World Electric Vehicle Journal

Article

A GRASP Approach for the Energy-Minimizing Electric Vehicle Routing Problem with Drones

Nikolaos A. Kyriakakis †, Themistoklis Stamadianos †, Magdalene Marinaki *,† and Yannis Marinakis †

School of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece; nkyriakakis@tuc.gr (N.A.K.); tstamadianos@tuc.gr (T.S.); marinakis@ergasya.tuc.gr (Y.M.)
* Correspondence: magda@dssl.tuc.gr
† These authors contributed equally to this work.

Abstract: This study addresses the Electric Vehicle Routing Problem with Drones (EVRPD) by implementing and comparing two variants of the Greedy Randomized Adaptive Search Procedure (GRASP). The primary objective of the EVRPD is to optimize the routing of a combined fleet of ground and aerial vehicles, with the aim of improving delivery efficiency and minimizing energy consumption, which is directly influenced by the weight of the packages. The study assumes a standardized packing system consisting of three weight classes, where deliveries are exclusively performed by drones, while ground vehicles function as mobile depots. The two employed GRASP variants vary in their methods of generating the Restricted Candidate List (RCL), with one utilizing a cardinality-based RCL and the other adopting a value-based RCL. To evaluate their performance, benchmark instances from the existing EVRPD literature are utilized, extensive computational experiments are conducted, and the obtained computational results are compared and discussed. The findings of the research highlight the considerable impact of RCL generation strategies on solution quality. Lastly, the study reports four new best-known values.

Keywords: drones; electric vehicle; unmanned aerial vehicle routing; GRASP

check for updates

Citation: Kyriakakis, N.A.; Stamadianos, T.; Marinaki, M.; Marinakis, Y. A GRASP Approach for the Energy-Minimizing Electric Vehicle Routing Problem with Drones. *World Electr. Veh. J.* **2023**, *14*, 354. https://doi.org/10.3390/wevj14120354

Academic Editors: Liguo Zang and Leilei Zhao

Received: 13 November 2023
Revised: 11 December 2023
Accepted: 17 December 2023
Published: 18 December 2023

1. Introduction

The global environmental crisis that humanity is currently facing, is multifaceted. Greenhouse Gas (GhG) emissions are one of the biggest contributors, and, according to the European Environment Agency, a great portion of them comes from road vehicles, especially trucks. This necessitates the use of alternative, greener means of transportation.

Given the combination of legislation that will soon prohibit the sale of new Internal Combustion Engine (ICE) vehicles and the incentives for alternative vehicles, logistics companies will start transitioning to Electric Vehicles (EVs). Nonetheless, this is not as simple as replacing the old vehicles with new ones. Despite the great progression in the field of EVs regarding energy capacity and recharging speeds, EVs cannot yet directly replace ICE vehicles. This indicates that, in order for EVs to succeed in their mission of greener transportation, they have to be handled differently.

This research explores the combination of two new types of vehicles with electric power sources: light-duty electric vans and Unmanned Aerial Vehicles (UAVs—also referred to as "drones"). Both vehicle types are constrained by the capacity and size of their battery, as well as the load they carry; however, each has its own strengths. Drones are the perfect option for the last-mile delivery of small items, weighing only a few kilograms, avoiding the use of large ground vehicles that contribute to traffic congestion and waste a lot of energy in delivering such small packages. Drones do not contribute to congestion, as they are airborne vehicles and spend only a fraction of the energy a large ground vehicle would. A side benefit is faster delivery times for both types of vehicles, as drones can inherently move faster, and ground vehicles have to make fewer stops. All of the above make them well-suited for city-logistics operations.

The Electric Vehicle Routing Problem with Drones (EVRPD) emerges as a compelling solution in contemporary transportation and logistics. Its potential for application in e-commerce, medical supply chains, urban logistics, sustainable product distribution, parcel delivery services, and more underscores its practical relevance. Its appeal lies in the ability to optimize routes for EVs and drones, aligning with the growing emphasis on sustainability. The reduced operating costs associated with EVs, coupled with their improved energy efficiency, present tangible economic benefits for logistics and transportation companies. The incorporation of drones addresses the critical last-mile delivery challenge, enhancing overall delivery efficiency and timeliness. Beyond the practical advantages, the appeal of EVRPD extends to technological innovation, compliance with environmental regulations, positive corporate image, and its role in urban planning and congestion mitigation. As organizations increasingly prioritize sustainability and seek innovative approaches to optimize supply chains, the integration of EVs and drones through EVRPD emerges as a strategic and attractive solution, promising enhanced operational efficiency and reduced environmental impact.

The EVRPD, as introduced in [1], employs light-duty EVs that transport both the drones and the items to predetermined sites, from where the drones launch to complete the last mile of the delivery. The EV stays and waits for all the drones that launched from it to return, and then proceeds to the next launch site. This operational scheme may expand the operational capabilities of drones, while simultaneously limiting the distances that the EVs have to travel. Ref. [1] aimed to concurrently minimize the energy consumption of both types of vehicles, with the objective function being influenced by the Energy-Minimizing VRP model, introduced in [2].

This paper presents the mathematical formulation for the problem, with discrete weight and quantity consideration, and implements two Greedy Randomized Adaptive Search Procedure (GRASP) algorithms, a value-based variant (GRASP-VL) and a cardinality-based (GRASP-CRD) variant. Each approach uses a different strategy for constructing the Restricted Candidate List (RCL), from which the next node to be visited is chosen. The choice among the elements of the RCL is unbiased for both variants. A local search procedure based on the well-known Variable Neighborhood Descent scheme is utilized to improve the generated solutions. Both algorithms were tested on the benchmark instances from [1], and the results were compared with their results. GRASP-CRD obtained four Best-Known Values (BKVs).

The key research gap this study revolves around is the seamless integration of EVs and drones into a unified optimization framework. The primary challenge is developing algorithms that can provide energy-efficient routing for both EVs and drones, considering their unique characteristics, constraints, and interactions. The challenge extends beyond ground-based EV routing, requiring cohesive solutions that incorporate the challenging nature of drones. Bridging these gaps will pave the way for a comprehensive and integrated solution and create a symbiotic relationship between EVs and drones for efficient and sustainable transportation. To this end, the present study proposes a solution method that improves upon previously best-known solutions in benchmark instances from the literature.

The rest of the paper has the following structure. Section 2 presents a literature review of the state-of-the-art VRP with Drones (VRPD) and the Electric VRP (EVRP). The EVRPD and its mathematical model are described in Section 3. The two proposed GRASP techniques for the EVRPD are described in Section 4. In Section 5, the experimental results are presented and compared. Finally, the conclusions and future research are discussed in Section 6.

2. Literature Review

Two main VRP variants can be recognized as highly related to the EVRPD variant: Electric VRP (EVRP) and VRP with Drones (VRPD). In the following subsections, these two variants are further discussed, along with some notable references to other influential papers.

2.1. Electric Vehicle Routing Problems

The earliest VRP considering Alternative Fuel Vehicles (AFVs) was presented in [3]. The variant they proposed could be used with any type of vehicle that is heavily constrained in the distance it can cover before requiring refuelling. They considered customer locations to be places of refuelling. In a later study, Green VRP (GVRP) was introduced in [4]. In GVRP, refuelling stops are not associated with customer locations, and the goal was to minimize the total traveled distance.

The first EVRP was introduced in [5] and it considered both Time Windows and Recharging Stations. The objective was the minimization of the total traveled distance. EVRPTW has been one of the most popular variants.

Energy consumption and replenishing have been the focus of many researchers on EVRPs. Four variants of the EVRPDTW were created by [6], each with a different charging scenario, which found that allowing multiple and partial recharges (PR) is the best option. Ref. [7] aimed to minimize the cost of recharging. Non-linear charging functions (NLCF) have been proposed by [8], solving the EVRP-NL. An NLCF was also employed in [9,10], which used a concave NLCF aiming to minimize the operational costs. Ref. [11] solved a variant with time-dependent waiting times for charging and highlighted the impact of charging delays and their related costs. Quick-charging technology and its impact have been included in the EVRP presented by [12]. In [13], factors related to charging, including energy costs and the converter efficiency of the EV, were taken into account. The approaches in [9,14] considered charging stations of limited capacity.

The State-of-Charge (SoC) is an important variable in routing problems which consider charging. The effects of charging and discharging have been studied in [15,16]. In [17], the energy of the EVs was calculated via a machine learning algorithm. Ref. [18], in their research, allowed the energy stored on the EVs to be returned to the grid.

EVRP has to account for more parameters than other variants, since the energy consumption and, subsequently, the range of the EVs are heavily affected by the load, speed, temperature, and other parameters. In [19,20], the energy consumption rate was a function of vehicle speed and load.

In [21], the impact of ambient temperature on vehicle operation was taken into consideration. Ref. [22] examined the influence of vehicle load on battery performance, utilizing real-life data. The objective of [23] was to reduce energy consumption rather than total distance traveled in Electric Vehicle Routing Problems (EVRP). The research accounted for factors such as energy, weight, speed, and road friction. In [24], a model for energy consumption was formulated, considering both topography and speed profiles.

There have been some studies where some parameters are not static. Ref. [25] investigated the effect of traffic by including a travel time function to alter the vehicle speed, depending on the time of the day. Ref. [26] explored Dynamic EVRP (DEVRP) and allowed for both public and private charging, with uncertain waiting times for public stations. Ref. [27] introduced a novel Fuzzy EVRPTW (FEVRPTW) with fuzzy service time, energy consumption, and travel times.

