in K \tag{29}$$

$$t_{i,k} \geq \tau_{i,r} - M(1 - \eta_{i,k,r}), \quad \forall i \in C^o \cup P, \forall r \in R, \forall k \in K \tag{30}$$

$$v_{i,d} \geq t_{i,k} + \iota - M(1 - \beta_{i,k,d}), \quad \forall i \in C^o \cup P, \forall d \in D, \forall k \in K \tag{31}$$

$$\tau_{i,r} \geq t_{i,k} + h - M(1 - \gamma_{i,k,r}), \quad \forall i \in C^o \cup P, \forall r \in R, \forall k \in K \tag{32}$$

$$x_{i,j,k} \in \{0,1\}, \quad i \neq j \forall i, j \in N \setminus C^o, \forall k \in K \tag{33}$$

$$z_{i,j,d}, \theta_{k,d}, \beta_{i,k,d}, \alpha_{i,k,d} \in \{0,1\}, \quad i \neq j \forall i, j \in C^o \cup P, \forall d \in D, \forall k \in K \tag{34}$$

$$y_{i,j,r}, \delta_{k,r}, \gamma_{i,k,r}, \eta_{i,k,r} \in \{0,1\}, \quad i \neq j \forall i, j \in C^o \cup P, \forall r \in R, \forall k \in K \tag{35}$$

$$t_{i,k}, v_{i,d}, \tau_{i,r}, w_i^r, w_i^d, \mathrm{ar}_{k,r}, \mathrm{ad}_{k,d}, \mathrm{b}_{k,p} \geq 0, \quad i \neq j \forall i, j \in C^o \cup P, \forall p \in P \forall d \in D, \forall r \in R, \forall k \in K \tag{36}$$

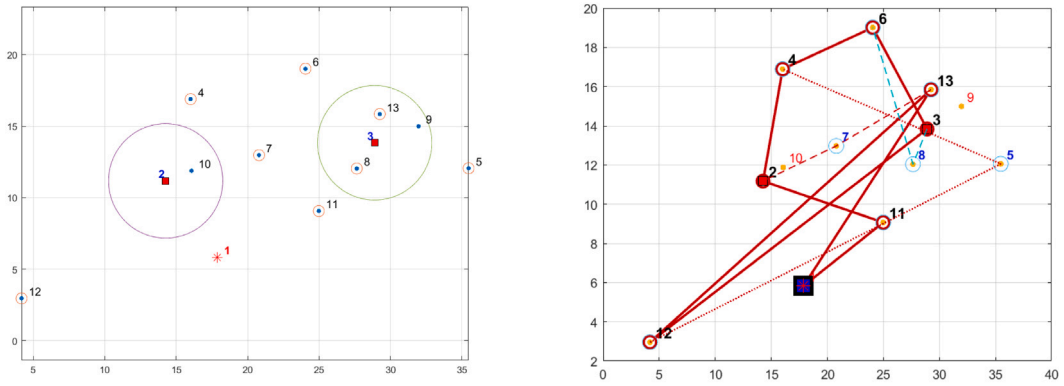
The objective function (1) minimizes total routing costs (TRC). The first term specifies the time-dependent usage costs of all trucks, i.e., for each truck the total time is calculated from the departure to arrival at the main depot. The second term sums up the total distance costs of all distances traveled by trucks. The third term calculates the costs of drone travel from truck stops to customers, including the drones' return time from the serving customer to the collection node. The fourth term calculates the costs of robots travel from truck stops to customers, including the robots' return time from the serving customer to the collection node. The fifth and sixth terms are considering the waiting time of drones and robots at collection node respectively, when they arrive earlier than the truck and wait to be collected. The last term adds the operation costs of selected depots for those customers who have chosen local depots for collecting their parcels. The second objective function, (2), maximize the total customer satisfaction (TCS).

Constrains (3),(4) and (5) represent the network flow constraints of the truck. Constraint (6) defines the network flow constraint of the robot. Constrains (7) and (8) ensure that a robot can be launched from or collected at a node only if it is carried by a truck which visits the node. Constraint (9) ensures that a robot cannot be carried by more than one truck. Constraint (10) represents the network flow constraint of the drone. Constrains (11) and (12) ensure that a drone can be launched from or collected at a node only if it is carried by a truck which visits that node. Constraint (13) ensures that a drone cannot be carried by more than one truck. Constraint (14) determines that all local depots with positive customer demands are visited by truck. Constraint (15) guarantees that all customers are either served by truck, drone/robot launched from the truck or by a local depot. Eq. (16) guarantees that a customer's node must be encompassed by a depot prior to receiving service from said depot. Constraint (17) ensures that the local depot capacity must be respected. Constraints (18) and (19) determine that a robot and a drone do not exceed their total battery endurance, respectively. Constraints (20) and (21) ensure that total capacity of shipments carried on each robot and each drone does not exceed the capacity of the robot and the drone, respectively. Constraint (22) ensures that total capacity of shipments and capacity occupation of all robots and drones carried on truck does not exceed the capacity of each truck, with (23),(24) and (25) calculating the truck capacity occupied by orders to be serviced by on-truck robots, on-truck drones and orders to be serviced by local depot(s), respectively. Constraints (26), (27) and (28) calculate the visiting time at nodes by trucks, drones and robots respectively and Constraints (29), (30), (31) and (32) link them together. Finally, Constraints (33), (34), (35) and (36) state the domain of decision variables.

4. Solution approach

The mixed truck and robot for last-mile delivery problem with single objective is an NP-hard optimization problem and exact solving of even small instances is hard and not promising Boysen et al. (2018), Heimfarth et al. (2022). Therefore, the MOVRP_DR, that is a generalized form of the truck and robot routing problem which considers a mixture of truck, drones and robots for last-mile delivery aiming to optimize two objectives (TRC and TCS) simultaneously is also an NP-hard optimization problem. In this research, an adaptation of the MOGA algorithm proposed by Mokhtari-Moghadam, Pourhejazy, and DeepakGupta (2023) is used to solve the MOVRP_DR model. However, to assign fitness to the individuals, we use the fitness assignment method of Strength Pareto Evolutionary Algorithm 2 (SPEA2) proposed by Zitzler, Laumanns, and Thiele (2001) , which incorporate both dominance-based fitness and density estimation to ensure a balance between convergence to the Pareto front and solution diversity. In addition, a local search algorithm (LSA) is designed to select the best nodes among the truck-visited nodes for launching and collecting of drones and robots.

In this section, first our approach for presenting a chromosome is discussed, then the method for decoding the chromosome will be explain. Finally, having found a mechanism to encode a chromosome and decode it into a solution of the LMD problem with drone-robot assisted delivery, then, the detailed description of AMOGA as an evolutionary algorithm is explained and applied to the problem for improving the quality of solutions according to Algorithm 1.



(a) (b)

Fig. 3. An instance of randomly generated location and solution.

Algorithm 1 Pseudo-code of AMOGA

Require: An instance of MOVRP_DR, AMOGA parameters
Ensure: A near optimal Truck-Drone-Robot Route
Initialization
Initialize parameters ($nPop, MaxIt, P_c, P_m, P_{ma}, T_{size}$);
Initialize population with the size of $nPop$;
- Generate random permutation of nodes;
- Apply assignment algorithm;
- Apply drone and robot drop-collection algorithm;

Calculate population objective functions values;
Assign fitness to the population using SPEA2, balancing dominance-based fitness with density estimation to ensure convergence and diversity;
Sort the initial generated population in ascending order with considering their ranks;

while not termination condition **do**
Crossover
Apply on the nodes sequencing vector
if $p \leq Pc$ **then**
- Choose two parent individuals through the Tournament selection technique;
- Select one crossover operator randomly among PMX and POX with equal probability;
- Apply crossover operator on the nodes sequencing part;
- Apply assignment algorithm;
- Apply drone and robot drop-collection algorithm;
- Calculate values of offsprings objective functions ;

end if
Mutation
Apply on the truck rout of nodes sequencing vector
if $p \leq Pm$ **then**
- Select one individual as parent by Tournament selection method;
- Select one mutation operator among Swapping, Insertion and Reversion mutation operators randomly;
- Apply mutation on the truck route part of the first vector
- Apply drone and robot drop-collection algorithm;
- Calculate value of offsprings objective function;

end if
Apply on the assignment vector
if $p \leq P_{ma}$ **then**
- Select one individual as parent by Tournament selection method;
- Select one mutation operator among Swapping, Insertion and Reversion mutation operators randomly ;
- Apply mutation on the assignment vector
- Apply drone and robot drop-collection algorithm;
- Calculate value of offsprings objective function;

end if
Combine all solutions (initial, recombination and mutation solutions);
Eliminate solutions with identical fitness values (objective functions);
Evaluate the remaining individuals by assigning fitness value to the population using SPEA2, balancing dominance-based fitness with density estimation to ensure convergence and diversity;
Apply LSA to refine some offspring solutions.
Arrange the individuals in ascending order based on their ranks;
Select best-ranked solutions with the size of $nPop$ for the next generation from the pool using Truncation selection;
Calculate fitness values of individuals using SPEA2 algorithm;
Sort population in ascending order according to their ranks.;
Save the Pareto-front solution;
end while

4.1. Chromosome representation

A new method is developed to utilize a multi-layer chromosome for encoding each solution of MOVRP_DR. It consists of two vectors that are (1) nodes sequencing vector (2) truck-drone-robot assignment vector and one cell array that is drone-robot drop-collect cell array. The length of each vector and cell array is equal to the total number of non-assigned orders (customers who choose door-delivery) and selected local depots (for customers who collect their order from local depots). To represent a chromosome explicitly and simplifying the description of different steps of encoding and decoding the instance number 3, *P*_03, from the test instances is selected. Fig. 3(a) shows the instance with one main depot (red star, index 1), two selected local depots (red square, indices 2 and 3), and 10 customers (dark blue circle point, indices 4 to 13), amongst which, customers within the covering areas of depots (large circle), nodes 9 and 10, have chosen local depots 3 and 2 for collecting their orders, respectively. The remaining customers opt for door-delivery that are shown with small orange circle around those nodes.

