Swarm and Evolutionary Computation 79 (2023) 101295

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Solving electric vehicle–drone routing problem using memetic algorithm

Setyo Tri Windras Mara *, Ruhul Sarker, Daryl Essam, Saber Elsayed

School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia

ARTICLE INFO

Keywords:
Vehicle routing problem with drones
Electric vehicles
Unmanned aerial vehicles
Last-mile logistics
Future transportation systems
Memetic algorithm

ABSTRACT

This paper considers a cooperative system between electric vehicles and drones in last-mile logistics operations. This form of cooperation is presented as electric vehicle–drone routing problem (EVDRP). In EVDRP, a set of electric vehicles are equipped with drones to deliver parcels to a set of customer nodes. Due to the battery limitation of electric vehicles, a set of available recharging stations is considered to recharge the vehicles' energy and/or to launch and retrieve drones. In this research, a mixed-integer programming model is formulated for EVDRP to minimize the total completion time. To solve the model, a memetic algorithm-based approach with four new problem-specific operators is developed. From comprehensive numerical experiments, the effectiveness of the algorithm is shown, and several useful managerial insights are derived.

1. Introduction

Managing the last-mile part of a logistics network is a complex task. Last-mile logistics comprises "*the very last stage of supply chain and logistics network that brings the goods to the doorstep of the consumers*" [1] and largely deals with the last stretch of a business-to-customer (B2C) delivery service. It is widely known as the most inefficient part of logistics networks, where some studies found that the costs of operational activities in last-mile logistics may account for up to 28% of the overall transportation costs [1] and up to 41% of the overall supply chain costs [2].

One of the major causes of operational inefficiency in last-mile logistics is the lack of economies of scale. Most B2C shipments in the last-mile logistics are performed as customer-direct deliveries [1]. This involves the delivery of small-size packages directly to the user's doorstep as opposed to bulk delivery to the retail stores, leading to an abundance of expensive and inefficient home deliveries [3].

Recently, the rising trend of e-commerce and online shopping has led to an increasing volume of customer direct deliveries. Since the start of the COVID-19 pandemic, there has been a further increase in home delivery, which is predicted to continue in post-pandemic [4]. This trend could become a major concern for both companies and society. Besides its costly nature, the increasing number of customer direct deliveries potentially adds extra burdens to the urban infrastructure and may lead to congestion and numerous health, environmental and safety problems in urban areas [3]. Considering the costs and negative externalities of this trend, optimizing the process within last-mile logistics operations becomes imperative.

In response to the challenge of managing last-mile logistics, researchers and practitioners have made a lot of effort. In recent years, several emerging technologies have been introduced as potential solutions to make last-mile logistics operations more effective and efficient. Recent work from Dong et al. [5] presented a review of nine major emerging technologies for last-mile logistics, such as automated robots, autonomous vehicles, drones, artificial intelligence, big data analytics, the internet of things, blockchain and electric vehicles (EVs). In their review, Dong et al. [5] emphasized the importance of studying the integration between various new technologies, as this will assist the convergence of these technologies for future implementation purposes. Responding to this query, this study aims to explore the cooperative operations between EVs and drones in last-mile logistics.

In recent years drones have received large amounts of attention from the operations research and computer science communities, due to their potential and capabilities for logistics. Compared to traditional vehicles for freight transportation (i.e. trucks, vans), drones are faster, can avoid traffic congestion in a road network, and incur lower acquisition and transportation costs [6]. Furthermore, a long list of well-known companies has taken their interests in drones as a future mainstream transportation mode, where several of them have also invested in notable drone-aided logistics projects to bring the implementation of drones to the mainstream, such as DHL with their "parcelcopter" program [7] and Walmart with their investment in DroneUp services [8].

Macrina et al. [9] reviewed the operational aspects of drones in logistics and classified them into two major classes: (i) trucks and

* Corresponding author.
E-mail addresses: s.windras_mara@adfa.edu.au (S.T.W. Mara), r.sarker@adfa.edu.au (R. Sarker), d.essam@adfa.edu.au (D. Essam), s.elsayed@unsw.edu.au (S. Elsayed).

https://doi.org/10.1016/j.swevo.2023.101295
Received 14 December 2022; Received in revised form 28 February 2023; Accepted 21 March 2023
Available online 4 April 2023

drones perform the deliveries [see 10], and (ii) only drones perform the deliveries [see 11]. Among those two classes, it has been noted that the first class (trucks and drones perform the deliveries) has drawn more attention from researchers and practitioners. The reasoning for this trend can be traced from the current limitations of drones to be solely implemented in a last-mile logistics system, as processing numerous shipments using only drones might lead to a scalability issue due to the significant upfront investment and current technical restrictions of drone technology, such as payload and battery limitations [3]. Correspondingly, the majority of studies on the operational aspect of drone-aided logistics systems have been focusing on the optimization of a coordinated system between trucks and drones. The main idea of this cooperation is to leverage the unique characteristics of each vehicle mode. In this regard, the technical restrictions of drones are compromised by embedding drones into trucks, which have much larger capacity and traveling range [10].

Among multiple operational models of truck–drone coordinated systems addressed in the literature, the cooperation between them in a tandem way is perhaps the most promising one. Studies on this tandem cooperation largely stem from two seminal studies from Murray and Chu [10], who introduced the concept of the flying sidekick traveling salesman problem (FSTSP), and Agatz et al. [12], who conceptualized a similar model named traveling salesman problem with drone (TSPD). Both FSTSP and TSPD can be loosely described as a coordinated logistics system where a single ground vehicle (GV) (i.e. truck) is equipped with a single aerial vehicle (i.e. drone) to perform parcel deliveries to customer locations. The aerial vehicle can make multiple deliveries (sorties) throughout the GV tour. In this regard, the operator of GV can launch the drone, use it to deliver a parcel, and then retrieve the drone back along the way. Within the last several years, this tandem-based model has been receiving substantial recognition from researchers and industry practitioners, as multiple derivation models and industrial implementations have been successfully introduced [see 13].

In this study, we consider the truck–drone tandem routing model, with the implementation of EVs as a GV, instead of the traditional internal combustion engine vehicles (ICEVs). Similar to drones, the deployment of EVs for commercial logistics purposes has been a recently emerging topic [14,15], as the reduction of transport emissions has become more crucial than ever. For reference, the transportation sector is widely known as a large contributor to greenhouse gas (GHG) emissions. This sector is responsible for 15% of total CO2 emissions [16], while a recent report from Nat [17] revealed that the transportation sector is responsible for about 18.9% of the whole GHG emissions of Australia. Considering this issue, EVs emerge as a promising technological solution to gradually substitute ICEVs. In the last several years, for instance, some major third-party logistics (3PL) companies, such as FedEx [18] and DHL [19], have also been noted for their initiatives to speed up the implementation of EVs in commercial logistics.

Nonetheless, it is observed that EV adoption in commercial logistics has been slower than expected. A handful of studies have identified several barriers to the adoption of EVs for commercial purposes, such as its high upfront investment, limited range, and its dependency on the limited necessary infrastructure, such as recharging stations [20]. However, considering the long-term importance of EVs in attaining sustainability in logistics, we believe that these present issues should not discourage the research community to study this topic.

To this end, although numerous studies have been done to analyze the operational aspect of both drone-aided [see 9] and EV-based [see 15] last-mile logistics systems, we notice that studies on the integration of them are still very scarce. Therefore, this study aims to both fill this gap and enhance the discussion of deploying EVs and drones in last-mile logistics. In this regard, we hypothesize that adding drones to a fleet of EVs can be a good alternative to enhance the attractiveness of adopting EVs.

The objectives of this study are threefold:

(i) *Constructing a mathematical model for EVDRP.*
Here, EVDRP is modeled as a mixed-integer linear programming (MILP) formulation to minimize the total completion time of delivery missions. This objective function is particularly selected to evaluate the impact of the EVDRP system from the perspective of performance, following the seminal study of Murray and Chu [10] in FSTSP. Compared to several related models available in the literature, the originality of our model can be explained as follows. First, our EVDRP model can be seen as a generalization of electric vehicle traveling salesman problem with drone (ETSPD) presented in Zhu et al. [21,22]. Second, our EVDRP model differs from the model presented by Kyriakakis et al. [23] in the sense that both EVs and drones can deliver parcels to the customer nodes, while Kyriakakis et al. [23] assumed that EVs are only used for transporting drones. We also consider the presence of recharging stations in our model, while Kyriakakis et al. [23] assumed that EVs have no energy limitation in their model.

(ii) *Developing an effective solution approach for EVDRP.*
To solve the developed EVDRP model, this study proposes a memetic algorithm (MA). MA is an extension of a genetic algorithm that incorporates the usage of local search operators to 'educate' solution vectors generated by genetic operators. Accordingly, an encoding scheme is developed here to represent an EVDRP solution as a solution vector (also called a 'chromosome'), and computational experiments are conducted to evaluate the performance of our proposed algorithm.

(iii) *Deriving insights from implementing the EVDRP model and solution approach.*
Based on the computational experimental study, this study provides some managerial insights on the integration of EVs and drones into last-mile delivery systems.

The remainder of this paper is structured as follows. Section 2 presents a literature review on related works. Section 3 formulates EVDRP and presents the corresponding mathematical model. Section 4 introduces the proposed solution method in detail. Section 5 describes the executed numerical experiments and presents the experiment results. Based on the results of our experiments, Section 6 presents a managerial discussion. Finally, Section 7 summarizes the findings of this study and discusses some future research directions.

2. Literature review

This section provides a review of previous works related to our study. This review starts with the discussions of previous literature on routing problems with cooperation between GVs and drones (Section 2.1), with an emphasis on TSPD and vehicle routing problem with drones (VRPD). Afterwards, the discussions move to the integrated models of EVs with drones (Section 2.2) and a review of solution methods in Section 2.3. Finally, Section 2.4 summarizes the contributions of this study.

2.1. Routing problems with cooperation between GVs and drones

The discourse on optimizing routing problems with cooperation between GVs and drones was first started by Murray and Chu [10]. Murray and Chu [10] introduced a seminal mathematical formulation of coordination between GV and drone, namely FSTSP. FSTSP considers a single truck assisted by a single drone to perform a parcel delivery task to a set of customer nodes in a graph. In this form of cooperation, a human operator can operate the attached drone by launching the drone from a launching node (a depot node or a customer node) to deliver a parcel to a single customer node. While the drone performs its sortie, the truck can serve other customer nodes before recollecting the drone in a rendezvous node, as long as the endurance constraint

of the drone is not violated. Murray and Chu [10] then presented a 'truck-first, drone-second' heuristic algorithm to solve the FSTSP, in which the route of the truck is generated first as a complete traveling salesman problem (TSP) tour. Afterwards, sortie operations are inserted iteratively by calculating the best possible savings that can be made from assigning drone-eligible customers to the drone. Another influential work from Agatz et al. [12] presented an integer programming formulation for a similar model to FSTSP, namely TSPD. By comparing their models to the classical TSP model, these two seminal studies successfully illustrated how the idea of using drones in last-mile logistics with the GV as its 'moving hub' could reduce the time required to complete all delivery tasks. For simplicity purposes, we will refer to both models as TSPD for the rest of this paper.

Since the studies from Murray and Chu [10], Agatz et al. [12], numerous studies have attempted to extend TSPD, and correspondingly, several notable extensions have been introduced. For instance, Ha et al. [24], Ha et al. [25] introduced a variant of TSPD with a cost-minimization objective. The min-cost TSPD from Ha et al. [24] aims to minimize the total operational costs, which comprises the traveling cost of the truck, traveling cost of the drone to perform the sorties, and the waiting costs incurred when a vehicle must wait for another to represent the parking and labor fees. Ha et al. [24] then proposed a greedy randomized adaptive search procedure (GRASP) algorithm as a solution method for the min-cost TSPD. Afterwards, another study from Ha et al. [25] continued the work of Ha et al. [24] by proposing a hybrid genetic algorithm (HGA) to solve min-cost TSPD, where the HGA was shown to be superior to GRASP by Ha et al. [24]. Marinelli et al. [26] extended TSPD by considering the capability of executing *en-route* operations, where the drone can be launched and retrieved along the arcs. In developing their model, Marinelli et al. [26] adapted the set partitioning formulation of TSPD [12], which partitioned a TSPD problem into a set of truck and drone subtours. For each partial subtour, a preprocessing calculation task is performed to calculate the optimal points to launch and retrieve the drone. As a solution method, Marinelli et al. [26] then adapted the GRASP algorithm of Ha et al. [24].

Jeon et al. [27] recently discussed an interesting extension of TSPD by considering a pickup-or-delivery scenario. In their model, right after finishing its line-haul delivery, the drone can visit another customer node to pick up a back-haul load before returning to the truck with the corresponding load. By implementing this scenario with a real-world case study, Jeon et al. [27] showed that the number of drone's empty traveling distances could be reduced. In a similar essence, several studies in TSPD [see 28,29] also considered the scenario where the drone is capable of visiting multiple customer nodes within a single sortie. These studies confirmed that allowing the drone to perform deliveries to more customer nodes leads to the reduction of total completion time and traveling cost, as drones are generally faster and have a lower travel cost per unit than trucks. Nevertheless, it is obvious that the adoption of this multi-visit scenario depends on the technical advancement (i.e. larger payload and battery capacity) and the corresponding acquisition price of drones [29]. Another study from Phan et al. [30] presented an idea to consider the situation where a single truck is equipped with multiple drones in a new model called TSP with multiple drones (TSP-mD). They proposed a MILP formulation for TSP-mD and developed a heuristic solution based on the adaptive large neighborhood search (ALNS) algorithm. A similar model was considered by two studies from Murray and Raj [31], Raj and Murray [32]. Murray and Raj [31] discussed the importance of proper scheduling of the retrievals, launches, and truck service, which might happen in a single customer node, and incorporated these decisions into their proposed MILP formulation. Then, Murray and Raj [31] presented a three-phased heuristic solution approach by decomposing TSP-mD into three decision steps: (i) initial assignment of customers to the truck, (ii) determine the sorties, and (iii) determine the activity start times for both truck and drones. The TSP-mD formulation from Murray and Raj [31] was then extended in Raj and Murray [32] by relaxing the assumption that drones fly at a certain speed in all arcs and considering the speed of drones as a decision variable. By allowing the drones to operate at a different speed for each arc (up to the maximum allowable speed), Raj and Murray [32] found that both the travel distance of truck and drone energy consumption per trip can be reduced.

Similar to traditional routing problems, one of the major extensions of TSPD is deploying multiple vehicles. This extension corresponds to the vehicle routing problem with drones (VRPD) models. VRPD was first discussed by Wang et al. [33], Poikonen et al. [34]. Wang et al. [33] performed a worst-case analysis to obtain the time savings bound that can be achieved by implementing a VRPD system; then, this study was continued in Poikonen et al. [34] who discussed the effect of drone battery limitation and different travel metrics between vehicles in VRPD. Schermer et al. [35] followed the idea of Marinelli et al. [26] by considering the ability to make *en-route* operations along the arcs in VRPD. Schermer et al. [35] proposed a hybrid variable neighborhood search (VNS) as a solution approach for the VRPD with *en-route* operators. Popovic et al. [36] then proposed a three-index mixed-integer quadratic programming formulation for VRPD, with an objective function to minimize the distribution cost based on the total distance traveled by vehicles and their total working time. Similarly, a seminal study from Sacramento et al. [37] separately proposed a MILP formulation to model VRPD. As a solution method for VRPD, Sacramento et al. [37] adopted the popular ALNS, in which Sacramento et al. [37] proved the effectiveness of their ALNS algorithm using a set of experiments on 112 test instances. Another study from Wang and Sheu [38] proposed an extension of VRPD by considering the capability of drones to make multiple visits in a single sortie [see 28,29,39]. In accordance to this, Wang and Sheu [38] built an arc-based MILP formulation for VRPD, then developed an exact branch-and-price algorithm as a solution method. Several studies also did consider the presence of customer time windows in their VRPD models. Di Puglia Pugliese and Guerriero [40] developed a MILP formulation for VRPD with time windows (VRPTWD), demonstrating the capability of the commercial solver CPLEX 12.5.1 to solve several small-size instances up to 10 customer nodes. More recently, the study from Kuo et al. [41] also proposed a MILP formulation of VRPTWD with an objective function to minimize the total traveling costs of the system. Kuo et al. [41] then developed a VNS-based heuristic approach for VRPTWD, which was shown to be more effective than the ALNS of Sacramento et al. [37].

2.2. Routing problems with cooperation between EVs with drones

The implementation of EVs as a more environmental-friendly transportation mode has been a hot topic in operations research and computer science within the last decade. A large number of research works have been dedicated to assessing the optimal way to implement EVs in logistics, particularly in routing, where several related surveys have been introduced to review this field of study [see 14,15].

Perhaps the most popular modeling variant in EV routing problems is an electric vehicle routing problem (EVRP), which deals with "finding optimized routes for EV, taking battery constraints and charging operations into account" [15]. Research on this topic was pioneered by two influential studies from Conrad [42], Erdoğan and Miller-Hooks [43]. Although both studies did not specifically state EV usage in their proposed models, they paved the way for other studies in EV routing problems afterwards. Conrad [42] introduced the recharging vehicle routing problem (VRP), where the vehicle has a limited range distance, requiring a refueling activity at the customer nodes. On the other hand, Erdoğan and Miller-Hooks [43] developed a model, called green VRP, which employs alternative-fueled vehicles instead of the traditional inner combustion vehicles. After these two studies, numerous studies have attempted to extend the state-of-the-art of EVRP modeling. In this regard, we list several notable works on this topic as follows. Schneider et al. [44] discussed an extension of EVRP under the presence of time windows and recharging stations to minimize

the total traveled distance by the set of EVs, where each EV has a limited driving range and may need to attend the intra-route recharging stations to finish its tour. Schneider et al. [44] noted that, although requiring a non-trivial establishment cost, employing these recharging stations is advantageous to avoid inefficient long detours to the depot. A heuristic algorithm based on a hybridization of VNS and tabu search was then proposed by Schneider et al. [44] as a solution method, where they demonstrated its applicability to optimize a set of problem instances with up to 100 customer nodes. In retrospect, considering intra-route recharging stations in Schneider et al. [44] has led to fruitful discussions in this field of study. Keskin and Çatay [45] considered the capability to perform a partial recharging policy in the recharging stations, relaxing the initial assumption in Schneider et al. [44] that the battery of EVs must be fully recharged after visiting a recharging station. Yang and Sun [46] discussed the usage of battery swapping stations within the graph, in which these swapping stations enable a short-time battery swap to reduce the recharging task, albeit with larger investment costs upfront to procure a set of batteries. In this regard, several works have also discussed the ideas to solve this upfront investment costs issue, such as the works of Koç et al. [47] that proposed an idea where several companies can jointly procure charging stations for their fleet of EVs and collaboratively use these stations. Furthermore, Schiffer et al. [48] presented a comprehensive review of research in routing problems with intra-route facilities.