As a means of separating long distance hauling from short distance deliveries, two-echelon models were developed. Ref. [28] solved a two-echelon EVRP (2e-EVRP), aiming to keep larger trucks out of city centers. They suggested that a battery capacity below 80 km would make the use of EVs an unviable option, while a battery capacity over 150 km would result in a big diminution of charging detours. Ref. [29] also solved a 2e-EVRP utilizing Battery-Swapping Stations in place of conventional charging stations.

2.2. Routing Problems with Drones

The first routing problem that included drones was presented in [30], referred to as the Traveling Salesman Problem with a Flying Sidekick (FSTSP), meaning only one ground vehicle was used. In this first variant, only one drone was used as well. The Multiple TSP with Drones was presented in [31], with drones having the ability to end their flight at any availablevehicle. In a later study, [32] solved the Multiple FSTSP and concluded that the optimal case is to allow both ground and aerial vehicles to make deliveries. Ref. [33] solved the same variant with variable drone speeds and concluded that the positive effect of the drones is only useful when the distances between customers are significant. Ref. [34] allowed drones to visit multiple customers in each flight. An alternative use for drones was presented in [35], where they were used to move items from the depot to the ground vehicles.

The VRP with Drones (VRPD) was introduced in [36], assuming the same speed for both types of vehicles. In the implementations of [37,38], trucks routed separately from the drone and the combined routes were determined and optimized later. In [39], the correlation between the maximum drone range and the benefits of using drones was analyzed. In the studies in [40,41], the last section of the delivery process involved only drones, with trucks serving solely as a mode of transportation to the launch site. Ref. [42] addressed the two vehicle types separately, but drones had the flexibility to execute multiple deliveries per flight and were free to return to any truck, enabling them to make multiple trips as needed. Ref. [43,44] created two-echelon variants and used the trucks as mobile depots, with the latter allowing drone deliveries directly from the depot. Ref. [45] aimed to minimize the number of drones.

The EVRPD is a recent addition to the VRP literature, introduced in [1]. In the subsequent work in [46], a new solution methodology was developed, providing a new BKV. A related variant to EVRPD was presented in [47], with a different operational approach, along with a memetic algorithm.

Despite the recent inclusion of drones in routing operations, there have been many reviews. In the review in [48], TSP with drones, VRP with drones, and Drone-Routing Problems were presented. Ref. [49] provided insights for real-world applications and highlighted the most important research gaps. Ref. [50], along with their review, suggested a taxonomy for the discussed problems. The most recent addition to the literature of drone integration in logistics operations is presented in [51], including both practical and theoretical dimensions of this issue, with a particular emphasis on the incorporation of autonomous vehicles. Furthermore, the study offers numerous suggestions for future research.

Table 1 offers a closer examination of prevalent routing problems with drones documented in the existing literature, highlighting the distinctive attributes inherent to each problem to better define the positioning of the present problem.

Table 1. Related VRP variants.

Problem	Description
Traveling Salesman Problem with a Flying sidekick (FSTSP)	The FSTSP involves optimizing the route of a traveling salesman who is accompanied by a flying sidekick, also known as the Traveling Salesman Problem with Drones (TSPD). The most common objectives are the minimization of cost and time.
Vehicle Routing Problem with Drones (VRPD)	In VRPD, the objective is to optimize the routes of a fleet of vehicles (e.g., trucks) along with the use of drones. Various extensions of VRP have been adapted for VRPD so far.
Drone Routing Problem (DRP)	The DRP focuses on optimizing the routes of drones for various applications. The primary objective is to find efficient paths for the drones to visit a set of locations, possibly taking into account additional constraints.
Electric Vehicle Routing Problem (EVRP)	EVRP is a variant of VRP that specifically addresses the unique characteristics and constraints associated with EVs. Charging time and energy consumption minimization are among the most common objectives.
Electric Vehicle Routing Problem with Drones (EVRPD)	EVRPD combines EVs and drones in routing applications, aiming to minimize the overall energy consumption and share the travel distance between two means of transportation.

3. The Electric Vehicle Routing Problem with Drones

The purpose of the Electric Vehicle Routing Problem with Drones, is to utilize two different types of EVs in combination, aiming to surpass the limitations they would face operating in solidarity. Given their electric nature, the minimization of the energy consumption is the objective. This may lead to longer travel distances in some cases; however, this is not a concern.

EVRPD, as introduced in [1], is structured as follows. Electric vans may carry up to a certain number of drones and their packages. The electric vans leave the depot loaded with everything and head to a launch site. There, the electric van stops, deploys the drones, and waits for their return. Once all the drones are back, the electric van heads to the next launch site. The electric vans end their trip back at the depot where they started.

3.1. EVRPD Route Example

The EVRPD takes into account both the transported weight and the number of packages. The packages to be delivered are assumed to fit in the special compartments of the drone and may belong to one of the three package classes, as described in Table 2. This is necessary to ensure a standardized method of loading, given the very small size of drones. The electric vans are also constrained by weight and quantity-carrying capabilities, although, given the larger available space, these constraints are sparsely applicable to practical cases. Both types of vehicles have a limited battery capacity that may not be replenished. The demand of each customer is considered to be one package belonging to one of the formed payload weight classes. Split deliveries are not considered.

Figure 1 illustrates a route example to better present the route weight, quantity, and cost calculation formulas.

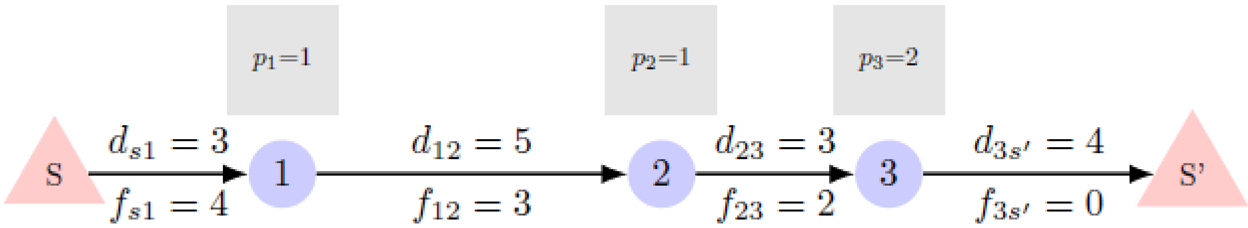

Figure 1. Route example of the EVRPD (d is the arc distance, f is the arc payload weight, p is the package's weight class, s, s' are the starting and ending nodes, respectively and 1, 2, 3 are the intermediate nodes.)

Table 2. Assumed package classes for drones.

Package Class	Weight Range (Weight Units)	Weight Accounted (Weight Units)
PC1	(0.0, 1.0]	1
PC2	(1.0, 2.0]	2
PC3	(2.0, 3.0]	3

All possible drone-loading cases are described in Table 3.

Table 3. List of all possible loading cases, with maximum payload weight of 4 units and maximum quantity of 3 items (PC: Package Class).

Case	Compartment 1	Compartment 2	Compartment 3	Packages (Quantity)	Payload (Weight Units)
1	PC 1	-	-	1	1.0
2	PC 1	PC 1	-	2	2.0

Table 3. *Cont.*

Case	Compartment 1	Compartment 2	Compartment 3	Packages (Quantity)	Payload (Weight Units)
3	PC 1	PC 1	PC 1	3	3.0
4	PC 1	PC 2	-	2	3.0
5	PC 1	PC 1	PC 2	3	4.0
6	PC 2	-	-	1	2.0
7	PC 2	PC 2	-	2	4.0
8	PC 3	-	-	1	3.0
9	PC 3	PC 1	-	2	4.0

In all types of EVs, energy consumption is highly related to the weight of the vehicle and the load it carries [51]. Weight is not the only parameter affecting energy consumption, but it is one of most influential and a parameter which can be managed [52]. Subsequently, energy consumption is one of the main concerns of this research.

The original EVRPD paper aimed to minimize the work needed to serve all the customers. In physics, work is described as the energy needed to move an object with a specified weight across a known distance. Unsurprisingly, greater weights necessitate more energy for transportation over a given distance, and, when an object of a specific weight is carried over longer distances, the overall energy consumption also increases.

$$ArcEnergy = ArcDistance \times (VehicleWeight + ArcPayload) \tag{1}$$

To have a baseline energy consumption of an empty vehicle, the $VehicleWeight$ is set to 1, since all vehicles of the same echelon are of a common type; thus, they have the same weight. The first echelon in the EVRPD comprises all the routing elements related to the EVs, while the second echelon addresses elements related to drones. By incorporating the weight of the EV, the energy function is able to determine the energy spent on arcs on which no payload is carried (i.e., when returning to the depot).

