sustainability

MDPI

Article

A Variable Neighborhood Descent Matheuristic for the Drone Routing Problem with Beehives Sharing

Maria Elena Bruni * and Sara Khodaparasti

Department of Mechanical, Energy and Management Engineering, University of Calabria, Via P. Bucci,
87036 Rende, CS, Italy

* Correspondence: mariaelena.bruni@unical.it

Abstract: In contemporary urban logistics, drones will become a preferred transportation mode for last-mile deliveries, as they have shown commercial potential and triple-bottom-line performance. Drones, in fact, address many challenges related to congestion and emissions and can streamline the last leg of the supply chain, while maintaining economic performance. Despite the common conviction that drones will reshape the future of deliveries, numerous hurdles prevent practical implementation of this futuristic vision. The sharing economy, referred to as a collaborative business model that foster sharing, exchanging and renting resources, could lead to operational improvements and enhance the cost control ability and the flexibility of companies using drones. For instance, the Amazon patent for drone beehives, which are fulfilment centers where drones can be restocked before flying out again for another delivery, could be established as a shared delivery systems where different freight carriers jointly deliver goods to customers. Only a few studies have addressed the problem of operating such facilities providing services to retail companies. In this paper, we formulate the problem as a deterministic location-routing model and derive its robust counterpart under the travel time uncertainty. To tackle the computational complexity of the model caused by the non-linear energy consumption rates in drone battery, we propose a tailored matheuristic combining variable neighborhood descent with a cut generation approach. The computational experiments show the efficiency of the solution approach especially compared to the Gurobi solver.

Keywords: drone routing problem; last-mile delivery; robust optimization; matheuristic

check for updates

Citation: Bruni, M.E.; Khodaparasti, S. A Variable Neighborhood Descent Matheuristic for the Drone Routing Problem with Beehives Sharing. *Sustainability* **2022**, *14*, 9978. https://doi.org/10.3390/su14169978

Academic Editor: Ripon Kumar Chakrabortty

Received: 27 June 2022
Accepted: 9 August 2022
Published: 12 August 2022
Corrected: 24 November 2022

1. Introduction

Last-mile delivery systems have increasingly exploited the use of drones in a number of applications to reduce delivery times, avoid traffic delays and potentially cut costs in the long term [1–3]. The actual implementation of fully drone-based last-mile solutions is challenged by the intrinsic characteristics of drones as small payload and limited battery capacity. Amazon released a number of patents trying to address these issues, envisioning multilevel fulfillment centers, also called Beehives (BHs), providing a number of services from package handling to recharging/refueling operations. These stations raise emerging problems relative to the management of the delivery services provided and the shared used of these big, capital intensive facilities. In this setting, each drone does not need to return to the departure depot for reloading and battery charging, but it can fly to other depots, enabling the shift from independent business operations to collaboration. Sharing BHs could sensibly reduce the total number of drones in the sky and the total operational costs of companies, but raises operational challenges to be addressed.

Although the idea of sharing BHs seems quite appealing, there are only a very few studies addressing the operational challenges in the last-mile drone delivery with sharing BHs. The distinguished study of Aurambout et al. [4] presents a framework to estimate the economic viability of potential drone BH locations across the European cities. Recently, Bruni. and Khodaparasti. [5] provided a mathematical programming approach for this complicated problem, considering some realistic features as drone energy consumption rates—which non-linearly depends on the payload, frame and battery mass, and travel time—and uncertainty in flight duration. We should mention that the model is quite involved, since it takes into account the non-linear nature of the energy constraints.

To the best of our knowledge, there are only a few studies that explicitly account for the energy constraints with non-linear rates [1,6]. They model the energy consumption rate in drone battery as a non-linear function in terms of payload and linear in terms of the travel time. However, a linear approximation approach is adopted to avoid the computational burden of dealing with non-linear energy constraints.

On the other hand, other studies replace the energy constraints with the more tractable flight range constraints to implicitly account for the limited drone battery charge [7].

The current paper aimed to fill this potential gap. In particular, we are interested in the development of efficient solution methods for the Drone Routing Problem with Shared BHs (DRP-ShaBH) [5]. We propose a tailored matheuristic approach that combines the Variable Neighborhood Descent (VND) [8], a metaheuristic commonly used as a local search for the Variable Neighborhood Search, with a cut generation procedure. The integration of VND presents numerous success cases in the literature but it must be carefully designed, since the computational cost of local search operators might compromise the entire optimization strategy.

The contributions of the paper is twofold:

- We formulate the drone delivery with shared BHs as a location-routing mathematical model where the non-linear nature of energy consumption rates in drone battery and the uncertainty of flight duration are taken into account.
- We develop an efficient matheuristic approach integrating the VND with a cut generation procedure that provides promising results outperforming the Gurobi off-the-shelf solver.

The rest of the paper is organized as follows. In Section 2, we review the relevant literature on drone-aided location-routing problems. In Section 3, we describe the DRP-ShaBH and formulate it as a deterministic mathematical program. Next, we derive the robust counterpart of the model under the uncertainty of flight duration. In Section 4, we describe the proposed matheuristic approach that tackles the computational complexity of the DRP-ShaBH exacerbated by the combined location and routing nature of the problem. In Section 5, we report the computational experiments conducted on a set on benchmark instances. Finally, we conclude the paper in Section 6 and outline avenues for future research.

2. Literature Review

In this section, we review the literature on drone location-routing problems in the last-mile delivery. We may recognize two different streams in the literature: (*i*) The contributions that adopt a hierarchical approach in which the location decisions and the operational drone-routing plans are tackled separately in a hierarchical fashion; (*ii*) the studies that address the strategic/tactical location plans and operational routing decisions simultaneously. As a matter of fact, the adoption of each stream is dependent on the application context.

Following the first stream, Kim et al. [9] proposed a set covering model to find the optimal locations used as drone depots in a healthcare context, followed by a multi-depot drone pickup and delivery model that minimizes the cost of used drones. The model accounts for the drone payload and the maximum drone flight time. A preprocessing algorithm, a partition method and a lagrangian relaxation method were developed to solve the model. Torabbeigi et al. [10] proposed two mathematical formulations involving strategic and operational plans to optimize the drone routes in a parcel delivery application. At the strategic level, a set covering model determines the minimum number of required depots to cover all customers, next, at the operational stage, a routing model is solved that finds the optimal drone routes minimizing the fleet size. The authors modeled the energy consumption as linear constraints where the consumption rates are expressed as a linear function of drone payload, drone mass, and travel time. The authors developed a variable preprocessing algorithm and a primal and dual bound generation procedure to accelerate the solution approach performance.