Nonetheless, literature on the integration between EVs and drones is still very scarce. Our literature review only found four works that deal with the implementation of EVs in this context. Baek et al. [49] first introduced an energy-efficient delivery system with a single EV and drone. In their model, Baek et al. [49] aimed to minimize the total energy consumption of the vehicles and assumed that the drone can serve a single customer node within a single sortie. Additionally, Baek et al. [49] also assumed that the drone battery cannot be replaced; thus, the drone will only be used until the battery is totally depleted. Then, as a solution approach, Baek et al. [49] developed a simple greedy heuristic approach. Another study from Zhu et al. [21] proposed a MILP formulation for ETSPD. Zhu et al. [21] considered the presence of several recharging stations on the graph, where the EV may need to visit these charging stations to recharge its battery. Zhu et al. [21] also assumed that both the EV and drone use the same energy source. As such, the energy state of EV is reduced correspondingly whenever the drone performs a sortie. In their study, Zhu et al. [21] also proposed an iterative heuristic algorithm, and they demonstrated its effectiveness in solving instances with up to 25 customer nodes. More recently, this study of Zhu et al. [21] was continued by Zhu et al. [22], who extended the ETSPD with a partial recharge policy. An algorithm based on ALNS and constraint programming was presented in Zhu et al. [22] where they showed that the performance of the ANLS is better than the baseline VNS algorithm. Meanwhile, another recent study from Kyriakakis et al. [23] attempted to scale up this cooperation by considering the presence of multiple EVs and drones. Kyriakakis et al. [23] presented a MILP formulation for EVDRP to minimize the total energy consumption of the system. A hybrid ant colony optimization (ACO) algorithm was presented as a heuristic approach for instances with up to 50 customer nodes. The effectiveness of the hybrid ACO was demonstrated by comparing its performance with several versions of ACO-based algorithms. However, it must be noted that Kyriakakis et al. [23] assumed the EVs are restricted for launching-and-retrieval purposes, as the customer nodes are served only by drones. In this regard, we believe that the model presented by Kyriakakis et al. [23] belongs to the class of 2-echelon routing problems [see 50] with EVs as a moving satellite. As such, for clarity, we will refer to the model from Kyriakakis et al. [23] as a 2-echelon EVDRP.

2.3. Review on relevant solution methods

In accordance to the proposal of our solution approach for EVDRP, a brief review of related solution methods is presented here. To start with, routing optimization problems are widely known as difficult combinatorial problems to solve. Even for its simplest form, such as TSP, the search space can grow exponentially as the scale of the problem increases. This limits the effectiveness of exact algorithms to be implemented, as finding the exact solution for problems with a realistic size is often impracticable. In this regard, developing a heuristic algorithm based on metaheuristics has been a popular alternative to finding a quasi-optimal solution in a relatively short time.

In the literature body of VRPD and its derivations (i.e. VRPTWD), some implementations of metaheuristics are available. We note that an iterative search equipped with neighborhood operators is a popular choice for this. For instance, the study from Schermer et al. [51] proposed the first heuristic algorithm for VRPD, in which they developed a two-phase heuristic with four different neighborhood moves to explore the search space. Another study from Schermer et al. [35] developed a hybrid VNS and tabu search to deal with an extension of VRPD with *en-route* operations. Afterwards, the aforementioned study from Sacramento et al. [37] developed an ANLS heuristic for VRPD, whilst similarly, the study from Coindreau et al. [52] proposed to solve VRPTWD with an ALNS algorithm. More recently, Kuo et al. [41] presented a VNS algorithm with eight different operators to solve VRPTWD, where they showed the superiority of VNS to ALNS in solving VRPTWD. Wu et al. [39] also proposed an improved variable neighborhood descent (IVND) approach for solving a collaborative truck–drone routing problem. In this approach, they incorporated a tabu list to avoid the local search returning back to a visited solution point. In addition, a repair strategy was also introduced by them to remove infeasible sorties in the current solution. Meanwhile, several studies also documented the usage of swarm intelligence techniques, particularly ACO and artificial bee colony (ABC). ACO was recently implemented by Huang et al. [53] who deal with solving VRPD. However, it needs to be noted that Huang et al. [53] considered a different problem structure from Sacramento et al. [37], Kuo et al. [41], in which Huang et al. [53] restricted that drones must return to the adjacent node after serving a certain drone node. In other words, Huang et al. [53] assumed a truck cannot serve any node while its corresponding drone performs a sortie. Another study from Han et al. [54] presented an ABC-based algorithm to solve a VRPTWD with multiple objectives, aiming to minimize the total energy consumption and the number of trucks deployed.

In this study, we attempted to propose a solution algorithm for EVDRP based on the paradigm of evolutionary algorithms. In this regard, we found only one previous study that implemented an evolutionary algorithm to solve problems related to VRPD. A recent study from Euchi and Sadok [55] proposed an MA-based algorithm to solve VRPD, namely a hybrid genetic-sweep algorithm. Euchi and Sadok [55] integrated a sweep algorithm to improve the performance of GA. Their computational experiments indicated that the proposed hybrid genetic-sweep achieved superior performance to the ALNS of Sacramento et al. [37]. Unfortunately, the authors did not present clear documentation on how they incorporated the sweep algorithm into the GA framework for solving VRPD and how they inducted the drone sorties into the generated truck tours.

2.4. Summary of contributions

In summary, the contributions of this study are as follows. First, this is the first study to extend the VRPD model [37] with the implementation of EVs. We differ in our contributions with Kyriakakis et al. [23] as their proposal has a stronger relationship with studies on 2-echelon routing problems with drones [see 56]. Borrowing the classification terminologies from Moshref-Javadi and Winkenbach [13], our EVDRP model can be classified as a "synchronized multi-modal/simultaneous"

(SM/S) model, while the model of Kyriakakis et al. [23] falls to the "synchronized multi-modal/non-simultaneous" (SM/N) class. Second, with respect to the studies from Baek et al. [49], Kyriakakis et al. [23], Zhu et al. [21], and Zhu et al. [22], our study can be seen as an early study on the integration between EVs and drones in a last-mile logistics system. In this regard, our model can also be seen as an extension of the ETSPD model presented by Zhu et al. [21]. Third, a MILP formulation of EVDRP is developed in this study. To the best of our knowledge, this is the first mathematical model for EVDRP with SM/S coordination available in the literature. Fourth, this study extends the solution scheme of VRPTWD from Kuo et al. [41] into a new solution scheme suitable for EVDRP. To the best of our knowledge, we are not aware of any previous studies that have provided a similar contribution. Lastly, we propose a MA as a solution approach for EVDRP, which is the first attempt to develop an evolutionary algorithm to solve the integration between EVs and drones in logistics. In addition, the proposed MA is also equipped with four new local search operators to explore the search space generated by the presence of recharging nodes.

3. Problem formulation

This section formally describes EVDRP and formulates the problem as a mathematical program. Here, several assumptions are taken into account in developing this model, which are as follows:

(i) A set of EVs and drones is available and each EV is equipped with one drone.

(ii) All EV routes must start and end in the depot node.

(iii) Along an EV tour, the drone can be deployed to perform multiple sorties. However, it should be launched from one node and retrieved at another node in the direction of the EV's path.

(iv) During a sortie, the corresponding EV can visit other nodes.

(v) Each drone must be launched from and be recollected by the same EV. These operations must be performed either in a depot node, a customer node, or a recharging station.

(vi) If one vehicle arrives earlier than the other one, it must wait at the rendezvous node for synchronization.

(vii) A drone can be safely landed at the rendezvous node before the arrival of EV.

(viii) Full recharging policy is applied in the recharging stations.

(ix) Drones always start their sortie in a fully charged state.

(x) Instant battery swap is assumed for drones, and then the swapped batteries are recharged using the energy of their corresponding EV.

(xi) All customer nodes must be visited once, and by only one EV-drone tandem.

(xii) During a single visit, the human operators of EV-drone tandem need to perform several tasks sequenced as follows: retrieving the drone (if any), fulfilling the customer order (if any), recharging the EV (if any), and launching the drone (if any).

Then, the EVDRP can be defined in an undirected graph $G = (V, A)$ by considering these assumptions. $V = V_0 \cup V_S \cup V_C$ is the set of all nodes in graph G, while set A consists of all arcs (i, j) between nodes in the set V. Accordingly, set $V_0 = \{0, r+n+1\}$ stands for the set of depot nodes and its dummy node. Set $V_S = \{1, \dots, r\}$ consists of all recharging station nodes. Set $V_C = \{r+1, \dots, r+n\}$ represents all the n customer nodes that need to be served.

The purpose of EVDRP is to fulfill all the customer demands $\{q_i | i \in V_C\}$ using a fleet of m EV-drone tandems $F = \{1, \dots, m\}$. Each EV has a limited payload capacity Q_t, while the drones also have a corresponding payload capacity Q_d. Given that $Q_t > Q_d$, some customer nodes might not be able to be served by a drone. Therefore, we introduce a subset $V_D \subseteq V_C$ to define all the customer nodes that can be served by a drone ($q_i \leq Q_d$).

Along its tour, the operator of the EV can launch and recollect the drones to perform a sortie operation multiple times. The sortie operation can be defined as a tuple $\langle i, j, k \rangle$, where the drone is launched at a launching node $i \in V_L = \{0, \dots, r+n\}$, flying to serve a customer node $j \in V_D$, then be recollected at the rendezvous node $k \in V_R = \{1, \dots, r+n+1\}$ before its flight endurance E_d runs out. After the retrieval, it is assumed that the drone batteries will be swapped and the used batteries are recharged using the energy from EV [see 21]. In a similar sense, the EVs also have a limited driving range, given by an energy capacity E_t. This energy is drained from the movement of EVs to perform their tour as well as recharge the drone batteries. As such, an EV might occasionally need to visit a recharging station before it runs out of energy. In this regard, we let h_t be the energy consumption rate of EVs, h_d be the energy consumption rate of drones, and R_t be the recharging rate of EV batteries.

The objective of this model is to minimize the total completion time required to serve all customer nodes. In order to calculate the objective function, let us introduce $t^t_{i,j}$ as the time required for EVs to traverse through arc $(i, j) \in A$, and $t^d_{i,j}$ as the corresponding travel time parameter of drones. The calculation of travel time is done using Eqs. (1) and (2), where $d^t_{i,j}$ and $d^d_{i,j}$ respectively stand for the travel distance between arc $(i, j) \in A$ using EVs and drones, while v_t and v_d represents the speed of EVs and drones. In addition, let us also introduce ρ_L and ρ_R as the time required to launch and recollect the drones, and s^t_i and s^d_i respectively as the time required to serve customer i using either EV or drone. Finally, using all the notations presented in Table 1, the MILP formulation of EVDRP can be fully presented as follows.

$$t^t_{i,j} = \frac{d^t_{i,j}}{v_t} \tag{1}$$

$$t^d_{i,j} = \frac{d^d_{i,j}}{v_d} \tag{2}$$

Objective function

$$\min Z = \sigma \tag{3}$$

subject to:

BASIC ROUTING CONSTRAINTS

$$\sum_{f \in F} \sum_{i \in V_L \setminus \{j\}} x_{i,j,f} + \sum_{f \in F} \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{i,j,k,f} = 1 \qquad \forall j \in V_C \tag{4}$$

$$\sum_{j \in V_N \cup \{r+n+1\}} x_{0,j,f} \leq 1 \qquad \forall f \in F \tag{5}$$

$$\sum_{i \in \{0\} \cup V_N} x_{i,r+n+1,f} \leq 1 \qquad \forall f \in F \tag{6}$$

$$\sum_{i \in V_L \setminus \{j\}} x_{i,j,f} - \sum_{k \in V_R \setminus \{j\}} x_{j,k,f} = 0 \qquad \forall f \in F,\ j \in V_N \tag{7}$$

$$u_{i,f} - u_{j,f} + 1 \leq M_1(1 - x_{i,j,f}) \qquad \forall f \in F,\ i \in V_L \setminus \{j\},\ j \in V_R \setminus \{i\} \tag{8}$$

$$u_{j,f} \leq M_1 \sum_{i \in V_L \setminus \{j\}} x_{i,j,f} \qquad \forall f \in F,\ j \in V_R \tag{9}$$

$$u_{j,f} - u_{i,f} \leq M_1 p_{i,j,f} \qquad \forall f \in F,\ i \in V_L \setminus \{j\},\ j \in V_N \setminus \{i\} \tag{10}$$

$$u_{j,f} - u_{i,f} \geq M_1(p_{i,j,f} - 1) + 1 \qquad \forall f \in F,\ i \in V_L \setminus \{j\},\ j \in V_N \setminus \{i\} \tag{11}$$

$$\sum_{j \in V_C} \Big(\sum_{k \in V_R \setminus \{j\}} q_j x_{j,k,f} + \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} q_j y_{i,j,k,f} \Big) \leq Q_t \qquad \forall f \in F \tag{12}$$

SORTIE CONSTRAINTS

$$y_{i,j,k,f} = 0 \qquad \forall f \in F,\ i \in V_L \setminus \{j,k\},\ j \in V_C \setminus \{V_D, i, k\},\ k \in V_R \setminus \{i,j\} \tag{13}$$

$$\sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{i,j,k,f} \leq 1 \qquad \forall f \in F,\ i \in V_L \tag{14}$$

$$\sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{i,j,k,f} \leq 1 \qquad \forall f \in F,\ k \in V_R \tag{15}$$

Table 1
Summary of notations for mathematical model.

Symbol	Description
Sets and notations	
n	Number of customer nodes to be served
r	Number of recharging stations available
m	Number of EV-drone tandems available
V	Set of all nodes, $V = V_0 \cup V_S \cup V_C$
V_0	Set of depot node and its dummy, $V_0 = \{0, r+n+1\}$
V_S	Set of recharging stations, $V_S = \{1, \dots, r\}$
V_C	Set of customer nodes, $V_C = \{r+1, \dots, r+n\}$
V_D	Subset of drone-eligible customers, $V_D \subseteq V_C$
V_L	Subset of launching nodes, $V_L = \{0, 1, \dots r+n\}$
V_R	Subset of rendezvous nodes, $V_R = \{1, \dots r+n+1\}$
A	Set of all (i, j) arcs, where $i, j \in V$ and $i \neq j$
F	Set of all EV-drone tandems, $F = \{1, \dots, f\}$
V_N	Set of all nodes except the depot nodes, $V_N = V_S \cup V_C$
Parameters	
$d^t_{i,j}$	Travel distance of EVs from nodes i to j
$d^d_{i,j}$	Travel distance of drones from nodes i to j
$t^t_{i,j}$	Travel time of EVs from nodes i to j
$t^d_{i,j}$	Travel time of drones from nodes i to j
q_i	Demand of node i
s^t_i	Service time incurred in node i if it is served by an EV
s^d_i	Service time incurred in node i if it is served by a drone
Q_t	Payload capacity of EVs
Q_d	Payload capacity of drones
E_t	Energy capacity of EVs
E_d	Flight endurance of drones
v_t	Travel speed of EVs
v_d	Travel speed of EVs
h_t	Energy consumption rate of EVs
h_d	Energy consumption rate of drones
R_t	Recharging rate of EVs at a recharging node
ρ_L	Setup time required to launch a drone
ρ_R	Setup time required to retrieve a drone
M_1, M_2	Sufficiently large numbers
Decision variables	
σ	Continuous: to indicate the completion time of all tandems
$x_{i,j,f}$	Binary: '1' if arc (i, j) is traveled by EV f, '0' otherwise
$y_{i,j,k,f}$	Binary: '1' if drone f performs sortie $< i, j, k >$, '0' otherwise
$p_{i,j,f}$	Binary: '1' if node i is served before j (not necessary to be adjacent), '0' otherwise
$a^t_{i,f}$	Continuous: to indicate the arrival time of EV f at node i
$a^d_{i,f}$	Continuous: to indicate the arrival time of drone f at node i
$e^t_{i,f}$	Continuous: to indicate the remaining battery capacity of EV f on arrival at node i
$u_{i,f}$	Integer: to indicate the position of node i in the tour of EV f

$$2(y_{i,j,k,f}) \leq \sum_{m \in V_R \setminus \{i\}} x_{i,m,f} + \sum_{m \in V_L \setminus \{k\}} x_{m,k,f} \quad \forall f \in F,\ i \in V_L \setminus \{j,k\},\ j \in V_C \setminus \{i,k\},\ k \in V_R \setminus \{i,j\} \tag{16}$$

$$y_{0,j,k,f} \leq \sum_{i \in V_L \setminus \{j,k\}} x_{i,k,f} \quad \forall f \in F,\ j \in V_D \setminus \{k\},\ k \in V_R \setminus \{j\} \tag{17}$$

TIMING CONTINUITY CONSTRAINTS

$$a^t_{i,f} + \rho_L \Big(\sum_{j \in V_D \setminus \{i,m\}} \sum_{m \in V_R \setminus \{i,j\}} y_{i,j,m,f} \Big) + t^t_{i,k} + \rho_R \Big(\sum_{m \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{m,k\}} y_{m,j,k,f} \Big) + s^t_k - M_2(1 - x_{i,k,f}) \leq a^t_{k,f} \quad \forall f \in F,\ i \in V_L \setminus \{k\},\ k \in V_R \setminus \{i, V_S\} \tag{18}$$

$$a^t_{i,f} + \rho_L \Big(\sum_{j \in V_D \setminus \{i,m\}} \sum_{m \in V_R \setminus \{i,j\}} y_{i,j,m,f} \Big) + t^t_{i,k} + \rho_R \Big(\sum_{m \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{m,k\}} y_{m,j,k,f} \Big) + \left(\frac{E_t - e^t_{k,f}}{R_t} \right) - M_2(1 - x_{i,k,f}) \leq a^t_{k,f} \quad \forall f \in F,\ i \in V_L \setminus \{k\},\ k \in (V_R \cap V_S) \setminus \{i\} \tag{19}$$

$$a^t_{i,f} + \rho_L + t^d_{i,j} + s^d_j - M_2 \Big(1 - \sum_{k \in V_R \setminus \{i,j\}} y_{i,j,k,f}\Big) \leq a^d_{j,f} \quad \forall f \in F,\ i \in V_L \setminus \{j\},\ j \in V_D \setminus \{i\} \tag{20}$$

$$a^d_{j,f} + t^d_{j,k} + \rho_R + s^t_k - M_2 \Big(1 - \sum_{i \in V_L \setminus \{j,k\}} y_{i,j,k,f}\Big) \leq a^d_{k,f} \quad \forall f \in F,\ j \in V_D \setminus \{k\},\ k \in V_R \setminus \{j, V_S\} \tag{21}$$

$$a^d_{j,f} + t^d_{j,k} + \rho_R + \Big(\frac{E_t - e^t_{k,f}}{R_t}\Big) - M_2 \Big(1 - \sum_{i \in V_L \setminus \{j,k\}} y_{i,j,k,f}\Big) \leq a^d_{k,f} \quad \forall f \in F,\ j \in V_D \setminus \{k\},\ k \in (V_R \cap V_S) \setminus \{j\} \tag{22}$$

TIMING SYNCHRONIZATION CONSTRAINTS

$$a^t_{i,f} - M_2 \Big(1 - \sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{i,j,k,f}\Big) \leq a^d_{i,f} \quad \forall f \in F,\ i \in V_L \tag{23}$$

$$a^t_{i,f} + M_2 \Big(1 - \sum_{j \in V_D \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y_{i,j,k,f}\Big) \geq a^d_{i,f} \quad \forall f \in F,\ i \in V_L \tag{24}$$

$$a^t_{k,f} - M_2 \Big(1 - \sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{i,j,k,f}\Big) \leq a^d_{k,f} \quad \forall f \in F,\ k \in V_R \tag{25}$$

$$a^t_{k,f} + M_2 \Big(1 - \sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in V_D \setminus \{i,k\}} y_{i,j,k,f}\Big) \geq a^d_{k,f} \quad \forall f \in F,\ k \in V_R \tag{26}$$

$$a^d_{k,f} - M_2(3 - p_{i,p,f} - \sum_{j \in V_C \setminus \{i,k\}} y_{i,j,k,f} - \sum_{q \in V_C \setminus \{p,r\}} \sum_{r \in V_R \setminus \{p,q\}} y_{p,q,r,f}) \leq a^d_{p,f}$$
$$\forall f \in F,\ i \in V_L \setminus \{p,k\},\ p \in V_N \setminus \{i,k\},\ k \in V_R \setminus \{i,p\} \quad (27)$$