The energy consumption e_{ij} for each arc (i, j) in the EVRPD route depicted in Figure 1 is determined based on the corresponding distance d_{ij} and is calculated in the following way:

$$\begin{aligned}
e_{s1} &= d_{s1} \times (1 + f_{s1}) = 3 \times 5 = 15 \\
e_{12} &= d_{12} \times (1 + f_{12}) = 5 \times 4 = 20 \\
e_{23} &= d_{23} \times (1 + f_{23}) = 3 \times 3 = 9 \\
e_{3s'} &= d_{3s'} \times (1 + f_{3s'}) = 4 \times 1 = 4
\end{aligned}$$

As expected, the sum of the above energy costs is the total route cost and can be calculated as follows:

$$C = \sum_{(i,j) \in route} e_{ij} = 48$$

Even if the distances considered are symmetrical, due to the cumulative nature of the payload weight, the reversal of the customers' order in the route changes the calculated energy cost. For the example given,

$$C_{reversed} = 4 \times 5 + 3 \times 3 + 5 \times 2 + 3 \times 1 = 42$$

Figure 2 illustrates an example of a complete EVRPD solution. Based on the presented energy cost function, the total energy cost of the solution can be calculated as the sum of the energy cost of all routes of all vehicles:

$$C_{Solution} = C_{EV_1^1} + C_{EV_2^1} + C_{D_1^1} + C_{D_2^1} + C_{D_3^1} + C_{D_1^2} + C_{D_2^2}$$

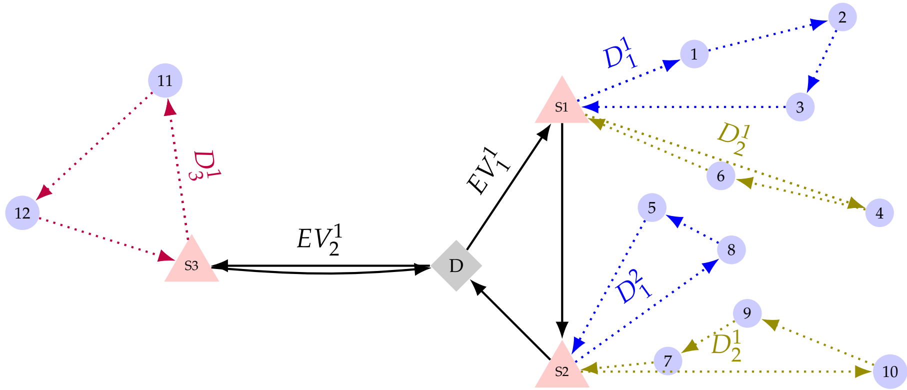

Figure 2. An example of the EVRPD solution.

3.2. Mathematical Formulation of the EVRPD

This subsection outlines the mathematical representation of the EVRPD, following the model presented in [46]. It is structured as a dual-level problem, wherein EVs navigate to their assigned stops, dispatch drones for deliveries, and wait for their return. After the drones finish their deliveries and return, the EVs can then move on to the next designated stop. The EV stops are comparable to the satellite locations present in standard two-echelon VRPs.

As with all VRPs, certain assumptions are made for the EVRPD and are presented below:

- Only drones may visit the customers;
- Each customer requires a single package;
- Each stop may be visited only once by an EV;
- The deployment and retrieval locations are identical;
- EVs remain stationary at the deployment/retrieval locations;
- Given sufficient range, drones may carry multiple deliveries;
- There is no limit to the concurrent number of drones operating from an EV;
- There are sufficient vehicles of both types to meet the demand;
- The handling time of drones is considered to be negligible;
- Ideal environmental conditions are assumed.

Most assumptions are targeted towards minimizing energy consumption, like restricting deliveries exclusively to drones and keeping EVs stationed at specific locations until the drones return.

The provided formulation incorporates ideas from [53], which first conceptualized two-echelon VRPs, [41], which addressed a VRP involving both aerial and ground vehicles, [22,54], where the transported cargo impacts energy requirement, and, finally, [29], which presented the battery-swapping variant of 2e-EVRP.

$V_D = \{v_D\}$ and $V'_D = \{v'_D\}$ denote the sets containing the depot node and its dummy, respectively. Set $V_S = \{v_{S1}, v_{S2}, ..., v_{Sn_s}\}$ is the set of n_s satellites and set $V'_S = \{v'_{S1}, v'_{S2}, ..., v'_{Sn_s}\}$ is its respective dummy. Set $V_C = \{v_{C1}, v_{C2}, ..., v_{Cn_c}\}$ denotes the set containing the n_c customer nodes. Set $A_1 = \{(i,j) | i \in V_D \cup V_S, j \in V'_D \cup V_S, i \neq j\}$ is the set of arcs connecting the elements of the first echelon and $A_2 = \{(i,j) | i \in V_C \cup V_S, j \in V_C \cup V'_S, i \neq j\}$ is the set of arcs for the second echelon's elements.

Let K^{EV} and K^D be the sets of k_{EV} EVs and k_d drones, respectively. Each EV has a payload quantity limit Q^{EV}, a payload weight limit W^{EV}, and an energy capacity limit E^{EV}. Respectively, each drone has similar limits denoted, as Q^D, W^D, and E^D. d_{ij} is the distance between nodes i and j, and p_i denotes the requested payload weight of customer i.

Binary decision variable z_{ijsk} indicates whether drone k travels along arc (i,j), while binary decision variable x_{ijk} signifies whether EV k traverses the same arc originating from s. The payload weight arriving at satellite i from vehicle k is represented by w_{ik}. Payload weights for EVs and drones are denoted as f^1_{ijk} and f^2_{ijsk} respectively, with f^1_{ijk} being the payload weight of EV k from node i to node j, and f^2_{ijsk} representing the payload weight of drone k traveling from node i to node j, originating from satellite s.

Additionally, two variables, TD_{ijs+} and TD_{ijs-}, establish connections between EVs and drones. TD_{ijs+} equals one if EV i is transporting drone j to satellite s, and TD_{ijs-} equals one if EV i is transporting drone j departing from satellite s.

The mathematical model of the EVRPD is expressed by the following equations:

$$\begin{aligned}\min f = &\sum_{(i,j)\in A_1}\sum_{k\in K^{EV}}(d_{ij}\times(1+f^1_{ijk})\times x_{ijk})\\ &+\sum_{(i,j)\in A_2}\sum_{k\in K^D}\sum_{s\in V_S}(d_{ij}\times(1+f^2_{ijsk})\times z_{ijsk})\end{aligned} \tag{2}$$

subject to

$$\sum_{j\in(V'_D\cup V_S)} x_{ijk} = \sum_{j\in(V'_D\cup V_S)} x_{jik}, \forall i\in(V_S\cup V_D), k\in K^{EV} \tag{3}$$

$$\sum_{j\in(V_C\cup V'_S)} z_{ijsk} = \sum_{j\in(V_C\cup V'_S)} z_{jisk}, \forall i\in V_C, s\in V_S, k\in K^D \tag{4}$$

$$\sum_{k\in K^D}\sum_{s\in V_S}\sum_{j\in(V_C\cup V'_S)} z_{ijsk} = 1, \forall i\in V_C \tag{5}$$

$$\sum_{i\in(V_S\cup V_D)} x_{isk} \le 1, \forall s\in V_S, k\in K^{EV} \tag{6}$$

$$\sum_{j\in(V_S\cup V'_D)} x_{v_d jk} = 1, \forall k\in K^{EV} \tag{7}$$

$$\sum_{i\in(V_D\cup V_S)} x_{iv'_d k} = 1, \forall k\in K^{EV} \tag{8}$$

$$w_{ik} = \sum_{j\in(V'_D\cup V_S)} f^1_{jik} - \sum_{j\in(V'_D\cup V_S)} f^1_{ijk}, \forall i\in V_S, k\in K^{EV} \tag{9}$$

$$0\le f^1_{ijk}\le W^{EV}\times x_{ijk}, \forall(i,j)\in A_1, k\in K^{EV} \tag{10}$$

$$\sum_{i\in(V_C\cup V_S)}\sum_{j\in V_C} z_{ijsk}\le Q^D, \forall s\in V_S, k\in K^D \tag{11}$$

$$\sum_{i\in(V_D\cup V_S)}\sum_{j\in V_S} x_{ijk}\le Q^{EV}, \forall k\in K^{EV} \tag{12}$$

$$p_i = \sum_{j\in(V_C\cup V'_S)} f^2_{jisk} - \sum_{j\in(V_C\cup V'_S)} f^2_{ijsk}, \forall i\in V_C, s\in V_S, k\in K^D \tag{13}$$

$$0\le f^2_{ijsk}\le W^D\times z_{ijsk}, \forall(i,j)\in A_2, s\in V_S, k\in K^D \tag{14}$$

$$\sum_{k\in K^D}\sum_{(i,j)\in A_2} p_i\times z_{ijsk}\times TD_{lks+} = w_{sl}, \forall s\in V_S, l\in K^{EV} \tag{15}$$

$$\sum_{i\in V_S}\sum_{j\in V_C}\sum_{k\in K^D} z_{ijsk}\times TD_{lks+}\le k_d, \forall s\in V_S, l\in K^{EV} \tag{16}$$

$$\sum_{i\in(V_D\cup V_S)}\sum_{j\in(V_C\cup V'_S)}(1+f^1_{ijk})\times d_{ij}\times x_{ijk}\le E^{EV}, \forall k\in K^{EV} \tag{17}$$

$$\sum_{(i,j)\in A_2}\sum_{s\in V_S}(1+f^2_{ijsk})\times d_{ij}\times z_{ijsk}\le E^D, \forall k\in K^D \tag{18}$$

$$\sum_{i\in(V_S\cup V'_S)}\sum_{j\in(V_S\cup V'_S)} z_{ijsk} = 0, \forall s\in V_S, \forall k\in K^D \tag{19}$$

$$TD_{ijs+} = TD_{ijs-}, \forall i\in K^{EV}, j\in K^D, s\in V_S \tag{20}$$

$$\sum_{i \in K^{EV}} TD_{ijs+} = 1, \forall j \in K^D, s \in V_S \tag{21}$$

$$TD_{ijs+} = TD_{ijs'+}, \forall i \in K^{EV}, j \in K^D, s \in V_S, s' \in \{V_S | s' \neq s\} \tag{22}$$

$$TD_{ijs+}, TD_{ijs-} \in \{0,1\}, \forall i \in K^{EV}, j \in K^D, s \in V_S \tag{23}$$

$$x_{ijk} \in \{0,1\}, \forall (i,j) \in A_1, k \in K^{EV} \tag{24}$$

$$z_{ijsk} \in \{0,1\}, \forall (i,j) \in A_2, s \in V_S, k \in K^D \tag{25}$$

Constraints (3) and (4) require every node to possess an equal number of incoming and outgoing connections pertaining to trucks and drones, respectively. Constraint (5) guarantees a single customer visit, while Constraint (6) ensures that drone deployment sites are accessed at most once by each EV. Constraints (7) and (8) affirm that every EV initiates its route from the depot and concludes it by returning to the same location.