Following the second stream, Liu et al. [11] proposed a location-routing model for a patrolling problem with drones to find the optimal location of drone launching bases and drone routes simultaneously. The aim is to minimize the total cost encompassing the base establishment cost, drone usage cost, and flight cost. The authors imposed some lower and upper bounds on the capacity of established bases and a maximum drone flight duration was considered to account for the limited battery charge. The authors developed two heuristic algorithms combined with local search that handles a test case with 25 target points and 5 potential base stations. In another study, Yakıcı [12] proposed a selective location-routing problem that seeks the optimal location of drone stations and optimal drone tours with the aim of maximizing the sum of importance values corresponding to the covered interest points. The model accounts for a maximum flight time and an upper bound for the number of selected stations. As the solution method, an ant colony optimization metaheuristic is designed. Kim et al. [7] proposed a drone routing model with multiple depots, multiple drones, allowing multiple UAVs to deliver goods to one customer at the same time. The objective function minimizes the delivery and drone usage costs. The drone payload capacity and a maximum flight distance are considered. The optimal location of selected depots, the fleet size, and the drone routes are the output of the problem. The model is solved for instances up to 75 customers. In another paper, Li et al. [13] studied a multi-depot drone routing problem minimizing the total number of dispatched drones and the total travel distance. The model accounts for a maximum drone flight time. Since all the depots are not required to be used, the optimal location of used depots is an output of the problem. To solve the problem, the authors presented a heuristic approach based on hybrid large neighborhood search. In a recent paper, Grogan et al. [14] addressed a drone application for relief operations after tornado. The authors presented a multi-depot routing problem to minimize the maximum route duration while imposing a maximum drone endurance limit for each drone. We classify the latter contribution as a location and fleet sizing model since the number of dispatched drones and occupied depots are the problem outputs.

To conclude, the energy consumption notion in the literature has been simplified or replaced by the flight range notion in order to avoid the computational burden caused by the non-linear energy constraints. This, clearly, makes the validity of the route plans and the BH site location decisions quite questionable. Moreover, the uncertainty and the fluctuations in the flight duration has not been addressed in the aforementioned studies. The current study bridges this gap and presents an efficient solution approach to handle the computational burden of a realistic model that explicitly accounts for complicating features such as the non-linear energy consumption rate in drone battery and the uncertain flight duration.

3. Drone Routing Problem with Shared Beehives

The DRP-ShaBH under the uncertainty of flight duration is a tailored location-routing problem, where the operational challenges related to the use of drones, among which energy consumption and the uncertainty in flight duration, are taken into account. We assume that a set D of distributed BHs are available as the drone stations to enhance last-mile delivery. We can consider any type of BHs, truck-based warehouses, local re-stocking stations for drones or fulfillment centers. The BHs offer landing, takeoff, package handling, and recharging services where each BH $f \in D$ can host up to N_{BH}^{f} drones. A limited number N_d of drones can shuttle back and forth between the BHs and a maximum number of N_s BHs can be used. A retail company uses the drones to deliver orders from BHs to a set C of customers. Each drone performs a distinct tour started from a specific BH, where the drone is loaded with the customers' orders up to its payload capacity Q. Next, the drone is launched to visit a subset of customers one by one. Upon arrival to each customer's site, the drone leaves the order and flies to visit the next customer. After delivering the order of the last customer within its tour, the drone is retrieved at one of the selected BHs, that could be different from the BH we used to launch the drone.

Figure 1 represents the delivery scheme for the DRP-ShaBH.

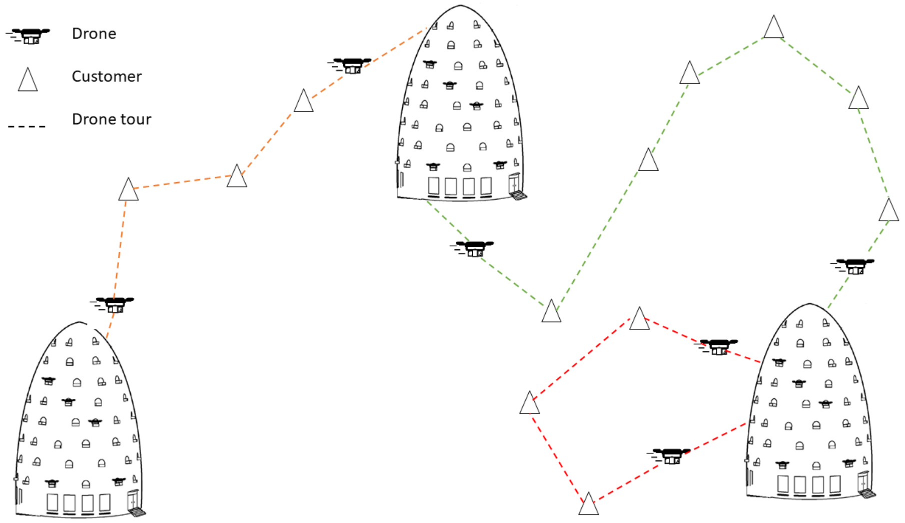

Figure 1. Scheme of the delivery system in DRP-ShaBH: A delivery example with 3 BHs, 3 drones, and 12 customers

Instead of assuming that drone endurance is limited by a fixed amount of distance, we consider energy consumption as a (non-linear) function of drone payload, frame and battery mass, and travel time.

In fact, Dorling et al. [1] model the energy consumption based on the main assumption that the power consumed during takeoff, or landing is approximately equivalent to the power consumed while hovering and the thrust balances the weight force.

In particular, let t_{ij} be the flight duration (travel time) between nodes i and j in the network, $W + M = \Gamma$ (W is the frame mass (in kg) and M (in kg) is the battery mass). The drone energy consumption along arc (i, j), while carrying a payload of d is expressed as $k(\Gamma + d)^{3/2}(t_{ij})$ where $k = \sqrt{\frac{g^3}{2\rho\xi h}}$ (here, g is the force applied by gravity per unit mass (in $\mathrm{N/kg}$), ρ is the fluid density of air (in $\mathrm{kg/m^3}$), ξ is the area of spinning blade disc (in $\mathrm{m^2}$), and h is the number of rotors) [1].