TIMING LIMITATION CONSTRAINTS

$$a^t_{0,f} = 0 \qquad \forall f \in F \quad (28)$$

$$a^d_{0,f} = 0 \qquad \forall f \in F \quad (29)$$

ENERGY CONSTRAINTS

$$e^t_{0,f} = E_t \qquad \forall f \in F \quad (30)$$

$$e^t_{j,f} - (h_t \cdot d^t_{j,k} \cdot x_{j,k,f}) - h_d \left(\sum_{i \in V_L \setminus \{m,k\}} \sum_{m \in V_D \setminus \{i,k\}} y_{i,m,k,f} \cdot (t^d_{i,m} + s^d_m + t^d_{m,k}) \right) + E_t(1 - x_{j,k,f}) \geq e^t_{k,f}$$
$$\forall f \in F,\ j \in V \setminus \{k, r+n+1, VS\},\ k \in V \setminus \{j, 0\} \quad (31)$$

$$E_t - (h_t \cdot d^t_{j,k} \cdot x_{j,k,f}) - h_d \left(\sum_{i \in V_L \setminus \{m,k\}} \sum_{m \in V_D \setminus \{i,k\}} y_{i,m,k,f} \cdot (t^d_{i,m} + s^d_m + t^d_{m,k}) \right) \geq e^t_{k,f}$$
$$\forall f \in F,\ j \in V_S \setminus \{k\},\ k \in V \setminus \{j, 0\} \quad (32)$$

$$t^d_{i,j} + s^d_j + t^d_{j,k} - M_2(1 - y_{i,j,k,f}) \leq E_d$$
$$\forall f \in F,\ i \in V_L \setminus \{j,k\},\ j \in V_D \setminus \{i,k\},\ k \in V_R \setminus \{i,j\} \quad (33)$$

STATE OF DECISION VARIABLES

$$\sigma \geq a^t_{r+n+1,f} \qquad \forall f \in F \quad (34)$$

$$\sigma \geq a^d_{r+n+1,f} \qquad \forall f \in F \quad (35)$$

$$x_{i,j,f} \in \{0,1\} \qquad \forall f \in F,\ i,j \in A \quad (36)$$

$$y_{i,j,k,f} \in \{0,1\} \qquad \forall f \in F,\ i \in V_L \setminus \{j,k\},\ j \in V_D \setminus \{i,k\},\ k \in V_R \setminus \{i,j\} \quad (37)$$

$$p_{i,j,f} \in \{0,1\} \qquad \forall f \in F,\ i \in V_L \setminus \{j\},\ j \in V_N \setminus \{i\} \quad (38)$$

$$u_{i,f} \in \mathbb{N} \qquad \forall f \in F,\ i \in V \quad (39)$$

$$a^t_{i,f}, a^d_{i,f}, e^t_{i,f} \in \mathbb{R}_+ \qquad \forall f \in F,\ i \in V \quad (40)$$

$$\sigma \in \mathbb{R}_+ \quad (41)$$

The objective function (3) aims to minimize the maximum completion time of all vehicle tours. This objective function is subject to several sets of constraints. Eqs. (4)–(12) are the basic routing constraints. Constraint (4) guarantees that each customer node is only visited once, either by an EV or a drone. Constraints (5) and (6) set that each tour must start and finish at the depot node. Constraint (7) is the flow conservation constraint to ensure that a vehicle entering any node (except the dummy depot node) will leave the corresponding node. Constraints (8) and (9) are subtours elimination constraints that ensure the connectivity of EV tours. Constraints (10) and (11) guarantee the sequence of EV tours so that each customer and the recharging node will not appear multiple times within a single tour. Constraint (12) is used to limit the payload capacity of EVs, which corresponds to the total demand of customer nodes served by both EV f and its corresponding drone.

Constraints (13)–(17) regulate the sortie operations. Constraint (13) guarantees that drones will only serve drone-eligible nodes. This guarantee is achieved by setting the value of $y_{i,j,k,f}$ to '0' for all $j \notin V_D$. In other words, this constraint serves as a payload capacity constraint for drones. Constraint (14) ensures that each drone can only be launched from each launching node in V_L at most once. Correspondingly, Constraint (15) ensures that each drone can only be retrieved from each rendezvous node in V_R at most once. Constraint (16) states that drones can only be launched and retrieved at two different nodes i and k, respectively. These nodes must be visited by the corresponding EV, although not necessarily in an adjacent order. Constraint (17) ensures that the EV must visit the rendezvous node $k \in V_R$ from any node when its corresponding drone is launched from the depot and finishes its sortie at node k.

Constraints (18)–(30) are a set of constraints related to the timing of tours. Constraints (18)–(22) impose the continuity of arrival time. Constraints (18) and (19) set the arrival time of EV f at node k to be larger than its arrival time at the predecessor node i. Constraint (18) regulates the situation when the successor node k is not a recharging node, while Constraint (19) deals with the condition when the successor node k is a recharging node. Constraint (20) guarantees that the arrival time of drone f at a drone-eligible node j is larger than the arrival time of its corresponding EV (and the drone itself) at the launching node i. Accordingly, Constraints (21) and (22) guarantee that the arrival time of drone f at a rendezvous node k is larger than the arrival time of drone f at the drone-visited node j. On the other hand, the synchronization of arrival time between EVs and drones is regulated by Constraints (23)–(27). Constraints (23) and (24) ensure that the arrival times of EV and drone f are the same at launching node i. Constraints (25) and (26) guarantee that the arrival times of EV and drone f are synchronized at rendezvous node k. Constraint (27) guarantees that EV f cannot launch its drone if the drone f is still on a sortie and has not been retrieved yet. Constraints (28) and (29) set the arrival time of all vehicles at the depot node $\{0\}$ as zero. Note that this formulation involves several big-M constraints of two types, M_1 and M_2. Here, M_1 is the bound on the number of nodes visited within a tour, while M_2 is the bound on the tour duration. Following Sacramento et al. [37],Kuo et al. [41], the value of M_1 can be efficiently set as $|V|$, which corresponds to the worst-case solution when a single tour is deployed to serve all customer nodes while visiting all recharging stations. The value of M_2 can be efficiently set as the maximum duration of a tour, which corresponds to the upper bound of σ. Sacramento et al. [37],Kuo et al. [41] set the value of M_2 by providing a maximum duration of a tour (i.e 480 min). However, in the case where the maximum tour duration is unbounded, a tight approximation of M_2 is desired. In this regard, we can approximate the value of M_2 with an optimal min-time TSP tour that visits all customer nodes.

Constraints (30)–(33) are the energy-related constraints. Constraint (30) regulates the battery level of EVs at the start of the tour, where all EVs are equipped with a fully-charged battery. Constraint (31) is employed to track the battery level of EVs after it traverses through arc (i,j). Similarly, Constraint (32) is the recharging constraint of EVs, which regulates the battery state of EV f at the successor node j after recharging node i. Constraint (33) is the endurance constraint of drones that sets the maximum flight time of drones. Finally, Constraints (34) and (35) set the value of σ as the maximum completion time of all tours, while Constraints (36)–(41) define the range value of all decision variables.

4. Solution method

This section describes the proposed solution method in detail. Section 4.1 presents the solution scheme developed in this study and the encoding–decoding process of this scheme. Section 4.2 provides a complete explanation of the proposed algorithm, and then, Section 4.3 explains the calculation of the objective function.

4.1. Solution representation

The solution scheme for EVDRP is developed based on the solution scheme of VRPTWD from Kuo et al. [41], which consists of two arrays with the same length. On one hand, the first array ('upper part', U) represents the delivery tours performed as well as the sequence of visits. This U array consists of:
(i) an integer permutation of n customer nodes,
(ii) $f+1$ depot nodes 0 to express the start and termination of each tour, and
(iii) m recharging nodes, each with an integer value $m_i \in \{0\} \cup V_S$ to represent the visitation of EVs to a recharging station.

Similarly, the second array ('lower part', L) contains $(n+f+1+m)$ items, where each item has a binary value $L_i \in \{0,1\}$. These values of L_i denote the mode of transport to visit the corresponding node in U_i. Fig. 1 illustrates a sample of an EVDRP case with 14 customer nodes, 3 recharging stations, and 3 EV-drone tandems. Accordingly, the corresponding solution scheme of the sample case is also provided in Fig. 1, in which the numerical encoding of the upper part is done using the same order as in the set V (see Table 1). From Fig. 1, it is shown how the presence of recharging stations in EVDRP enlarges the solution space in several ways, as these recharging stations could also be used as a launching node and rendezvous node. This is similar to the usage of parking lots to aid a drone-based routing delivery [57]. Then, in order to provide a clear understanding of how this solution scheme works, we refer readers to Appendix A.

4.2. Memetic algorithm

In this study, we developed a memetic algorithm (MA) for solving our EVDRP model. MA is a sub-branch of evolutionary algorithms that hybridizes an evolutionary framework with a set of local search components [58]. The effectiveness of MA has been shown in numerous previous studies related to EVDRP [see 59], where this algorithm framework was notably cited as "perhaps the most successful variant of evolutionary algorithms in combinatorial optimization" by Sörensen and Sevaux [60].

This section describes the proposed MA. First, a visual presentation of the algorithm is presented in Fig. 2, while the complete pseudocode is given in Algorithm 1. The proposed algorithm comprises three main components: generating an initial population of individuals, search/genetic operators, and a local search scheme. These components are discussed below.

4.2.1. Initialization procedure

One key to the successful implementation of MA is providing a good starting point [58]. In this regard, as seen in Fig. 2, providing initial solutions with good quality for EVDRP can be achieved by decomposing EVDRP into four sub-problems: (i) clustering the customer nodes, (ii) finding the optimal vehicle tour for each cluster, (iii) inserting recharging nodes, and (iv) inserting the drone sorties. These sub-problems are solved sequentially, which results in a feasible solution for EVDRP.

Constrained clustering

Decomposing an instance into several clusters and solving each cluster as a TSP has been known as one traditional technique in solving VRP [61]. This technique aims to deconstruct VRP into a set of smaller problems that can be easily solved, and as the structure of EVDRP is related to VRP, this technique is very suitable for creating the base solution structure for EVDRP.

In the first step, we derive several clusters from the present EVDRP instance. These clusters are derived with a consideration of EV payload capacity Q_t for each cluster, such that the total customer demand of each cluster is not violating Q_t. As such, we developed a constrained clustering procedure by simply modifying the classic K-means procedure with a consideration of the total demand for each cluster. In this regard, the procedure appends each customer node to the nearest cluster only if the capacity constraint is not violated, otherwise, it resorts to the next best available cluster.

Algorithm 1: Pseudocode of MA

Input: EVDRP instance, MA parameters $(POP_s, C_R, M_R, EL_R, POOL_s)$, N_{LIST}
Result: S^*_{EVDRP}, Z^*_{EVDRP}

$P \leftarrow \emptyset$
$P \leftarrow$ initializePopulation(POP_s)
$P \leftarrow$ fitnessEvaluation(P)
while *termination criterion is not met* **do**
  $MP \leftarrow$ createMatingPool$(P, POOL_s)$
  $P_{new} \leftarrow \emptyset$
  $P_{new} \leftarrow$ elitism(P_{new}, EL_R)
  while $|P_{new}| < POP_s$ **do**
    $Parent_A, Parent_B \leftarrow$ parentSelection(MP)
    $S_{EVDRP} \leftarrow$ orderCrossover$(C_R, Parent_A, Parent_B)$
    $S_{EVDRP} \leftarrow$ swapMutation(M_R, S_{EVDRP})
    $S_{EVDRP}, Z_{EVDRP} \leftarrow$ localSearch(S_{EVDRP}, N_{LIST})
    $P_{new} \leftarrow P_{new} \cup S_{EVDRP}, Z_{EVDRP}$
  end
  $P \leftarrow P_{new}$
end
$S^*_{EVDRP}, Z^*_{EVDRP} \leftarrow$ findBestSolution(P)

Notably, the important issues in this step are to decide the value of K and the distance function used. To decide the value of K, we observed that deploying all available vehicles $(K = f)$ tends to reduce delivery time. Thus, we set $(K = f)$ as the starting point and derive V_K as the set of customer nodes divided into K clusters. On the other hand, we deploy $t^t_{i,j}$ as the distance function for this sub-problem, as the aim of this step is to derive a set of EV tours.

Finding the optimal vehicle tour for each cluster

After generating a set of clusters, the second step is to generate a TSP tour for each cluster. This procedure aims to minimize the travel time in each path, as the ultimate goal is to minimize the completion time of EVDRP. Here, we deploy Concorde solver [62] to solve each cluster optimally, resulting in K different TSP solutions S_{TSP}, one for each cluster. Additionally, the complete formulation of the implemented TSP is presented in Appendix B.

Inserting recharging nodes

With K different S_{TSP} on hand, we now insert the recharging node into the tour (if needed) and generate the corresponding electric vehicle traveling salesman problem (ETSP) solution S_{ETSP}. In this regard, a simple constructive algorithm is derived for this purpose, as described in Algorithm 2. For each cluster, this procedure starts by recording the nearest recharging node for each customer node, we let R_i be the nearest recharging node from node i. Then, array $S = [0]$ is produced to contain the generated S_{ETSP}, while we let e_i record the battery state of EV at the ith node in S and dummy set T to simplify the presentation of S_{TSP}. For each step of EV, we evaluate whether the EV could travel from the current node to the next node in T (T_i) and proceed to the nearest recharging node from T_i. If this task could not be completed, the EV should first detour to the nearest recharging station from the current node and then travel to T_i afterwards. In this way, we ensure that the generated solution will never run out of battery.

Inserting drone sorties

The final step of this initialization procedure is to inject drone sorties into the S_{ETSP} of each cluster and produce an ETSPD solution S_{ETSPD} for each cluster. Altogether, these sets of S_{ETSPD} correspond to a complete solution for EVDRP S_{EVDRP}. The aim of inserting drone

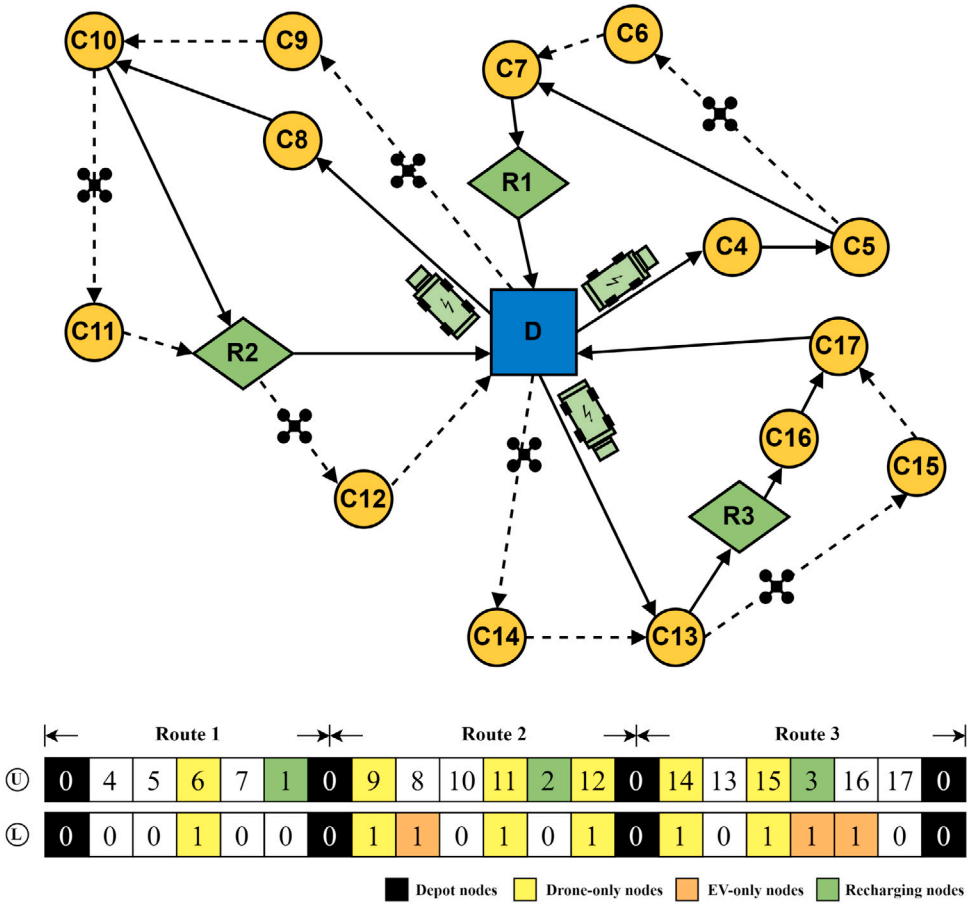

Fig. 1. Illustration of an EVDRP case and the corresponding encoding scheme.

Algorithm 2: Constructive algorithm to insert recharging nodes

Input: $S_{TSP}, d^t_{i,j}, E_t, h_t, R_i$
Result: S_{ETSP}

$T \leftarrow S_{TSP}$
$S \leftarrow [0]$
$e_0 \leftarrow E_t$
$i \leftarrow 1,\ j \leftarrow 1$
while $i \leq |T|$ **do**
  if $e^t_{j-1} - \left(d^t_{(T_{j-1},T_i)} + d^t_{(T_i,R_{T_i})}\right) h_t > 0$ **then**
    $S \leftarrow S \cup T_i$
    $e_j \leftarrow e_{j-1} - d^t_{(T_{j-1},T_i)} h_t$
    $j \leftarrow j + 1$
  else
    $S \leftarrow S \cup R_{T_i}$
    $S \leftarrow S \cup T_i$
    $e_j \leftarrow e_{j-1} - d^t_{(T_{j-1},R_{T_{j-1}})} h_t$
    $e_{j+1} \leftarrow E_t - d^t_{(R_{T_{j-1}},T_i)} h_t$
    $j \leftarrow j + 2$
  end
  $i \leftarrow i + 1$
end
$S_{ETSP} \leftarrow S$

sorties into S_{ETSP} is obvious: to exploit the faster nature of drones and reduce the total completion time of the tour. In this regard, we derive a MILP formulation for drone insertion by adapting a similar MILP formulation presented by Es Yurek and Ozmutlu [63], where the complete presentation of this formulation is available in Appendix C.

Injecting noise

At this point, it is important to note that the presented decomposition procedure leads to a single S_{EVDRP}. On the other hand, MA belongs to the class of population-based algorithms which consists of multiple solution vectors in a single population. In this regard, we inject noise into the other solution vectors with a simple ejection chain move as presented in Fig. 3. In the presented example, we consider a feasible EVDRP solution with three active routes. This move works by selecting one random node for each active route, then, starting from the first route, the selected node is ejected from its current position and is inserted into the next one.

4.2.2. Genetic operators

As an extension of GA, the core structure of MA consists of a set of genetic operators which resemble the traditional structure of GA. In this regard, the genetic operators include the creation of a mating pool for parent selection, a crossover operator to inherit the features of parents, and a mutation operator to introduce random variations during the searching process. In addition, our approach includes an elitism operator to conserve some high-performance solution vectors.