The calculation of the payload weight transported to each satellite location by EVs is determined by Constraint (9), and (10) sets a limit on the EV payload, ensuring it is within capacity limits when an arc is traversed (or 0 if not). Constraint (11) restricts the payload quantity of each drone at each satellite location to remain within capacity limits. Similarly, Constraint (12) sets the limits for the payload quantity of each EV. Constraints (13) and (14) limit the payload weight carried by drones, while Constraint (15) ensures payload consistency between trucks and drones. The availability of drones for each EV is controlled by Constraint (16). Energy consumption values for trucks and drones are regulated by Constraints (17) and (18), respectively. To avoid any connections between satellite locations and their dummies, Constraint (19) is implemented. Constraints (20)–(22) establish unique pairings between each drone and a truck. Decision variables are bounded by Constraints (23)–(25).

4. The proposed GRASP Approach

4.1. Greedy Randomized Adaptive Search Procedure

GRASP, or the Greedy Randomized Adaptive Search Procedure, is a powerful iterative optimization algorithm that follows a two-phase approach in each iteration: solution construction and local search. The method, introduced in [55], aims to find high-quality solutions to combinatorial optimization problems. In the solution-construction phase, GRASP employs a combination of greedy and random methods to create feasible solutions. This is achieved by iteratively adding nodes to an incomplete solution based on a Restricted Candidate List (RCL). The RCL is determined using a greedy function that identifies the best candidate nodes, and, from these candidates, a node is chosen randomly. This randomized greedy mechanism allows for adaptability, introducing an element of randomness into the construction process.

The adaptability of GRASP is further emphasized by the heuristic method in the construction phase, which updates the benefits of each element during each iteration. This dynamic adjustment accounts for changes from previous nodes, ensuring that the algorithm remains responsive to the evolving state of the solution space. As a result, GRASP is capable of generating diverse solutions at each iteration, contributing to its exploration–exploitation balance. In the local-search phase, the constructed solution undergoes iterative improvement within its neighborhood until a local minimum is reached. The overall process repeats until a termination criterion, such as a maximum number of iterations, is met, producing high-quality solutions for combinatorial optimization problems.

Algorithm 1 provides a closer look into the procedural steps of the GRASP algorithm, showcasing its structured approach to solution construction and local search. The algorithm's adaptability, randomness, and iterative refinement make GRASP a versatile and effective tool for solving a wide range of optimization problems.

Algorithm 1 Overall GRASP Algorithm.

```
Input: instance, parameters
Result: S_best
while Maximum number of iterations are not reached do
    repeat
        L ← InitializeAvailableNodes(instance,parameters);
        S ← {};
        while Not all customer have been visited do
            RCL ← GenerateRCL(L, instance, parameters) ; // According to the variant
            nxt ← randomChoice(RCL);
            S ← addNode(S, nxt);
            L ← UpdateAvailableNodes(L, nxt);
    until Until a feasible solution is generated;
    S_improved ← LocalSearch(S, N = {N_1, N_2, ..N_k_max}, LSiters);
    if Cost(S_improved < Cost(S_best) then
        S_best ← S_improved;
return S_best;
```

After its initial introduction, the GRASP algorithm has led to the creation of other similar methods, such as Path Relinking [56,57] and Expanding Neighborhood Search [58], which have been proposed for the second phase of the algorithm. Parallel multi-threaded implementations have also been proposed, as the GRASP can be considered a parallel multi-start algorithm. The communication needs among threads in GRASP iterations are confined to identifying program termination and collecting the optimal solution discovered across all threads [59,60].

This paper incorporates two variations of the GRASP for the EVRPD, each employing a distinct strategy for populating the RCL. The first variant, GRASP-VL, uses a value-based criterion to populate the RCL based on a percentage a between the best and worst candidate values. The second variant, GRASP-RCD, uses a cardinality-based criterion to populate the RCL, adding the n best candidates to the list. For both GRASP-VL and GRASP-RCD implementations, the choice among the elements of the formed RCL is random, without bias.

4.2. GRASP-VL RCL Construction

In populating the RCL, the GRASP-VL employs a parameter $a \in [0, 1]$ to decide the eligibility of a node for inclusion, considering its distance from the last inserted node (denoted as i). The terms d_{max} and d_{min} represent the maximum and minimum distances among the current node and potential nodes to be visited.

Therefore, a candidate node $l \in L$, where L is the list containing all the nodes not yet visited, is incorporated into the RCL only in the case were the condition in Equation (26) stands true:

$$d_{il} \leq d_{min} + a(d_{max} - d_{min}) \tag{26}$$

The RCL populating procedure for GRASP-VL is presented in Algorithm 2.

4.3. GRASP-CRD RCL Construction

The GRASP-CRD variant uses a parameter n to set a fixed maximum size for the RCL. If the candidate nodes in list L are fewer than the value of n, then the size of RCL will be equal to $|L|$. The RCL is then populated based on the minimum distance between the last inserted nodes and the elements of the candidate nodes. Algorithm 3 presents the RCL population method for the GRASP-CRD.

Algorithm 2 Value-based RCL construction.

Data: d, i, a, L
Result: RCL
$d_{min} \leftarrow \min\{d_{il} | l \in L\}$;
$d_{max} \leftarrow \max\{d_{il} | l \in L\}$;
$RCL \leftarrow \{\}$;
for l *in* L **do**
  if $d_{il} \leq d_{min} + a(d_{max} - d_{min})$ **then**
    $RCL \leftarrow RCL \cup \{l\}$;
return RCL;

Algorithm 3 Cardinality-based RCL construction.

Data: d, i, n, L
Result: RCL
$RCL \leftarrow \{\}$;
for $k \leftarrow 1$ *to* $min(|L|, n)$ **do**
  $d_{min} \leftarrow \min\{d_{il} | l \in L\}$;
  $l_{min} \leftarrow \{l \in L | d_{il} = d_{min}\}$;
  $RCL \leftarrow RCL \cup \{l_{min}\}$;
  $L \leftarrow L - \{l_{min}\}$;
return RCL;

4.4. Local Search

The second stage of the GRASP algorithm employs an intensive local search process based on the Variable Neighborhood Descent (VND) structure. VND is a variant with deterministic behaviour derived from the widely recognized Variable Neighborhood Search framework initially introduced in [61]. This application adopts a Pipe-VND strategy [62], wherein a neighborhood is revisited as long as it enhances the solution. After reaching a local minimum within the current neighborhood, the algorithm moves to the subsequent one. The algorithm stops when there are no more improvements to be made in the last neighborhood.

Let $N = \{N_1, N_2, \ldots N_{k_{max}}\}$ represent an operator set that map a given solution S to a neighborhood structure $N_k(S)$. With $R^S = \{R_1, R_2, \ldots R_m\}$ denoting the set of routes for solution S, Algorithm 4 outlines the local search structure.

The operators used by the local search procedure include both intra-route and inter-route neighborhood structures for both drone and EV routes. The following list presents the operators used:

1. Intra-EV-Intra-Drone-Intra-route Swap 1-1: Swaps positions of two customers belonging to the same drone route;
2. Intra-EV-Intra-Drone-*Inter*-route Exchange 1-1: Exchanges positions of two customers belonging to two different routes of the same drone;
3. Intra-EV-Intra-Drone-*Inter*-route Relocation 1-0: Relocates a customer to another route of the same drone;
4. Intra-EV-*Inter*-Drone-*Inter*-route Exchange 1-1: Exchanges positions of two customers belonging to routes of two different drones;
5. Intra-EV-*Inter*-Drone-*Inter*-route Relocation 1-0: Relocates a customer to a route of a different drone;

6. *Inter*-EV-*Inter*-Drone-*Inter*-route Exchange 1-1: Exchange 1-1: Exchanges positions of two customers belonging to two drones' routes, and those drones belong to different EVs;
7. *Inter*-EV-*Inter*-Drone-*Inter*-route Relocation 1-0: Relocates a customer to a route of a drone which belongs to a different EV;
8. EV-route-Intra-route 2-Opt: Performs the 2-opt operator in the EV's route.

Algorithm 4 Local search.

```
Data: S, N = {N_1, N_2, ..N_{k_max}}, LSiters
Result: S
for iter ← 1 to LSiters do
    for k ← 1 to k_max do
        R_i, R_j ← ChooseTwoRandomRoutes(S, k);
        /* candidate routes depended on current k                */
        improved ← True;
        repeat
            S' ← N_k(S, R_i, R_j);
            if cost(S) < cost(S') then
                S← S';
            else
                improved ← False;
        until improved = False;
return S
```

5. Computational Results

The proposed GRASP algorithms undergo testing on 24 instances, as documented in [1]. The proposed GRASP implementations are written in C++ and compiled using GCC 11.2. The experimentation is conducted on a system equipped with a 2014 Intel® Core i7-4770 CPU (3.40 GHz) and 7.7 GB RAM, running the Fedora Workstation 35 OS. For each algorithm, each instance is solved 15 times during the experiments.