The mathematical formulation of DRP-ShaBH is essentially developed based on the multi-layered network, which has been proved to be superior to other models, for a wide range of routing problems (for more information about the multi-layered network, please see [15–19]). The model takes the advantage of position-dependent binary variables: x_i^r which indicates if customer $i \in C$ is visited in position r along the drone tour (meaning that customer i is the r^{th}-last customer served by the drone within its trip), y_{ij}^r to show that customer $j \in C$ is visited right after node $i \in C \cup D$ in position r and w_{fj}^r which tells if customer $j \in C$ is the first one to be visited by a drone launched from BH $f \in D$ and there are $r - 1$ customers left to be visited along the tour. Binary variable z_f indicates if BH $f \in D$ is used or not. The total load carried by the drone upon departure from customer i to reach customer j is denoted by u_{ij}; if customer j is the first customer after the drone departure from BH f, its load is denoted by v_{fj}. The accumulated energy consumption upon arrival at node $i \in C \cup D$ is denoted by e_i. Let $D' = \{0, m+1\}$ denote the set of auxiliary BHs introduced to specify the ending points of the drone trip and $L = \{1, \ldots, r, \ldots, N\}$ be the set of positions where $N \leq |C| - N_d + 1$. Table A1 in Appendix A summarizes the notation used in the mathematical formulation of the problem. The DRP-ShaBH is formulated as follows:

$$\min \sum_{f\in D}\sum_{j\in C}\sum_{r\in L\setminus\{1\}} r\, t_{fj}\, w^r_{fj} + \sum_{i\in C}\sum_{\substack{j\in C\\ j\neq i}}\sum_{r\in L} r\, t_{ij}\, y^r_{ij} \tag{1}$$

$$\sum_{r\in L} x^r_i = 1 \quad i\in C \tag{2}$$

$$\sum_{r\in L}\sum_{j\in C} y^r_{0j} = N_d \tag{3}$$

$$\sum_{i\in C} x^1_i = \sum_{r\in L}\sum_{j\in C} y^r_{0j} \tag{4}$$

$$\sum_{\substack{j\in C\\ j\neq i}} y^r_{ij} = x^{r+1}_i \quad i\in C,\ r\in L\setminus\{N\} \tag{5}$$

$$\sum_{\substack{i\in C\cup\{0\}\\ i\neq j}} y^r_{ij} = x^r_j \quad j\in C,\ r\in L\setminus\{N\} \tag{6}$$

$$y^N_{0j} = x^N_j \quad j\in C \tag{7}$$

$$\sum_{f\in D} w^r_{fj} \geq y^r_{0j} \quad j\in C,\ r\in L\setminus\{1\} \tag{8}$$

$$z_f \geq w^r_{fj} \quad f\in D,\ j\in C,\ r\in L\setminus\{1\} \tag{9}$$

$$\sum_{f\in D} z_f \leq N_s \tag{10}$$

$$\sum_{j\in C}\sum_{r\in L} w^r_{fj} \leq N^f_{BH} \quad f\in D \tag{11}$$

$$x^1_j = y^0_{j(m+1)} \quad j\in C \tag{12}$$

$$y^0_{j(m+1)} \geq \sum_{f\in D} w^1_{jf} \quad j\in C \tag{13}$$

$$\sum_{j\in C} w^1_{jf} \leq \sum_{r\in L\setminus\{1\}}\sum_{j\in C} w^r_{fj} \quad f\in D \tag{14}$$

$$v_{fj} \geq d_j \sum_{r\in L\setminus\{1\}} w^r_{fj} \quad f\in D,\ j\in C \tag{15}$$

$$v_{fj} \leq Q \sum_{r\in L\setminus\{1\}} w^r_{fj} \quad f\in D,\ j\in C \tag{16}$$

$$u_{ij} \geq d_j \sum_{r\in L} y^r_{ij} \quad i,j\in C,\ i\neq j \tag{17}$$

$$u_{ij} \leq (Q-d_i) \sum_{r\in L\setminus\{N\}} y^r_{ij} \quad i,j\in C,\ i\neq j \tag{18}$$

$$\sum_{f\in D} v_{fi} + \sum_{\substack{j\in C\\ j\neq i}} u_{ji} - \sum_{\substack{j\in C\\ j\neq i}} u_{ij} = d_i \quad i\in C \tag{19}$$

$$e_f + k(\Gamma + v_{fj})^{3/2}(t_{fj}) \leq e_j + M'_{fj}(1 - \sum_{r\in L\setminus\{1\}} w^r_{fj}) \quad f\in D,\ j\in C \tag{20}$$

$$e_i + k(\Gamma + u_{ij})^{3/2}(t_{ij}) \leq e_j + M''_{ij}(1 - \sum_{r\in L\setminus\{N\}} y^r_{ij}) \quad i,j\in C,\ i\neq j \tag{21}$$

$$e_j + k(\Gamma)^{3/2} \sum_{f\in D} (t_{jf}) w^1_{jf} \leq B \quad j\in C \tag{22}$$

$$e_f = 0 \qquad f\in D \tag{23}$$

$$z_f \in \{0,1\} \quad f\in D \tag{24}$$

$$x^r_i \in \{0,1\} \quad i\in C,\ r\in L \tag{25}$$

$$y^r_{0j} \in \{0,1\} \quad j \in C,\ r \in L \tag{26}$$

$$y^0_{j(m+1)} \in \{0,1\} \quad j \in C \tag{27}$$

$$y^r_{ij} \in \{0,1\} \quad i,j \in C,\ i \neq j,\ r \in L \setminus \{N\} \tag{28}$$

$$w^r_{fj} \in \{0,1\} \quad f \in D,\ j \in C,\ r \in L \setminus \{1\} \tag{29}$$

$$w^1_{if} \in \{0,1\} \quad i \in C,\ f \in D \tag{30}$$

$$u_{ij} \geq 0 \quad i,\ j \in C,\ i \neq j \tag{31}$$

$$v_{fj} \geq 0 \quad f \in D,\ j \in C \tag{32}$$

$$e_i \geq 0 \quad i \quad \in C \tag{33}$$

where M'_{fj} and M''_{ij} are "bigM" numbers and the variables definitions are as in Table A1.