For each iteration of MA, we generate a mating pool MP using a tournament selection technique. The description of this classic technique is fairly straightforward: two solution vectors are randomly selected from the current population set P, then the performance (i.e. objective value or fitness value) of these vectors is compared and the solution vector with better performance is asserted into MP. This 'battle' is repeated until the maximum size of MP is achieved. In this

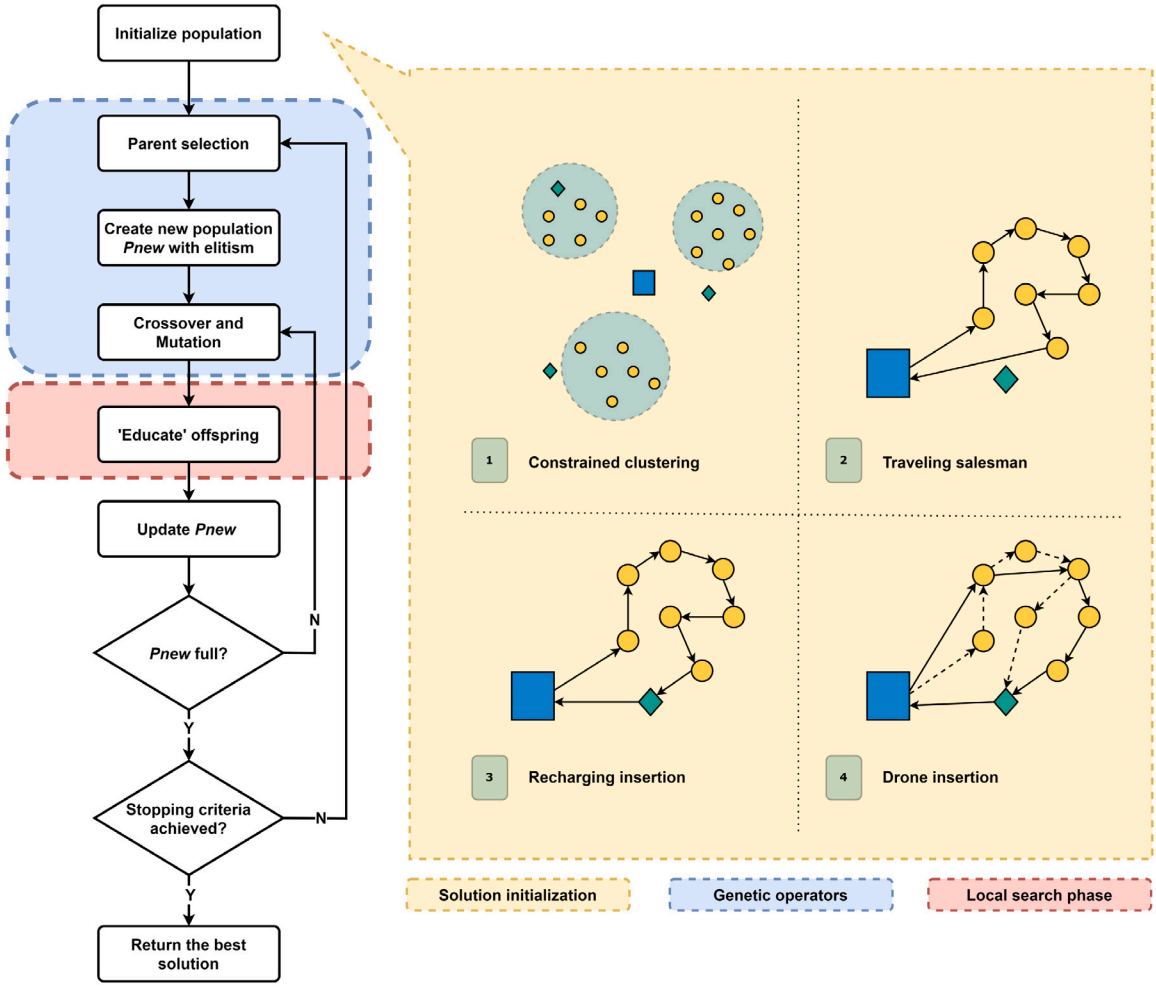

Fig. 2. Flowchart of memetic algorithm for EVDRP.

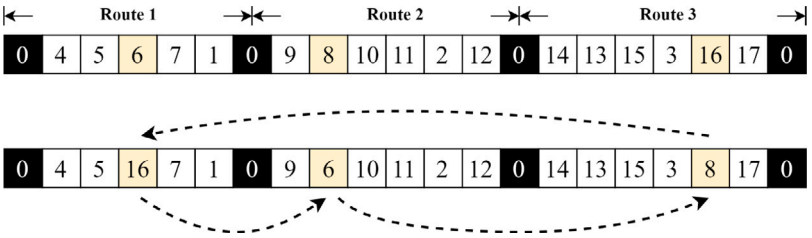

Fig. 3. Ejection chain.

case, the size of MP is limited to $POOL_s \cdot POP_s$. Afterwards, an empty population set P_{new} is created. As mentioned before, a simple elitism procedure is introduced here. In this regard, we simply append a set of solution vectors with the highest performance into the new population P_{new}. The number of these 'elite' vectors is controlled with an elitism rate EL_R.

The next step is to fulfill P_{new} with a set of offspring using a crossover operator. Looking at the literature, it is obvious that various crossover operators have been introduced. This ranges from fairly simple concepts like n-point crossover to more sophisticated operators like order crossover or route-copy crossover [64]. After several pilot tests, we resort to using the 'order crossover' technique. Order crossover is a classic crossover technique that has been used by multiple previous studies. However, due to the complex structure of EVDRP, the implementation of this crossover technique is non-trivial. In this regard, we modified the implementation of order crossover to make it feasible to use it for EVDRP, which is a minor contribution to this study.

Fig. 4 presents our implementation of order crossover. This procedure starts by randomly selecting two parents (A and B) from MP, then we divide these parents into two parts: U and L, following the

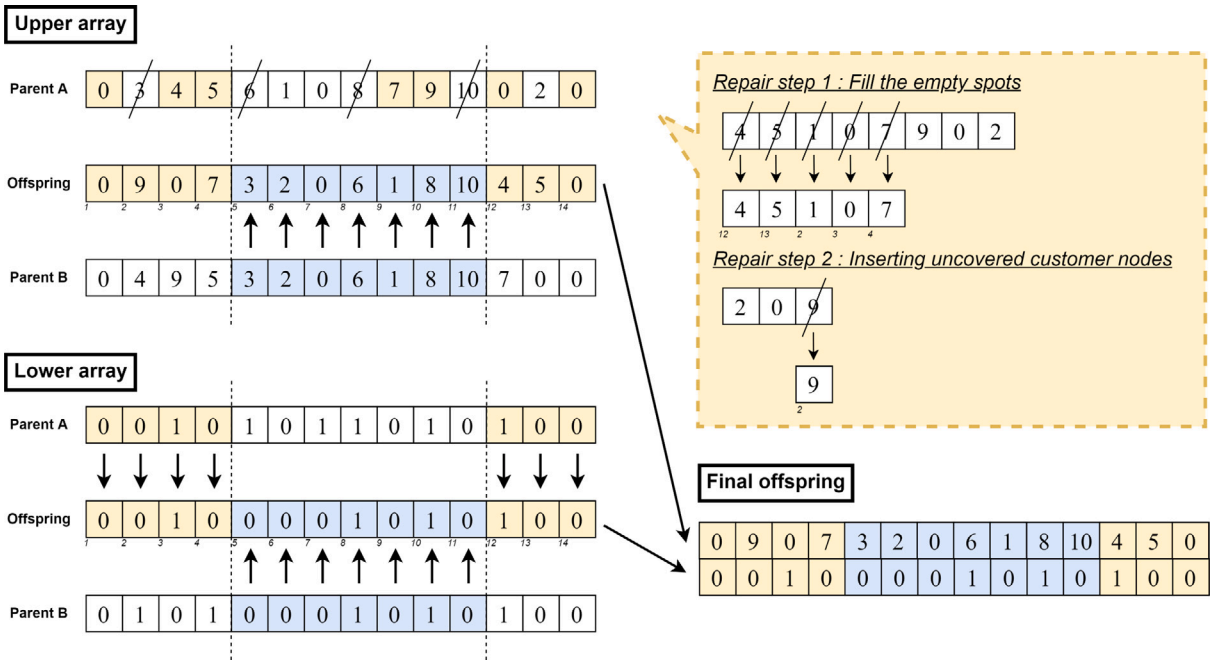

Fig. 4. Order crossover.

encoding scheme used in this study (see Fig. 1) Afterwards, two random split points (sp_1 and sp_2) are selected to decide the parts that would be inherited by the produced offspring from each parent. For the lower array, the process is not complex as it is similar to traditional two-point crossover: Genes that are included between sp_1 and sp_2 inherit the structure of Parent B, while the other genes are generated from Parent A. Meanwhile, the procedure for the upper array is fairly complex due to the requirement to make a feasible encoding scheme (i.e. all customer nodes must be presented without any duplication). Similar to the lower one, the crossover procedure for the upper array starts with inheriting the structure of genes between sp_1 and sp_2 from Parent B, then customer nodes that appeared in this inherited structure are expelled from Parent A. Afterwards, two repair steps are executed to fix the offspring structure (note that the produced offspring could still be infeasible due to the violation of certain constraints, such as capacity constraint). The first step is executed by iteratively filling the empty spots in the offspring with the remaining genes in Parent A. This step starts from $sp_2 + 1$, then encompasses all the remaining genes from there until sp_2, except for the first and last nodes. Then, the second step works by backwardly inserting the remaining customer nodes (if any) to the extra recharging nodes presented in the current state of the offspring. Lastly, the final state of the offspring is created by vertically concatenating the produced upper and lower array.

On the other hand, our selection of mutation operator is quite simple. The proposed MA deploys a swap mutation to provide random components to the offspring generated from the crossover step. This simple technique is executed by selecting two random nodes from the upper array, and then the positions of these nodes in the current tours are swapped.

4.2.3. Local search procedure

The extra step to improve (or 'educate') the solutions generated by genetic operators is the distinct aspect of MA. This improvement is achieved by executing a set of local search moves to improve the quality of the solution vector and increase the exploitation of the search space for each offspring, resembling the process of *memes* receiving an education during their life. Oftentimes, these local search moves are selected or developed based on problem-specific knowledge. This inclusion of problem-specific knowledge in the local search components is noted as the key to a successful implementation of MA [65].

As we are dealing with a problem with a complex structure like EVDRP, our design of MA here employs multiple local search components. In total, there are 11 neighborhood moves included in this algorithm. Seven of them are adapted from Kuo et al. [41] who investigated VRPTWD, these are (i) swap node, (ii) swap whole, (iii) insertion node, (iv) reverse node, (v) reverse whole, (vi) remove sortie node, and (vii) add sortie node. Meanwhile, four of the deployed neighborhood moves are developed in this study to deal with the presence of recharging stations in EVDRP, these are (viii) recharging insertion, (ix) change station, (x) moving station, and (xi) remove station. Fig. 5 visually describes each neighborhood move used in our proposal.

Another issue to be defined is the procedure used to execute these education components. In this regard, we resort to a randomized variable neighborhood descent (VND) as a local search procedure. This selection was motivated by the successful implementation of this procedure in Kuo et al. [41], as well as the simplicity to develop this procedure. Algorithm 3 presents the pseudocode of VND, where it starts with an EVDRP solution S_{EVDRP} and N_{LIST} that encompasses all the neighborhood moves used. Then, the order of neighborhood moves is shuffled, and the S_{EVDRP} is evaluated. Afterwards, the present solution is explored with the current order of N_{LIST} and the solution with the best fitness value is returned back to the main iteration of MA.

4.3. Calculation of objective function

This subsection discusses the calculation of the objective function. In developing a metaheuristic-based solution approach for optimization problems, verifying the procedure to calculate the objective function is crucial in order to ensure that, for the same solution structure, the deployed solution representation returns the same objective value Z, as the MILP. In particular, this step is non-trivial when dealing with a complex optimization problem such as EVDRP, which comprises numerous problem constraints.

As indicated in the proposed MILP formulation, EVDRP deals with four major constraints that need to be managed closely. These are as follows:

(i) endurance constraint of EVs,

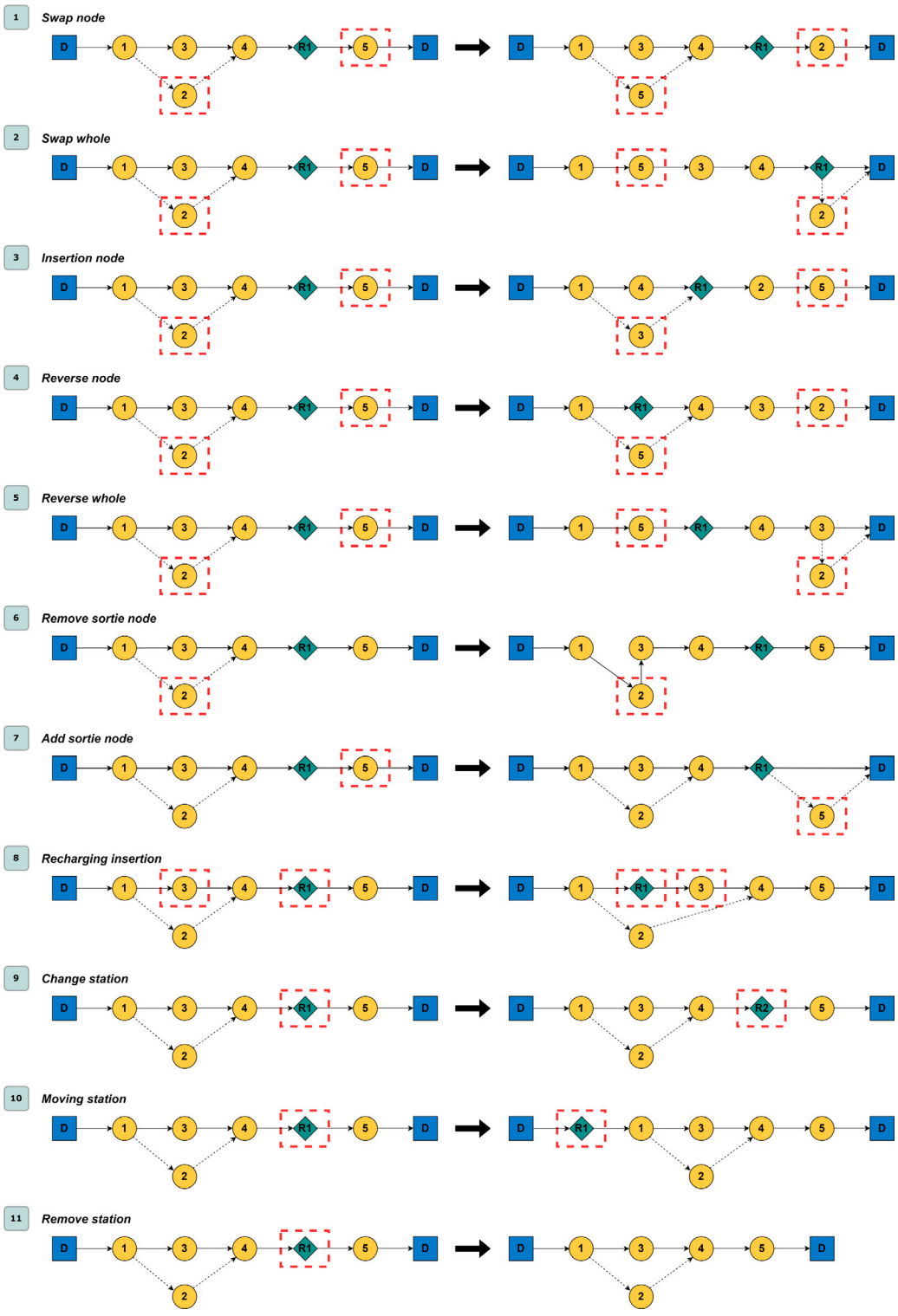

Fig. 5. Illustration of neighborhood moves.

(ii) endurance constraint of drones,
(iii) payload constraint of EVs, and
(iv) payload constraint of drones.

In order to handle these constraints, we deployed the classic technique of penalizing the infeasible solutions that violate constraints, so that we could separate the infeasible search space from the feasible one.

In this regard, the objective value Z is transformed into a penalized objective value as in Eq. (42) by adding penalty values. The penalty values are calculated as in Eqs. (42)–(46), where P_v is a pre-defined penalty value to adjust the scale of violations. In this study, we simply set the value of P_v to a large value (i.e. $max_{\{i,j\in A\}} t^t_{i,j}$) in order to separate the feasible solutions from other potential solutions that violate the problem constraints. Meanwhile, P_{Et}, P_{Ed}, P_{Qt}, and P_{Qd} are respectively defined as a total violation of the endurance constraint of EVs, endurance constraint of drones, payload constraint of EVs, and payload constraint of drones. Furthermore, it should be noted that if the

Algorithm 3: Variable neighborhood descent

Input: S_{EVDRP}, N_{LIST}
Result: S_{EVDRP}, Z_{EVDRP}
$N_{LIST} \leftarrow \text{shuffle}(N_{LIST})$
$Z_{EVDRP} \leftarrow \text{fitnessEvaluation}(S_{EVDRP})$
$k \leftarrow 1$
while $k \leq |N_{LIST}|$ **do**
  $S_{new} \leftarrow \text{perform}(N_{LIST}, k)$
  $Z_{new} \leftarrow \text{fitnessEvaluation}(S_{new})$
  if Z_{new} *is better than* Z_{EVDRP} **then**
    $Z_{EVDRP} \leftarrow Z_{new}$
    $S_{EVDRP} \leftarrow S_{new}$
    $N_{LIST} \leftarrow \text{shuffle}(N_{LIST})$
    $k \leftarrow 1$
  else
    $k \leftarrow k + 1$
  end
end

fitness value is used instead of the objective value Z, the corresponding fitness value can simply be calculated with the formula of $(1/(1+Z))$.

$$Z = \sigma + P_v \cdot (P_{Et} + P_{Ed} + P_{Qt} + P_{Qd}) \tag{42}$$

$$P_{Et} = \sum_{i \in V_f^t} \sum_{f \in F} max\left(0, -a_{i,f}^t\right) \tag{43}$$

$$P_{Ed} = \sum_{i \in V_f^d} \sum_{f \in F} max\left(0, -a_{i,f}^d\right) \tag{44}$$

$$P_{Qt} = \sum_{f \in F} max\left(0, \sum_{i \in V_f^t} q_i - Q_t\right) \tag{45}$$

$$P_{Qd} = \sum_{f \in F} max\left(0, \sum_{i \in V_f^d} q_i - Q_d\right) \tag{46}$$

5. Numerical experiments

This section presents the results of our numerical experiments. The main purpose of this section is to evaluate the performance of the proposed algorithm. In this regard, we compare the performance of the proposed MA with the VNS from Kuo et al. [41] and IVND with tabu list and repair strategy from Wu et al. [39]. VNS and IVND are selected as baseline algorithms, based on their applicability in solving EVDRP, as they were proposed for problems with a related structure. Note that VNS was proposed by Kuo et al. [41] to solve VRPTWD, while IVND was proposed by Wu et al. [39] to solve a collaborative truck–drone routing problem.

This section starts with a presentation of the experiment settings in Section 5.1, which encompass the selection of test instances and problem parameters related to EVDRP. Then, Section 5.2 presents the parameter settings for the algorithms. Afterwards, the complete experiment results are discussed in Section 5.3.

5.1. Experiment settings

In this study, all algorithms were coded in Julia v1.7.2 and implemented on a personal computer with AMD Ryzen 5 1600 Six-Core Processor 3.2 GHz, 16 GB of DDR4 memory, NVIDIA Quadro K-1200 4 GB graphic processing unit and a Windows 11 operating system. The mathematical model is written in Python 3 and was solved using GUROBI 9.5.1 on the same computer with a runtime limit of 3 h (10800 s).

Table 2
Summary of problem parameters for EVDRP.

Parameter	Value	Reference
$d_{i,j}^t$	Calculated by Manhattan distance	Murray and Chu [10]
$d_{i,j}^d$	Calculated by Euclidean distance	Murray and Chu [10]
s_i^t	1 min	Murray and Chu [10]
s_i^d	1 min	Murray and Chu [10]
Q_t	650 kg	Collie [66]
Q_d	2.7 kg	DJI [67]
E_t	33 kWh	Collie [66]
E_d	55 min	DJI [67]
v_t	35 mph	Sacramento et al. [37]
v_d	50 mph	Sacramento et al. [37]
h_t	0.159 kWh/km	Collie [66]
h_d	0.01 kWh/min	DJI [67]
R_t	300 kWh (5 kW/min)	Randall [68]
ρ_L	1 min	Kuo et al. [41]
ρ_R	1 min	Kuo et al. [41]

Table 3
Parameter settings.