The characteristics of each instance are presented in Table 4.

Table 4. Characteristics of each instance.

Instance	Number of Customers	Satellite Positions	Drones per EV	Number of EVs
EVRPD-n22-k4-s10-14	21	2	3	2
EVRPD-n22-k4-s11-12	21	2	3	2
EVRPD-n22-k4-s12-16	21	2	3	2
EVRPD-n22-k4-s6-17	21	2	3	2
EVRPD-n22-k4-s8-14	21	2	3	2
EVRPD-n22-k4-s9-19	21	2	3	2
EVRPD-n33-k4-s1-9	32	2	3	2
EVRPD-n33-k4-s14-22	32	2	3	2
EVRPD-n33-k4-s2-13	32	2	3	2
EVRPD-n33-k4-s3-17	32	2	3	2
EVRPD-n33-k4-s4-5	32	2	3	2
EVRPD-n33-k4-s7-25	32	2	3	2
EVRPD-n51-k5-s11-19	50	2	3	3
EVRPD-n51-k5-s11-19-27-47	50	4	3	3
EVRPD-n51-k5-s2-17	50	2	3	3
EVRPD-n51-k5-s2-4-17-46	50	4	3	3

Table 4. *Cont.*

Instance	Number of Customers	Satellite Positions	Drones per EV	Number of EVs
EVRPD-n51-k5-s27-47	50	2	3	3
EVRPD-n51-k5-s32-37	50	2	4	3
EVRPD-n51-k5-s4-46	50	2	3	3
EVRPD-n51-k5-s6-12	50	2	3	3
EVRPD-n51-k5-s6-12-32-37	50	4	3	3

5.1. Parameter Sensitivity

Both implemented GRASP variants share the parameters of *Maximum Number of Iterations*, equal to 10,000, and *Number of Local Search Iterations*, equal to 50. Additionally, each has a unique parameter that greatly affects the quality of the obtained solutions. In order to investigate the sensitivity of the algorithms to those parameters, different values are tested.

For the GRASP-VL variant, the parameter a values tested, controlling greediness, are $\{0.2, 0.3, 0.4, 0.5\}$. For the GRASP-CRD, the parameter n values tested, controlling the size of the RCL, are $\{2, 3, 5, 8\}$ (see Figure 3).

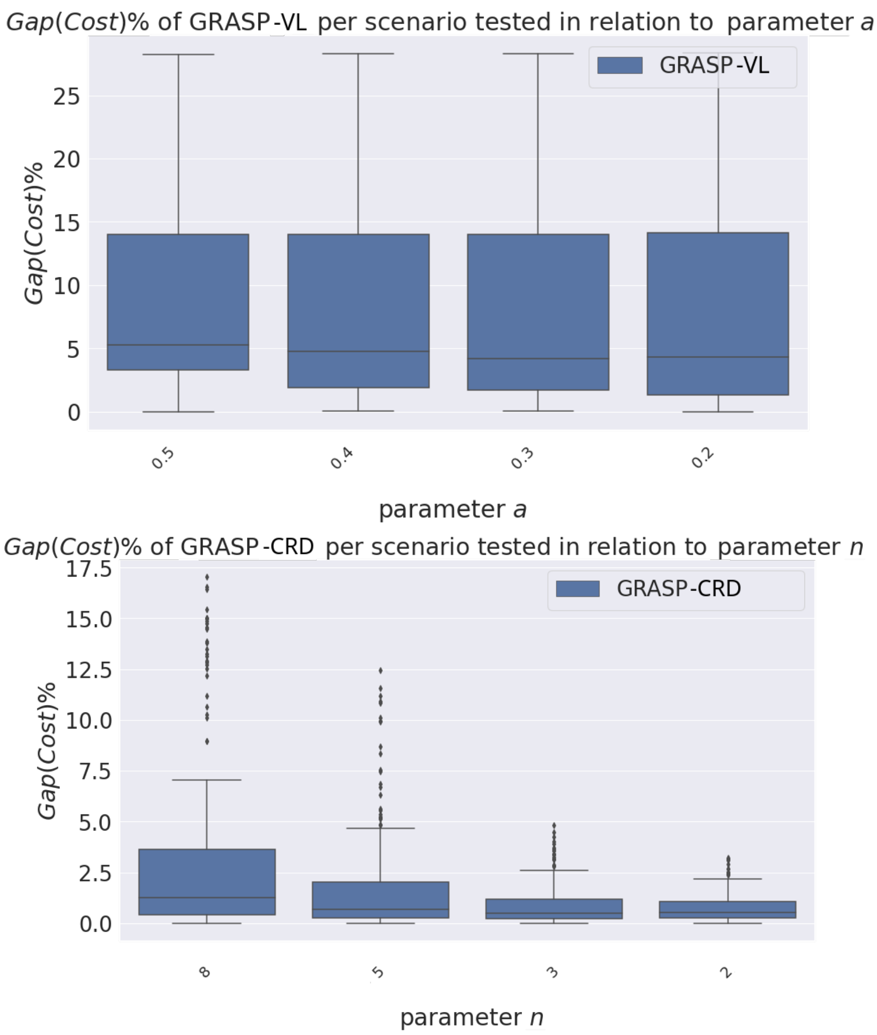

Figure 3. Gap% of solutions to the best solution found for each instance with different parameter settings.

The GRASP-VL algorithm is mostly insensitive to changes in parameter a. Although lower values seem to improve the obtained solutions, the results remain similar despite the vast range of tested values. This may be attributed to the large number of nodes in the RCL at the beginning of the solution construction, due to the clustered nature of the instances, combined with the equal chance of them being selected.

The GRASP-CRD approach utilizes a fixed maximum number of nodes for the RCL at each step using parameter n. The algorithm is highly sensitive to this number, showcasing a clear trend towards smaller values benefiting the overall results the most.

Limiting the choices of nodes at each step is observed to be beneficial, as, due to the weight, quantity, and range constraints, the improvement capabilities of the local search procedure on bad solutions are limited.

5.2. Experimental Results

Table 5 presents the results obtained by each algorithm for the EVRPD instances. Column 1 denotes the instance name, and Column 2 denotes the BKVs found in the literature [1]. Columns 3 and 6 present the best solution values found by each GRASP variant. Columns 4 and 7 present the average solution values obtained in the experiments. Columns 5 and 8 indicate the average computational time required for each run.

The results of the experiments on the EVRPD instances are outlined in Table 5. The first column presents the instance names, while the second column lists the BKVs from the literature [1,46]. For each GRASP variant, the third and sixth columns display the best solutions achieved, while the fourth and seventh columns show the average solution values of the experiments. The fifth and eighth columns provide insights into the average CPU time of each run.

Table 5. Computational results for the EVRPD instances (BKV obtained in bold).

Instance		GRASP-CRD			GRASP-VL		
	BKV	$Cost_{best}$	$Cost_{avg}$	$T_{avg}(s)$	$Cost_{best}$	$Cost_{avg}$	$T_{avg}(s)$
EVRPD-n22-k4-s10-14	1144.28	**1136.29**	1142.48	23.51	1293.90	1295.43	22.10
EVRPD-n22-k4-s11-12	1403.94	1405.41	1410.21	23.14	1463.78	1464.55	21.18
EVRPD-n22-k4-s12-16	1240.95	**1239.74**	1247.42	24.56	1421.66	1423.80	22.34
EVRPD-n22-k4-s6-17	1610.70	**1602.32**	1617.77	23.06	2053.43	2054.62	22.95
EVRPD-n22-k4-s8-14	1191.20	**1189.32**	1192.77	24.19	1293.83	1295.47	23.19
EVRPD-n22-k4-s9-19	1873.95	1874.20	1879.94	22.87	2278.81	2279.25	20.49
EVRPD-n33-k4-s1-9	3599.16	3615.09	3622.40	40.14	3793.41	3798.25	36.91
EVRPD-n33-k4-s14-22	4033.19	4038.24	4049.83	40.44	4038.50	4046.52	37.28
EVRPD-n33-k4-s2-13	3428.85	3434.82	3442.55	38.95	3461.27	3505.41	34.31
EVRPD-n33-k4-s3-17	3307.26	3331.42	3375.77	40.01	3328.80	3366.99	36.02
EVRPD-n33-k4-s4-5	3795.61	3810.39	3878.80	42.30	4591.91	4613.57	38.16
EVRPD-n33-k4-s7-25	3819.62	3821.37	3830.03	39.41	3909.77	3922.74	34.20
EVRPD-n51-k5-s11-19	3061.89	3104.68	3144.38	59.59	3300.97	3389.15	51.03
EVRPD-n51-k5-s11-19-27-47	1916.57	1968.93	2089.22	56.66	1958.55	1992.22	49.86
EVRPD-n51-k5-s2-17	2891.04	2953.84	3000.79	56.59	3038.33	3122.75	51.22
EVRPD-n51-k5-s2-4-17-46	2895.94	2960.97	3028.70	55.39	3014.60	3089.14	49.29
EVRPD-n51-k5-s27-47	1917.50	1975.07	2070.98	59.00	1950.62	1983.73	49.16
EVRPD-n51-k5-s32-37	4918.59	4998.20	5074.82	62.82	4973.86	5030.61	54.48
EVRPD-n51-k5-s4-46	4170.25	4205.47	4233.21	58.56	5066.18	5148.52	51.53
EVRPD-n51-k5-s6-12	2540.91	2634.67	2674.66	54.15	2739.78	2769.80	49.82
EVRPD-n51-k5-s6-12-32-37	2543.73	2582.77	2723.38	56.93	2609.31	2669.19	49.00
Average		2756.34	2796.67	42.96	2932.44	2964.84	38.31

The performance of the GRASP-CRD variant demonstrates a significant advantage over the GRASP-VL variant for most instances. This can be attributed to the cardinality-based approach employed by GRASP-CRD, which utilizes a fixed maximum number of nodes to populate the RCL. This approach allows for the generation of high-quality solutions that are further improved by the LS procedure. To the contrary, the value-based approach utilized by GRASP-VL presents a larger number of choices during each step of solution construction. In combination with the unbiased choice method, this approach tends to yield relatively inferior solutions. Moreover, due to the very constrained nature of the problem, the local-search procedure fails to improve the solution, causing the algorithm to become trapped in locally optimal solutions.