The objective function (1) represents the total waiting times of the customers. Constraints (2) ensure that each customer is visited exactly once. Constraint (3) ensures the use of exactly N_d drones. Constraints (4)–(6) guarantee the continuity of each tour. Constraints (7) force nodes to be connected to node 0 at level N. Constraints (8) and (9) allow starting a drone route only from used facilities. Constraints (10) and (11) represent the restriction on the maximum number of BHs used and the number of drones that each BH can handle. Constraints (12) and (13) define the relation between the last customer visited and the destination BH. Constraints (14) determine the return of drones to BHs already used as drone launch sites. Constraints (15)–(19) are used to limit the drone payload [20], while (20)–(23) represent the drone energy consumption. Finally, the set of constraints (24)–(33) establish the nature of the variables.

As is known, weather conditions, wind direction and aerial congestion can influence the travel time of the drone and, hence, the energy consumption. In this paper, we consider drone travel time between nodes i and j as an uncertain parameter denoted by $\tilde{t}_{ij}$ that belongs to a box uncertainty set $U_\infty = \{\tilde{t} | \, ||\tilde{t}||_\infty \leq \Psi\}$, where Ψ is an adjustable parameter controlling the size of the uncertainty set. We refer the reader to [21] for more information on uncertainty sets in robust optimization.

Following the robust paradigm in [22], the robust counterpart of the DRP-ShaBH is expressed as follows.

$$\min \sum_{f \in D} \sum_{j \in C} \sum_{r \in L \setminus \{1\}} r\,(\bar{t}_{fj} + \Psi \hat{t}_{fj})\, w^r_{fj} + \sum_{i \in C} \sum_{\substack{j \in C \\ j \neq i}} \sum_{r \in L} r\,(\bar{t}_{ij} + \Psi \hat{t}_{ij})\, y^r_{ij} \tag{34}$$

$$e_f + k(\Gamma + v_{fj})^{3/2}(\bar{t}_{fj} + \Psi \hat{t}_{fj}) \leq e_j + M'_{fj}(1 - \sum_{r \in L \setminus \{1\}} w^r_{fj}) \ f \in D,\ j \in C \tag{35}$$

$$e_i + k(\Gamma + u_{ij})^{3/2}(\bar{t}_{ij} + \Psi \hat{t}_{ij}) \leq e_j + M''_{ij}(1 - \sum_{r \in L \setminus \{N\}} y^r_{ij}) \ i,\ j \in C,\ i \neq j \tag{36}$$

$$e_j + k\Gamma^{3/2} \sum_{f \in D} (\bar{t}_{jf} + \Psi \hat{t}_{jf}) w^1_{jf} \leq B \ j \in C \tag{37}$$

(2)–(19), (23)–(33)

where the uncertain parameter $\tilde{t}_{ij}$ falls within the symmetric box $[\bar{t}_{ij} - \Psi\,\frac{\hat{t}_{ij}}{2}, \bar{t}_{ij} + \Psi\,\frac{\hat{t}_{ij}}{2}]$.

Needless to say, the computational intractability of the model is exacerbated by the presence of non-linear constraints (35)–(37). To overcome this difficulty, we develop an efficient VND-based matheuristic which is discussed in Section 4.

4. The VND-Based Matheuristic

In this section, we outline the proposed matheuristic integrating the VND heuristic with a cutting plane approach. The main idea behind the proposed VND-based Matheuristic (VND-M) is to apply VND as a local search heuristic that explores different neighborhood structures in a deterministic way. In particular, we define each neighborhood by constraints which are derived based on the last information on the current solution. To be more precise, we construct two types of constraints, known as "no-good" cuts, that exclude the current solution from the search space [23].

To derive the cuts, we decompose the problem into a master problem ignoring the complicating non-linear energy constraints (35)–(37), and a subproblem that checks the feasibility of the current solution with respect to the energy constraints. The master problem is solved to optimality using an off-the-shelf MIP solver (such as Gurobi) and based on the information we get from the optimal solution, inequalities cutting off the current solution from the feasible space are added as constraints into the master problem. In the following, we explain how the inequalities are generated.

From now on, we call any feasible solution of the master problem that is also feasible for the subproblem an "energy-feasible" candidate.

Once the master problem is solved, an integer feasible solution $p = (\hat{x}, \hat{y}, \hat{w}, \hat{z}, \hat{u}, \hat{v})$ with the corresponding objective value Z^p is found. Next, we evaluate the feasibility of solution p with respect to the subproblem (35)–(37). Two mutually exclusive cases may occur:

(1) The current solution p is not an energy-feasible solution and the battery capacity for some of the drone tours is violated (we refer to the tours exceeding the battery charge as the "infeasible tours"). In this case, corresponding to each infeasible tour $\mathscr{P}_\pi$, $\pi \in \{1, \cdots, N_d\}$ in p, a valid no-good cut (38) is generated and added to the master problem.

$$\mathscr{N}\mathscr{G}(\pi) : \sum_{\substack{(f,j)\in\mathscr{P}_\pi \\ f\in D}} \sum_{r\in L} (1 - w^r_{fj}) + \sum_{\substack{(i,j)\in\mathscr{P}_\pi \\ i,j\in C}} \sum_{r\in L} (1 - y^r_{ij}) + \sum_{\substack{(i,f)\in\mathscr{P}_\pi \\ f\in D}} (1 - w^1_{if}) \geq rhs \tag{38}$$

Note that rhs is a pre-specified integer input parameter. From now on, we refer to (38) as the cut type I. Any cut in the form of (38) is, in fact, a valid cut that ensures corresponding to each infeasible tour $\mathscr{P}_\pi$, at least rhs customers/BHs are removed. Obviously, the higher the value of rhs is, the more emphasis on seeking diverse solutions is imposed. Once the cuts are generated, we can specify the structure of neighborhood as follows:

Let $\{\mathscr{P}_\pi\}_{\pi=1}^{s}$ be the set of infeasible tours in p, the neighborhood $\mathscr{N}_{rhs}(p)$ ($\mathscr{N}_{rhs} \subseteq X$) is defined as $\mathscr{N}_{rhs}(p) = \{\bar{p} | \bar{p} \in \bigcap_{\pi=1}^{s} \mathscr{N}\mathscr{G}(\pi)\}$ where X denotes the master problem solution space. We should note that the parameter rhs is interpreted as the size of neighborhood $\mathscr{N}_{rhs}$.