Method	Parameter	Name	Value			Reference
MA	C_R	Crossover rate	0.6	0.7	**0.8**	Gonçalves et al. [69]
	M_R	Mutation rate	0.1	0.2	**0.3**	Gonçalves et al. [69]
	POP_S	Population size	**20**	40	60	Sörensen and Sevaux [60]
	EL_R	Elitism rate	0.1	**0.2**	0.3	Gonçalves et al. [69]
	$POOL_S$	Mating pool size	0.3	0.5	**0.7**	Gonçalves et al. [69]
IVND	$TEMP_{start}$	Start temperature	0.005			Wu et al. [39]
	$TEMP_{end}$	End temperature	$1 \cdot 10^{-5}$			
	$COOL_R$	Cooling rate	0.98			
	$TABU_L$	Tabu length	$0.3 \cdot n$			

Since EVDRP is a new problem, to the best of our knowledge a dedicated dataset for this problem is not available at this moment. Therefore, we derived a set of test instances adapted from a related problem for this study. Here, we considered the location demography of the nodes (depot, recharging stations, and customer nodes) from the popular EVRP dataset from Schneider et al. [44]. Then, a total of 128 test instances with up to 100 customer nodes and up to 21 recharging nodes are generated to evaluate the performance of our algorithm. In addition, several problem-specific parameters are also added to the dataset in order to make the EVDRP instances more realistic. These parameters are listed in Table 2.

5.2. Parameter settings

Metaheuristic algorithms are known to be sensitive to the setting of parameters. Therefore, in order to find a suitable configuration for the proposed MA, we resorted to a systematic parameter tuning technique, namely ParamILS (iterated local search for parameter tuning) [70]. ParamILS works by treating parameter tuning tasks as an optimization problem. It starts from a default configuration of parameters for a certain algorithm. Then, the configuration is optimized iteratively with a simple local search algorithm [71].

In order to avoid overfitting and to keep the computational cost inexpensive, ParamILS is executed on a subset of 10% instances from the considered dataset, selected randomly. For each iteration, the parameter setting is perturbed, and then, the performance of the current setting in these selected instances is compared to the performance of the current-best setting. Additionally, as we used a simple random mutation to perturb the parameter setting, our pilot study revealed that the searching process could potentially evaluate a certain setting that had been evaluated before (looping), which wastes computational resources. Thus, in order to avoid the presence of such a loop, we modified ParamILS with a short-term Tabu list memory. Furthermore,

Algorithm 4: ParamILS

Input: $param_{default}$, $T_{list} \leftarrow \emptyset$
Result: $param_{set}$

$param_{set} \leftarrow param_{default}$
$F_{set} \leftarrow Test(param_{set})$
$T_{list} \leftarrow Update(T_{list}, param_{set})$
$param_{new} \leftarrow param_{set}$
while *termination criterion* $= false$ **do**
  while $param_{new}$ *in* T_{list} **do**
    $param_{new} \leftarrow Perturb(param_{new})$
  end
  $F_{new} \leftarrow Test(param_{new})$
  if F_{new} *is better than* F_{set} **then**
    $param_{set} \leftarrow param_{new}$
    $F_{set} \leftarrow F_{new}$
  end
  $T_{list} \leftarrow Update(T_{list}, param_{new})$
end

Algorithm 4 presents the pseudocode of our ParamILS implementation, where we set the size of the tabu list as 20 solution vectors and the termination criterion of ParamILS as 100 iterations by considering the search space size for this parameter tuning task. On the other hand, Table 3 lists the parameters of MA alongside its considered range of values, where the chosen values are indicated in **bold**. In addition, Table 3 also lists the parameter settings of IVND. It is interesting here to report that the design of VNS from Kuo et al. [41] does not involve any parameters.

5.3. Experimental studies

Here, the experiment results are provided and analyzed. The presentation starts with a discussion of the algorithm design process. Afterwards, we discuss the verification process and implementation of our proposed algorithm to solve EVDRP instances with various sizes.

5.3.1. Algorithm design

Metaheuristic-based algorithms can be seen as a collection of operators that work in a systematic way to optimize an optimization problem presented. To illustrate, MA comprises a combination of several genetic operators (i.e. mating selection, crossover, and mutation) with a local search operator or even a combination of several local search operators. Looking back at the literature body of some metaheuristics, it is clear that numerous works have proposed their own design of MA, oftentimes with a proposal of new operators. This phenomenon obviously results in a wide range of options that could be selected to develop an algorithm. However, from our literature review, it is clear that arguments on selecting certain (combinations of) operators within an algorithm are rarely reported. In this regard, although it is certainly impossible to evaluate all available operators in the literature, we argue that a report of systematic design testing is required to validate the algorithm design.

In this study, we implemented a simple 'one-selection-at-a-time' testing process [44] to come up with our final design of MA. We started with designing several potential options for each component. Afterwards, using a 'flavor' obtained from a small-scale pilot study as a base design, we started to test and compare the performance of each option for each component. For each component, we incorporated the option with the best performance, then we moved to test the next component using the updated design. Furthermore, similar to the parameter tuning process, in order to reduce the computational cost and avoid overfitting the design, we performed the design test on a subset of 10% test instances, selected randomly, and for each design option, we executed the algorithm with 10 times repetition. Table 4 lists the considered options for each component of MA, alongside the selected option indicated with **bold**. Then, Fig. 6 reports the design testing results. Note that an option with a relative performance value of '1' indicates superior performance to the other options, where the relative performance value is calculated based on the produced fitness value.

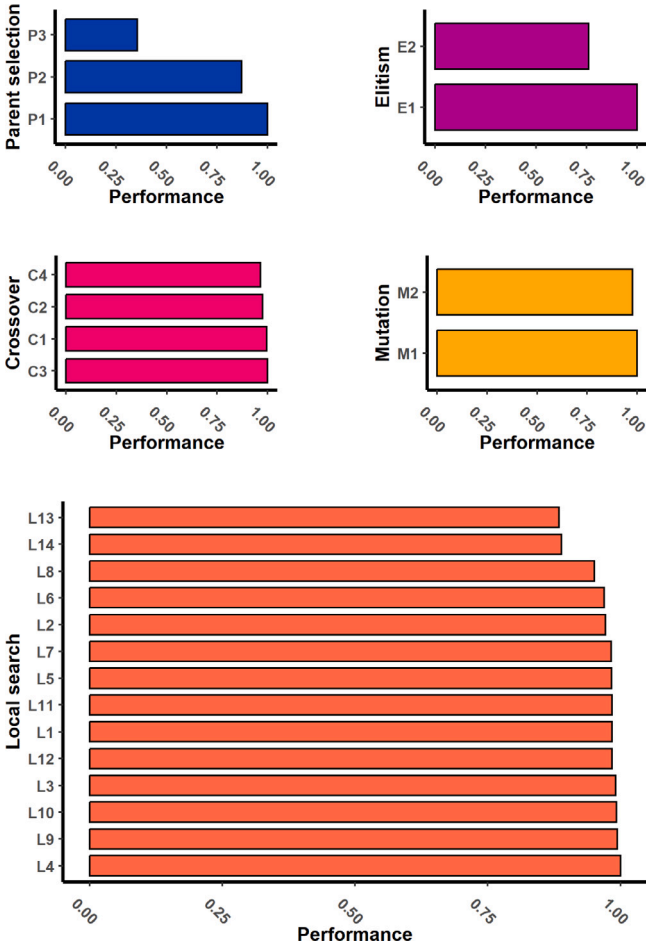

Fig. 6. Summary of design testing results.

5.3.2. Comparative analysis

In order to evaluate the performance of our algorithm, we compare the performance of MA with VNS from Kuo et al. [41] and IVND [39]. At this point, it is worth mentioning that the design of VNS and IVND are fully adopted from Kuo et al. [41], Wu et al. [39]. Then, similar to the proposed MA, both VNS and IVND are injected with four additional neighborhood moves related to recharging stations (recharging insertion, change station, moving station, and remove station) alongside their original neighborhood moves. In addition, all the considered algorithms were tested with the same fitness evaluation budget (300,000 times for small-size instances and 1,000,000 times for large-size instances) obtained from our pilot studies and are executed with 10 repetitions. These aforementioned settings are presented to maintain the fairness of the experiments.

The first step is to verify that the considered algorithms perform correctly. In this regard, the proposed MA can be verified by comparing

Table 4
Design options for proposed MA.

Parent selection	Elitism	Crossover	Mutation
Tournament selection (P1)	**Applied (E1)**	Single point (C1)	**Swap (M1)**
Roulette wheel (P2)	Not applied (E2)	Two point (C2)	Inverse (M2)
Universal sampling (P3)		**Order (C3)**	
		Route copy (C4)	
Local search			
(without) Swap node (L1)		(without) Add sortie node (L8)	
(without) Swap whole (L2)		(without) Recharging insertion (L9)	
(without) Insertion node (L3)		(without) Change station (L10)	
(without) Insertion whole (L4)		(without) Moving station (L11)	
(without) Reverse node (L5)		(without) Remove station (L12)	
(without) Reverse whole (L6)		Use all moves (L13)	
(without) Remove sortie node (L7)		Not applied (L14)	

its performance in solving EVDRP instances to the results from the commercial software, GUROBI. In this study, the comparison was performed in a set of 72 small-size instances with 5–12 customer nodes and 3–7 recharging nodes. Additionally, we also performed verification of the performance of VNS and IVND to solve EVDRP instances.

The complete experiment results for the small-size instances are presented in Appendix D. For each instance, we return the number of customer nodes (n), number of recharging nodes (r), number of available vehicles (f), and the proportion of drone-eligible customer nodes ($\%VD$), and the objective value found by GUROBI (Z^*_{GRB}), alongside its solution gap and running time. Afterwards, we calculate the gap between Z^*_{GRB} with the best solution found by the alternative approaches. In this regard, let Δ^*_A be the gap between Z^*_{GRB} with the best objective values found by an alternative approach 'A' in 10 runs, while $\overline{\Delta}_A$ is the gap between Z^*_{GRB} with the corresponding average objective values of approach 'A'. The calculation of Δ^*_A and $\overline{\Delta}_A$ are presented in Eqs. (47) and (48), where Z^*_A and $\overline{Z}_A$ refer to the best and average objective values found by the alternative approach 'A', while Z^* and $\overline{Z}$ respectively refer to the best and average benchmark values (i.e. GUROBI in this case). In addition, note that positive values on Δ^*_A and $\overline{\Delta}_A$ indicate that the alternative approach 'A' finds a better solution value. In this sense, we could refer to these gap values as the improvement made by approach 'A' with respect to the benchmark value.

$$\Delta^*_A = \frac{Z^* - Z^*_A}{Z^*} \cdot 100\% \tag{47}$$

$$\overline{\Delta}_A = \frac{\overline{Z} - \overline{Z}_A}{\overline{Z}} \cdot 100\% \tag{48}$$

Looking at the values of Δ^* and $\overline{\Delta}$, Appendix D verifies the performance of MA, VNS and IVND to be used as a solver for EVDRP. It is shown that all these approaches can find the comparable quality of solutions to GUROBI for small instances, where overall, the average of Δ^* is 2.36% (1.48% for VNS, −5.10% for IVND) and the average of $\overline{\Delta}$ is 1.08% (0.09% for VNS, −19.93% for IVND). In addition, it is also observed that within 3 h of runtime, GUROBI was not able to find the optimal solutions for some instances with 7 and 8 customer nodes, and all instances with a larger number of customer nodes. In this regard, MA, VNS and IVND were able to find competitive solutions for these instances in a shorter running time.

Moving on, our next task is to validate the performance of our proposed algorithm. With regard to the performance of GUROBI in small-size instances (see Appendix D), we only considered comparing the performance of MA with VNS [41] and IVND [39] for solving the large-size instances of EVDRP. The complete experimental results for large-size instances are presented in Appendix E. Similar to the previous experiment, for each instance, we return the description of the instance (n, r, f and $\%VD$) and the performance comparison between approaches. On this matter, we introduce $Z^*_{BEST} = \min_{o \in O}\{Z^*_o\}$ and $Z^*_{AVG} = \min_{o \in O}\{\overline{Z}_o\}$ as the benchmarks to compare the performance of MA, VNS and IVND, where O refers to methods considered in the experiments. Then, based on Z^*_{BEST} and Z^*_{AVG}, we calculate the solution gaps of MA, VNS and IVND using Eqs. (47) and (48). In this regard, note that a zero value in Δ^*_A ($\overline{\Delta}_A$) means that approach 'A' is able to find the Z^*_{BEST} (Z^*_{AVG}) value.

Overall, it is shown that the proposed MA has superior performance for solving large-scale instances of EVDRP. In terms of running time, it is revealed that with the same fitness evaluation budget, MA and VNS have comparable running times, while IVND requires a longer time to return a solution due to the incorporation of a tabu list and repair strategy. In detail, MA requires a 1.95% lower running time compared to VNS and is 20.18% faster than IVND. This finding is quite surprising since MA is a population-based approach dealing with the evolution of multiple solution vectors, while both VNS and IVND are iterative neighborhood-based approaches with a single solution vector, albeit it must be noted that the structure of IVND involves the usage of a tabu list and a repair strategy that contribute to its computational expenses. This finding, alongside our pilot design testing in Section 5.3.1, indicates that the typical strategy to implement MA with a small population [see 58,60] is a good option for developing an efficient evolutionary algorithm without compromising the quality of solutions. Furthermore, in terms of the quality of solutions, it is observed that MA has a superior performance to both VNS and IVND, as both Δ^*_{MA} and $\overline{\Delta}_{MA}$ return a closer gap to Z^*_{BEST} and Z^*_{AVG} than the corresponding gap values of VNS and IVND. In detail, MA obtained a 1.03% gap to Z^*_{BEST} (5.92% for VNS, 14.38% for IVND) and 0.67% gap to the Z^*_{AVG} (4.68% for VNS, 20.89% for IVND). It is also observed that MA outperformed both VNS and IVND in about 72%–77% of large-size instances, which demonstrates the consistency of this approach in finding high-quality solutions for EVDRP.

Moreover, in order to confirm the results of this experiment, we perform a statistical analysis of the obtained results. The guidelines from Derrac et al. [72] have been closely followed and two sequential statistical tests have been performed using STAC (Statistical Tests for Algorithms Comparison) platform [73] for this purpose. First, Friedman's non-parametric test for multiple comparisons was executed to check if there are any significant differences among algorithms. With a 95% confidence level, it can be concluded from this test that a significant difference among algorithms exists (with a computed p-value of 0.00000) and MA is ranked above both VNS and IVND. Looking at the result of Friendman's test, we perform a post-hoc procedure afterwards with MA as the control algorithm. For this purpose, Hochberg's post-hoc test is executed and the result has confirmed that the performance of MA is significantly better than both VNS (adjusted p-value = 0.00270) and IVND (adjusted p-value = 0.00270). All in all, Table 5 summarizes the statistical testing results and it can be concluded that this experiment has validated the effectiveness of MA as a solver for EVDRP. Related to this conclusion, we demonstrate the applicability of MA for solving EVDRP by visualizing the obtained solutions for several instances in Appendix F.

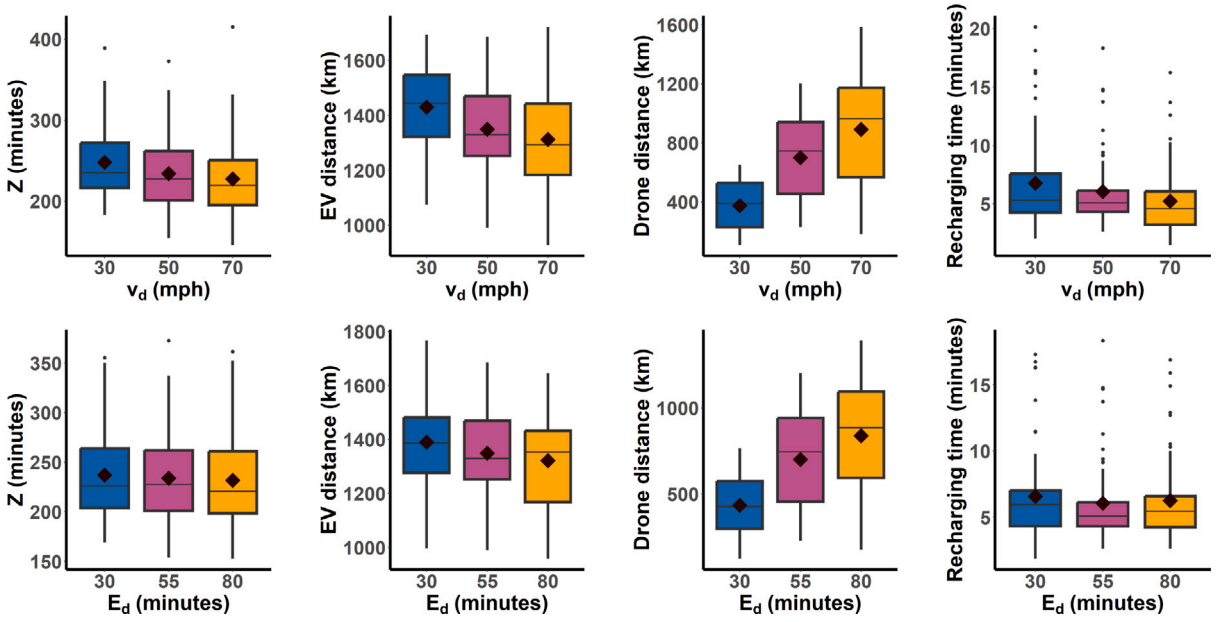

Fig. 7. Impact of drone parameters: (i) speed v_d and (ii) battery endurance E_d.

Table 5
Summary of statistical testing results.

Friedman's non-parametric test	
Algorithm	Ranking
MA	1.3320
VNS	1.7070
IVND	2.9609
p-value = 0.0000	
Hochberg's post-hoc test	
Comparison	Adjusted p-value
MA vs IVND	0.0027
MA vs VNS	0.0027

6. Managerial discussions

In this section, we discuss some interesting insights revealed from the implementation of EVDRP. These insights are derived from extensive experiments using scenario analysis to compare EVDRP with related models (i.e. EVRP and VRPD) and a series of sensitivity analyses to capture the impact of various problem parameters in EVDRP. For this purpose, all additional experiments and comparisons were performed to the large-size instances in order to minimize the impact of problem size on the numerical results. Furthermore, the complete experiment results are presented in Appendix G.

6.1. Impact of the drone technical parameters

The implementation of drones in last-mile logistics is intriguing since drones have distinct advantages to other transportation modes (i.e. faster speed, environmentally friendly), yet also possess obvious limitations (i.e. payload, endurance) [13]. Therefore, we are interested to observe the impact of altering the value of several parameters related to drones' technical aspects on the performance of an EVDRP-based delivery system.

Fig. 7 summarizes the results of sensitivity analysis to drone parameters, in which we executed two sets of sensitivity analysis to v_d and E_d. From Fig. 7, it is noticeable that as the value of v_d increases from 30 mph to 70 mph, the delivery mission tends to be completed faster (reduction of Z by 8.19%). This finding could easily be attributed to the increasing drone usage (travel distance of drones) that rises to 138%, especially when drones are faster than EVs (note that we set $v_t = 35$ mph, see Table 2). In this regard, the optimal solutions for EVDRP tend to deploy drones as frequently as possible in order to reduce the total completion time, which in turn will also reduce the total distance traveled by EVs. Then, similar patterns are also observed in altering the value of E_d. The total completion time tends to be slightly reduced when the battery endurance of drones increases (2.23% reduction from $E_d = 30$ to $E_d = 80$ minutes) and the usage of drones becomes more frequent since the increasing battery endurance also means that drones could travel through more arcs (i, j).