The GRASP-CRD variant obtained 4 new BKVs in the small instances of 22 nodes. It is worth noting that the previous BKVs were attained through the implementation of Ant Colony Optimization (ACO) algorithms, as described in [1]. Additionally, one BKV was obtained through the employment of the Bee Colony Optimization (BCO) algorithm outlined in [46]. Unlike the ACO approaches, which employ a pheromone-based memory structure, the memory-less strategy of GRASP is more closely related to the BCO method and does not suffer from premature convergence to good solutions.

On the other hand, the GRASP-VL variant outperformed the GRASP-CRD variant in four instances. Specifically, in the solution of the EVRPD-n33-k4-s3-17 instance, the value-based variant utilized up to five drones per Electric Vehicle (EV), while the cardinality-based approach required up to six drones per EV.

Figures 4 and 5 demonstrate the results of Table 5 in terms of value gaps to the previously known best solution. The first figures displays the gaps% of the best solutions found and the second one displays the gaps% of the average solution found for each instance. When comparing the two figures, it is evident that both the best and the average result comparison showcase similar behavior between the same instances. In addition, despite the fact that GRASP-VL did not obtain any new optimal solutions, there are some cases in which the gap between the two GRASP methods was relatively small.

In terms of computational time, the GRASP-VL variant exhibited faster performance on average compared to GRASP-CRD in the tested instances. Part of this difference may be attributed to the complexity of the RCL construction procedure employed by each variant. The cardinality-based approach involves sorting the available nodes, resulting in a complexity of $\mathcal{O}(nlog(n))$, where n represents the number of elements. Conversely, the value-based approach utilizes a partition operation, which has a linear complexity of $\mathcal{O}(n)$. Figure 6 visualizes the execution time differences between the two GRASP variants.

In the industry, it is common among decision-makers to prioritize obtaining a satisfactory solution promptly rather than seeking the globally optimal one. Despite the GRASP-VL being approximately 12% quicker, its average solution quality lagged by about 6% compared to the GRASP-CRD variant. Given that the execution time differences among the GRASP variants are comparable, opting for the GRASP-CRD variant, which delivered significantly superior results, would be the most suitable choice for practical applications.

5.3. Comparison with Other Approaches

Table 6 presents the results of the two GRASP implementations compared to the other approaches found in the literature. Although the GRASP-CRD was able to obtain new BKVs for the small EVRPD instances, the lack of memory structure has a negative impact on the quality of results compared to the ACO for larger instances.

The comparative results between the ACO, BCO, and GRASP approaches indicate that the ability to resume the search and exploration of the solution space around previously identified promising areas is important in order to overcome local minimums. The lack of such a mechanism in the implemented GRASP-CRD algorithm provides better exploration opportunities in small instances, but hinders its ability to effectively search the solution space in larger instances. This behavior is also observed in the BCO results.

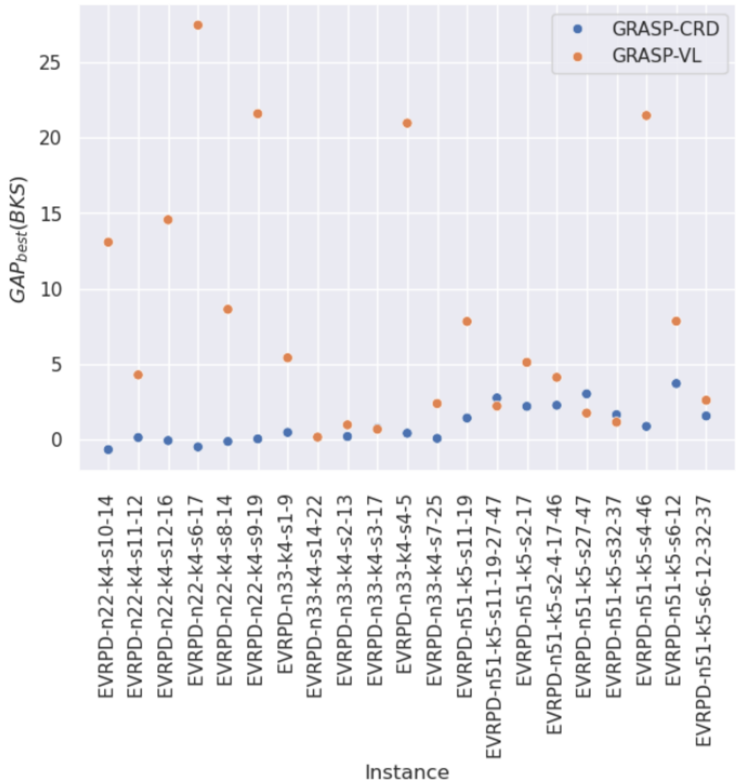

Figure 4. Gap% of best solution obtained to the best previously known solution for each instance.

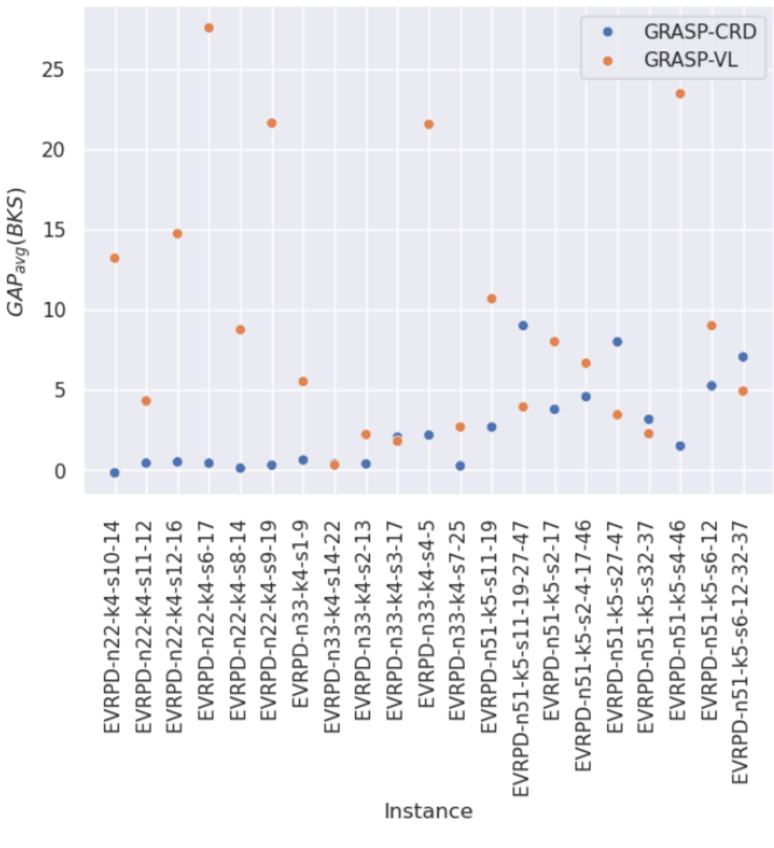

Figure 5. Gap% of average solution obtained to the best previously known solution for each instance.