(2) The current solution p is an energy-feasible solution, not necessarily optimal. Therefore, we cut off the current solution from the search space introducing a single no-good cut cut type II removing the current solution from the search space.

$$\mathscr{N}\mathscr{G}(1) : \sum_{(i,j)\in S} \sum_{r\in L} (1 - y^r_{ij}) + \sum_{(i,j)\notin S} \sum_{r\in L} y^r_{ij} \geq rhs \tag{39}$$

where $S = \{(i,j) \in \cup_\pi \mathscr{P}_\pi | i, j \in C, \hat{y}^r_{ij} = 1\}$. The neighborhood $\mathscr{N}'_{rhs}(p)$ is defined as $\mathscr{N}'_{rhs}(p) \subseteq X$, $\mathscr{N}'_{rhs}(p) = \{\bar{p} | \bar{p} \in \mathscr{N}\mathscr{G}(1)\}$.

Algorithm 1 describes the cut generation procedure. We should note that, each time the master problem is solved, we invoke a build route procedure to retrieve the routes $\mathscr{P}_\pi$ from p. The scheme of the procedure is shown in Algorithm 2. Thanks to the build route procedure, the energy feasibility check of a master problem solution is easily done in polynomial time.

Algorithm 1 Generate cut (p) procedure

```
**Input:** Solution $p$
**Output:** Set of cuts $\mathscr{C}$
$\mathscr{C} \leftarrow \emptyset$, $\mathscr{P}_\pi \leftarrow \emptyset$
$\mathscr{P}_\pi \leftarrow$ *build-route*$(p)$
**if** $p$ is energy-feasible **then**
    Generate type II cut $\mathscr{NG}(1)$ as in (39)
    $\mathscr{C} \leftarrow \mathscr{C} \cup \{\mathscr{NG}(1)\}$
**else**
    **for** $\pi \in \cup_{\pi=1}^{s} \mathscr{P}_\pi$ (set of infeasible tours) **do**
        Generate type I cut $\mathscr{NG}(\pi)$ as in (38)
        $\mathscr{C} \leftarrow \mathscr{C} \cup \{\mathscr{NG}(\pi)\}$
    **end for**
**end if**
**Return** $\mathscr{C}$
```

Algorithm 2 Pseudocode of build route (p) procedure

```
$\pi \leftarrow 0$, $flag \leftarrow 0$
**for** $c \in C$ **do**
    **if** $x_c^1 = 1$ **then**
        $\pi \leftarrow \pi + 1$
        $\mathscr{P}_\pi \leftarrow \emptyset$
        $f' \leftarrow \{f \in D | \, w_{cf'}^1 = 1\}$
        $\mathscr{P}_\pi \leftarrow \mathscr{P}_\pi \cup \{(c, f')\}$
        **repeat**
            $c' \leftarrow \{i \in C \cup \{0\} | y_{ic}^r = 1\}$
            **if** $c' = 0$ **then**
                $flag \leftarrow 1$
                **break**
            **end if**
            $\mathscr{P}_\pi \leftarrow \mathscr{P}_\pi \cup \{(c', c)\}$
            $c \leftarrow c'$
        **until** $flag = 0$
        $f'' \leftarrow \{f \in D | \, w_{fc}^r = 1\}$
        $\mathscr{P}_\pi \leftarrow \mathscr{P}_\pi \cup \{(f'', c)\}$
    **end if**
**end for**
**Return** $\{\mathscr{P}_\pi\}_\pi$
```

The general sketch of the VND-M method is shown is Algorithm 3. The algorithm takes the maximum number of neighborhoods to explore as the input which is denoted by $\mathscr{N}$ and returns the incumbent p^* with the corresponding objective value Z^*. First, in Line 4, the master problem is solved and an integer feasible solution p with objective value Z^p is found. Next, the feasibility subproblem is solved to check the energy feasibility of p. If p is energy feasible, the incumbent and its objective value are updated (Lines 5–7). Lines 8–27 describe the scheme of VND heuristic. Starting from the first neighborhood specified by its right hand side of $rhs = 1$ in line 9, the corresponding neighborhood around p is explored; this requires the definition of the proper cut (Line 11) which is then added into the master problem (Line 12) which is solved again to find a new solution p'. Three different cases are relevant:

(*i*) The new solution p' is energy-feasible, therefore, we replace the current solution p with p' and update the incumbent, if necessary (Lines 13–17).

(*ii*) The new solution p' is not energy-feasible but the amount of violation is within an acceptable pre-specified threshold α. In such case, we replace the current solution p with p' (Line 19) . This can be regarded as a diversification mechanism to allow a controlled exploration of infeasible promising solutions.

(*iii*) None of the two previous cases holds. Hence, we move to the next neighborhood (Line 21).

This procedure is repeated until all the $\mathcal{N}$ neighborhoods are explored. We also keep track of the incumbent updates to early end the algorithm whenever no improving solution within the last β consecutive iterations is found. As for the stopping criterion, a time limit is imposed.

