applied sciences

MDPI

Article

Multiobjective Scheduling of Logistics UAVs Based on Variable Neighborhood Search †

Yixuan Li [1], Xiaoxiang Yuan [1,2], Jie Zhu [1], Haiping Huang [1,2,*] and Min Wu [1,2]

1 School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; yixuanlitj@163.com (Y.L.); Y1569140217@163.com (X.Y.); zhujie@njupt.edu.cn (J.Z.); wumin@njupt.edu.cn (M.W.)

2 Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, China

* Correspondence: hhp@njupt.edu.cn

† This paper is an extended version of paper published in 10th International Symposium, PAAP 2019, held in Guangzhou, China, 12–14 December 2019. Yixuan Li, Xiaoxiang Yuan, Jie Zhu, Haiping Huang, Min Wu, 2019.

Received: 9 April 2020; Accepted: 19 May 2020; Published: 21 May 2020

Abstract: This study focuses on the issue of logistics Unmanned Aerial Vehicle (UAV) distribution in urban environment and an automatic delivery system to support the delivery of packages. It can effectively integrate existing facilities and be easily deployed. There is a scheduling problem in this system with multiple UAVs and multiple flights. We manage to optimize the two objectives of customer satisfaction and total completion time. The scheduling problem is formulated to a Mixed Integer Linear Programming (MILP), and we propose a multiple objectives decision-making method. A special encoding method suitable for the small scale problem is presented and Variable Neighborhood Search (VNS) algorithm framework is used to generate the approximate optimal solution for this problem. In experiments, we calibrate the important parameter and analyze the robustness of the algorithm. The experimental results show that the proposed algorithms are efficient for this problem.

Keywords: logistics; unmanned aerial vehicle; simulated annealing; variable neighborhood search

1. Introduction

With the popularity of the Internet, more and more people choose online shopping. A large number of online orders have brought great pressure to the express industry. Therefore, the transformation of the production pattern needs to be carried out for traditional delivery companies. As one of the autonomous things, UAVs are considered to be the top strategic technology for logistics industry [1]. Both governments and enterprises have been heavily investing in the development of Unmanned Aerial Vehicles (UAVs) [2]. Recently, with the maturity of UAV technology, it has been widely used in various fields such as communication platforms, precision agriculture, surveillance and monitoring, and cargo delivery; the UAV-assisted logistics systems have drawn significant research interests [3]. It is worth noting that in different application scenarios, the number of UAVs included in the UAV-assisted system is different (see Figure 1 for an illustration). Teams of UAVs can be dispatched, for instance, as providing service to disaster-affected areas, detecting environment conditions or as an aerial sensor network, collecting data in large areas. UAV logistics is also an emerging application scenario, and most commercial UAV logistics auxiliary systems are exactly multi UAV cooperative systems, whose execution efficiency depends on the number of UAVs. It uses UAVs to improve the efficiency of the logistics distribution and schedule, reduce operating costs, and optimize the link of terminal distribution.

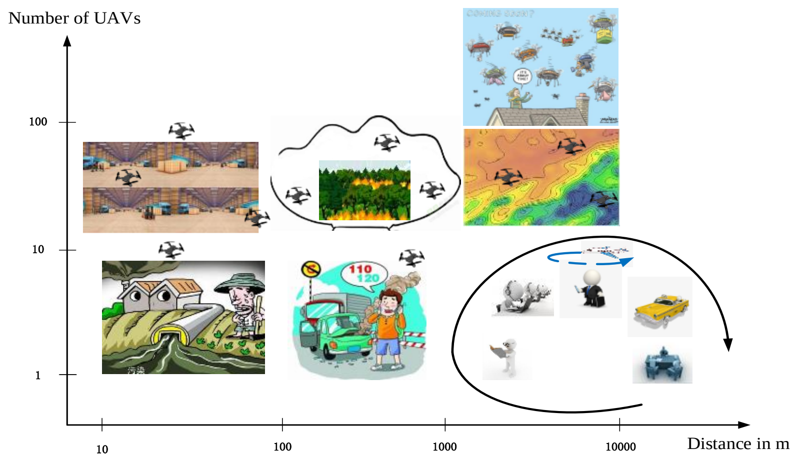

Figure 1. Application areas over a range of distance vs. number of Unmanned Aerial Vehicles (UAVs).