Related to the previous point, a comparable result is also shown when the proportion of drone-eligible nodes $\%VD$ increases. From Fig. 8, it can be seen that the increasing proportion of drone-eligible nodes from 'Low' to 'High' could increase the utilization rate of drones by 38.71% and, in turn, could reduce the total completion time of delivery missions (by 5.54%). Moreover, the increase in the drone utilization rate could also reduce the total times required to recharge EVs, as EVs travel less distance in the presence of drones. In addition, these patterns observed here are in-line with previous findings on problems with a related structure, such as traveling repairman problem [74] and VRPD [see 37,41]. This indicates that the general characteristics of drone implementation remain the same as in EV-based logistics systems. Nevertheless, the attractiveness of this option obviously needs to be assessed further, especially from a financial point-of-view since acquiring drones provides an extra burden to firms' expense [24].

6.2. Impact of EV technical parameters

Afterwards, the impact of altering parameters related to EVs' technical aspects is analyzed. In this regard, we focus our analysis on two parameters: R_t and E_t. For clarity, we neglected v_t since altering the speed of EVs is essentially changing the ratio between v_t and v_d, which is the same as our previous analysis in Fig. 7. Whereas, increasing the consumption rate of EVs, h_t, is essentially decreasing the value of E_t.

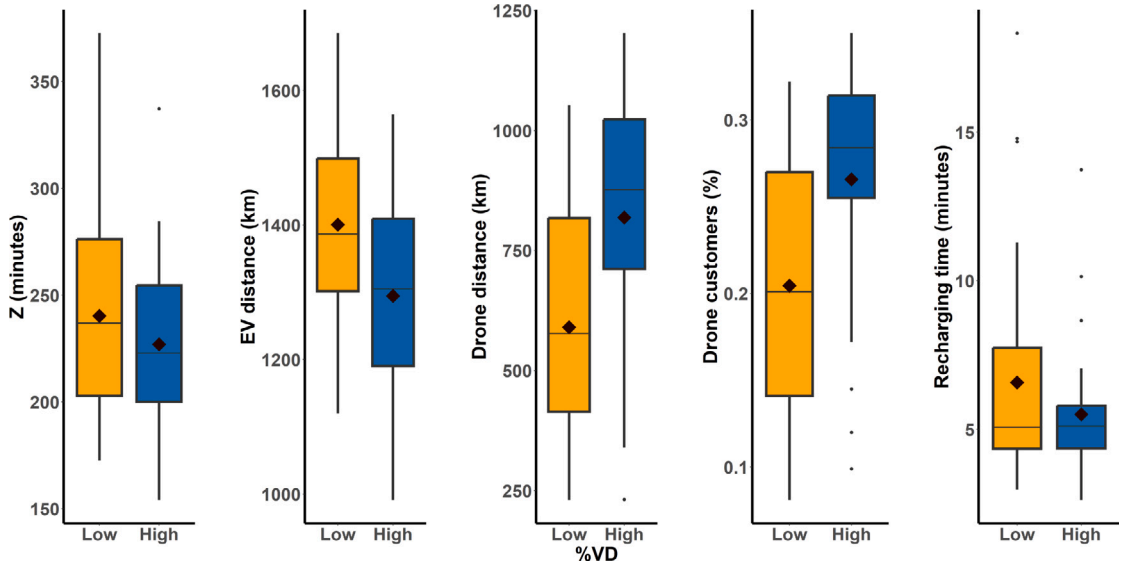

Fig. 8. Impact of the proportion of drone-eligible customers (%*VD*), Low = instances with $\%VD \leq 50\%$ drone-eligible nodes, High = instances with $\%VD > 50\%$.

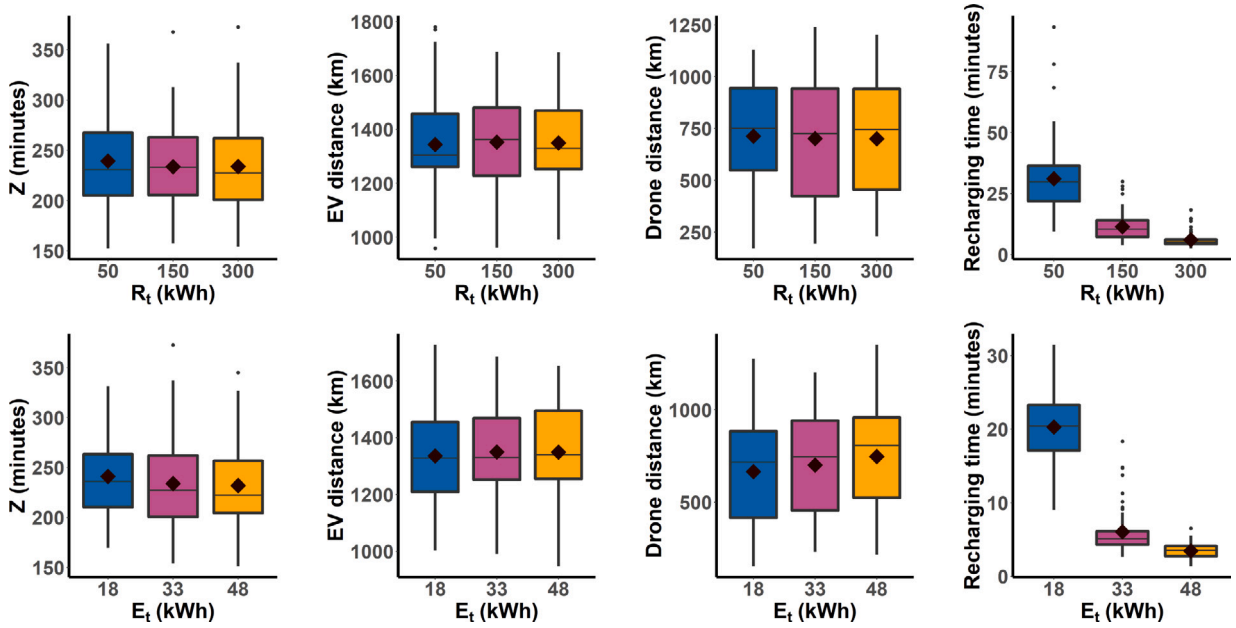

Fig. 9. Impact of EV parameters: (i) recharging rate R_t and (ii) battery endurance E_t.

The summary of results is presented visually in Fig. 9. From Fig. 9, it is noticeable that the most obvious impact of increasing both R_t and E_t is the reduction of total recharging times required. As the driving range (or battery endurance) of EVs increases, EVs require less frequent visits

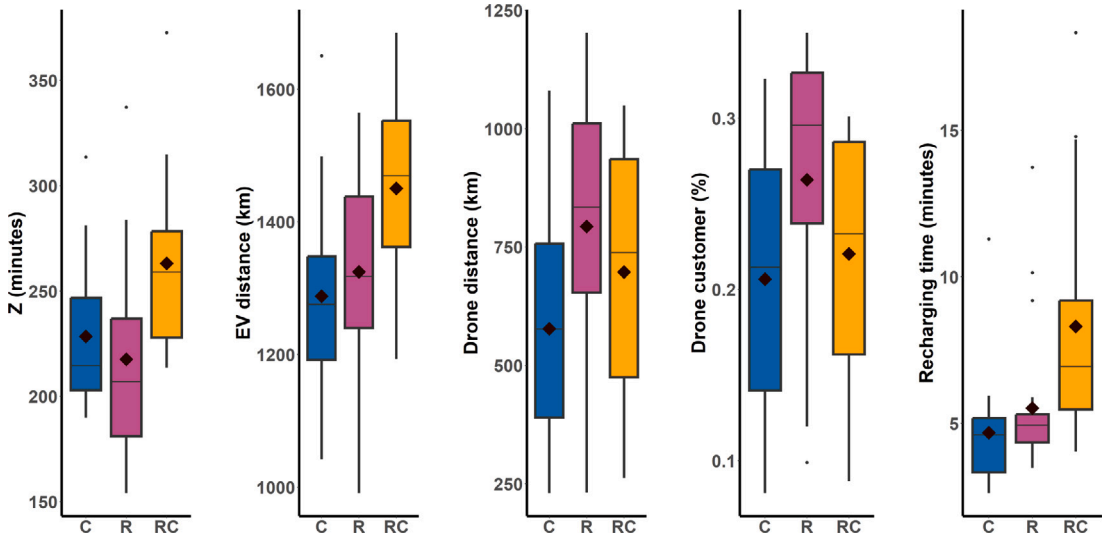

Fig. 10. Impact of customer distribution, C = Clustered, R = Random, RC = Random-clustered.

to the recharging station, and consequently, spend less time recharging their energy. In addition, a reduction in total recharging times also tends to decrease the total completion time of the delivery mission, since the vehicles spend less time detouring and recharging the battery. From these observations, we can conclude that advancements in recharging technology and battery capacity could potentially enhance the performance of delivery systems with EVs. Nevertheless, it must be admitted that the reduction of Z obtained by increasing the value of R_t and E_t in EVDRP is not as large as anticipated (2.28% improvement of Z from $R_t = 50$ to $R_t = 300$ kWh, 3.81% improvement of Z from $E_t = 18$ to $E_t = 48$ kWh. These findings could be attributed to the fact that EVDRP employs not only EVs, but also drones. In this regard, the deployment rate of EVs in the optimal solutions for EVDRP is naturally smaller than in EVRP [37], which seems to minimize the impact of a faster recharging technology and the increased driving range of EVs. Conversely, this finding could also be interpreted that implementing drones could be an interesting option for decision-makers to improve the performance of EV-based logistics systems, *in lieu* of waiting for advancements in recharging and battery technology. For clarity, note that for the baseline case that we set in this study, $R_t = 300$ kWh, can be classified as an ultra-fast recharging technology, which admittedly is expensive to acquire for daily commercial operations [75].

6.3. Impact of customer distribution

Another aspect to analyze in routing problems is the impact of customer spatial distribution on generated solutions. In general, the spatial distribution of customer nodes can be classified into three different classes: clustered, random and random-clustered [76]. Clustered instances constitute problem instances where the location of customer nodes forms several clusters, random instances correspond to problem instances where the customer nodes are located randomly throughout the maps, while a random-clustered instance means that half of the customer nodes are clustered and the other half are randomly located. In the original EVRP dataset of Schneider et al. [44], instances with clustered, random, and random-clustered distributions are respectively indicated with a code 'c', 'r', and 'rc', which we retained for the modified instances used in this study (see Appendix E).

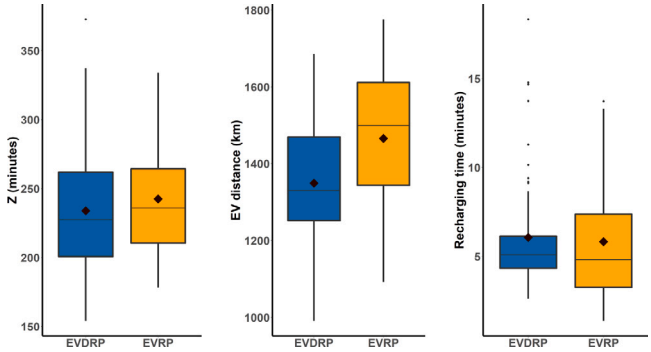

Fig. 11. Summary of comparison between EVDRP and EVRP.

The comparison between those three distributions is presented in Fig. 10. Overall, it is observed that the solutions for clustered instances tend to have lower travel distances for both EVs and drones, which is reasonable since the structure of clustered instances will logically produce several tours where each tour serves a subset of customer nodes located nearby to each other (within the same cluster). Conversely, solutions for randomly distributed instances have lower completion times. This could be explained by considering that customer nodes in random instances tend to be located at a considerable distance from each other, therefore, implementing drones (which have faster speed than EVs) in randomly distributed instances produces a higher time margin than in clustered instances. In fact, we could also observe in Fig. 10 that solutions for random instances tend to deploy drones more frequently.

6.4. Impact of deploying drones into the system

On implementing a last-mile delivery system with a combination of GVs and drones, one particular issue to be answered is obviously

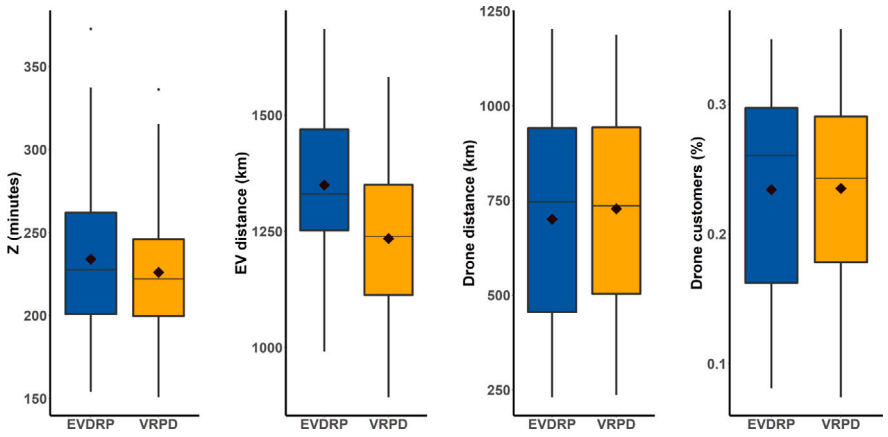

Fig. 12. Summary of comparison between EVDRP and VRPD.

the impact of deploying drones. In this regard, the impact of drones in EVDRP can be assessed by comparing the performance of EVDRP with EVRP [15], which constitutes an EVDRP system with no drone-eligible customer nodes ($\%VD = 0$).

Fig. 11 presents a summary of the comparison between EVDRP and EVRP. First, it is obvious that the implementation of EVDRP systems tends to reduce the distance traveled by EVs (7.92% reduction) and produce better performance in terms of total completion time (3.52% improvement). These findings could easily be attributed to the fact that drones possess faster speeds than EVs and could be exploited by decision-makers to serve a subset of drone-eligible customer nodes. This explanation is supported by another observation on the effect of $\%VD$, where it is noticeable that the margins of Z and EV distance between EVDRP and EVRP enlarges in favor of EVDRP as the proportion of drone-eligible nodes increases. Similarly, an analogous pattern is also presented in the comparison of EV recharging times. In instances with a low proportion of $\%VD$, EVRP might produce solutions with lower recharging times than EVDRP. However, as the proportion of drone-eligible nodes increases, the total recharging times of EVs in EVDRP solutions decrease to a lower level than in EVRP solutions. Therefore, it can be concluded that the presence of drones in EVDRP can improve delivery performance by reducing the total completion time. In addition, the usage of drones might also diminish the system's operational cost by reducing the travel distance of EVs and the required recharging times, which might be in line with the finding of Ha et al. [24], Sacramento et al. [37] in ICEV-based systems. Though, it is obvious that the attractiveness of drones in EV-based logistics systems largely depends on economies of scale (i.e. *how often drones could be used to serve customer nodes?*).

6.5. Impact of deploying EVs into the system

In addition, it is also interesting to compare EVDRP with VRPD [37], which practically shares a similar problem structure due to the simultaneous deployment of EVs and drones. Fig. 12 summarizes the comparison between EVDRP and VRPD. Performance-wise, it is noticeable that EVDRP systems have a longer completion time (3.41% worse than VRPD) and deploy their GVs more frequently than VRPD (8.52% longer distance). These observations could be attributed to the battery limitation of EVs. Since the driving range of EVs is generally shorter than ICEVs (which in the case of VRPD is assumed to be unlimited) [15], EVs are occasionally forced to perform a recharging detour before continuing their tour, which in turn will enlarge its completion time and the length of the tour. Related to these issues, observation of the distance traversed by drones also reveals that drones in EVDRP tend to travel a lesser distance. However, observation on $\%D$ reveals that both EVDRP and VRPD have almost similar percentages of customer nodes served using drones, which indicates that the solution pattern for those models is similar. A possible explanation for this finding is the ability to launch and retrieve drones from recharging stations in EVDRP, which can reduce the distance traveled by drones when the corresponding recharging station is located in an advantageous place.

On one hand, the finding that EVDRP systems have a longer completion time than VRPD admittedly indicates that the usage of EVs tends to reduce the performance level of delivery systems. On the other hand, regarding the distance traversed by EVs, one thing to consider is that EVs generally have lower transportation cost per distance unit, and obviously, lower pollution levels than ICEVs [15]. Therefore, the trade-off between time performance and potential cost and emission reductions would be an important consideration for decision-makers. In this regard, as discussed rigorously by Anosike et al. [77], alongside governmental intervention in regulation, scientific innovations to increase the driving range of EVs will play a key role in increasing the adoption rate of EVs in commercial logistics.

7. Conclusions

In this paper, we discuss the integration between EVs and drones in last-mile logistics. This research is motivated by several previous findings in research streams of both EV-based and drone-aided logistics systems, where it has been shown that both contemporary transportation modes have the potential to improve the current state of a last-mile logistics system. Given the ever-increasing complexity of last-mile logistics networks nowadays, which has long been known as the most inefficient part of logistics networks, it is obvious that a fast, efficient and more environmentally friendly logistics solution is needed.

In the present study, the integration between EVs and drones is modeled as EVDRP with simultaneous cooperation and recharging stations within the graph. The problem is formulated as a MILP. Then, we developed a heuristic algorithm as a solver for large-scale instances

based on MA, whereas within the proposed MA, several problem-specific operators are designed and tested. The performance of MA was then validated through extensive computational comparisons with state-of-the-art solution approaches. Moreover, a set of managerial insights was derived from applying MA for solving EVDRP and several related problems. In this regard, we confirm that implementing drones into EV-based logistics systems retains the benefits and characteristics of implementing drones into ICEV-based logistics systems, as the integration of EVs and drones in EVDRP produces superior performance to logistics systems without drones. Meanwhile, comparing EVDRP with cooperation between traditional ICEVs and drones (VRPD) reveals that EVDRP solutions remain competitive in terms of time performance.

With the idea of combining multiple disruptive technologies, the contributions made by this research represent a forward step in pursuing sustainability in logistics. In this sense, we believe that numerous future works are needed to further assess the feasibility of this idea, such as integrating different operational variants into the EVDRP model, assessing the EVDRP model with different objective functions, and incorporating multiple objective functions simultaneously. Nevertheless, as the first step, we believe that further analysis of EVDRP from a financial point-of-view is needed to assess whether this system is practicable to be adopted for commercial purposes.

CRediT authorship contribution statement

Setyo Tri Windras Mara: Conceptualization, Methodology, Software, Data curation, Investigation, Formal analysis, Validation, Writing – original draft. **Ruhul Sarker:** Conceptualization, Supervision, Writing – review & editing. **Daryl Essam:** Conceptualization, Supervision, Writing – review & editing. **Saber Elsayed:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Appendix A. Decoding the solution scheme

See Supplementary Material associated with this manuscript.

Appendix B. MILP formulation for TSP

See Supplementary Material associated with this manuscript.

Appendix C. MILP formulation for drone insertion

This part defines the MILP formulation for the drone insertion sub-problem. This formulation is implemented to the ETSP solution of each cluster in order to inject drone sorties into S_{ETSP}. As such, the solution from this sub-problem corresponds to a solution for ETSPD.