First vector of the chromosome shows the sequence by which customers and local depots are visited by truck, drone or robot. Because the main depot (node 1) is the start and finish node that is always assigned to the truck, in the chromosome representation it is ignored for simplicity of coding. To generate a random solution, the order by which nodes will be visited by truck, drones or robots are permuted, as depicted in Fig. 2 (first vector).

The second vector of the chromosome represents the assignment of truck, drones and robots. In this example, the number 1 in the Truck-Drone-Robot assignment vector shows the truck, which means truck 1 is assigned to nodes 13, 12, 3, 6, 4, 2 and 11. The numbers 2 and 3 are for robots and finally, number 4 shows the drone. It is clear that robot 1 and robot 2 (indices 2 and 3) are allocated to nodes 7 and 8 respectively, while drone 1 (index 4) is allocated to the node 5. There are some assumptions and constraints that need to be considered while assigning the truck, drones and robots. It is assumed that the first and last nodes must be visited by truck. Also, the local depot must be visited by truck only. Regarding the constraints, robots can carry at most two orders (Q_r) and drones are allowed to serve only one customer due to current technology limitations. Algorithm 3 shows the Truck-Drone-Robot assignment algorithm.

The third part of the chromosome is devoted to the drop and collection nodes of drones and robots. Algorithm 4 shows the algorithm for drone and robot drop-collection nodes. The combination of the two vectors and one cell array of encoding chromosome is shown at Fig. 2, which shows that robot 1 (index=2) is dropped at node 13 and collected at node 2, robot 2 (index=3) is dropped at node 3 and collected at

node 6, and drone 1 (index=4) is dropped at node 12 and collected at node 4.

4.2. Assign customers to local depots

The algorithm for selecting the local depot is demonstrated in Algorithm 2. In fact, for convenience to the customers, it is assumed that there are some local depots where customers have an option to collect their order such as Amazon locker. Therefore, those customers who prefer to pick their orders through the nearest local depot could select the local depot considering the local depots capacity and vicinity (within 4 km radius). In this case, there are three states for local depots:

- Local depot is not selected by any customer, and all customers within the vicinity of a local depot prefer to received their order at home.
- Local depot is selected by one or more customers.
- Local depot is chosen by all customers within the vicinity for collecting their orders.

Algorithm 2 Local depot selection algorithm

Require: d_{ij} , Q_p, q_j, λ_{ij}, and Preference rate (Pr)
Ensure: list of selected local depots P and the non-assigned customers N'
SLD = {} ▷ Selected local depots
AC_i = {} ▷ Assigned customers to the local depot i
for each local depot i in P **do**
  Sort all customers distance from local depot i in ascending order
  for each customer j in C **do**
    if $j \notin AC$ & Remaining capacity of local depot $i \leq q_j$ & $\lambda_{ij} = 1$ **then**
      Generate a random number $r \in [0, 1]$
      if $r \geq Pr$ **then**
        $AC_i \leftarrow j$
        $SLD \leftarrow i$
      end if
    end if
  end for
end for
$N' = SLD \bigcup (C - AC)$

Algorithm 3 Truck-Drone-Robot Assignment Algorithm

Require: nodes sequencing vector , R, C_R
Ensure: Truck-drone-robot assignment;
Create an empty vector with the length of nodes sequencing vector;
Assign the first node, the last node and the local depots nodes to truck;
Assign the remaining nodes to the truck, drones, and robots randomly.
for each robot r **do**
  if number of assigned nodes $\geq Q_r + 1$ **then**
    Select two nodes randomly and assign to the robot;
    Assign the remaining nodes to the truck.
  end if
end for
for each Drone d **do**
  if number of assigned nodes $\geq Q_d + 1$ **then**
    Select one node randomly and assign it to the drone;
    Assign the remaining nodes to the truck.
  end if
end for

Algorithm 4 Drone-robot drop-collection algorithm

Require: Node sequencing and truck-drone-robot assignment vectors
Ensure: Drone-robots drop-collection nodes
Create an empty cell array with the length of nodes sequencing vector
for each drone $d \neq \{\}$ **do**
  Select two nodes from the assignment vector ($\equiv 1$) randomly
  Assign index of drone to the first selected node for dropping
  Assign index of drone to the second selected node for collecting
end for
for each robot $r \neq \{\}$ **do**
  Select two nodes from the assignment vector ($\equiv 1$) randomly
  Assign index of robot to the first selected node for dropping
  Assign index of robot to the second selected node for collecting
end for

4.3. Solution decoding

The algorithm for decoding of the chromosome is presented in Algorithm 5. The algorithm decodes the randomly generated chromosome into a feasible or non-feasible but may not optimal solution for the LMD using the drones and robots assisted truck (Fig. 3(b)). For instance, starting reading the chromosome from left to right, the first element in the second vector is equal to 2, which means the first node which is 13 should be visited by truck. Therefore, truck start its journey from the main depot to visit node 13. Arriving at first node, then the first element of the third vector is checked. If it is not empty, the truck needs to drop the robot or drone or both (according to the assigned index) and then do services to the customer. In this example the first element of the third vector is equal to 2, so truck drop robot (robot index 2: R2) then do service the customer and continue its journey to the next assigned node, which is 12. At node (13), R2 starts its journey and visits and services the only assigned node 7, then according to the third vector, R2 needs to be collected at node 2, therefore, after servicing the last assigned nodes (here only node 7), it continues its journey to the collected node 2. At this node, if truck arrives earlier than robot, it needs to wait until the robot arrives and collects it and vice versa.

Arriving truck at node 12, it drops drone (index 4 : D4) and then do services to the customer. At node 12, D4 starts its journey to visit assigned node 5 (according to the second vector), then to be collected continue its journey to node 4. This process continue until truck visit the last node 11, before returning to the main depot.

4.4. Population initialization

In the context of heuristic approaches for solving real-world problems, generating initial populations randomly can lead the diversity and quality solution. Therefore, here, to provide such features in the initial population, the sequence of nodes in the first chromosome vector are simply permuted that followed by the random assignment of truck, robots and drones to each node using Algorithm 3 which shaped the second chromosome vector. Finally, the dropping and collecting nodes of drones and robots are randomly selected from the truck route using Algorithm 4.

4.5. Fitness value calculation and pareto front generation

In multi-objective optimization, solutions with higher fitness values have a greater likelihood of being selected for survival and reproduction in subsequent generations. In this study, two objective functions are used to evaluate and assign fitness to solutions: total route costs (TRC) and total customer satisfaction (TCS). To calculate fitness, we adopt the Strength Pareto Evolutionary Algorithm 2 (SPEA2) fitness assignment approach as proposed by Zitzler et al. (2001). This method

Algorithm 5 Chromosome decoding

Require: d_{ij} , DB, RB, o_i, v_k, v_d, v_r, h, ι and cost variables.
Ensure: Total Route Costs (TRC)and Total Customer Satisfaction (TCS)
Initialize vectors for recording Truck Arrival Time (TAT), Truck Setup Time (TDT), Truck Service Time (TST), Truck Collection Time (TCT), Truck Waiting Time (TWT), Drone and Robot Release Time (DRT and RRT), Drone and Robot Collection Time (DCT and RCT), Drone and Robot Travel Times (DTT and RTT), and Drone and Robot Violation Battery Times (DVB and RVB).
Initialize table for drone and robot arriving time (DAT and RAT), drone and robot finish service time (DFST and RFST),and customer's time window.
for each node on the truck route **do**
$TAT = d/v_k$ ▷ truck travelling time from main depot to the first node
DN = find indices of drone(s) and robot(s) need drop at current node from drone-robot drop-collection array
if $DN \neq \{\}$ **then**
for each robot r in DN **do**
$TDT = TAT + h$
$RRT = TDT$
$RAT = RRT + d/v_r$ ▷ Travelling time to the assigned node(s)
$RFST = RAT + o_i^r$
end for
for each drone d in DN **do**
$TDT = TAT + \iota$
$DRT = TDT$
$DAT = DRT + d/v_d$ ▷ Travelling time to the assigned node(s)
$DFST = DAT + o_i^d$
end for
else
$TDT = TAT$
end if
$TST = TDT + o_i^k$
CN = find indices of drone(s) and robot(s) need collection from drone-robot drop-collection array
if $CN \neq \{\}$ **then**
for each drone d in CN **do**
$DCT = DFST$ (last visited node) + drone travelling time to the collection node
$DTT = DCT - DRT$
$DVB = \max\{0, (DCT/DB - 1)\}$
$TWT = \max\{0, (DCT - TST)\}$
$DWT = \max\{0, (TST - DCT)\}$
end for
for each robot r **do**
$RCT = RFST$ (last visited node) + robot travelling time to the collection node
$RTT = RCT - RRT$
$RVB = \max\{0, (RCT/RB - 1)\}$
$TWT = \max\{0, (RCT - TST)\}$
$RWT = \max\{0, (TST - RCT)\}$
end for
$TCT = TST + \max(TWT)$
else
TCT = TST
end if
end for
$TAT = TCT + d/v_k$ ▷ Return time at main depot
calculate TRC and TCS according to the equation (1) and (2), respectively.

incorporates both dominance-based fitness and density estimation, ensuring a balance between convergence towards the Pareto front and maintaining diversity among solutions. First, raw fitness is calculated using dominance count techniques, followed by density estimation based on the kth nearest neighbor distance in the objective space. The final fitness value of each individual is obtained by combining the raw fitness and the density measure, effectively promoting a diverse and high-quality solution set while mitigating computational overhead. By comparing pairs of individuals in the population, non-dominated individuals that contribute to constructing the (near-) Pareto front are identified, which guide the search process. Through successive generations, the proposed algorithm iteratively refines the population, aiming to discover solutions closer to the true Pareto-optimal front.

4.6. Selection methods

To select parents from the pool for the reproduction phase, the Tournament Selection technique is used here. The size of a tournament must be chosen precisely as the larger tournament size gives more weight to the elite-selection policy and hence the fastest convergence of the algorithm and on the other hands, choosing small tournament size improves diversity and slow algorithm convergence. After extensive computational experimenting, the tournament size in this research is set to 3, i.e. ($T_{size} = 3$).