In this paper, we mainly present an automatic delivery system for logistics UAVs in urban environment. The algorithm frameworks based on Variable Neighborhood Search (VNS) are proposed to solve the small scale scheduling problem of this system, and we consider several important optimization objectives. In the traditional logistics industry, all packages that need to be delivered are stored in the trans-shipment depot. These packages in the trans-shipment depot are classified by their destinations. Multiple couriers deliver these packages, and each courier is responsible for the delivery of packages within a certain area. In the above system, a group of UAVs are used to replace a Courier. Some autonomous express cabinets are set up near the package's destinations. These UAVs drop some packages from the trans-shipment depot into express cabinets according to the task scheduling results. The aforementioned UAVs and express cabinets are all connected to the Internet. The execution status of the system is obtained by the dispatch center in real time. The dispatch center performs the task scheduling at intervals and reports the status of the system to the relevant staff. The execution process of the system is shown in Figure 2: the dispatch center uploads the data to the Internet. The trans-shipment depot S_0, express cabinets from S_1 to S_9, and UAVs from u_1 to u_5 all connected with the Internet. Packages are sorted in S_0 according to instructions downloaded from the Internet; UAVs and express cabinets can realize reasonable dispatching through the interaction data from the dispatch center.

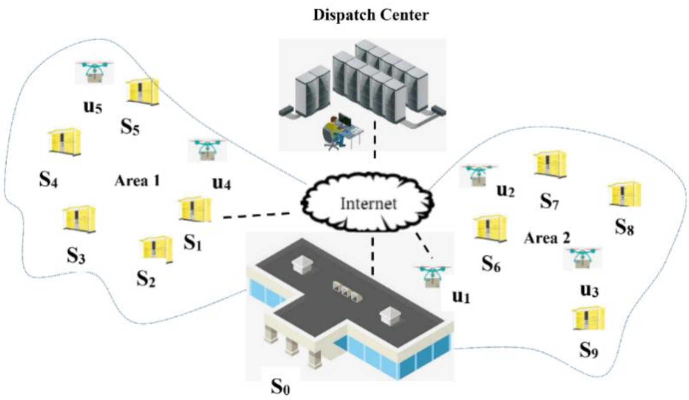

Figure 2. The execution process of the delivery system.

The scheduling problem of logistics UAVs is a special kind of Unmanned Aerial Vehicle routing problem (UAVRP). Similar to the traditional vehicle routing problem (VRP), the UAVRP adds some new features such as limitations of flight time and load capacity. VRP is well known as an NP-hard problem. As a variant of VRP, the scheduling problem of logistics UAVs is also NP-hard. The VNS algorithm framework is used to address the computational intractability and search for the approximate optimal solution of the proposed problem.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Section 3 describes and formalizes the problem under study. An algorithm framework is proposed for the considered problem in Section 4. Section 5 evaluates the performance of the proposal under different workload scenarios followed by conclusions in Section 6.

2. Related Works and Contributions

The traditional logistics scheduling problem is a special Vehicle Routing Problem (VRP). A great number of studies have focused on the field of VRP. Dantzig and Ramser [4] first introduced the truck dispatching problem and proposed a procedure to search for the near-optimal solution. Chao et al. [5] considered the VRP with simultaneous pickup–delivery and time windows, and a parallel Simulated Annealing algorithm was proposed for the problem. Jiahai et al. [6] present a multiobjective and multidepot VRP. They developed a two-stage multiobjective evolutionary algorithm to deal with this problem. Li et al. [7] constructed the multiobjective VRP mathematical model of military logistics in wartime and improved the NSGA II algorithm by the introduction of Greedy algorithm, in order to solve the proposed multiobjective VRP problem. Wang et al. [8] proposed the improved Ant Colony Algorithm to solve the classical Vehicle Routing Problem, where experimental analysis showed that the proposed algorithm was better than the traditional Ant Colony Algorithm in optimum value and the rate of convergence. Hosseinabadi et al. [9] proposed a variable neighborhood search algorithm to solve a kind of capacitated location-routing problem (CLRP). Ziyao Li [10] put forward an improved ACO algorithm and built a path model to solve practical vehicle route problem in the emergency rescue event. Jun Zheng [11] set up a vehicle routing problem (VRP) model with multiple fuzzy time windows, based on time-varying traffic flow. They also proposed an improved ITO algorithm to reduce the distribution cost and consumer dissatisfaction.