Algorithm 3 VND-M

```
 1: Input: $\mathcal{N}$, $\alpha$, $\beta$
 2: Output: $p^*$, $Z^*$
 3: Initialization: $p^* \leftarrow 0$, $Z^* \leftarrow \infty$
 4: Find an integer feasible solution $p$ for the master problem
 5: if $p$ is energy feasible then
 6:     $Z^* \leftarrow Z^p$, $p^* \leftarrow p$
 7: end if
 8: while not stopping criterion do
 9:     $rhs \leftarrow 1$
10:     while $rhs \leq \mathcal{N}$ do
11:         Generate cut $(p)$
12:         Solve the master problem with cuts and let $(p', Z^{p'})$ be the optimal solution
13:         if $p'$ is energy-feasible then
14:             $p \leftarrow p'$
15:             if $Z^p < Z^*$ then
16:                 $Z^* \leftarrow Z^p$
17:             end if
18:         else if $p'$ energy-violation is within a predefined threshold $\alpha$ then
19:             $p \leftarrow p'$
20:         else
21:             $rhs \leftarrow rhs + 1$
22:         end if
23:         if No improvement within the last consecutive $\beta$ iterations then
24:             break
25:         end if
26:     end while
27: end while
28: Return $p^*$, $Z^*$
```

5. Computational Experiments

In this section, we report on the computational study carried out to assess the validity of the proposed matheuristic approach. All the experiments were executed on an Intel Core i7-10750H, with 2.60 GHz CPU and 16 GB RAM working under Windows 10. The model and the VND-M algorithm were implemented in the algebraic modeling language AIMMS 4.79.2.5. Since model (34)–(37), (2)–(19), (23)–(33), is a non-linear MIP model , we solved it using the AIMMS Outer Approximation (AOA) algorithm. Gurobi 9.0.1 was used instead to solve the master problem. A time limit of 500 s was set for both the exact and the matheuristic approach.

We adopted the set of A1 instances up to 40 customers from [6]. The coordinates of the customer are the same as those reported in [6]. Regarding the customer demand, we considered a random demand distribution varying between $[0.1, 0.7]$ kg for the first 40% of customers and $[0.1, 1.5]$ kg for the rest of customer. We considered up to 5 potential BHs with two spatial configurations (close to the center of the delivery area or in the outskirts), referred to as centered and marginal. In the problem analysis, values of 2, 3, 5, 6 were set as the maximum number of drones per BH, depending on the problem size. Finally, we considered a homogeneous fleet of 2, 3, 4, 7 or 8 Alta 8 drones with the following characteristics: $W = 6.2\,(\text{kg})$, $M = 2.8\,(\text{kg})$, $Q = 9.1\,(\text{kg})$, $h = 8$, $\rho = 1.204\,(\text{kg/m}^3)$, $\xi = 0.1256\ (\text{m}^2)$, $B = 0.355\,(\text{kWh})$. As for the heuristic, we set the following input values $\mathcal{N} = 5$, $\alpha = 0.01\,B$, $\beta = 10$.

Tables 1 and 2 report the computational results under the centered and marginal configurations, respectively. Columns 1–5 present the characteristic of the instances considered; the objective function value (Obj) and the solution time in seconds (CPU) for both the proposed VND-M and the AOA are reported in columns 6–7 and 8–9, respectively. Column 10 with heading LB displays a lower bound derived from a linear relaxation of the master problem where all the integer variables except the z_f variables are relaxed. Needless to say that the lower bounds for cases solved to optimality are not relevant and therefore have not been reported. Finally, the last column with heading *Gap* represents the relative gap (in percentage) with respect to the solution found by AOA, if any; otherwise, the gap has been calculated with respect to the lower bounds LB.

As the results show, AOA provides the optimal solution for instances up to 20 customers. As a general observation, we note that the AOA computational time is also quite limited, especially considering that this problem is a tactical operational problem. Interestingly, for all instances solved to optimality by AOA, the VND-M finds the optimal solution, too (except for instance A1-15-5 with the centered configuration for which the optimality gap is quite low 0.85%). In terms of the solution time, for the smallest instances with 10 customers, the VND-M CPU time is higher than the AOA's but with the increase of instance size to 15 and 20, the VND-M CPU time outperforms the AOA significantly. For instances with more than 30 customers, the AOA fails to find any feasible solution within the time limit (those specified by TL that accounts for Time Limit). On the contrary, the proposed VND-M is quite fast and reports promising results for all instances confirmed by the average gap of at most 2.06% and 5.65% and the CPU time of 97.96 and 35.57 s under the centered and marginal configurations, respectively.

The position of the BHs does not seem to influence a lot the total latency since on average the percentage difference is quite low and below 5%.

Table 1. Computational results for instances under centered setting.

					VND-M		**AOA**			
Instance	$\|C\|$	$\|D\|$	N_d	N^f_{BH}	**Obj**	**CPU**	**Obj**	**CPU**	**LB**	$Gap(\%)$
A1-10-1	10	5	2	2	2.02	17.89	2.02	7.09		0.00
A1-10-2					1.80	1.01	1.80	2.02		0.00
A1-10-3					1.84	43.51	1.84	4.11		0.00
A1-10-4					1.57	2.32	1.57	2.77		0.00
A1-10-5					1.96	0.55	1.96	2.75		0.00
A1-15-1	15	5	3	2	2.09	1.87	2.09	33.38		0.00
A1-15-2					2.52	2.65	2.52	48.49		0.00
A1-15-3					2.00	3.00	2.00	37.94		0.00
A1-15-4					2.24	1.49	2.24	42.86		0.00
A1-15-5					2.35	4.03	2.33	47.55		0.85
A1-20-1	20	5	4	3	2.60	13.10	2.60	336.83		0.00
A1-20-2					2.67	6.74	2.67	442.88		0.00
A1-20-3					2.68	8.33	2.68	496.25		0.00
A1-20-4					2.38	6.58	-	TL	2.38	0.00
A1-20-5					2.72	9.35	-	TL	2.61	4.04

Table 1. *Cont.*

					VND-M		**AOA**			
Instance	$\lvert C \rvert$	$\lvert D \rvert$	N_d	N_{BH}^f	**Obj**	**CPU**	**Obj**	**CPU**	**LB**	$Gap(\%)$
A1-25-1	25	5	7	5	4.42	11.24	-	TL	4.01	9.28
A1-25-2					4.21	64.47	-	TL	4.20	0.24
A1-25-3					4.25	TL	-	TL	3.60	15.29
A1-25-4					3.91	15.11	-	TL	3.89	0.51
A1-25-5					4.21	4.98	-	TL	3.99	5.23
A1-30-1	30	5	8	6	4.31	44.59	-	TL	4.20	2.55
A1-30-2					3.56	28.57	-	TL	3.51	1.40
A1-30-3					4.89	236.27	-	TL	4.80	1.84
A1-30-4					4.48	2.74	-	TL	4.39	2.01
A1-30-5					3.66	50.79	-	TL	3.60	1.64
A1-35-1	35	5	8	6	4.61	166.97	-	TL	4.55	1.30
A1-35-2					5.86	384.00	-	TL	5.59	4.61
A1-35-3					4.95	66.00	-	TL	4.87	1.62
A1-35-4					4.91	52.98	-	TL	4.85	1.22
A1-35-5					5.57	39.88	-	TL	5.51	1.08
A1-40-1	40	5	8	6	7.14	235.75	-	TL	6.89	3.50
A1-40-2					5.65	181.48	-	TL	5.61	0.71
A1-40-3					7.01	TL	-	TL	6.82	2.71
A1-40-4					6.66	TL	-	TL	6.20	6.91
A1-40-5					6.18	220.52	-	TL	5.95	3.72
Average						97.96				2.06

Table 2. Computational results for instances under marginal setting.