After each generation, the truncation selection method is utilized to pick the best-ranked individuals from the pool for the subsequent generation. Initially, the old population, pop_{old}, is combined with the newly generated population from recombination and mutation, pop_{new}. Then, to enhance efficiency, individuals with identical objective functions (TRT and TCS) are eliminated. Lastly, the truncation selection process is carried out to select the top-ranked solutions with a size of $nPop$ for the next generation.

4.7. Crossover operators

Crossover, also known as recombination, is an evolutionary algorithm operator that utilizes the genetic information from two selected parents to generate offspring. In the proposed AMOGA algorithm, two types of crossovers, namely partial mapping crossover (PMX) (Goldberg & Lingle, 1985) and preserving order-based crossover (POX) (Kacem, Hammadi, & Borne, 2002) are employed on the node sequencing vector of selected parents. Following the creation of the first offspring vector, Algorithm 3 and Algorithm 4 are utilized to finalize the truck-drone-robot assignment vector and drone-robot drop-collection cell array of the new chromosomes.

To implement PMX crossover on the chromosome's node sequencing vector, a process is followed where two parents (P1 and P2) are chosen from the current population using the tournament selection method. Subsequently, two empty offspring (referred to as Off1 and Off2) are created, each with the same length as the chromosome. Then, two random cut points (c1 and c2) are selected. After that, having formed the mapping relationship of genes between the c1 and c2, they then are exchanged and transferred into Off1 and Off2 in order to create partial offspring. Finally, the empty positions of offspring Off1 and off2 are filled with P2 and P1 respectively. In case of any repeated gene in the offspring while filling the empty positions, the genes from the mapping relationship will be used. To clarify the procedure of PMX crossover, an illustration is depicted in Fig. 4.

The POX crossover involves creating two offspring (Off1 and Off2) from two selected parents (P1 and P2) by applying it to the node sequencing vector of chromosome. In this process, two sub-nodes (sn1 and sn2) are randomly generated from all nodes in the first vector, and two empty vectors with the same length as Off1 and Off2 are also created. Genes from P1 and P2 that match sn1 and sn2 are transferred to the corresponding positions in Off1 and Off2. Subsequently, genes belonging to sn1 and sn2 are removed from parents P2 and P1, respectively. The remaining genes of P2 and P1 are then transferred to Off1 and Off2 in the same order as they appear in the parents. The application of POX crossover is depicted in Fig. 5.

It is notable that, to make it simple to understand, in this example, selected nodes are between 1 and 10 for a graphic representation of how PMX and POX crossovers are applied in the proposed AMOGA.

4.8. Mutation operators

To improve search diversity and avoid premature convergence, the mutation operator is employed. In this study, to produce an offspring, three mutation operators named swap, insertion and reversion, are employed.

4.8.1. Apply mutation operators on the truck route of node sequencing vector

A parent is chosen from the current population through the tournament selection method, followed by the random selection of one of the mutation operators with an equal probability. This selected mutation operator is then applied to the chosen parent to generate

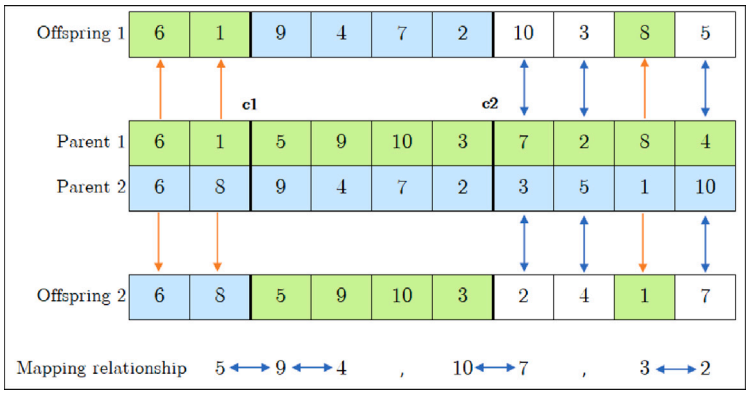


Fig. 4. PMX crossover operator.

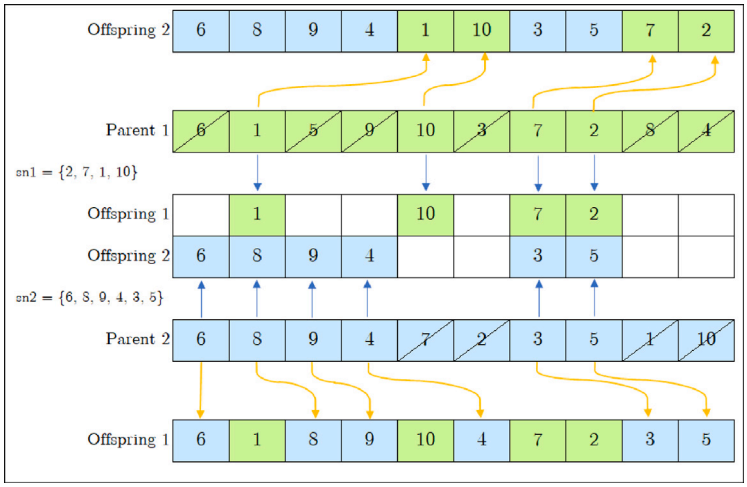


Fig. 5. POX crossover operator.

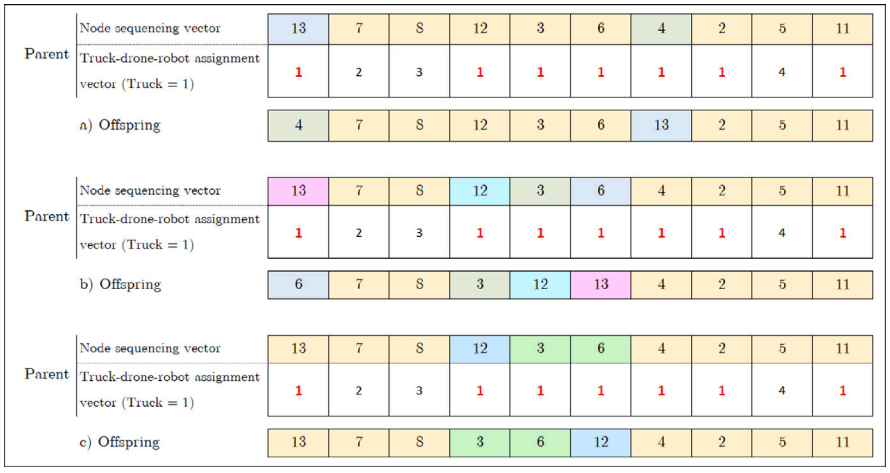


Fig. 6. Illustration of the mutation operators: (a) swap, (b) reversion, and (c) insertion on the truck route of first vector of chromosome.

a new offspring. Fig. 6 shows the procedure of applying mutation operators on the first vector of chromosome of the sample instance of Fig. 2 . After generating the first vector of offspring, the truck-drone-robot assignment vector of parent is not changed. Therefore, this vector is copied completely to the offspring. However, drone-robot drop-collection cell array of the new chromosome needs to be updated using Algorithm 4.

4.8.2. Apply mutation operators on the truck-drone-robot assignment vector

To apply mutation operators on the truck-drone-robot assignment vector, a parent is selected from the population using tournament selection method and then one of the mutation operators with an equal probability is chosen randomly. It is worth noting that according to the assumptions described earlier, in the problem description and mathematical modeling section, the first and last nodes of each

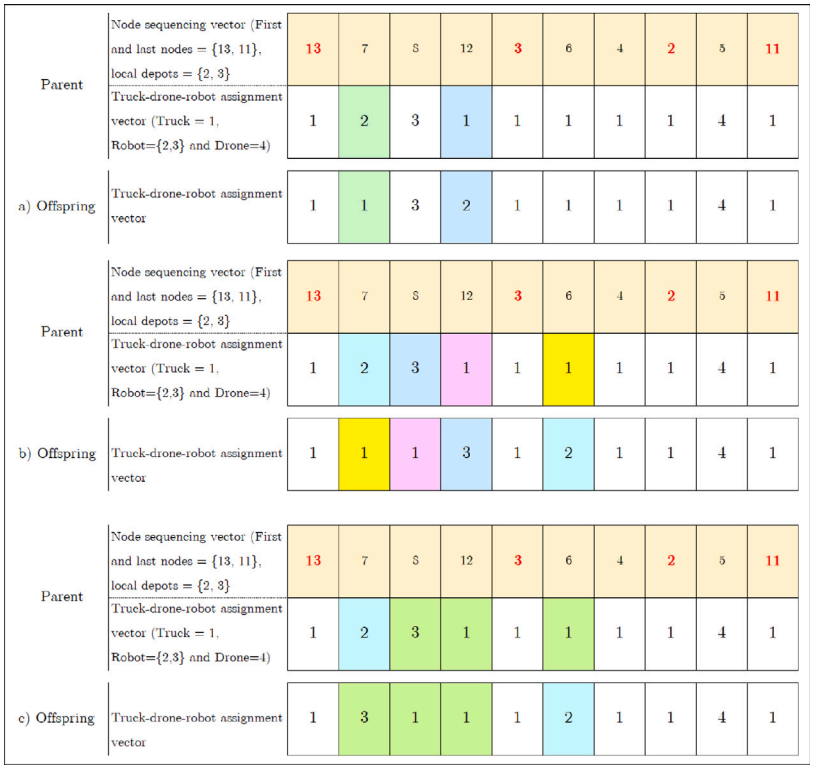


Fig. 7. Illustration of the mutation operators: (a) swap, (b) reversion, and (c) insertion on the assignment vector of chromosome.

route, including the local depot(s), must be served by truck. Therefore, mutation operators are respecting these assumptions while they are implemented on the second vector, which means the assigned truck to those nodes will not change. Fig. 7 shows the procedure of applying mutation operators to the second vector of chromosome. Having mutated the assignment vector, the sequencing vector remains unchanged in the offspring, but the drone-robot drop-collection cell array require updating using Algorithm 4.