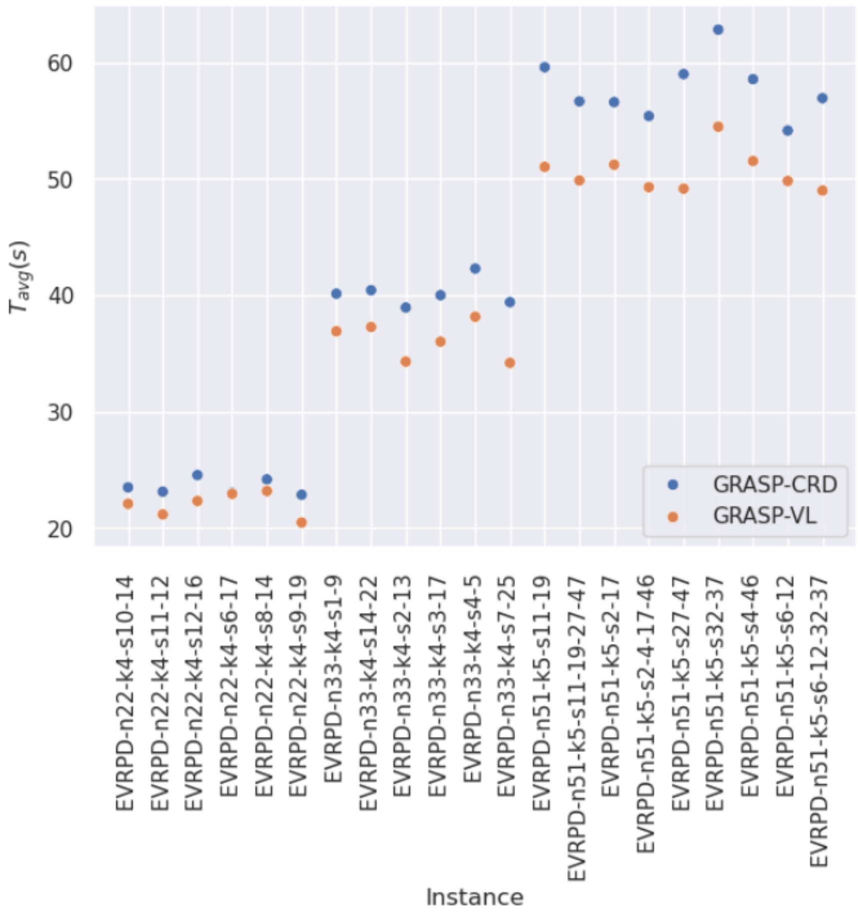

Figure 6. Average computational time required for each instance per algorithm variant.

Table 6. Comprehensive results of different approaches for the EVRPD.

Algorithm	Avg. $Cost_{best}$	Avg. $Cost_{avg}$	Avg. $Time_{avg}$
ACS	2736.84	2749.76	167.99
HACS	2738.77	2752.68	73.72
MMAS	2730.26	2745.36	41.75
HMMAS	2734.24	2752.78	43.39
BCO	2737.87	2763.89	43.81
GRASP-CRD	2756.34	2796.67	42.96
GRASP-VL	2932.44	2964.84	38.31

The GRASP-CRD approach proves to be the superior algorithm among the two GRASP variants examined. This superiority is evident in terms of both the quality of solutions obtained (both in terms of best and average) and the number of BKVs achieved. Therefore, when conducting statistical comparisons with other approaches from the literature, the GRASP-CRD variant is selected as the benchmark.

The statistical analysis comparing the implemented GRASP-CRD with other algorithms from the literature is presented in Table 7. The non-parametric Wilcoxon signed-rank test was utilized for this comparison. The initial row denotes the number of instances included in the test, and the following row shows the W-value. The third row presents the significance level α_s of the test, and row four presents the p-value. The last row of the table indicates if the null hypothesis H_0 can be rejected while limiting risk of rejecting H_0 when it is true to lower than 5%.

Table 7. Wilcoxon signed-rank test of GRASP-CRD compared to other algorithms on EVRPD instances.

Other Algorithm	ACS	ACSVND	MMAS	MMASVND	BCO
Number of instances	21	21	21	21	21
W-value	54.0	48.0	21.0	30.0	24.0
Significance level α_s	0.05	0.05	0.05	0.05	0.05
p-value	0.0319	0.0333	0.0017	0.0018	0.0007
H_0 rejected	Yes	Yes	Yes	Yes	Yes

The null hypothesis H_0 makes the assumption that the true mean of the algorithm being compared is equal to the mean of GRASP-CRD, and H_1 assumes a difference in the true means of the two algorithms. The test results validate the observations made regarding the average values obtained by the algorithms. The findings indicate that the null hypothesis can be rejected with less than 5% risk when GRASP-CRD is compared to the ACO implementations. Consequently, despite producing new BKVs for four instances, the performance of GRASP-CRD is statistically inferior to the ACO methods, especially MMAS and MMASVND, which obtained p-values less than 1%.

The comparison with BCO resulted in the smallest p-value, meaning the two algorithms have a statistically significant difference in their results. Nonetheless, the average performance of BCO, as presented in Table 6, is superior to that of GRASP, while GRASP offered more new BKVs compared to BCO.

6. Conclusions

The EVRPD is a complex VRP, combining EVs and drones, that considers package weights as the most controllable element of the energy expenditure. These electricity-powered vehicles are state-of-the-art vehicles, which will become more prevalent in the coming years as logistic companies transition to more environmentally sustainable operations. While drones offer numerous advantages in terms of environmental impact, cost-effectiveness, and service quality, their limited operational range presents a significant constraint that requires careful consideration. To overcome this limitation, the EVRPD employs a hybrid approach, utilizing both drones and EVs to share the delivery workload.

This paper presented the EVRPD model, which considers payload weight and payload quantity individually, incorporating both aspects into the model's constraints. Furthermore, two variants of the GRASP metaheuristic algorithm were implemented for solving the EVRPD. Each algorithm incorporated a different strategy for the construction of the RCL. GRASP-VL followed a value-based approach, while GRASP-CRD used a cardinality-based scheme. In both variants, the customer selection for the RCL was random and unbiased. To better exploit the generated solution, a local search procedure based on the VND algorithm was utilized as the second phase of the GRASP metaheuristic.

The GRASP-CRD approach achieves 4 new best solutions on small instances with 22 nodes. However, for larger instances, the cardinality-based variant does not exhibit competitive performance compared to Ant Colony Optimization (ACO) approaches found in the existing literature. While GRASP-VL does not yield significant results for most of the tested instances, it outperforms the GRASP-CRD variant in four instances.

These findings emphasize the significance of obtaining high-quality solutions to ensure the effectiveness of the local search procedure. While this is generally true for any VRP, it holds even greater importance for the EVRPD due to its inherent complexity and constrained nature. In practical applications, decision makers are not interested in finding the globally optimal solution, but rather a good enough solution in a timely manner. Although the GRASP-VL was about 12% faster, its average solution quality was about 6% worse than the GRASP-CRD variant. Since the differences in execution time of the implemented GRASP variants are on the same order of magnitude, the GRASP-CRD variant, which provided significantly better results, would be the best choice to adapt for a real-life scenario.

The EVRPD can serve as a foundation for future research extensions that incorporate additional aspects of electric vehicles, such as charging time and battery swapping. Considering the uncertainties surrounding charging station availability and actual vehicle range, exploring a stochastic variant of the problem would be worthwhile.

Author Contributions: N.A.K.: Conceptualization, Methodology, Software, Investigation, Writing—original draft preparation; T.S.: Conceptualization, Methodology, Software, Investigation, Writing—original draft preparation; M.M.: Conceptualization, Supervision, Methodology, Writing—original draft preparation, Writing—review and editing; Y.M.: Writing—review and editing, Supervision, Conceptualization, Methodology. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/vrp2einfo.html (accessed on 17 December 2023)).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kyriakakis, N.A.; Stamadianos, T.; Marinaki, M.; Marinakis, Y. The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. *Clean. Logist. Supply Chain.* **2022**, *4*, 100041. [CrossRef]
2. Kara, İ.; Kara, B.Y.; Yetis, M.K. Energy Minimizing Vehicle Routing Problem. In Proceedings of the Combinatorial Optimization and Applications, Xi'an, China, 14–16 August 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 62–71.
3. Conrad, R.G.; Figliozzi, M.A. The recharging vehicle routing problem. In Proceedings of the 2011 Industrial Engineering Research Conference (IISE), Reno, NV, USA, 21–25 May 2011; p. 8.
4. Erdoğan, S.; Miller-Hooks, E. A green vehicle routing problem. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 100–114. [CrossRef]
5. Schneider, M.; Stenger, A.; Goeke, D. The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transp. Sci.* **2014**, *48*, 500–520. [CrossRef]
6. Desaulniers, G.; Errico, F.; Irnich, S.; Schneider, M. Exact algorithms for electric vehicle-routing problems with time windows. *Oper. Res.* **2016**, *64*, 1388–1405. [CrossRef]
7. Keskin, M.; Çatay, B. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Comput. Oper. Res.* **2018**, *100*, 172–188. [CrossRef]
8. Montoya, A.; Guéret, C.; Mendoza, J.E.; Villegas, J.G. The electric vehicle routing problem with nonlinear charging function. *Transp. Res. Part B Methodol.* **2017**, *103*, 87–110. [CrossRef]
9. Froger, A.; Mendoza, J.E.; Jabali, O.; Laporte, G. *A Matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations*; Research Report; Centre Interuniversitaire de Recherche sur les Reseaux D'entreprise, la Logistique et le Transport (CIRRELT): Montreal, QC, Canada, 2017.
10. Zuo, X.; Xiao, Y.; You, M.; Kaku, I.; Xu, Y. A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function. *J. Clean. Prod.* **2019**, *236*, 117687. [CrossRef]
11. Keskin, M.; Laporte, G.; Çatay, B. Electric Vehicle Routing Problem with Time-Dependent Waiting Times at Recharging Stations. *Comput. Oper. Res.* **2019**, *107*, 77–94. [CrossRef]
12. Çatay, B.; Keskin, M. The impact of quick charging stations on the route planning of electric vehicles. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 152–157.
13. Ferro, G.; Paolucci, M.; Robba, M. An Optimization Model For Electrical Vehicles Routing with time of use energy pricing and partial Recharging. *IFAC-PapersOnLine* **2018**, *51*, 212–217. [CrossRef]
14. Ding, N.; Batta, R.; Kwon, C.; et al. *Conflict-Free Electric Vehicle Routing Problem with Capacitated Charging Stations and Partial Recharge*; Technical Report; Department of Industrial and Systems Engineering, University at Buffalo: Buffalo, NY, USA, 2015.
15. Chakraborty, N.; Mondal, A.; Mondal, S. Intelligent charge scheduling and eco-routing mechanism for electric vehicles: A multi-objective heuristic approach. *Sustain. Cities Soc.* **2021**, *69*, 102820. [CrossRef]
16. Keskin, M.; Çatay, B.; Laporte, G. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. *Comput. Oper. Res.* **2021**, *125*, 105060. [CrossRef]
17. Basso, R.; Kulcsár, B.; Sanchez-Diaz, I. Electric vehicle routing problem with machine learning for energy prediction. *Transp. Res. Part B Methodol.* **2021**, *145*, 24–55. [CrossRef]
18. Lin, B.; Ghaddar, B.; Nathwani, J. Electric vehicle routing with charging/discharging under time-variant electricity prices. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103285. [CrossRef]
19. Xiao, Y.; Zuo, X.; Kaku, I.; Zhou, S.; Pan, X. Development of energy consumption optimization model for the electric vehicle routing problem with time windows. *J. Clean. Prod.* **2019**, *225*, 647–663. [CrossRef]