As a special kind of VRP, UAV routing and scheduling problem has been applied in various scenarios. Dorling et al. [12] proposed two multitrip VRPs for UAV delivery that minimizes cost subject to a delivery time limit or minimizes the overall delivery time subject to a budget constraint. Kim and Morrison [13] present a scenario in which UAVs were needed to provide continuous service and docking locations. They used a scheduling method to solve this problem. Zeng et al. [14] developed a nonlinear model to schedule UAV resources to the battlefield. Denis et al. [15] proposed the network-centric multiagent system for real-time task scheduling of the UAV group, which achieved coordination and control of several UAVs in the group in order to perform joint tasks. Ahmadian et al. [16] proposed a mixed-integer programming model to determine the optimal collision-free schedule for multiple UAVs. The approach reduced the possibility of collision between UAVs by creating a gap between their arrival intervals to each node. Ahmed et al. [17] proposed an efficient algorithm based on successive convex approximation and classical Dinkelbach method to reduce UAV flight energy consumption. Xu et al. [18] suggested a trajectory planning method for two-UAV cooperative target locating. Ghazzai et al. [19] designed a generic scheduling framework of a fleet of micro UAVs. The framework employed multiple UAVs in sequential and parallel ways to perform a mission over a long period of time. A mixed integer linear programming (MILP) problem aiming at minimizing the total energy consumption was formulated after a series of linearization steps. The generic UAV scheduling framework can be applied in multiple domains comprising short and/or long-term UAV missions while ensuring uninterrupted service.

Although UAVs can be used in many scenarios, there are few studies on the application of logistics UAV in urban environment. The scheduling method proposed in VRP related research cannot be directly applied to UAV logistics scheduling. The traditional VRP Problem includes customer set and vehicle set, and the main constraints include customer demand and actual road constraints, focusing on path planning. Whereas in UAVRP problems, UAV's operation track is relatively more flexible, smaller in size and greater in number. UAV-based scheduling problems also have some specific constraints, such as the power and capacity constraints, which need to be considered comprehensively for package loading and path planning. Therefore, a suitable logistics UAV scheduling scheme is essential to replace the work of couriers.

We refer to the methods presented in the above literates, and the main contributions of the present study are summarized as follows. (i) We first present the automatic delivery system with logistics UAVs to improve the efficiency of the logistics distribution and reduce operating costs. (ii) Mixed Integer Linear Programming (MILP) is used to describe the scheduling problem of the system. (iii) We propose a special coding method for this small-scale problem. (iv) The algorithm frameworks based on VNS is presented, as an extension of the previous version [20], to search for approximate optimal solution and make the decision between multiple objectives.

3. Problem Description and Model

Packages in the trans-shipment depot need to be delivered to multiple areas. In the traditional logistics delivery scenario, one courier is responsible for a certain small area. Many couriers work together to complete the delivery of all packages in the trans-shipment depot. Multiple UAVs are being used to replace one courier. The above mentioned automatic delivery system for logistics UAVs is used to support the execution of the above workflow. Package tasks for one courier are scheduled by this system respectively.

In order to deploy the UAV delivery system on existing resources, the following components need to be used in logistics distribution. (i) Some package loading equipment in the trans-shipment depot, they are responsible for loading packages into UAVs and replacing UAV's battery for their next flight. (ii) Each of these UAVs will complete the delivery according to the route specified by the dispatch center. These package tasks in a certain area require multiple UAVs to collaborate. Furthermore, these package tasks are usually completed through multiple flights. (iii) Some express cabinets are placed near these package destinations. The express cabinet supports UAVs to automatically place the package inside it, and it allows customers to pick up their packages at any time after delivery. (iv) The dispatch center performs resource scheduling in units of areas. It updates the status of the transit warehouse, UAVs and express cabinets in real time, and notifies the corresponding staff in time when the system fails.