5. Experimentation and computational results

5.1. Description on test algorithms and comparison metrics

To validate the effectiveness and efficiency of the proposed AMOGA for solving various models of MOVRP, we compare it with a well-known multi-objective evolutionary algorithms, NSGA-II proposed by Deb, Pratap, Agarwal, and Meyarivan (2002) on various scales of MOVRP_DR model. The algorithms are coded and compiled on MATLAB R2020a and run on a personal computer with AMD Rayzen 3, 2.60 GHz CPU, and 12 GB of RAM.

To evaluate the performance of multi-objective algorithms, several metrics are commonly employed. A detailed explanation of each metric can be found in Mokhtari-Moghadam et al. (2023). A brief overview of these metrics is provided below:

- Spacing Metric (SM): Assesses the uniformity of non-dominated solutions. Lower values indicate better spread uniformity.
- Diversification Metric (DM): Evaluates the heterogeneity of non-dominated solutions. Higher values signify better diversity.
- Mean Ideal Distance (MID): Measures the proximity of solutions to the ideal point. Smaller values reflect better performance.
- Quality Metric (QM): Represents the percentage of unique solutions contributed by each algorithm. Higher values indicate more distinctive solutions.
- Inverted Generational Distance (IGD): Combines convergence and diversity by measuring the distance from the obtained solutions to a reference Pareto-front. Lower values are preferred.

5.2. Test instances

Test data are randomly generated, but in such a way that they are representative of real world drone-robot-assisted delivery problems. Table 3 shows the levels and ranges of the factors determining the configuration of the test instances. The test problems are denoted by (the number of the main depot).(the number of orders).(the number of the local depots).(the number of trucks).(the number of drones).(the number of robots). For example, a problem with 1 main depot, 10 customers, 2 local depots, 1 truck, 1 drone and 2 robots can be denoted by MOVRP_DR_1 ·10 ·2 ·1 ·1 ·2.

The main depot, local depots, and customer locations are randomly distributed within an area measuring 40 km in length and 20 km in width. The velocities of the truck and robot, as well as the drop-off and collection times for the robot, and the delivery service times for both the truck and robot, are detailed in Simoni et al. (2020). The drone's speed information is sourced from Kitjacharoenchai et al. (2019). Furthermore, the expenses associated with the truck and robot are determined based on the findings of Ostermeier et al. (2023), with the currency being converted from € to £.

5.3. Parameter settings and stopping criterion

Upon delineating the test problems, the calibration of parameters in any evolutionary algorithm is crucial, as it directly impacts the algorithm's performance. Consequently, a series of extensive trial-and-error experimental tests were carried out to fine-tune the parameters of the AMOGA and NSGA-II algorithms. These parameters include the population size, tournament selection size, as well as the crossover

Table 3
Summary of test data.

Factors	Levels	Number of levels
Number of order	Ranges from 8 to 120	17
Demand's quantity	Discrete uniform [5, 10]	1
Number of truck	1	1
Number of robot	1, 2, 3, 4, and 5	5
Number of drone	1, 2, 3, 4, and 5	5
Number of main depot	1	1
Number of local depot	1, 2, 3, 4, 5, and 6	6
Truck speed (m/sec)	6	1
Robot speed (m/sec)	1	1
Drone speed (m/sec)	9	1
Truck delivery time (minute)	6	1
Robot delivery time (minute)	3	1
Drone delivery time (minute)	3	1
Average drone/robot drop-off time (minute)	1	1
Customer desired time window (minute)	120	1
Customer tolerable time window (minute)	180	1
Truck cost (£/hr)	30	1
Robot cost (£/hr)	1	1
Drone cost (£/hr)	5	1
Local depot fixed cost (£/selected depot)	10	1

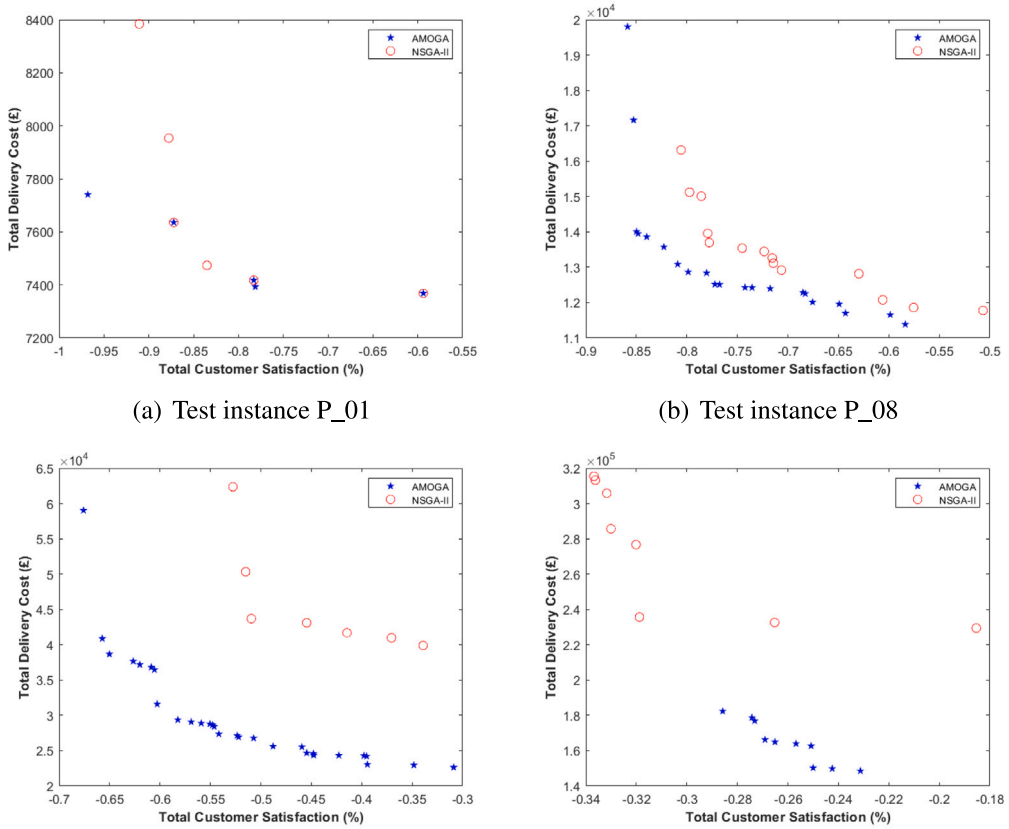


Fig. 8. Non-dominated solutions of various test instances achieved by AMOGA and NSGA-II.

and mutation probabilities, with values set at 50, 3, 0.4, and 0.3, respectively.

The termination criterion is established based on the maximum number of iterations for each test problem. Should the algorithm meet this condition, the process will cease, and the Pareto front solutions will be presented as the final outcome. Given the varying scales of problems, larger-sized instances may require additional iterations or extended run-time for effective resolution. Consequently, the maximum number of iterations for the test cases is set between 200 and 3000, depending on the problem size.

5.4. Computational results and discussion

In this section, the performance of the proposed AMOGA in solving various MOVRP models is evaluated. Initially, AMOGA is benchmarked

Table 4
Computational results of AMOGA and NSGA-II based on performance metrics (best in bold).