At first, we label the nodes in S_{ETSP} as $[1,\dots,\ell]$ according to its order within the EV tour, such that when $S_{ETSP} = [0,3,2,1,5,4,7,6,0]$, then $\ell = 9$ and we label each node as $[1,2,3,4,5,6,7,8,9]$ where both '1' and '9' stand for depot node '0'. In accordance, we store these labels into $\tau \subseteq V$, a subset of all customer nodes included in cluster i and depot node ($\tau = V_i^t \cup V_0$). Then, we define:

- $\upsilon_L = \{1,\dots,\ell-1\}$ as a set of launching nodes in τ
- $\upsilon_R = \{2,\dots,\ell\}$ as a set of rendezvous nodes in τ
- $\upsilon_D = (S_{ETSP})_\tau \cup V_D$ as a set of drone-eligible nodes included in τ
- α_τ as a set of (i,j) arcs in τ, where $i<j$
- α_τ^s as a set of feasible (i,j,k) arcs in τ, where $i<j<k$
- $\chi_{i,j}$ as a decision variable to decide if arc (i,j) is traversed by EV or not
- $\mu_{i,j,k}$ as a decision variable to decide if sortie arc (i,j,k) is traversed by drone or not
- W_i as a decision variable to define the waiting time of EV at node i

Using these notations, we can present the MILP of the drone insertion sub-problem as follows.

$$Z_{ETSPD} = \sum_{i,j\in\alpha_\tau} \chi_{i,j} + \sum_{i,j,k\in\alpha_\tau^s} (\rho_L+\rho_R)\mu_{i,j,k} + \sum_{k\in\upsilon_R} W_k \tag{C.1}$$

subject to:

$$\sum_{\substack{i\in[1,\dots,j-1]\\ i<j}} \chi_{i,j} + \sum_{i\in[1,\dots,j-1]} \sum_{\substack{k\in[j+1,\dots,\ell]\\ (i,j,k)\in\alpha_\tau^s}} \mu_{i,j,k} = 1 \qquad \forall\, j\in[2,\dots,\ell-1] \tag{C.2}$$

$$\sum_{\substack{i\in\upsilon_L\\ i<j}} \chi_{i,j} \le 1 \qquad \forall\, j\in\upsilon_R \tag{C.3}$$

$$\sum_{i,j,k\in\alpha_\tau^s} \mu_{i,j,k} \le \sum_{\substack{j\in\tau\\ i<j}} \chi_{i,j} \qquad \forall\, i\in\upsilon_L \tag{C.4}$$

$$\sum_{(i,j,k)\in\alpha_\tau^s} \mu_{i,j,k} \le \sum_{\substack{j\in\tau\\ j<k}} \chi_{j,k} \qquad \forall\, k\in\upsilon_R \tag{C.5}$$

$$(t^d_{\tau_i,\tau_j} + s^d_j + t^d_{\tau_j,\tau_k})\mu_{i,j,k} \le E_d \qquad \forall\,(i,j,k)\in\alpha_\tau^s \tag{C.6}$$

$$\sum_{(p,q,r)\in\alpha_\tau^s} \mu_{p,q,r} \le \sum_{\substack{(i,j,k)\in\alpha_\tau^s\\ i<p,k\le p}} \mu_{i,j,k} \qquad \forall\, p\in\upsilon_L,\ p>1 \tag{C.7}$$

$$\mu_{p,q,r} + \mu_{i,j,k} \le 1 \qquad \forall\,(i,j,k)\in\alpha_\tau^s,\ (p,q,r)\in\alpha_\tau^s,\ i\le p, j\ne q, k>p \tag{C.8}$$

$$\mu_{p,q,r} + \mu_{i,j,k} \le 1 \qquad \forall\,(i,j,k)\in\alpha_\tau^s,\ (p,q,r)\in\alpha_\tau^s,\ i<r, j\ne q, k\ge r \tag{C.9}$$

$$(t^d_{\tau_i,\tau_j} + s^d_j + t^d_{\tau_j,\tau_k})\mu_{i,j,k} + \sum_{p\in[i,\dots,k-1]} \sum_{\substack{q\in[i+1,\dots,k]\\ p<q}} \chi_{p,q} \le W_k \qquad \forall\,(i,j,k)\in\alpha_\tau^s \tag{C.10}$$

$$W_i \ge 0 \qquad \forall\, i\in\upsilon_R \tag{C.11}$$

$$\chi_{i,j}\in\{0,1\} \qquad \forall\,(i,j)\in\alpha_\tau \tag{C.12}$$

$$\mu_{i,j,k}\in\{0,1\} \qquad \forall\,(i,j,k)\in\alpha_\tau^s \tag{C.13}$$

Objective function (C.1) regulates the MILP objective, which is to minimize the total completion time of ETSPD tour within the cluster. Borrowing the finding of Dell'Amico et al. [78], the calculation of total completion time of a truck–drone delivery system can be transformed into the sum of total travel time of truck, the setup times required to launch–retrieve drones, and the waiting time incurred when the truck arrives earlier than its corresponding drone.

This objective function is subjected to a set of constraints. Constraint (C.2) ensures that all customer nodes within the cluster are visited once, either by an EV or a drone. Constraint (C.3) controls the continuity of EV tour arcs. Constraints (C.4) and (C.5) guarantee that drones can only be launched and retrieved from nodes visited by EV. Constraint (C.6) is the endurance constraint for drone sorties. Constraints (C.7), (C.8) and (C.9) are used to regulate the continuity of drone sorties, so that drone cannot be re-launched if it is still performing a sortie. Constraints (C.9) and (C.10) define the value of EV waiting time in rendezvous node i, while Constraints (C.11) and (C.12) define the range value of decision variables.

Appendix D. Experiment results for small-size instances

Name	n	r	f	%VD	GUROBI			MA			VNS [41]			IVND [39]		
					Z^*_{GRB}	Gap (%)	time (s)	Δ^*_{MA}	$\bar{\Delta}_{MA}$	time (s)	Δ^*_{VNS}	$\bar{\Delta}_{VNS}$	time (s)	Δ^*_{IVND}	$\bar{\Delta}_{IVND}$	time (s)
c101C5_S3	5	3	2	25	135.22	0.00	17.70	0.00	0.00	13.15	0.00	0.00	12.12	0.00	−27.65	20.52
c103C5_S2	5	2	2	50	109.52	0.00	8.29	0.00	0.00	13.27	0.00	0.00	11.71	−16.55	−36.21	21.49
c206C5_S4	5	4	2	75	94.48	0.00	479.89	0.00	0.00	12.48	0.00	0.00	11.43	0.00	−28.44	21.19
c208C5_S3	5	3	2	100	91.57	0.00	22.70	0.00	0.00	13.00	0.00	0.00	11.94	−50.11	−70.63	21.14
r104C5_S3	5	3	3	25	52.13	0.00	44.08	0.00	0.00	13.41	0.00	0.00	11.63	−16.10	−30.70	22.47
r105C5_S3	5	3	3	50	75.56	0.00	165.10	0.00	0.00	12.79	0.00	0.00	11.17	0.00	−1.92	21.66
r202C5_S3	5	3	3	75	113.78	0.00	176.26	0.00	0.00	12.75	0.00	0.00	12.31	0.00	−3.80	13.96
r203C5_S3	5	3	3	100	58.39	0.00	118.33	0.00	0.00	13.03	0.00	0.00	12.45	−33.73	−50.05	14.33
rc105C5_S4	5	4	4	25	113.91	0.00	5076.27	0.00	0.00	14.19	0.00	0.00	11.73	0.00	−1.02	15.24
rc108C5_S4	5	4	4	50	160.78	0.00	4874.35	0.00	0.00	14.20	0.00	0.00	12.56	0.00	−0.19	15.28
rc204C5_S4	5	4	4	75	107.52	0.00	2355.18	0.00	0.00	13.41	0.00	0.00	11.90	−1.86	−10.25	14.79
rc208C5_S3	5	3	4	100	136.09	0.00	61.93	0.00	0.00	14.31	0.00	0.00	12.86	−12.52	−23.28	15.29
c101C6_S3	6	3	2	25	172.43	0.00	91.05	0.00	0.00	13.79	0.00	0.00	13.69	0.00	−22.62	15.35
c103C6_S2	6	2	2	50	108.52	0.00	52.83	0.00	0.00	13.09	0.00	0.00	11.78	0.00	−9.34	14.49
c206C6_S4	6	4	2	75	110.52	0.00	3160.98	0.00	0.00	14.78	0.00	0.00	13.20	−4.76	−7.36	15.71
c208C6_S3	6	3	2	100	90.29	0.00	204.91	2.16	2.16	14.47	2.16	2.16	13.22	0.00	−27.15	15.71
r104C6_S3	6	3	3	25	124.56	0.00	542.65	0.00	0.00	14.88	0.00	0.00	13.48	0.00	−3.48	15.62
r105C6_S3	6	3	3	50	96.87	0.00	1682.14	0.00	0.00	13.77	0.00	0.00	12.25	0.00	−10.84	14.95
r202C6_S3	6	3	3	75	87.58	0.00	504.57	0.00	0.00	14.01	0.00	0.00	12.48	0.00	−21.24	15.05
r203C6_S3	6	3	3	100	77.56	0.00	536.14	0.00	0.00	16.11	0.00	0.00	14.07	−13.05	−44.62	16.35
rc105C6_S4	6	4	4	25	96.87	0.00	4143.37	0.00	0.00	15.97	0.00	0.00	13.43	0.00	−9.86	16.68
rc108C6_S4	6	4	4	50	139.48	0.00	3931.86	0.00	0.00	15.38	0.00	0.00	13.99	0.00	−0.14	16.60
rc204C6_S4	6	4	4	75	105.39	0.00	4891.92	0.00	0.00	16.05	0.00	0.00	13.93	0.00	0.00	16.53
rc208C6_S3	6	3	4	100	128.83	6.99	10800.00	5.91	5.91	15.91	5.91	5.91	14.17	0.00	−7.08	16.66
c101C7_S3	7	3	2	25	109.52	0.00	2464.96	0.00	0.00	15.21	0.00	0.00	13.32	0.00	−10.12	15.76
c103C7_S2	7	2	2	50	205.39	0.00	196.94	0.00	0.00	15.22	0.00	−0.05	15.08	−4.70	−8.24	16.55
c206C7_S4	7	4	2	75	138.61	100.00	10800.00	1.31	1.31	16.54	−12.30	−12.34	14.66	−12.37	−39.38	16.49
c208C7_S3	7	3	2	100	93.35	0.00	2715.31	0.45	0.09	16.96	0.45	0.31	14.70	−6.88	−35.33	17.04
r104C7_S3	7	3	3	25	124.56	0.00	6295.27	0.00	0.00	16.97	0.00	0.00	14.61	0.00	−0.08	17.08
r105C7_S3	7	3	3	50	86.96	100.00	10800.00	0.00	−0.03	16.55	0.00	0.00	14.08	−0.30	−9.49	16.37
r202C7_S3	7	3	3	75	73.23	0.00	8875.27	0.00	−0.01	16.67	−0.02	−0.02	14.21	−19.11	−34.18	16.75
r203C7_S3	7	3	3	100	73.30	100.00	10800.00	0.00	0.00	16.38	0.00	0.00	14.22	−12.14	−20.82	16.74
rc105C7_S4	7	4	4	25	151.13	27.87	10800.00	0.00	0.00	16.41	0.00	0.00	15.16	0.00	−8.10	17.65
rc108C7_S4	7	4	4	50	111.78	100.00	10800.00	0.00	0.00	16.35	0.00	0.00	14.74	0.00	−15.54	17.80
rc204C7_S4	7	4	4	75	171.43	100.00	10800.00	0.00	0.00	17.06	0.00	0.00	14.87	0.00	−0.23	18.60
rc208C7_S3	7	3	4	100	123.17	1.70	10800.00	0.07	0.07	17.58	−1.62	−5.75	14.52	−4.59	−7.94	17.94
c101C8_S3	8	3	2	25	143.48	0.00	5175.27	0.00	−1.12	16.76	0.00	−1.12	16.09	−6.87	−24.68	17.63
c103C8_S2	8	2	2	50	130.83	0.00	2188.78	0.00	−0.08	16.66	0.00	−0.08	15.60	−0.76	−28.96	17.61
c206C8_S4	8	4	2	75	154.39	100.00	10800.00	0.00	−0.52	17.96	−1.18	−1.27	16.20	−1.30	−18.75	18.36
c208C8_S3	8	3	2	100	147.57	0.00	5214.69	0.00	−1.27	18.48	0.00	−0.68	16.50	−31.46	−49.05	17.86
r104C8_S3	8	3	3	25	94.74	100.00	10800.00	0.00	0.00	17.81	0.00	0.00	15.77	0.00	−5.00	17.91
r105C8_S3	8	3	3	50	81.83	100.00	10800.00	0.00	0.00	17.97	0.00	0.00	15.85	−6.43	−23.40	17.47
r202C8_S3	8	3	3	75	92.51	100.00	10800.00	−0.01	−0.03	19.65	−0.01	−0.01	16.20	−6.65	−28.57	18.76
r203C8_S3	8	3	3	100	79.70	100.00	10800.00	0.00	0.00	19.30	0.00	−0.68	16.47	−13.37	−40.99	19.50
rc105C8_S4	8	4	4	25	128.83	100.00	10800.00	0.00	0.00	19.23	0.00	0.00	16.44	0.00	−19.67	20.78
rc108C8_S4	8	4	4	50	150.13	100.00	10800.00	0.00	0.00	18.62	0.00	0.00	16.55	0.00	−12.61	19.42
rc204C8_S4	8	4	4	75	96.87	100.00	10800.00	0.00	0.00	19.82	0.00	0.00	15.98	−4.05	−7.81	19.36
rc208C8_S3	8	3	4	100	103.72	100.00	10800.00	0.09	0.09	19.89	−5.05	−14.26	16.67	−5.60	−23.86	19.48
c101C10_S5	10	5	2	25	163.21	100.00	10800.00	−0.35	−0.89	19.97	0.00	−3.26	19.72	−6.65	−31.77	20.19
c104C10_S4	10	4	2	50	221.30	100.00	10800.00	27.01	27.01	19.02	27.01	27.01	17.70	27.01	8.83	19.89
c202C10_S5	10	5	2	75	195.74	100.00	10800.00	0.00	0.00	19.15	0.51	0.05	18.01	0.00	−15.12	20.04
c205C10_S3	10	3	2	100	149.87	100.00	10800.00	0.00	−6.91	20.27	−7.44	−10.20	19.94	−14.46	−28.39	21.28
r102C10_S4	10	4	3	25	122.30	100.00	10800.00	0.00	0.00	19.17	0.00	0.00	17.44	0.00	−9.71	20.20
r103C10_S3	10	3	3	50	148.00	100.00	10800.00	0.00	0.00	20.76	0.00	0.00	18.55	−1.35	−13.61	20.89
r201C10_S4	10	4	3	75	100.67	100.00	10800.00	0.00	−3.44	20.87	−3.60	−5.20	18.70	−4.44	−44.82	21.12
r203C10_S5	10	5	3	100	88.22	100.00	10800.00	0.00	0.00	21.03	0.00	0.00	19.70	−12.68	−29.40	21.24
rc102C10_S4	10	4	4	25	119.17	100.00	10800.00	−1.68	−2.35	20.59	0.00	−2.60	19.04	−3.58	−16.52	21.60
rc108C10_S4	10	4	4	50	149.64	100.00	10800.00	13.24	3.75	20.39	−0.32	−0.32	19.34	−0.32	−10.83	21.44
rc201C10_S4	10	4	4	75	110.29	100.00	10800.00	0.00	−5.04	21.31	0.00	−2.32	20.04	−16.57	−44.53	21.94

rc205C10_S4	10	4	4	100	137.22	100.00	10800.00	0.00	0.00	21.32	0.00	0.00	19.35	0.00	−20.23	21.76
c103C15_S5	15	5	2	25	275.70	100.00	10800.00	23.33	17.13	24.24	23.69	20.57	23.62	11.41	−4.64	25.88
c106C15_S3	15	3	2	50	186.42	100.00	10800.00	8.93	6.11	26.15	7.45	−1.22	25.99	0.71	−25.16	27.13
c202C15_S5	15	5	2	75	201.74	100.00	10800.00	6.62	2.85	25.77	6.62	1.37	26.04	−5.92	−34.72	26.66
c208C15_S4	15	4	2	100	147.91	100.00	10800.00	4.95	1.94	27.07	5.62	1.64	26.35	−21.05	−36.65	27.44
r102C15_S8	15	8	3	25	153.00	100.00	10800.00	0.68	−4.44	25.59	−0.65	−13.24	23.35	−6.31	−17.38	26.94
r105C15_S6	15	6	3	50	114.65	100.00	10800.00	0.87	−0.23	26.54	1.02	−6.64	25.09	−18.98	−32.27	27.85
r202C15_S6	15	6	3	75	139.52	100.00	10800.00	16.84	9.79	26.90	16.84	7.89	24.93	−6.79	−23.34	28.74
r209C15_S5	15	5	3	100	92.22	100.00	10800.00	13.86	11.00	27.25	14.38	10.17	25.56	0.00	−12.55	28.11
rc103C15_S5	15	5	4	25	167.04	100.00	10800.00	5.78	0.86	26.69	5.18	0.74	24.90	−0.60	−15.05	27.95
rc108C15_S5	15	5	4	50	177.69	100.00	10800.00	15.66	6.89	27.89	4.87	4.06	26.88	2.40	−22.15	28.67
rc202C15_S5	15	5	4	75	171.43	100.00	10800.00	16.52	2.12	27.15	0.00	0.00	26.48	0.00	−8.99	27.99
rc204C15_S7	15	7	4	100	123.66	100.00	10800.00	7.46	5.14	28.10	17.35	5.66	27.44	−4.17	−25.29	28.81
Average								**2.36**	**1.08**	**18.35**	**1.48**	**0.09**	**16.75**	**−5.10**	**−19.93**	**19.72**