We consider the scheduling problem of the system in a certain area. Since one courier has to deliver a certain number of packages in a day, there is a small-scale scheduling problem in the specific area that needs to be addressed. The search ability of three algorithm frameworks proposed in this section are adequate for this problem. The customer satisfaction and the total completion time, which are critical to the problem, are selected as the two optimization objectives for this problem. The scheduling problem is modeled as a MILP. Notations are summarized in Table 1.

Table 1. Notations description.

Notation	Description
h	Number of UAVs
u_k	The k^{th} UAV
$U = \{u_1, u_2, \ldots, u_h\}$	Set of UAVs
e	Maximum flight time of UAV
c	Load capacity of UAV
v	Average speed of UAV
t^d	Time for UAV to load packages in s_0
t^c	Time for UAV to unload packages into $s_1, \ldots, s_m$
n	Number of packages
p_i	The i^{th} package
$P = \{p_1, p_2, \ldots, p_n\}$	Set of packages
w_i	Weight of p_i
l_i	Destination of p_i
t_i^0	Best delivery time of p_i
t_i	Actual delivery time of p_i
g	Maximum weight of packages

Table 1. *Cont.*

Notation	Description
G	Total weight of packages
m	Number of express cabinet
s_j	The j^{th} station
$S = \{s_0, s_1, \ldots, s_m\}$	Set of stations
$d = \left(s_j, s_{j'}\right)$	Distance between s_j and $s_{j'}$
r	Radius of area
t	Initial temperature of SA framework
S^{uav}	The scheduling scheme
T_k	Number of flights for u_k
$T_{k,f}$	Number of packages delivered by u_k's f^{th} flight
$p_{k,f,b}$	Number of b^{th} package on u_k's f^{th} flight
$l_{k,f,b}$	Destination of b^{th} package on u_k's f^{th} flight
$t_{k,f}$	Start time of u_k's f^{th} flight
$q_{k,f}$	Task execution duration of u_k's f^{th} flight
t_k^e	Time for u_k to complete all its tasks
A	Customer satisfaction
B	Total completion time
B^{max}	Maximum total completion time

There is a small scale problem. All packages that a courier needs to deliver are represented by $P = \{p_1, p_2, \ldots, p_n\}$. A set of stations $S = \{s_0\} \cup \{s_1, s_2, \ldots, s_m\}$ are set up in the scenario, where s_0 is the trans-shipment depot and the set of $\{s_1, s_2, \ldots, s_m\}$ represents these express cabinets in the area. $l_{k,f,b}$ represents the destination of the b^{th} package delivered by u_k on f^{th} flight. We determine a minimum circle that covers all these stations, and r is the radius of the circle. According to customer requirements, the optimal delivery time t_i^0 is set for each package, and the actual delivery time of the package is represented as t_i. A set of UAVs $U = \{u_1, u_2, \ldots, u_h\}$ is used to accomplish these package tasks. S^{uav} is the solution of the task scheduling for all UAVs. It contains the scheduling results of multiple flights and is an intuitive representation of the scheduling scheme. Customer satisfaction A and total completion time B are two optimization objectives.

Based on the above descriptions and notations, the following mathematical model is established.

$$Maximum\ A = \sum_{i=1}^{n} \max\left(0, t_i - t_i^0\right) \tag{1}$$

$$Minimum\ B = \max_{1 \le k \le n} t_k^e \tag{2}$$

Subject to:

$$\max_{1 \le i \le n} w_i \le c, \forall k, f, \tag{3}$$

$$\textstyle\sum_{b=1}^{T_{k,f}} w_{p_{k,f,b}} \le c, \forall k, f, \tag{4}$$

$$\textstyle\sum_{b=0}^{T_{k,f}-1} d\left(l_{k,f,b}, l_{k,f,b+1}\right) + d\left(l_{k,f,T_{k,f}}, s_0\right) \le v \times e, \forall k, f, \tag{5}$$