Instances name	Size	*SM*		*DM*		*MID*		*QM*		*IGD*		Run time (sec)	
		AMOGA	*NSGA − II*	*AMOGA*	*NSGA − II*	*AMOGA*	*NSGA − II*	*AMOGA*	*NSGA − II*	*AMOGA*	*NSGA − II*	*AMOGA*	*NSGA − II*
P_01	MOVRP_DR_1.8.1.1.1.1.	**0.73**	1.78	372.41	**1015.86**	**142.84**	345.24	**0.50**	**0.50**	**15.96**	18.54	25.36	**23.07**
P_02	MOVRP_DR_1.10.2.1.2.1	**0.83**	1.43	1692.29	**2594.10**	**713.39**	852.92	**0.24**	0.76	**24.21**	46.73	50.75	**37.66**
P_03	MOVRP_DR_1.10.2.1.2.2	1.12	**0.00**	**3506.31**	0.00	1018.16	**681.47**	**0.75**	0.25	**32.59**	1036.34	56.16	**45.90**
P_04	MOVRP_DR_1.12.2.1.1.2	**0.43**	1.42	1261.97	**10876.03**	**750.67**	1716.36	0.44	**0.56**	**61.51**	69.17	54.23	**45.71**
P_05	MOVRP_DR_1.12.2.1.2.1	**0.86**	1.69	2785.98	**23990.70**	**1091.94**	3479.48	**0.50**	**0.50**	**87.26**	159.79	60.57	**48.74**
P_06	MOVRP_DR_1.12.2.1.2.2	1.32	**0.84**	**11501.07**	3388.90	**2253.48**	2691.37	**0.83**	0.17	**5.96**	1029.56	72.14	**51.83**
P_07	MOVRP_DR_1.15.3.1.2.1	**0.95**	1.21	3979.75	**5287.85**	**2483.26**	3589.50	**0.92**	0.08	**0.00**	459.50	76.03	**63.74**
P_08	MOVRP_DR_1.15.3.1.2.2	1.19	**0.85**	**8418.86**	4539.76	**1790.50**	2109.33	**0.95**	0.05	**0.00**	350.97	76.63	**62.65**
P_09	MOVRP_DR_1.15.3.1.2.3	1.01	**0.97**	6087.36	**8510.34**	**1825.69**	2297.65	**0.84**	0.16	**26.07**	983.88	81.73	**67.08**
P_10	MOVRP_DR_1.18.3.1.2.3	0.86	**0.74**	**10889.86**	10250.47	**3763.23**	4476.52	**0.64**	0.36	**107.53**	379.36	149.08	**124.13**
P_11	MOVRP_DR_1.18.3.1.3.2	0.88	**0.66**	4672.07	**7592.52**	2618.71	**1820.44**	**0.83**	0.17	**35.29**	728.28	146.88	**131.05**
P_12	MOVRP_DR_1.18.3.1.3.3	0.57	**0.37**	4221.71	**7922.20**	**2172.05**	3050.17	**0.86**	0.14	**0.00**	1221.28	152.29	**135.07**
P_13	MOVRP_DR_1.20.3.1.2.3	**0.45**	1.11	11634.29	**30898.30**	6983.81	**6069.32**	0.43	**0.57**	**258.82**	633.10	227.11	**189.34**
P_14	MOVRP_DR_1.20.3.1.3.2	1.09	**0.63**	17846.89	**20658.03**	**5073.85**	12052.67	**0.83**	0.17	**167.41**	1516.18	212.89	**183.09**
P_15	MOVRP_DR_1.20.3.1.3.3	**0.81**	1.07	13166.13	**17616.40**	**5086.70**	8059.82	**0.82**	0.18	**398.29**	963.26	228.71	**191.34**
P_16	MOVRP_DR_1.25.3.1.2.3	0.56	**0.32**	3936.13	**15450.26**	**1796.19**	12153.66	**0.96**	0.04	**0.00**	6247.43	305.94	**242.41**
P_17	MOVRP_DR_1.25.3.1.3.2	**0.89**	0.96	5846.86	**17224.91**	**1667.52**	12141.41	**0.91**	0.09	**1218.04**	7283.06	301.56	**232.31**
P_18	MOVRP_DR_1.25.3.1.3.3	**0.82**	1.12	4214.95	**5035.18**	**905.50**	1038.01	**0.79**	0.21	**31.71**	288.65	195.72	**174.82**
P_19	MOVRP_DR_1.30.3.1.2.3	1.09	**0.96**	3021.90	**5115.11**	**2056.58**	2794.56	**0.92**	0.08	**37.42**	726.71	215.36	**184.73**
P_20	MOVRP_DR_1.30.3.1.3.2	1.00	**0.72**	3296.94	**5092.08**	**1329.44**	1825.32	**0.77**	0.23	**49.99**	89.47	215.64	**196.27**
P_21	MOVRP_DR_1.30.3.1.3.3	1.22	**1.00**	**36405.45**	22476.28	**7226.40**	23383.35	**0.97**	0.03	**0.00**	10852.33	402.99	**324.12**
P_22	MOVRP_DR_1.40.4.1.3.3	**0.66**	0.76	8593.66	**12234.12**	**1806.44**	2039.78	**0.92**	0.08	**16.89**	40.71	106.29	**88.22**
P_23	MOVRP_DR_1.50.4.1.4.4	**0.51**	1.38	4471.60	**9194.52**	**1856.62**	3054.32	**0.80**	0.20	**12.11**	56.29	103.51	**84.09**
P_24	MOVRP_DR_1.60.4.1.4.4	0.93	**0.79**	10984.12	**13456.92**	**2132.81**	3124.56	**0.69**	0.31	**37.67**	132.50	187.32	**161.23**
P_25	MOVRP_DR_1.70.4.1.4.4	**0.74**	1.12	8241.09	**17159.85**	**2785.64**	4919.76	**0.81**	0.19	**51.45**	153.12	220.08	**176.72**
P_26	MOVRP_DR_1.80.5.1.5.5	1.02	**0.95**	7356.19	**13867.24**	**3282.11**	4562.37	**0.85**	0.15	**62.74**	187.81	303.33	**252.71**
P_27	MOVRP_DR_1.90.5.1.5.5	0.93	**0.66**	6232.55	**12841.70**	**2461.55**	3199.25	**0.94**	0.06	**28.17**	145.80	271.18	**230.23**
P_28	MOVRP_DR_1.100.5.1.5.5	**0.52**	1.03	12115.23	**21850.90**	**2583.76**	4175.83	**0.93**	0.07	**45.96**	129.72	310.85	**259.48**
P_29	MOVRP_DR_1.110.6.1.5.5	1.07	**0.95**	5689.81	**14825.33**	**3182.45**	4126.15	**0.91**	0.09	**77.94**	220.12	362.77	**307.71**
P_30	MOVRP_DR_1.120.6.1.5.5	**0.77**	1.22	5409.64	**11673.46**	**1960.59**	3275.52	**0.88**	0.12	**64.72**	98.53	323.74	**286.18**

against NSGA-II across 30 test instances of the MOVRP_DR model using five performance metrics. Subsequently, the proposed algorithm is applied to other variants of the MOVRP model for further analysis.

5.4.1. Compare AMOGA and NSGA-II

The comparison between AMOGA and NSGA-II is based on 30 test instances of the MOVRP_DR model, analyzed using five key performance metrics. The detailed results are provided in Table 4. The best-obtained solutions are highlighted in bold. When evaluating the first criterion, SM, it is evident that both algorithms achieve a uniform spread of non-dominated solutions in nearly half of the test instances. For the second metric, DM, NSGA-II demonstrates superior diversity in the Pareto-front solutions across the majority of test instances. However, in terms of proximity and quality of the non-dominated solutions, as assessed by the MID and QM metrics, AMOGA outperforms NSGA-II, achieving solutions closer to the ideal point. Lastly, for the IGD metric, the results indicate that AMOGA consistently outperforms NSGA-II across all test instances, demonstrating superior convergence and diversity of solutions. In terms of runtime, AMOGA lags behind NSGA-II, likely due to the additional computational demands of the local search algorithm and the potentially time-intensive process of assigning specific fitness values.

To visually compare the Pareto-front solutions generated by the benchmark algorithms, four test instances of varying scales — P_01, P_08, P_21, and P_30 — are presented in Fig. 8. For clarity, the values of customer satisfaction have been converted to their negative counterparts.

Overall, the majority of the non-dominated solutions obtained by AMOGA surpass those of NSGA-II in terms of total delivery cost and customer satisfaction. This performance advantage can be attributed to AMOGA's integration of a local search algorithm and advanced fitness value assignment methods, which effectively determine the optimal launching and collecting nodes for drones and robots. These optimizations reduce truck waiting times and minimize travel distances for autonomous vehicles, leading to lower overall delivery costs and improved customer satisfaction. These results indicate that the proposed AMOGA is a more robust and effective approach for addressing the studied problem across varying scales.

5.4.2. Compare delivery modes

Building on the promising performance of our proposed algorithm co mpared to NSGA-II, we apply it in this section to 30 test instances of varying scales for the MOVRP_DR model. To benchmark the outcomes of our MOVRP_DR model against relevant studies in the literature, which focus exclusively on drone-assisted (MOVRP_D) or robot-assisted (MOVRP_R) last-mile delivery, we conduct extensive computational experiments. The primary goal of this research is to compare different models and their objective function values, specifically Total Routing Cost (TRC) and Total Customer Satisfaction (TCS). To facilitate this, we opted to use the TRC and TCS directly for comparison. A challenge arises when comparing objective values due to the presence of multiple non-dominated solutions in most test problems. These solutions are distributed across multi-dimensional spaces based on the number of objective functions, complicating direct comparisons. To address this, we focus on boundary solutions in cases where multiple non-dominated

Table 5
Number of assisted drone(s) and/or robot(s) for different models.

Instance no.	MOVRP_DR		MOVRP_D		MOVRP_R	
	D	R	D	R	D	R
*P*_01	1	1	2	0	0	2
*P*_02	1	2	3	0	0	3
*P*_03	2	2	4	0	0	4
*P*_04	2	1	3	0	0	3
*P*_05	1	2	3	0	0	3
*P*_06	2	2	4	0	0	4
*P*_07	2	2	4	0	0	4
*P*_08	1	2	3	0	0	3
*P*_09	3	2	5	0	0	5
*P*_10	3	2	5	0	0	5
*P*_11	2	3	5	0	0	5
*P*_12	3	3	6	0	0	6
*P*_13	3	2	5	0	0	5
*P*_14	2	3	5	0	0	5
*P*_15	3	3	6	0	0	6
*P*_16	3	2	5	0	0	5
*P*_17	2	3	5	0	0	5
*P*_18	3	3	6	0	0	6
*P*_19	3	2	5	0	0	5
*P*_20	2	3	5	0	0	5
*P*_21	3	3	6	0	0	6
*P*_22	3	3	6	0	0	6
*P*_23	4	4	8	0	0	8
*P*_24	4	4	8	0	0	8
*P*_25	4	4	8	0	0	8
*P*_26	5	5	10	0	0	10
*P*_27	5	5	10	0	0	10
*P*_28	5	5	10	0	0	10
*P*_29	5	5	10	0	0	10
*P*_30	5	5	10	0	0	10

Table 6
Computational Results of AMOGA for MOVRP_DR, MOVRP_D, and MOVRP_R.