20. Shao, S.; Guan, W.; Bi, J. Electric vehicle-routing problem with charging demands and energy consumption. *IET Intell. Transp. Syst.* **2017**, *12*, 202–212. [CrossRef]
21. Rastani, S.; Y'uksel, T.; Çatay, B. Effects of ambient temperature on the route planning of electric freight vehicles. *Transp. Res. Part D Transp. Environ.* **2019**, *74*, 124–141. [CrossRef]
22. Lin, J.; Zhou, W.; Wolfson, O. Electric Vehicle Routing Problem. *Transp. Res. Procedia* **2016**, *12*, 508–521. [CrossRef]
23. Zhang, S.; Gajpal, Y.; Appadoo, S.; Abdulkader, M. Electric vehicle routing problem with recharging stations for minimizing energy consumption. *Int. J. Prod. Econ.* **2018**, *203*, 404–413. [CrossRef]
24. Basso, R.; Kulcsár, B.; Egardt, B.; Lindroth, P.; Sanchez-Diaz, I. Energy consumption estimation integrated into the Electric Vehicle Routing Problem. *Transp. Res. Part D Transp. Environ.* **2019**, *69*, 141–167. [CrossRef]
25. Omidvar, A.; Tavakkoli-Moghaddam, R. Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. In Proceedings of the 2012 IEEE International Energy Conference and Exhibition (ENERGYCON), Florence, Italy, 9–12 September 2012; pp. 1089–1094.
26. Kullman, N.D.; Goodson, J.; Mendoza, J.E. *Dynamic Electric Vehicle Routing: Heuristics and Dual Bounds*; Hal Science: Lyon , France, 2018. *working paper or preprint*.
27. Zhang, S.; Chen, M.; Zhang, W.; Zhuang, X. Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert Syst. Appl.* **2020**, *145*, 113123. [CrossRef]
28. Breunig, U.; Baldacci, R.; Hartl, R.F.; Vidal, T. The electric two-echelon vehicle routing problem. *Comput. Oper. Res.* **2019**, *103*, 198–210. [CrossRef]
29. Jie, W.; Yang, J.; Zhang, M.; Huang, Y. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *Eur. J. Oper. Res.* **2019**, *272*, 879–904. [CrossRef]
30. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. [CrossRef]
31. Kitjacharoenchai, P.; Ventresca, M.; Moshref-Javadi, M.; Lee, S.; Tanchoco, J.M.; Brunese, P.A. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Comput. Ind. Eng.* **2019**, *129*, 14–30. [CrossRef]
32. Murray, C.C.; Raj, R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 368–398. [CrossRef]
33. Raj, R.; Murray, C. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102813. [CrossRef]
34. Luo, Z.; Poon, M.; Zhang, Z.; Liu, Z.; Lim, A. The Multi-visit Traveling Salesman Problem with Multi-Drones. *Transp. Res. Part C Emerg. Technol.* **2021**, *128*, 103172. [CrossRef]
35. Pina-Pardo, J.C.; Silva, D.F.; Smith, A.E. The traveling salesman problem with release dates and drone resupply. *Comput. Oper. Res.* **2021**, *129*, 105170. [CrossRef]
36. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [CrossRef]
37. Schermer, D.; Moeini, M.; Wendt, O. A matheuristic for the vehicle routing problem with drones and its variants. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 166–204. [CrossRef]
38. Hu, M.; Liu, W.; Lu, J.; Fu, R.; Peng, K.; Ma, X.; Liu, J. On the joint design of routing and scheduling for vehicle-assisted multi-UAV inspection. *Future Gener. Comput. Syst.* **2019**, *94*, 214–223. [CrossRef]
39. Sacramento, D.; Pisinger, D.; Ropke, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 289–315. [CrossRef]
40. Rossello, N.B.; Garone, E. Carrier-vehicle system for delivery in city environments. *IFAC-PapersOnLine* **2020**, *53*, 15253–15258. [CrossRef]
41. Karak, A.; Abdelghany, K. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 427–449. [CrossRef]
42. Nguyen, M.A.; Dang, G.T.H.; Hà, M.H.; Pham, M.T. The min-cost parallel drone scheduling vehicle routing problem. *Eur. J. Oper. Res.* **2021**. [CrossRef]
43. Kitjacharoenchai, P.; Min, B.C.; Lee, S. Two echelon vehicle routing problem with drones in last mile delivery. *Int. J. Prod. Econ.* **2020**, *225*, 107598. [CrossRef]
44. Li, H.; Wang, H.; Chen, J.; Bai, M. Two-echelon vehicle routing problem with time windows and mobile satellites. *Transp. Res. Part B Methodol.* **2020**, *138*, 179–201. [CrossRef]
45. Liu, C.; Chen, H.; Li, X.; Liu, Z. A scheduling decision support model for minimizing the number of drones with dynamic package arrivals and personalized deadlines. *Expert Syst. Appl.* **2021**, *167*, 114157. [CrossRef]
46. Kyriakakis, N.A.; Stamadianos, T.; Marinaki, M.; Matsatsinis, N.; Marinakis, Y. A Bee Colony Optimization Approach for the Electric Vehicle Routing Problem with Drones. In Proceedings of the Machine Learning, Optimization, and Data Science: 8th International Workshop (LOD 2022), Certosa di Pontignano, Italy, 19–22 September 2022; Revised Selected Papers, Part II; Springer: Berlin/Heidelberg, Germany, 2023; pp. 219–233.
47. Mara, S.T.W.; Sarker, R.; Essam, D.; Elsayed, S. Solving electric vehicle–drone routing problem using memetic algorithm. *Swarm Evol. Comput.* **2023**, *79*, 101295. [CrossRef]
48. Macrina, G.; Pugliese, L.D.P.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102762. [CrossRef]

49. Chung, S.H.; Sah, B.; Lee, J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Comput. Oper. Res.* **2020**, *123*, 105004. [CrossRef]
50. Moshref-Javadi, M.; Winkenbach, M. Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Syst. Appl.* **2021**, *177*, 114854. [CrossRef]
51. Stamadianos, T.; Kyriakakis, N.A.; Marinaki, M.; Marinakis, Y. Routing Problems with Electric and Autonomous Vehicles: Review and Potential for Future Research. *Oper. Res. Forum* **2023**, *4*, 46. [CrossRef]
52. Kyriakakis, N.A.; Aronis, S.; Marinaki, M.; Marinakis, Y. A GRASP/VND algorithm for the energy minimizing drone routing problem with pickups and deliveries. *Comput. Ind. Eng.* **2023**, *182* 109340. [CrossRef]
53. Perboli, G.; Tadei, R.; Vigo, D. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transp. Sci.* **2011**, *45*, 364–380. [CrossRef]
54. Kancharla, S.R.; Ramadurai, G. Electric vehicle routing problem with non-linear charging and load-dependent discharging. *Expert Syst. Appl.* **2020**, *160*, 113714. [CrossRef]
55. Feo, T.; Resende, M. Greedy Randomized Adaptive Search Procedures. *J. Glob. Optim.* **1995**, *6*, 109–133. [CrossRef]
56. Laguna, M.; Marti, R. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. *INFORMS J. Comput.* **1999**, *11*, 44–52. [CrossRef]
57. Aiex, R.; Resende, M.; Pardalos, P.; Toraldo, G. GRASP with Path Relinking for Three-Index Assignment. *INFORMS J. Comput.* **2005**, *17*, 224–247. [CrossRef]
58. Marinakis, Y. Multiple Phase Neighborhood Search-GRASP for the Capacitated Vehicle Routing Problem. *Expert Syst. Appl.* **2012**, *39*, 6807 – 6815. [CrossRef]
59. Resende, M.G.C.; Ribeiro, C.C. Parallel GRASP heuristics. In *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*; Springer: New York, NY, USA, 2016; pp. 205–227. [CrossRef]
60. Kyriakakis, N.A.; Marinaki, M.; Matsatsinis, N.; Marinakis, Y. A cumulative unmanned aerial vehicle routing problem approach for humanitarian coverage path planning. *Eur. J. Oper. Res.* **2022**, *300*, 992–1004. [CrossRef]
61. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
62. Mjirda, A.; Todosijević, R.; Hanafi, S.; Hansen, P.; Mladenović, N. Sequential variable neighborhood descent variants: An empirical study on the traveling salesman problem. *Int. Trans. Oper. Res.* **2017**, *24*, 615–633. [CrossRef]