					VND-M		**AOA**			
Instance	$\lvert C \rvert$	$\lvert D \rvert$	N_d	N_{BH}^f	**Obj**	**CPU**	**Obj**	**CPU**	**LB**	$Gap(\%)$
A1-10-1	10	5	2	2	1.64	35.67	1.64	2.85		0.00
A1-10-2					1.56	7.30	1.56	2.68		0.00
A1-10-3					1.40	3.02	1.40	1.87		0.00
A1-10-4					1.13	2.26	1.13	1.96		0.00
A1-10-5					1.86	3.31	1.86	2.81		0.00
A1-15-1	15	5	3	2	2.05	2.09	2.05	35.10		0.00
A1-15-2					2.25	3.24	2.25	30.17		0.00
A1-15-3					2.13	5.30	2.13	32.12		0.00
A1-15-4					2.35	3.94	2.35	41.78		0.00
A1-15-5					1.87	3.31	1.87	51.74		0.00
A1-20-1	20	5	4	3	2.59	5.07	2.59	305.30		0.00
A1-20-2					2.56	8.78	2.56	440.93		0.00
A1-20-3					2.68	10.75	2.68	425.31		0.00
A1-20-4					2.49	6.50	-	TL	2.19	12.16
A1-20-5					2.83	9.38	-	TL	2.31	18.49
A1-25-1	25	5	7	5	3.41	1.71	-	TL	3.00	12.05
A1-25-2					4.10	21.79	-	TL	3.95	3.69
A1-25-3					3.62	7.27	-	TL	3.01	16.97
A1-25-4					3.47	3.25	-	TL	3.20	7.73
A1-25-5					3.63	27.93	-	TL	3.23	11.10

Table 2. *Cont.*

					VND-M		AOA			
Instance	$\lvert C \rvert$	$\lvert D \rvert$	N_d	N_{BH}^{f}	**Obj**	**CPU**	**Obj**	**CPU**	**LB**	$Gap(\%)$
A1-30-1					3.55	2.46	-	TL	3.29	7.11
A1-30-2	30	5	8	6	3.49	2.20	-	TL	3.22	7.66
A1-30-3					3.92	3.64	-	TL	3.74	4.57
A1-30-4					3.65	2.26	-	TL	3.20	12.37
A1-30-5					4.03	2.45	-	TL	3.27	18.91
A1-35-1					4.13	155.01	-	TL	4.01	2.76
A1-35-2	35	5	8	6	5.15	107.19	-	TL	5.12	0.54
A1-35-3					4.38	80.72	-	TL	4.33	1.01
A1-35-4					4.95	66.00	-	TL	3.46	30.14
A1-35-5					4.78	30.97	-	TL	3.70	22.61
A1-40-1					4.87	107.53	-	TL	4.78	1.87
A1-40-2	40	5	8	6	5.22	160.56	-	TL	5.17	0.98
A1-40-3					4.72	112.01	-	TL	4.71	0.28
A1-40-4					5.08	153.48	-	TL	4.95	2.68
A1-40-5					4.96	86.51	-	TL	4.87	1.95
Average						35.57				5.65

6. Conclusions

In this paper, we have computationally addressed a new emerging problem in drone-aided last-mile logistics modeled as a variant of the location-routing problem, called DRP-ShaBH. The main characteristics of this new problem are the drones intrinsic properties, as the limited battery capacity, payload, and the weather-related uncertainties that lead to uncertain flight duration.

The model introduced can be solved to optimality for instances with limited size up to 20 customers. For larger instances, we have proposed a matheuristic approach based on a variable neighborhood descent heuristic and cutting planes. The efficiency of the proposed solution approach is investigated on instances with up to 40 customers.

There are several fruitful directions for future research based on the present study. To have a clear idea about the solution approach efficiency, we may investigate the performance of the proposed matheuristic compared with the state-of-the-art metaheuristics especially for large instances. In addition, it would be interesting to study the DRP-ShaBH in the multi-objective context where the stakeholder aims to optimize multiple objectives simultaneously, for example a customer-centric objective such as the total waiting times and a server-centric criterion such as the total system cost. Last but not least, the tariff definition for beehive usage is another interesting topic that is the focus of pricing problems in bi-level context. This requires the reformulation of the current problem as a bi-level model to capture the interaction between the beehive owner and the logistics manager.

Author Contributions: Conceptualization, M.E.B. and S.K.; methodology, M.E.B. and S.K.; software, S.K.; validation, M.E.B. and S.K.; formal analysis, M.E.B. and S.K.; investigation, M.E.B. and S.K.; resources, M.E.B. and S.K.; data curation, M.E.B. and S.K.; writing—original draft preparation, M.E.B. and S.K.; writing—review and editing, M.E.B. and S.K.; visualization, M.E.B. and S.K.; supervision, M.E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A

Table A1. Notation for the mathematical model of DRP-ShaBH .