$$t_k^e = t_{k,T_k} + q_{k,T_k}, \forall k, \tag{6}$$

$$t_{k,f} = \begin{cases} 0, & f = 1 \\ t_{k,f-1} + q_{k,f-1}, & f \ge 2 \end{cases}, \forall k, \tag{7}$$

$$q_{k,f} = \textstyle\sum_{b=0}^{T_{k,f}-1} d\left(l_{k,f,b}, l_{k,f,b+1}\right) + d\left(l_{k,f,T_{k,f}}, s_0\right) + t^d + T_{k,f} \times t^c, \forall k, f, \tag{8}$$

$$p_{k,f,b} = \textstyle\sum_{i=1}^{n} i \times x_{k,f,b,i}, \forall k, f, b, \tag{9}$$

$$\textstyle\sum_{i=1}^{n} x_{k,f,b,i} = 1, \forall k, f, b, \tag{10}$$

$$\sum_{k=1}^{h}\sum_{f=1}^{T_k}\sum_{b=1}^{T_{k,f}} x_{k,f,b,i} = 1, \forall i, \tag{11}$$

$$x_{k,f,b,i} \in \{0, 1\},$$

Equations (1) and (2) describe how optimization objectives are calculated. Customer satisfaction A is significantly more important than total completion time B. Constraint (3) is the weight limit of packages. Constraints (4–5) ensure the maximum payload and flight distance of UAVs. Equations (6)–(9) present the calculation methods of the relevant parameters. $x_{k,f,b,i}$ is a decision variable, taking 1 if b^{th} package on u_k's f^{th} flight is p_i, taking 0 otherwise. That taking 1 means loading the package, and taking 0 means unloading the package. Constraints associated with decision variables are 10–11.

4. Proposed Algorithm

In this section, the proposed algorithm will be described in detail. The key problem of the three algorithm frameworks proposed is to determine an appropriate encoding method for the small scale problem. A new encoding method is introduced, as the scheduling scheme $S^{uav} = \{< 5, 2, 3 >, < 4, 1 >\}$. It consists of two vectors, and each of the vectors determines the order in which a UAV delivers packages. For example, the distribution scheme of the above example S^{uav} is that UAV u_1 delivers packages p_5, p_2, p_3, and UAV u_2 delivers packages p_4, p_1, respectively. Each UAV completes its package tasks through multiple flights. The number of flights is determined by the load capacity of the UAV c. The encoding method is essentially a sequence of packages. It can accurately assign each package task and find a better solution when the search space is small.

As our algorithm is a hybrid of Initial Solution Generation Algorithm (ISG) and heuristic algorithms, we first provide a brief description about ISG algorithm as follows.

Once a better initial solution can be found to participate in the iterations, more satisfactory efficiency will be achieved. Since the initial solution to the scheduling scheme S^{uav} is essentially a sort of packages, ISG is used to generate it. The shortest path through all express cabinets is calculated by a simple Simulated Annealing (SA) algorithm (Lines 1–15). According to the shortest path and the best delivery time t_i^0, the elements in S^{init} are ordered as the initial solution (Lines 16–18).

Initial Solution Generation Algorithm *ISG*()	
1	$L \leftarrow \{s_1, s_2, \ldots, s_m\}$;/* L records the shortest path that traverses $\{s_1, s_2, \ldots, s_m\}$ */
2	$L^{current} \leftarrow L$;
2	$S^{init} \leftarrow P$;
3	$t \leftarrow 80$;/* t represents the initial temperature */
4	**for** $\varphi = 80$; $\varphi > 1$; $\varphi \leftarrow 0.9 \times \varphi$ **do**
5	**for** $\tau = 0$; $\tau < 500$; $\tau \leftarrow \tau + 1$ **do**
6	Swapping two random disjoint subsequences of $L^{current}$ to generate L^{new};
7	**if** $R(L^{new}) < R(L^{current})$ **then**/* $R(L^{new})$ is the path length of L^{new} */
8	$L^{current} \leftarrow L^{new}$;
9	**if** $R(L^{current}) < R(L)$ **then**
10	$L \leftarrow L^{current}$;
11	**else**
12	$p \leftarrow e^{\frac{-1 \times R(L\prime)}{0.8 \times \varphi}}$; /* Accept an inferior solution with the probability p */
13	A random number $p\prime$ is generated between 0 and 1;
14	**if** $p\prime < p$ **then**
15	$L^{current} \leftarrow L^{new}$;
16	The packages in S^{init} are sorted according to L;
17	For packages with the same destination in S^{init}, sort by t_i^0;
18	**return** S^{init};