Instances no.	Size	MOVRP_DR				MOVRP_D				MOVRP_R			
		TCS(b)	TCS(w)	TRC(b)	TRC(w)	TCS(b)	TCS(w)	TRC(b)	TRC(w)	TTCS(b)	TCS(w)	TRC(b)	TRC(w)
P_01	MOVRP_1.8.1.D.R	**0.96**	0.75	**7787.32**	**7916.62**	0.91	**0.85**	7855.30	8384.22	0.84	0.70	8231.52	9171.41
P_02	MOVRP_1.10.2.D.R	0.97	0.63	8243.64	9762.58	**0.99**	**0.92**	**7165.56**	**9542.92**	0.90	0.71	9277.80	20924.48
P_03	MOVRP_1.10.2.D.R	0.95	**0.85**	**7115.66**	10621.97	**1.00**	**0.85**	**7115.66**	**9843.41**	0.84	0.70	9391.95	22737.56
P_04	MOVRP_1.12.2.D.R	**0.97**	0.76	**7932.17**	16620.58	0.95	**0.90**	8279.27	**10143.25**	**0.97**	0.51	8259.99	17256.84
P_05	MOVRP_1.12.2.D.R	0.82	**0.76**	11084.28	18085.44	**0.83**	0.42	10313.46	**15226.63**	0.78	0.36	**9872.41**	19737.44
P_06	MOVRP_1.12.2.D.R	0.88	0.60	9972.69	16333.09	**0.89**	**0.60**	**8824.48**	**13190.22**	0.81	0.52	10208.42	13927.28
P_07	MOVRP_1.15.3.D.R	**0.92**	**0.75**	10112.88	17944.24	0.86	0.67	**9384.71**	**15280.88**	0.89	0.61	10954.67	20489.15
P_08	MOVRP_1.15.3.D.R l	**0.91**	**0.47**	11993.10	20334.62	0.77	0.43	**10393.55**	**15425.85**	0.84	0.39	14214.40	21944.57
P_09	MOVRP_1.15.3.D.R	**0.96**	**0.49**	**10135.39**	17804.08	0.95	0.43	11458.30	**14223.20**	0.91	0.42	13243.89	21751.16
P_10	MOVRP_1.18.3.D.R	**0.94**	0.44	14165.63	25225.86	0.85	**0.50**	**11790.10**	**19460.82**	0.76	0.56	14171.47	42035.17
P_11	MOVRP_1.18.3.D.R	**0.92**	**0.46**	11568.78	**18454.83**	0.73	0.39	**11560.79**	20154.74	0.89	0.43	13898.12	19226.00
P_12	MOVRP_1.18.3.D.R	**0.87**	**0.54**	16147.92	21483.93	0.81	0.33	**15023.81**	**20329.40**	0.81	0.63	16595.50	30015.79
P_13	MOVRP_1.20.3.D.R	0.84	**0.62**	**14616.62**	**23733.48**	**0.86**	0.57	16985.77	26582.92	0.79	0.46	16896.10	25328.62
P_14	MOVRP_1.20.3.D.R	**0.90**	**0.31**	**16107.89**	36483.43	0.86	0.26	16179.58	**25944.23**	0.72	0.18	17905.59	29514.94
P_15	MOVRP_1.20.3.D.R	**0.83**	**0.42**	16764.18	30024.11	0.66	0.33	**16373.21**	**25866.00**	0.79	0.35	20452.28	48163.47
P_16	MOVRP_1.25.3.D.R	**0.84**	**0.47**	17546.66	**27284.48**	0.67	0.28	**16914.23**	28139.93	0.81	0.49	21076.82	28628.47
P_17	MOVRP_1.25.3.D.R	**0.74**	0.28	**18711.75**	48247.02	0.65	**0.31**	23311.52	42833.56	0.71	0.39	23758.24	**31811.29**
P_18	MOVRP_1.25.3.D.R	**0.85**	0.30	18435.09	34171.27	0.82	**0.48**	18100.60	39604.55	0.70	0.48	**17800.12**	**26675.44**
P_19	MOVRP_1.30.3.D.R	**0.78**	0.19	**19395.92**	43893.75	0.77	0.23	22241.33	42965.39	0.70	**0.36**	22823.04	**29317.30**
P_20	MOVRP_1.30.3.D.R	**0.75**	**0.47**	24874.19	35554.15	0.74	0.22	21439.85	40186.98	0.66	0.33	**20702.97**	**34877.51**
P_21	MOVRP_1.30.3.D.R	**0.81**	0.21	**19304.05**	40123.28	0.67	**0.32**	20729.47	43768.33	0.59	0.31	22684.04	**32877.01**
P_22	MOVRP_1.40.4.D.R	**0.48**	**0.15**	**24947.80**	52176.71	0.44	0.13	**28699.50**	45812.55	0.42	0.15	30453.40	**41692.97**
P_23	MOVRP_1.50.4.D.R	**0.50**	0.17	**32832.27**	**49551.23**	0.41	0.22	47801.66	85077.46	0.38	**0.23**	36392.28	51689.37
P_24	MOVRP_1.60.4.D.R	**0.56**	**0.32**	**38664.95**	99910.06	0.54	0.23	43634.19	96341.48	0.42	0.28	42016.45	**53633.76**
P_25	MOVRP_1.70.4.D.R	**0.29**	0.12	70625.63	84919.56	0.29	**0.13**	73030.75	171055.81	0.22	0.12	**56767.95**	**78115.03**
P_26	MOVRP_1.80.5.D.R	0.44	**0.36**	90300.17	123166.32	**0.45**	0.25	93308.30	149388.46	0.40	0.33	**69190.33**	**95954.28**
P_27	MOVRP_1.90.5.D.R	0.27	**0.18**	82034.19	101822.77	**0.31**	0.09	106511.38	212266.49	0.23	0.13	**73930.08**	**82100.15**
P_28	MOVRP_1.100.5.D.R	0.22	0.08	104925.46	176272.50	**0.27**	**0.11**	137719.11	240616.59	0.18	0.10	**84106.71**	**113974.35**
P_29	MOVRP_1.110.5.D.R	**0.36**	**0.24**	120944.26	204092.88	0.33	0.22	173151.68	224855.73	0.35	0.24	**88473.73**	**157495.27**
P_30	MOVRP_1.120.5.D.R	0.29	**0.23**	148558.46	182380.20	**0.32**	0.22	182677.99	222725.23	0.28	0.24	**118532.28**	**128555.65**

solutions exist. This approach is justified, as any additional solutions are encompassed within these boundaries.

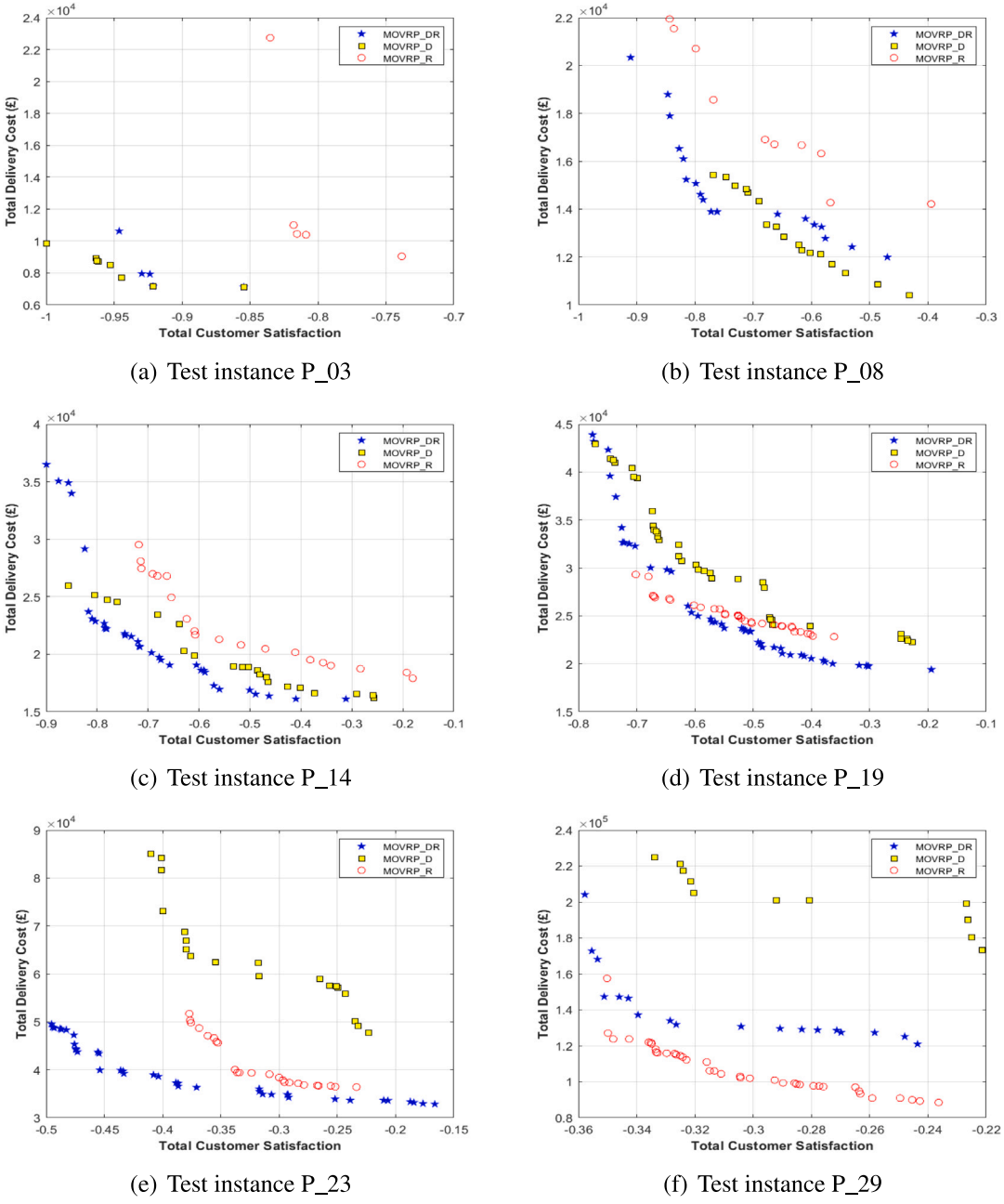
To make a fair comparison between various models, we assume the number(s) of assistants (drones and/or robots) for each assisted model is(are) equal. For example, in instance 1 (P_01), the MOVRP_DR model uses one drone and one robot, so the assigned assistants for its counterpart, MOVRP_R, should be equal to 2 robots instead of 1. Table 5 shows the number of assistants in terms of drone(s) and/or robot(s) in each test instance for different models. It is worth noting that the numbers of main depots, demand nodes, customer's local depot, and truck are presented in Table 6 for each testing example. Therefore, in the following sections, to avoid duplication, the numbers of drones and robots in the configuration (size) column are denoted as *D* and *R*, that need to be replaced with the appropriate number of drones and robots according to Table 5 for each model.

Table 6 presents a summary and comparison of the computational findings of AMOGA across various models. The first two columns show the test problem number and the model's configuration size. Columns 3 to 6 display the results of the MOVRP_DR model solved by our algorithm. Columns 7 to 10 present the outcomes of the MOVRP_D model, while columns 11 to 14 show the results of the MOVRP_R model. The abbreviations TCS(b) and TCS(w) denote the best and worst overall customer satisfaction in percentage, respectively. TRC(b) and TRC(w) represent the best and worst total route costs in British Pound Sterling (£). The best-obtained solutions are highlighted in bold.

In general, the findings indicate that in cases where there is a limited number of demand points distributed across various coordinates, the superiority of the MOVRP_D model emerges owing to the benefits associated with drones as previously discussed. Nevertheless, with an escalation in the number of clients and encountering moderate density in specific areas along with distant demand nodes, the performance of the MOVRP_DR model surpasses that of its counterparts as it integrates both drones and robots. Lastly, in densely populated regions, the MOVRP_R model proves to be more economically efficient. In addition, as the number of demand nodes increase, the total customer satisfaction will decrease.