Appendix E. Experiment results for large-size instances

Name	n	r	f	%VD	Z^*_{BEST}	Z^*_{AVG}	MA			VNS [41]			IVND [39]		
							Δ^*_{MA}	$\overline{\Delta}_{MA}$	time (s)	Δ^*_{VNS}	$\overline{\Delta}_{VNS}$	time (s)	Δ^*_{IVND}	$\overline{\Delta}_{IVND}$	time (s)
c101_21	100	21	5	25	248.17	311.62	0.00	−0.65	660.14	−10.35	0.00	666.51	−33.86	−16.07	747.48
c102_21	100	21	5	50	244.30	281.23	0.00	0.00	660.73	−1.17	−4.28	676.75	−3.57	−17.61	759.98
c103_21	100	21	5	75	227.13	255.20	0.00	−2.50	682.83	−2.87	0.00	685.49	−9.49	−16.62	780.61
c104_21	100	21	5	100	188.04	256.05	0.00	0.00	690.85	−23.05	−0.08	673.39	−23.20	−11.05	787.89
c105_21	100	21	6	25	211.13	237.00	0.00	0.00	648.56	−9.10	−11.61	643.52	−8.48	−16.94	757.84
c106_21	100	21	6	50	203.43	237.67	0.00	0.00	665.72	−10.24	−7.07	664.32	−16.33	−21.30	770.25
c107_21	100	21	6	75	182.92	225.06	0.00	0.00	672.20	−14.80	−2.10	667.63	−25.33	−14.42	776.05
c108_21	100	21	6	100	175.36	205.62	−10.16	0.00	687.83	−10.12	−3.84	676.04	0.00	−33.49	786.17
c109_21	100	21	7	25	183.43	214.44	0.00	0.00	646.34	−7.70	−2.44	644.85	−21.71	−22.90	755.86
c201_21	100	21	7	50	197.82	242.60	−9.76	−1.74	668.99	0.00	0.00	667.35	−13.97	−16.46	773.93
c202_21	100	21	7	75	196.17	214.77	0.00	0.00	688.76	−0.05	−4.74	685.96	−1.33	−24.82	805.03
c203_21	100	21	7	100	172.78	201.68	0.00	0.00	683.36	−8.76	−8.19	686.59	−21.97	−30.44	818.21
c204_21	100	21	8	25	175.35	205.29	0.00	0.00	659.38	−13.69	−8.22	666.63	−9.89	−18.54	772.54
c205_21	100	21	8	50	184.43	197.29	0.00	0.00	666.57	−5.78	−7.63	682.77	−9.92	−19.59	791.20
c206_21	100	21	8	75	177.78	194.03	−0.15	0.00	681.21	0.00	−8.26	689.08	−11.40	−23.92	805.51
c207_21	100	21	8	100	161.66	189.82	−6.72	0.00	699.62	0.00	−2.72	685.01	−19.14	−25.52	818.33
c208_21	100	21	9	25	185.09	202.90	0.00	0.00	655.29	−2.70	−5.73	653.88	−7.48	−16.83	776.47
r101_21	100	21	5	25	247.43	283.85	0.00	0.00	671.66	−6.89	−2.89	681.67	−18.49	−34.00	797.06
r102_21	100	21	5	50	238.55	264.92	−1.28	−5.13	685.64	0.00	0.00	697.15	−9.03	−19.27	805.98
r103_21	100	21	5	75	204.58	233.76	0.00	0.00	693.49	−8.26	−2.83	702.24	−26.38	−41.37	828.33
r104_21	100	21	5	100	279.22	317.63	−2.22	−6.20	631.55	0.00	0.00	671.04	−18.78	−16.41	773.36
r105_21	100	21	6	25	221.80	251.18	−0.08	0.00	657.34	0.00	−3.22	692.70	−0.42	−13.03	802.38
r106_21	100	21	6	50	207.61	237.92	−0.98	−0.81	673.66	0.00	0.00	699.91	−8.25	−19.16	995.73
r107_21	100	21	6	75	196.37	219.75	−2.07	−1.46	677.80	0.00	0.00	696.09	−0.50	−17.90	1003.49
r108_21	100	21	6	100	233.06	270.10	−5.27	−4.91	621.51	0.00	0.00	662.11	−10.65	−3.13	928.90
r109_21	100	21	7	25	190.56	229.76	0.00	0.00	650.67	−6.50	−1.56	681.54	−24.28	−18.52	962.34
r110_21	100	21	7	50	191.54	206.97	0.00	0.00	681.19	−1.13	−3.05	701.07	−10.50	−19.78	1002.69
r111_21	100	21	7	75	171.87	196.61	0.00	0.00	681.67	−11.47	−2.70	703.58	−7.76	−13.76	1030.14
r112_21	100	21	7	100	210.61	234.02	0.00	0.00	621.07	−0.14	−2.59	657.66	−1.61	−9.32	962.56
r201_21	100	21	8	25	177.74	196.69	0.00	0.00	674.54	−7.93	−22.31	681.11	−15.24	−18.06	987.81
r202_21	100	21	8	50	169.78	187.81	0.00	0.00	672.62	−9.26	−9.05	693.13	−14.09	−21.39	1003.93
r203_21	100	21	8	75	161.61	181.87	0.00	0.00	684.24	−6.98	−5.56	701.30	−13.90	−24.59	969.90
r204_21	100	21	8	100	191.82	213.26	−1.36	0.00	642.83	0.00	−1.50	668.77	−13.12	−17.13	854.52
r205_21	100	21	9	25	163.26	180.18	0.00	0.00	672.83	−9.21	−14.22	689.80	−2.77	−11.55	889.99
r206_21	100	21	9	50	163.26	174.99	0.00	0.00	683.53	−5.79	−7.53	688.10	−17.63	−29.57	963.88
r207_21	100	21	9	75	154.22	170.00	0.00	0.00	677.43	−2.03	−3.91	694.44	−14.50	−19.44	893.46
r208_21	100	21	9	100	185.56	198.32	0.00	0.00	626.89	−5.03	−7.64	655.71	−2.76	−5.61	829.84
r209_21	100	21	10	25	154.35	177.59	0.00	0.00	668.47	−18.59	−10.97	674.14	−20.62	−14.71	865.95
r210_21	100	21	10	50	156.00	172.53	0.00	0.00	676.04	−8.03	−8.27	696.33	−4.15	−16.09	899.34
r211_21	100	21	10	75	141.65	154.05	0.00	0.00	679.49	−1.39	−9.11	726.66	−21.77	−24.35	923.56
rc101_21	100	21	5	25	307.64	338.26	0.00	−10.19	632.65	−1.38	0.00	674.96	−14.50	−16.29	849.84
rc102_21	100	21	5	50	277.78	314.99	0.00	0.00	653.15	−4.06	−4.04	705.51	−14.70	−15.63	862.30
rc103_21	100	21	5	75	233.13	265.63	0.00	0.00	676.24	−16.84	−14.70	716.25	−49.55	−58.36	886.56
rc104_21	100	21	5	100	233.92	284.65	0.00	0.00	695.46	−5.83	−1.88	705.21	−18.31	−27.17	929.93
rc105_21	100	21	6	25	275.26	305.31	0.00	−2.66	636.12	−0.50	0.00	650.21	−14.82	−10.37	848.94

rc106_21	100	21	6	50	240.82	276.22	0.00	0.00	648.40	−9.40	−4.74	660.91	−25.50	−30.32	857.34
rc107_21	100	21	6	75	228.13	262.96	0.00	0.00	668.86	−7.47	−2.30	685.29	−4.60	−21.30	723.36
rc108_21	100	21	6	100	202.09	252.46	0.00	−0.28	690.79	−14.14	0.00	689.05	−20.40	−35.93	729.59
rc201_21	100	21	7	25	235.09	266.19	0.00	0.00	625.07	−7.95	−4.18	645.71	−17.32	−16.46	698.16
rc202_21	100	21	7	50	231.04	255.18	0.00	0.00	650.53	−8.73	−6.66	654.94	−6.39	−22.16	711.66
rc203_21	100	21	7	75	213.90	231.46	−0.86	0.00	678.18	0.00	−2.10	676.09	−12.22	−21.32	722.43
rc204_21	100	21	7	100	186.33	223.79	−7.30	−1.25	690.20	0.00	0.00	685.20	−23.04	−28.53	731.77
rc205_21	100	21	8	25	192.09	228.15	0.00	0.00	629.03	−16.64	−5.27	649.26	−9.39	−16.08	700.63
rc206_21	100	21	8	50	203.09	226.94	0.00	0.00	633.95	−9.83	−6.35	662.70	−8.82	−19.61	717.40
rc207_21	100	21	8	75	185.34	217.53	−3.50	0.00	677.17	0.00	−8.37	678.28	−21.96	−25.15	746.36
rc208_21	100	21	8	100	184.43	213.61	−6.04	0.00	695.50	0.00	−5.62	674.87	−29.47	−30.66	761.45
Average							**−1.03**	**−0.67**	**666.18**	**−5.92**	**−4.68**	**679.40**	**−14.38**	**−20.89**	**834.57**

Appendix F. Visualization of results

See Supplementary Material associated with this manuscript.

Appendix G. Experiment results for managerial discussions

See Supplementary Material associated with this manuscript.

Appendix H. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.swevo.2023.101295.

References

[1] Y. Deutsch, B. Golany, A parcel locker network as a solution to the logistics last mile problem, Int. J. Prod. Res. 56 (1–2) (2018) 251–261.
[2] K. Jacobs, S. Warner, M. Rietra, L. Mazza, J. Buvat, A. Khadikar, S. Cherian, Y. Khemka, The last-mile delivery challenge, Technical report, 2018.
[3] N. Boysen, S. Fedtke, S. Schwerdfeger, Last-mile delivery concepts: a survey from an operational research perspective, OR Spectrum 43 (1) (2021) 1–58.
[4] Australia Post Group, 2022 Inside Australian online shopping: ecommerce industry report, 2022.
[5] C. Dong, A. Akram, D. Andersson, P.O. Arnäs, G. Stefansson, The impact of emerging and disruptive technologies on freight transportation in the digital era: current state and future trends, Int. J. Logist. Manag. 32 (2) (2021) 386–412.
[6] S. Poikonen, J.F. Campbell, Future directions in drone routing research, Networks 77 (1) (2021) 116–126.
[7] DHL, Unmanned aerial vehicles in logistics: A DHL perspective on implications and use cases for the logistics industry, Technical report, 90, 2014, pp. 418–420.
[8] D. Guggina, We're bringing the convenience of drone delivery to 4 million U.S. households in partnership with droneup, 2022.
[9] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, G. Laporte, Drone-aided routing: A literature review, Transp. Res. C 120 (February) (2020) 102762.
[10] C.C. Murray, A.G. Chu, The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, Transp. Res. C 54 (2015) 86–109.
[11] K. Dorling, J. Heinrichs, G.G. Messier, S. Magierowski, Vehicle routing problems for drone delivery, IEEE Trans. Syst. Man Cybern. 47 (1) (2017) 70–85.
[12] N. Agatz, P. Bouman, M. Schmidt, Optimization approaches for the traveling salesman problem with drone, Transp. Sci. 52 (4) (2018) 965–981.
[13] M. Moshref-Javadi, M. Winkenbach, Applications and Research avenues for drone-based models in logistics: A classification and review, Expert Syst. Appl. 177 (March) (2021) 114854.
[14] Y. Xiao, Y. Zhang, I. Kaku, R. Kang, X. Pan, Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption, Renew. Sustain. Energy Rev. 151 (2021).
[15] I. Kucukoglu, R. Dewil, D. Cattrysse, The electric vehicle routing problem and its variations: A literature review, Comput. Ind. Eng. 161 (July) (2021) 107650.
[16] J.P. Rodrigue, C. Comtois, B. Slack, The Geography of Transport Systems, Routledge, 2016.
[17] National Greenhouse Gas Inventory, Quarterly update of Australia's national greenhouse gas inventory : December 2020, 2020.
[18] A.J. Hawkins, FedEx receives its first electric delivery vans from GM's BrightDrop, 2021.
[19] DHL, DHL powers up its offering for electric vehicle logistics, 2021.
[20] V. Cantillo, J. Amaya, I. Serrano, V. Cantillo-Garcia, J. Galvan, Influencing factors of trucking companies willingness to shift to alternative fuel vehicles, Transp. Res. Part E-Logist. Transp. Rev. 163 (2022).
[21] T. Zhu, S.D. Boyles, A. Unnikrishnan, Electric vehicle traveling salesman problem with drone, 2022.
[22] T. Zhu, S.D. Boyles, A. Unnikrishnan, Electric vehicle traveling salesman problem with drone with partial recharge policy, 2022, pp. 1–34.
[23] N.A. Kyriakakis, T. Stamadianos, M. Marinaki, Y. Marinakis, The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries, Cleaner Logist. Supply Chain 4 (March) (2022) 100041.
[24] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, On the min-cost Traveling Salesman Problem with Drone, Transp. Res. C 86 (2018) 597–621.
[25] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, A hybrid genetic algorithm for the traveling salesman problem with drone, J. Heuristics 26 (2) (2020) 219–247.
[26] M. Marinelli, L. Caggiani, M. Ottomanelli, M. Dell'Orco, En route truck-drone parcel delivery for optimal vehicle routing strategies, IET Intell. Transp. Syst. 12 (4) (2018) 253–261.
[27] A. Jeon, J. Kang, B. Choi, N. Kim, J. Eun, T. Cheong, Unmanned aerial vehicle last-mile delivery considering backhauls, IEEE Access 9 (2021) 85017–85033.
[28] P.L. Gonzalez-R, D. Canca, J.L. Andrade-Pineda, M. Calle, J.M. Leon-Blanco, Truck-drone team logistics: A heuristic approach to multi-drop route planning, Transp. Res. C 114 (2020) 657–680.
[29] S.T.W. Mara, A.P. Rifai, B.M. Sopha, An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops, Expert Syst. Appl. (2022) 117647.
[30] A.T. Phan, T.D. Nguyen, Q.D. Pham, Traveling salesman problem with multiple drones, in: ACM International Conference Proceeding Series, 2018, pp. 46–53.
[31] C.C. Murray, R. Raj, The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones, Transp. Res. C 110 (November 2019) (2020) 368–398.
[32] R. Raj, C. Murray, The multiple flying sidekicks traveling salesman problem with variable drone speeds, Transp. Res. C 120 (March) (2020) 102813.
[33] X. Wang, S. Poikonen, B. Golden, The vehicle routing problem with drones: several worst-case results, Optim. Lett. 11 (4) (2017) 679–697.
[34] S. Poikonen, X. Wang, B. Golden, The vehicle routing problem with drones: Extended models and connections, Networks 70 (1) (2017) 34–43.
[35] D. Schermer, M. Moeini, O. Wendt, A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations, Comput. Oper. Res. 109 (2019) 134–158.
[36] D. Popovic, M. Kovac, N. Bjelic, A miqp model for solving the vehicle routing problem with drones, in: Proceedings of the 4th Logistics International Conference, 2019, pp. 53–62.
[37] D. Sacramento, D. Pisinger, S. Ropke, An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones, Transp. Res. C 102 (2019) 289–315.
[38] Z. Wang, J.B. Sheu, Vehicle routing problem with drones, Transp. Res. B 122 (2019) 350–364.
[39] G. Wu, N. Mao, Q. Luo, B. Xu, J. Shi, P.N. Suganthan, Collaborative truck-drone routing for contactless parcel delivery during the epidemic, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 25077–25091.
[40] L. Di Puglia Pugliese, F. Guerriero, Last-mile deliveries by using drones and classical vehicles, Springer Proc. Math. Stat. 217 (2017) 557–565.
[41] R.J. Kuo, S.H. Lu, P.Y. Lai, S.T.W. Mara, Vehicle routing problem with drones considering time windows, Expert Syst. Appl. 191 (2022).
[42] R. Conrad, The recharging vehicle routing problem, in: Proceedings of the 2011 Industrial Engineering Research Conference, Vol. 8, IISE Norcross, GA, 2011.
[43] S. Erdoğan, E. Miller-Hooks, A green vehicle routing problem, Transp. Res. Part E: Logist. Transp. Rev. 48 (1) (2012) 100–114.
[44] M. Schneider, A. Stenger, D. Goeke, The electric vehicle-routing problem with time windows and recharging stations, Transp. Sci. 48 (4) (2014) 500–520.
[45] M. Keskin, B. Çatay, Partial recharge strategies for the electric vehicle routing problem with time windows, Transp. Res. C 65 (2016) 111–127.
[46] J. Yang, H. Sun, Battery swap station location-routing problem with capacitated electric vehicles, Comput. Oper. Res. 55 (2015) 217–232.
[47] Ç. Koç, O. Jabali, J.E. Mendoza, G. Laporte, The electric vehicle routing problem with shared charging stations, Int. Trans. Oper. Res. 26 (4) (2019) 1211–1243.
[48] M. Schiffer, M. Schneider, G. Walther, G. Laporte, Vehicle routing and location routing with intermediate stops: A review, Transp. Sci. 53 (2) (2019) 319–343.

[49] D. Baek, Y. Chen, N. Chang, E. Macii, M. Poncino, Energy-efficient coordinated electric truck-drone hybrid delivery service planning, in: 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2020, 2020.
[50] R. Cuda, G. Guastaroba, M.G. Speranza, A survey on two-echelon routing problems, Comput. Oper. Res. 55 (2015) 185–199.
[51] D. Schermer, M. Moeini, O. Wendt, Algorithms for Solving the Vehicle Routing Problem with Drones, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10751 LNAI, 2018, pp. 352–361.
[52] M.A. Coindreau, O. Gallay, N. Zufferey, Parcel delivery cost minimization with time window constraints using trucks and drones, Networks 78 (4) (2021) 400–420.
[53] S.H. Huang, Y.H. Huang, C.A. Blazquez, C.Y. Chen, Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm, Adv. Eng. Inform. 51 (2022).
[54] Y.Q. Han, J.Q. Li, Z. Liu, C. Liu, J. Tian, Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones, Int. J. Adv. Robot. Syst. 17 (2) (2020).
[55] J. Euchi, A. Sadok, Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones, Phys. Commun. 44 (2021).
[56] H. Li, J. Chen, F. Wang, M. Bai, Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review, European J. Oper. Res. 294 (3) (2021) 1078–1095.
[57] J. Gomez-Lagos, A. Candia-Vejar, F. Encina, A new truck-drone routing problem for parcel delivery services aided by parking lots, IEEE Access 9 (2021) 11091–11108.
[58] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: A literature review, Swarm Evol. Comput. 2 (2012) 1–14.
[59] I. Karaoglan, F. Altiparmak, A memetic algorithm for the capacitated location-routing problem with mixed backhauls, Comput. Oper. Res. 55 (2015) 200–216.
[60] K. Sörensen, M. Sevaux, MA|PM: Memetic algorithms with population management, Comput. Oper. Res. 33 (5) (2006) 1214–1225.
[61] M.O. Ball, Heuristics based on mathematical programming, Surv. Oper. Res. Manag. Sci. 16 (1) (2011) 21–38.
[62] D.L. Applegate, R.E. Bixby, V. Chvátal, W. Cook, D.G. Espinoza, M. Goycoolea, K. Helsgaun, Certification of an optimal TSP tour through 85,900 cities, Oper. Res. Lett. 37 (1) (2009) 11–15.
[63] E. Es Yurek, H.C. Ozmutlu, A decomposition-based iterative optimization algorithm for traveling salesman problem with drone, Transp. Res. C 91 (July 2017) (2018) 249–262.
[64] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools Appl. 80 (5) (2021) 8091–8126.
[65] P. Moscato, C. Cotta, An accelerated introduction to memetic algorithms, Internat. Ser. Oper. Res. Management Sci. 272 (2019) 275–309.
[66] S. Collie, Renault Kangoo ZE: Electric van priced from $52,527 for private buyers, 2021.
[67] DJI, DJI Matrice 300 RTK, 2022.
[68] C. Randall, Alpitronic's hyperchargers pass plug&charge audit, 2022.
[69] J.F. Gonçalves, M.G. Resende, M.D. Costa, A biased random-key genetic algorithm for the minimization of open stacks problem, Int. Trans. Oper. Res. 23 (1–2) (2016) 25–46.
[70] F. Hutter, H.H. Hoos, T. Stützte, Automatic algorithm configuration based on local search, in: Proceedings of the National Conference on Artificial Intelligence, Vol. 2, 2007, pp. 1152–1157.
[71] E. Montero, M.C. Riff, B. Neveu, A beginner's guide to tuning methods, Appl. Soft Comput. 17 (2014) 39–51.
[72] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[73] I. Rodriguez-Fdez, A. Canosa, M. Mucientes, A. Bugarin, STAC: A web platform for the comparison of algorithms using statistical tests, in: IEEE International Conference on Fuzzy Systems, Vol. 2015-November, IEEE, 2015.
[74] M. Moshref-Javadi, A. Hemmati, M. Winkenbach, A truck and drones model for last-mile delivery: A mathematical model and heuristic approach, Appl. Math. Model. 80 (2020) 290–318.
[75] S. LaMonaca, L. Ryan, The state of play in electric vehicle charging services – A review of infrastructure provision, players, and policies, Renew. Sustain. Energy Rev. 154 (November 2021) (2022) 111733.
[76] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, A. Subramanian, New benchmark instances for the Capacitated Vehicle Routing Problem, European J. Oper. Res. 257 (3) (2017) 845–858.
[77] A. Anosike, H. Loomes, C.K. Udokporo, J.A. Garza-Reyes, Exploring the challenges of electric vehicle adoption in final mile parcel delivery, Int. J. Logist. Res. Appl. (2021) 1–25.
[78] M. Dell'Amico, R. Montemanni, S. Novellani, Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem, Optim. Lett. (2019).