4.1. Local Search Algorithm Framework

Local Search algorithm (*LS*) is applied to improve the initial solution. A new solution in the local search is generated by random swapping. The approximate optimal solution to the scheduling scheme S^{uav} is calculated by iteration (Lines 2–6).

Local Search algorithm framework $LS()$

1 $S^{uav} \leftarrow ISG()$;
2 **While** not Termination Criterion **do**
3   Generation of candidate neighbor S^{uav};
4   **if** $Fit(S^{new}) < Fit(S^{uav})$ **then**
5             $S^{uav} \leftarrow S^{new}$;
6   **return** S^{uav};

where *Fit* function is the fitness of the solution defined as a linear weighted sum:

$$Fit = \omega A + (1 - \omega)B \; (\omega \in (0, 1)) \tag{12}$$

4.2. Simulated Annealing Algorithm Framework

As our algorithm is a hybrid of Initial Solution Generation Algorithm (ISG) and heuristic algorithms, we first provide a brief description about ISG algorithm as follows. Simulated Annealing algorithm is a greedy algorithm. It is derived from the principle of solid annealing. SA accepts an inferior solution with the certain probability, and this probability varies with the temperature. The SA algorithm framework is used to compute the integrated objective C. We obtain the approximate optimal solution by the Simulated Annealing algorithm framework. ISG is used to generate the initial solution (Line 1). The approximate optimal solution to the scheduling scheme S^{uav} is calculated by iteration (Lines 2–14). In the iterative process, the global optimal solution is recorded (Lines 8–9) and an inferior solution is accepted with probability p (Lines 11–14).

Simulated Annealing algorithm framework $SA()$

1 $S^{uav} \leftarrow ISG()$;
2 $S^{current} \leftarrow S^{uav}$;
3 **for** $\varphi = t$; $\varphi > 1$; $\varphi \leftarrow 0.9 \times \varphi$ **do**
4   **for** $\tau = 0$; $\tau < 1000$; $\tau \leftarrow \tau + 1$ **do**
5   Swapping two random disjoint subsequences of $S^{current}$ to generate S^{new};
6   **if** $Fit(S^{new}) < Fit\left(S^{current}\right)$ **then**/* $Fit(S^{new})$ is the fitness of S^{new} */
7             $S^{current} \leftarrow S^{new}$;
8   **if** $Fit\left(S^{current}\right) < Fit(S^{uav})$ **then**
9                 $S^{uav} \leftarrow S^{current}$;
10   **else**
11             $p \leftarrow e^{\frac{-1 \times R(L\prime)}{0.8 \times \varphi}}$; /* Accept an inferior solution with the probability p */
12   A random number $p\prime$ is generated between 0 and 1;
13   **if** $p\prime < p$ **then**
14                 $S^{current} \leftarrow S^{new}$;
15   **return** S^{uav};

4.3. Variable Neighborhood Search Algorithm Framework

Variable Neighborhood Search (VNS) is a metaheuristic with the systematic change of neighborhood in a search. VNS was first proposed for solving the traveling salesman problem and its effectiveness is illustrated in [21]. Since TSP can be taken as a subproblem of the considered problem in this paper, we chose VNS to improve our previous work. VNS can escape from the local optimum by restructuring the neighborhood. Different neighborhoods supply different candidate solutions, thus it is possible to find a better solution.

ISG is used to generate the initial solution (Line 2). The approximate optimal solution to the scheduling scheme S^{uav} is calculated by iteration (Lines 4–12). In the iterative process, the neighborhood is obtained by shaking (Line 5), the local optimum $S^{uav\prime\prime}$ is calculated by iteration (Lines 6–7), and the global optimal solution is recorded (Lines 8–9).