Examining the intricacies of the outcomes reveals that the performance of the MOVRP_D model surpasses the other three models in majority of instances P_01 to P_06 in terms of both TCS and TRC objective functions, due to the longer distance between nodes in a fixed area. Because a drone is 1.5 times faster than a truck and its cost is only 1/6 of that of the truck, it, obviously, serves the assigned nodes in less time and lower cost. Despite the higher cost of employing drones compared to robots, drones can efficiently serve distant customers, thereby saving time and cost for trucks, which are more expensive. Comparing the results of MOVRP_R from applying AMOGA to the aforementioned instances indicate that the MOVRP_DR model outperforms due to its ability to maximize customer satisfaction and also reduce total route costs by utilizing drones. For instance, in Fig. 10(a), it is evident that drone services demand point 12, which is distant from other nodes, leading to improvement in both objective values. To visualize and compare the non-dominated solutions obtained for each model, Fig. 9 presents the outcomes yielded by AMOGA for all four models across various test instances. For clarity, the values of customer satisfaction have been converted to their negative counterparts. For example, Fig. 9(a) corresponding to instance P_03, illustrate that the solutions of the MOVRP_D model (yellow square) dominate those of its counterparts when there are few dispersed demand points.

With a slight increase in the density of demand points, from 12 to 18, the outcomes achieved by solving the MOVRP_DR model for test problems P_07 to P_12 surpass those of the MOVRP_D model and other counterparts, particularly in terms of customer satisfaction (TCS). However, due to the inherent trade-off between objective functions, the MOVRP_D model demonstrates superior performance over the other two approaches in minimizing total route cost (TRC). In essence, the MOVRP_DR model prioritizes maximizing customer satisfaction, which comes at the expense of higher costs compared to the MOVRP_D model. The latter leverages a greater number of drones to reduce truck waiting times, thereby minimizing route costs.



(a) Test instance P_03 (b) Test instance P_08

(c) Test instance P_14 (d) Test instance P_19

(e) Test instance P_23 (f) Test instance P_29

Fig. 9. Non-dominated solutions of various test instances using AMOGA.

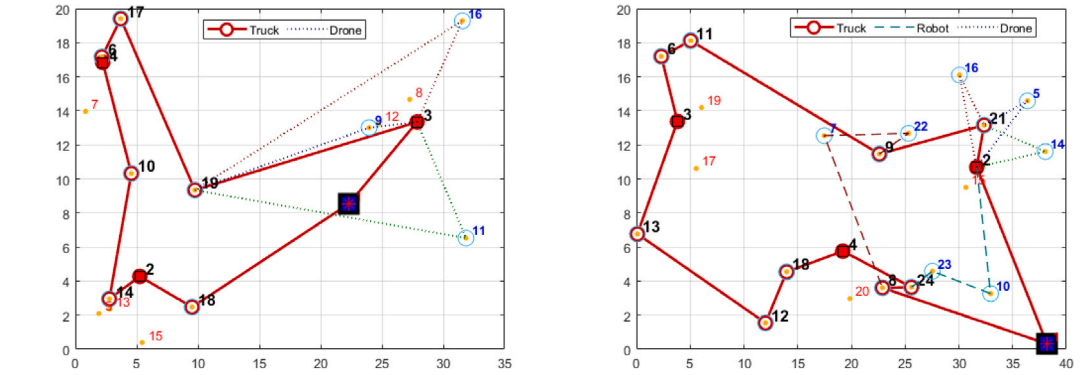
Table 7
Sensitivity analysis of customer time window for the MOVRP_DR model on test instance P_30.

Scenario No	Desired time window (hour)	Tolerable time window (hour)	TCS_best (%)	TCS_worst (%)	TRC_best	TRC_worst	TCS_best change (%)	TRC_best change (%)
1	2	3	28.58%	23.13%	148,558.46	182,380.20	0.00%	0.00%
2	3	4.5	29.02%	10.44%	119,701.19	211,344.21	1.55%	19.42%
3	4.5	6.5	36.19%	12.38%	114,245.39	127,930.70	24.69%	4.56%
4	6	8	37.64%	15.72%	121,873.85	218,630.99	4.01%	−6.68%
5	8	10	51.17%	16.41%	115,677.56	260,207.49	35.93%	5.08%

Fig. 9(b), corresponding to instance P_08, illustrates this dynamic: the solutions of the MOVRP_DR model (blue pentagrams) dominate those of its counterparts in terms of TCS, while the MOVRP_D model dominates in terms of TRC.

With a moderate increase in customer nodes, the outcomes achieved by solving the MOVRP_DR model for most test problems (P_13toP_24) surpass those of the MOVRP_D and MOVRP_R models. By effectively utilizing both drones and robots, the MOVRP_DR model achieves higher customer satisfaction while reducing total costs.

Fig. 9(c–e) compares the non-dominated solutions obtained for each model, highlighting the superior performance of the MOVRP_DR model (blue pentagrams). These solutions trend closer to the lower-left corner of the plots, indicating better trade-offs between customer satisfaction and total route costs (TRC). In contrast, the solutions of the MOVRP_D



(a) Test instance P_09 with TRC: 10135.39 (£) and TCS: 0.48

(b) Test instance P_13 with TRC: 14616.62 (£) and TCS: 0.62

Fig. 10. Routes of truck, robots, and drones for the MOVRP-DR model.

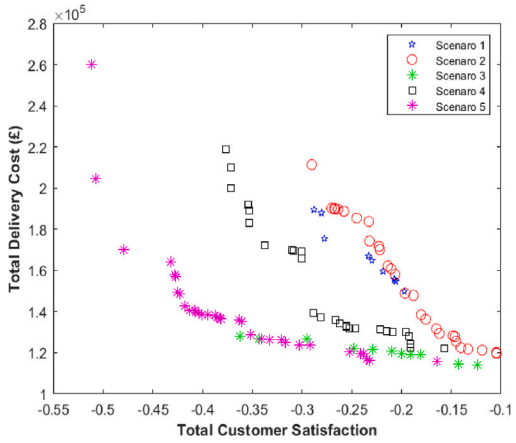


Fig. 11. Sensitivity analysis of customer time window for the MOVRP_DR model on test instance P_30.

model (yellow squares) deviate further from this region, particularly in terms of TRC. An interesting observation is the shift in the Pareto front of the MOVRP_R model towards the lower region, narrowing the gap with the MOVRP_DR solutions in terms of the total cost objective function.

As the density of customers within a specific area increases further (test instances P_25-P_30), it becomes evident that the cost objective of the MOVRP_R model outperforms the total route cost (TRC) objective of the other models. In densely populated areas, the MOVRP_R model capitalizes on the advantages of robots, which can serve nearby demand nodes at a lower cost. Additionally, due to their slower speed compared to trucks and drones, robots experience reduced waiting times when synchronizing with trucks, further enhancing efficiency. Interestingly, the customer satisfaction scores (TCS) for both the $MOVRP\_D$ and $MOVRP\_DR$ models tend to converge in these scenarios. As described in the test instances section, the demand nodes are randomly distributed within a fixed area of 40×20 across all test cases. This setup inherently results in higher node density as the number of demand points increases, making the use of robots particularly advantageous in such settings.

It is also important to note that the number of assistive units (robots and drones) remains consistent across all models. In the proposed MOVRP_DR model, this predetermined allocation of robots and drones limits flexibility. However, if additional robots or drones were available, the model could potentially further optimize assistance levels to achieve reductions in both objective functions, enhancing performance across all metrics.

Fig. 10 illustrates the routes of the truck, drones, and robots for test instances P_09 and P_13 in the MOVRP_DR model. For example, in Fig. 10(b), the bold red line represents the truck's route. Starting from the main depot (blue square: node 1), the truck travels to node 2 (local depot), where it launches three drones and one robot. After serving its assigned customers, the truck proceeds to node 21 to collect the drones.

Meanwhile, Robot 1 (depicted by a green dashed line) departs from the local depot to serve customers at nodes 10 and 23. It then moves to node 24, where it is later retrieved by the truck. After completing its tasks at node 21, the truck visits node 9, where Robot 2 (dark red dashed line) is deployed to serve nodes 22 and 7. The truck serves node 9, then continues its route, visiting nodes 2, 11, 6, 3, 13, 12, 18, 4, 24, and 8, where it collects Robot 2 and serves the final node. Finally, the truck returns to the main depot.

It is worth noting that nodes 15, 19, 17, and 20 represent customers who opted to pick up their orders directly from the local depot.

5.4.3. Sensitivity analysis

As the concluding phase of the numerical assessments, a sensitivity analysis is performed to evaluate how the objective functions — delivery cost and customer satisfaction — respond to variations in the length of the customer time window. Specifically, as the number of customers increases, companies often extend the delivery time slots due to resource limitations, which helps to reduce overall logistics costs. To examine this relationship, we analyze the impact of the time window length in a large-scale scenario consisting of 120 customer nodes within a high-population-density area, as summarized in Table 7. When the duration of the time window increases, a higher number of orders can be delivered on time, leading to an increase in customer satisfaction. This added flexibility in the time window also contributes to a reduction in total delivery costs. For example, doubling the time window length can lead to an approximate 25 percent improvement in the values of both objective functions. Fig. 11 illustrates how both objectives respond to variations in the time window length, demonstrating notable improvements in their respective values.

6. Conclusion

Autonomous technologies have shown great promise to improve supply chain cost-effectiveness and responsiveness. The transition to drone and robot-assisted truck delivery calls for improved operations management know-how and decision support. This study is one of the first attempts at the integrated planning of DART operations for last-mile parcel delivery. A new multi-objective optimization problem and

a solution algorithm are developed, and the DART delivery scheme is compared with two baselines.