Sets	
C	Set of customers
D	Set of potential BHs
D'	Set of two auxiliary BHs $(D' = \{0, m+1\})$
$L = \{1, \cdots, N\}$	Set of positions, indexed by r
Parameters	
N_d	Number of drones
d_i	Demand of customer i
Q	Drone payload capacity
k	Energy consumption-related parameter $k = \sqrt{\frac{g^3}{2\rho\xi h}}$
W	Drone frame mass
M	Drone battery mass
g	Force applied by gravity per unit mass
ρ	Fluid density of air
ξ	Area of spinning blade disc
h	Number of rotors
Γ	Total mass of an unloaded drone $(W + M)$
N_{BH}^f	Maximum number of drones that BH f can host
N_s	Maximum number of BHs to use
t_{ij}	Travel time along arc (i, j)
B	Drone battery charge
M'_{fj}, M''_{ij}	BigM numbers
Decision variables	
x_i^r	Binary variable that takes 1 if customer i is visited in position r; otherwise, 0
y_{ij}^r	Binary variable that takes 1 if customer j is visited right after customer i in position r and there are exactly $r-1$ customers left to be visited after; otherwise, 0
w_{fj}^r	Binary variable that takes 1 if customer j is visited right after BH f and there are exactly $r-1$, $(r \neq 1)$ customers left to be visited after; otherwise, 0
v_{fj}	Total load carried by the drone traveling over arc (f, j) upon departure from BH f
u_{ij}	Total load carried by the drone traveling over arc (i, j) after serving customer i
z_f	Binary variable takes 1 if BH f is used; otherwise, 0
e_i	The accumulated energy consumption upon arrival at node $i \in C \cup D$

References

1. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 70–85. [CrossRef]
2. Choi, Y.; Schonfeld, P.M. Optimization of multi-package drone deliveries considering battery capacity. In Proceedings of the 96th Annual Meeting of the Transportation Research Board, Washington, DC, USA, 8–12 January 2017; pp. 8–12.
3. Troudi, A.; Addouche, S.A.; Dellagi, S.; Mhamedi, A.E. Sizing of the drone delivery fleet considering energy autonomy. *Sustainability* **2018**, *10*, 3344. [CrossRef]
4. Aurambout, J.P.; Gkoumas, K.; Ciuffo, B. Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities. *Eur. Transp. Res. Rev.* **2019**, *11*, 1–21. [CrossRef]
5. Bruni, M.; Khodaparasti, S. Addressing the Challenges of Last-mile: The Drone Routing Problem with Shared Fulfillment Centers. In Proceedings of the ICORES 2022: 11th International Conference on Operations Research and Enterprise Systems, online, 3–5 February 2022; pp. 362–367. Available online: https://www.unical.it/storage/publications/149141/ (accessed on 8 August 2022).
6. Cheng, C.; Adulyasak, Y.; Rousseau, L.M. Drone routing with energy function: Formulation and exact algorithm. *Transp. Res. Part B Methodol.* **2020**, *139*, 364–387. [CrossRef]
7. Kim, S.; Kwak, J.H.; Oh, B.; Lee, D.H.; Lee, D. An Optimal Routing Algorithm for Unmanned Aerial Vehicles. *Sensors* **2021**, *21*, 1219. [CrossRef]
8. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
9. Kim, S.J.; Lim, G.J.; Cho, J.; Côté, M.J. Drone-aided healthcare services for patients with chronic diseases in rural areas. *J. Intell. Robot. Syst.* **2017**, *88*, 163–180. [CrossRef]

10. Torabbeigi, M.; Lim, G.J.; Kim, S.J. Drone delivery scheduling optimization considering payload-induced battery consumption rates. *J. Intell. Robot. Syst.* **2020**, *97*, 471–487. [CrossRef]
11. Liu, Y.; Liu, Z.; Shi, J.; Wu, G.; Chen, C. Optimization of base location and patrol routes for unmanned aerial vehicles in border intelligence, surveillance, and reconnaissance. *J. Adv. Transp.* **2019**, *2019*, 9063232. [CrossRef]
12. Yakıcı, E. Solving location and routing problem for UAVs. *Comput. Ind. Eng.* **2016**, *102*, 294–301. [CrossRef]
13. Li, X.; Li, P.; Zhao, Y.; Zhang, L.; Dong, Y. A Hybrid Large Neighborhood Search Algorithm for Solving the Multi Depot UAV Swarm Routing Problem. *IEEE Access* **2021**, *9*, 104115–104126. [CrossRef]
14. Grogan, S.; Pellerin, R.; Gamache, M. Using tornado-related weather data to route unmanned aerial vehicles to locate damage and victims. *Spectrum* **2021**, *43*, 905–939. [CrossRef]
15. Bruni, M.E.; Khodaparasti, S.; Demeulemeester, E. The distributionally robust machine scheduling problem with job selection and sequence-dependent setup times. *Comput. Oper. Res.* **2020**, *123*, 105017. [CrossRef]
16. Nucamendi-Guillén, S.; Martínez-Salazar, I.; Angel-Bello, F.; Moreno-Vega, J.M. A mixed integer formulation and an efficient metaheuristic procedure for the k-travelling repairmen problem. *J. Oper. Res. Soc.* **2016**, *67*, 1121–1134. [CrossRef]
17. Nucamendi-Guillén, S.; Martínez-Salazar, I.; Khodaparasti, S.; Bruni, M.E. New formulations and solution approaches for the latency location routing problem. *Comput. Oper. Res.* **2022**, *143*, 105767. [CrossRef]
18. Bruni, M.E.; Khodaparasti, S.; Martínez-Salazar, I.; Nucamendi-Guillén, S. The multi-depot k-traveling repairman problem. *Optim. Lett.* **2022** , 1–29. [CrossRef]
19. Bruni, M.E.; Beraldi, P.; Khodaparasti, S. A hybrid reactive GRASP heuristic for the risk-averse k-traveling repairman problem with profits. *Comput. Oper. Res.* **2020**, *115*, 104854. [CrossRef]
20. Nucamendi-Guillén, S.; Angel-Bello, F.; Martínez-Salazar, I.; Cordero-Franco, A.E. The cumulative capacitated vehicle routing problem: New formulations and iterated greedy algorithms. *Expert Syst. Appl.* **2018**, *113*, 315–327. [CrossRef]
21. Ben-Tal, A.; Ghaoui, L.E.; Nemirovski, A. *Robust Optimization*; Princeton University Press: Princeton, NY, USA , 2009. [CrossRef]
22. Ben-Tal, A.; Den Hertog, D.; Vial, J.P. Deriving robust counterparts of nonlinear uncertain inequalities. *Math. Program.* **2015**, *149*, 265–299. [CrossRef]
23. Balas, E.; Jeroslow, R. Canonical cuts on the unit hypercube. *SIAM J. Appl. Math.* **1972**, *23*, 61–69. [CrossRef]