The numerical experiments showed that DART delivery is beneficial when the demand network comprises both densely populated areas and distant nodes. Drone-assisted truck delivery performed best for small cases with a dispersed demand population. Addressing the limitations of drones, especially the load capacity and endurance may make this mode prevalent in the future of last-mile parcel delivery. Robot-assisted delivery is more competitive in highly dense urban areas where the company should extend customer reach. Although improving robots' load capacity is comparatively easier than in drones, addressing the travel speed limitations may worsen robots' endurance and face challenges over the pedestrian safety concerns. Overall, drone and robot integration seem needed where LMD demands are highly volatile. This integration adds a high level of flexibility to planning the delivery operations.

This study used random test instances to test the developed optimization method for planning LMD operations. Further analysis is required to compare AMOGA's performance with other multi-objective solution algorithms considering very large instances and a case study. Besides, the optimization problem can be extended by considering heterogeneous trucks, drones, and robots. Finally, decomposition-based and set partitioning methods can be used to improve the efficiency of the optimization method for commercial purposes.

CRediT authorship contribution statement

Ali Mokhtari-Moghadam: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Visualization. **Abdellah Salhi:** Conceptualization, Resources, Supervision, Writing – review & editing, Funding acquisition. **Xinan Yang:** Supervision, Project administration, Writing – review & editing. **Trung Thanh Nguyen:** Validation, Supervision, Writing – review & editing. **Pourya Pourhejazy:** Investigation, Methodology, Writing – review & editing, Visualization.

Declaration of competing interest

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

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, *52*(4), 965–981. http://dx.doi.org/10.1287/trsc.2017.0791, URL https://ideas.repec.org/a/inm/ortrsc/v52y2018i4p965-981.html.

Alverhed, E., Hellgren, S., Isaksson, H., Olsson, L., Palmqvist, H., & Flodén, J. (2024). Autonomous last-mile delivery robots: a literature review. *European Transport Research Review*, http://dx.doi.org/10.1186/s12544-023-00629-7.

Andreas, K. (2024). Sustainability and new technologies: Last-mile delivery in the context of smart cities. *Sustainability*, *16*(18), http://dx.doi.org/10.3390/su16188037, URL https://www.mdpi.com/2071-1050/16/18/8037.

Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, *72*(4), 528–542. http://dx.doi.org/10.1002/net.21864, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.21864.

Boysen, N., Fedtke, S., & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, *43*(1), 1–58. http://dx.doi.org/10.1007/s00291-020-00607-8.

Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, *271*(3), 1085–1099. http://dx.doi.org/10.1016/j.ejor.2018.05.058, URL https://www.sciencedirect.com/science/article/pii/S0377221718304776.

Chang, Y. S., & Lee, H. J. (2018). Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications*, *104*, 307–317. http://dx.doi.org/10.1016/j.eswa.2018.03.032, URL https://www.sciencedirect.com/science/article/pii/S0957417418301775.

Chen, C., Demir, E., & Huang, Y. (2021). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European Journal of Operational Research*, *294*(3), 1164–1180. http://dx.doi.org/10.1016/j.ejor.2021.02.027, URL https://www.sciencedirect.com/science/article/pii/S037722172100120X.

Das, D. N., Sewani, R., Wang, J., & Tiwari, M. K. (2021). Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, *22*, 5772–5782, URL https://api.semanticscholar.org/CorpusID:219501280.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. http://dx.doi.org/10.1109/4235.996017.

Figliozzi, M. A. (2020). Carbon emissions reductions in last mile and grocery deliveries utilizing air and ground autonomous vehicles. *Transportation Research Part D: Transport and Environment*, *85*, Article 102443. http://dx.doi.org/10.1016/j.trd.2020.102443, URL https://www.sciencedirect.com/science/article/pii/S1361920920306301.

Goldberg, D. E., & Lingle, R. (1985). Alleles, loci, and the traveling salesman problem. In *Proceedings of the 1st international conference on genetic algorithms and their applications* (pp. 154–159). Lawrence Erlbaum Associates, Publishers.

He, X., He, F., Li, L., Zhang, L., & Xiao, G. (2022). A route network planning method for urban air delivery. *Transportation Research Part E: Logistics and Transportation Review*, *166*, Article 102872. http://dx.doi.org/10.1016/j.tre.2022.102872, URL https://www.sciencedirect.com/science/article/pii/S1366554522002526.

Heimfarth, A., Ostermeier, M., & Hübner, A. (2022). A mixed truck and robot delivery approach for the daily supply of customers. *European Journal of Operational Research*, *303*(1), 401–421. http://dx.doi.org/10.1016/j.ejor.2022.02.028, URL https://www.sciencedirect.com/science/article/pii/S0377221722001333.

Jingi, A. M., & Yang, X. (2023). Robot-assisted delivery problems and their exact solutions. In B. Dorronsoro, F. Chicano, G. Danoy, & E.-G. Talbi (Eds.), *Optimization and learning* (pp. 341–353). Cham: Springer Nature Switzerland.

Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *32*(1), 1–13. http://dx.doi.org/10.1109/TSMCC.2002.1009117.

Kim, W., & Hur, S. H. (2024). Why and why not? Systematic review of the willingness to accept drone and robot deliveries. *Transportation Research Record*, Article 03611981241248439. http://dx.doi.org/10.1177/03611981241248439.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, *129*, 14–30. http://dx.doi.org/10.1016/j.cie.2019.01.020, URL https://www.sciencedirect.com/science/article/pii/S0360835219300245.

Kloster, K., Moeini, M., Vigo, D., & Wendt, O. (2023). The multiple traveling salesman problem in presence of drone- and robot-supported packet stations. *European Journal of Operational Research*, *305*(2), 630–643. http://dx.doi.org/10.1016/j.ejor.2022.06.004, URL https://www.sciencedirect.com/science/article/pii/S0377221722004593.

Lemardelé, C., Estrada, M., Pagès, L., & Bachofner, M. (2021). Potentialities of drones and ground autonomous delivery devices for last-mile logistics. *Transportation Research Part E: Logistics and Transportation Review*, *149*, Article 102325. http://dx.doi.org/10.1016/j.tre.2021.102325, URL https://www.sciencedirect.com/science/article/pii/S1366554521000995.

Liu, D., Yan, P., Pu, Z., Wang, Y., & Kaisar, E. I. (2021). Hybrid artificial immune algorithm for optimizing a van-robot E-grocery delivery system. *Transportation Research Part E: Logistics and Transportation Review*, *154*, Article 102466. http://dx.doi.org/10.1016/j.tre.2021.102466, URL https://www.sciencedirect.com/science/article/pii/S1366554521002295.

Luo, Q., Wu, G., Ji, B., Wang, L., & Suganthan, P. N. (2022). Hybrid multi-objective optimization approach with Pareto local search for collaborative truck-drone routing problems considering flexible time windows. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 13011–13025. http://dx.doi.org/10.1109/TITS.2021.3119080.

Mohammad, W. A., Diab, Y. N., Elomri, A., & Triki, C. (2023). Innovative solutions in last mile delivery: concepts, practices, challenges, and future directions. *Supply Chain Forum: An International Journal*, *24*, 151–169, URL https://api.semanticscholar.org/CorpusID:256680263.

Mokhtari-Moghadam, A., Pourhejazy, P., & DeepakGupta (2023). Integrating sustainability into production scheduling in hybrid flow-shop environments. *Environmental Science and Pollution Research*, http://dx.doi.org/10.1007/s11356-023-26986-3.

Morim, A., Campuzano, G., Amorim, P., Mes, M., & Lalla-Ruiz, E. (2024). The drone-assisted vehicle routing problem with robot stations. *Expert Systems with Applications*, *238*, Article 121741. http://dx.doi.org/10.1016/j.eswa.2023.121741, URL https://www.sciencedirect.com/science/article/pii/S0957417423022431.

Ostermeier, M., Heimfarth, A., & Hübner, A. (2023). The multi-vehicle truck-and-robot routing problem for last-mile delivery. *European Journal of Operational Research*, *310*(2), 680–697. http://dx.doi.org/10.1016/j.ejor.2023.03.031, URL https://www.sciencedirect.com/science/article/pii/S0377221723002552.

Ramos, T. R. P., & Vigo, D. (2023). A new hybrid distribution paradigm: Integrating drones in medicines delivery. *Expert Systems with Applications*, *234*, Article 120992. http://dx.doi.org/10.1016/j.eswa.2023.120992, URL https://www.sciencedirect.com/science/article/pii/S095741742301494X.

Said, M., Aeschliman, S., & Stathopoulos, A. (2023). Robots at your doorstep: acceptance of near-future technologies for automated parcel delivery. *Scientific Reports*, *13*, 18556. http://dx.doi.org/10.1038/s41598-023-45371-1.

Simoni, M. D., Kutanoglu, E., & Claudel, C. G. (2020). Optimization and analysis of a robot-assisted last mile delivery system. *Transportation Research Part E: Logistics and Transportation Review*, *142*, Article 102049. http://dx.doi.org/10.1016/j.tre.2020.102049, URL https://www.sciencedirect.com/science/article/pii/S1366554520307006.

Statista (2022). Projected market size of the autonomous last mile delivery worldwide from 2020 to 2028. URL https://www.statista.com/statistics/1103574/autonomous-last-mile-delivery-market-size-worldwide. (Last Accessed 24 August 2023).

Tu, P. A., Dat, N. T., & Dung, P. Q. (2018). Traveling salesman problem with multiple drones. In *Proceedings of the 9th international symposium on information and communication technology* (pp. 46–53). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3287921.3287932.

Wang, K., Pesch, E., Kress, D., Fridman, I., & Boysen, N. (2022). The piggyback transportation problem: Transporting drones launched from a flying warehouse. *European Journal of Operational Research*, *296*(2), 504–519. http://dx.doi.org/10.1016/j.ejor.2021.03.064, URL https://www.sciencedirect.com/science/article/pii/S0377221721003106.

Yu, S., Puchinger, J., & Sun, S. (2022). Van-based robot hybrid pickup and delivery routing problem. *European Journal of Operational Research*, *298*(3), 894–914. http://dx.doi.org/10.1016/j.ejor.2021.06.009, URL https://www.sciencedirect.com/science/article/pii/S0377221721005154.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. In *TIK report*. ETH Zurich, Computer Engineering and Networks Laboratory, URL https://api.semanticscholar.org/CorpusID:16584254.