Variable Neighborhood Search algorithm framework $VNS()$

1 Select the set of neighborhood structures $N_k,\ k = 1, \ldots, k_{max}$;
2 $S^{uav} \leftarrow ISG()$;
3 Set k=1;
4 **While** $k \le k_{max}$ **do**
5 Generate a point $S^{uav\prime}$ randomly from N_k (S^{uav}); /* Shaking Step*/
6 **While** not Termination Criterion **do**
/* Obtain local optimum $S^{uav\prime\prime}$ from an initial solution $S^{uav\prime}$*/
7 Generation of candidate neighbor $S^{uav\prime\prime}$;
8 **if** $Fit(S^{uav\prime\prime}) < Fit(S^{uav})$ **then**
9 $S^{uav} \leftarrow S^{uav\prime\prime}, k = 1$;
10 **else**
11 $k = k + 1$;
12 **return** S^{uav};

As our algorithm is a hybrid of Initial Solution Generation Algorithm (ISG) and heuristic algorithms, we first provide a brief description about ISG algorithm as follows.

4.4. Time Complexity Analysis

(1) Since the time complexity of the ISG algorithm is related to the number of packages n, the number of UAVs h and the number of stations $|S|$, it can be denoted as $O(n + h * |S|)$.

(2) The time complexity of the SA algorithm is related to the ISG algorithm, the initial temperature t and the number of iterations τ. Therefore, the time complexity is $O\left(\left(\log_{0.9} \frac{1}{t}\right) * \tau * (n + h * |S|)\right)$.

(3) The time complexity of the VNS algorithm is related to the ISG algorithm, the number of neighbors $|N_k|$, and the number of neighbors I_k per exploration. Therefore, the time complexity is $O(\sum_{k=1}^{k=k_{max}} I_k * |N_k| * (n + h * |S|))$.

5. Simulated Experiments

In this section, numerical results are reported. We first calibrate the important parameter initial temperature t of the SA algorithm framework. Then, the robustness of the algorithm is analyzed for all instance parameters. All experiments are coded in Java and run on a PC with an Intel(R) Core(TM) i5-7500 CPU @3.40Ghz, 8GB of RAM. The version of Integrated Development Environment (IDE) is eclipse Jee Oxygen April 2018 ×64. Relative Percentage Deviation (RPD) is used to evaluate the performance of the algorithm. The value of RPD reflects the difference between one solution and the optimal solution in the same instance. The better algorithm corresponds to a smaller PRD value. By observing the change of the value, the difference between the optimization effects of algorithms can be clearly observed.

$$\mathrm{RPD} = \frac{Fit(C) - Fit(C\prime)}{Fit(C\prime)}, \tag{13}$$

Equation (13) depicts the calculation of the RPD value. In Equation (13), $Fit(C\prime)$ represents the optimal fitness for the same instances.

5.1. Parameter Calibration

The initial temperature t is critical to the performance of the SA algorithm framework. We use a large number of instances to determine the value of t. To the best of our knowledge, no uniform testing benchmark is available for the considered problem. So, we combined some practical scenarios in logistics transportation to generate test instances, such as [22] and [23].

Parameters are designed as follows. (i) We consider the regional parameters with the area radius $r \in \{1\text{ km}, 2\text{ km}, 3\text{ km}\}$ and the number of stations $m \in \{5, 10, 15\}$. (ii) The weight of each package w_i is generated randomly in interval $[0.05,\ g]$, and the parameter g represents the maximum weight of all packages. The package related parameters are set as $g \in \{3\text{ kg}, 4\text{ kg}, 5\text{ kg}\}$ and $G \in \{80, 100, 120\}$. (iii) The number of UAVs $h \in \{3, 4, 5\}$. The special logistic UAV is selected with maximum flight time $e = 0.67$ h, load capacity $c = 5$ kg, and average speed $v = 48$ km/h.

There are $3 \times 3 \times 3 \times 3 \times 3 = 243$ instance combinations. For each instance combination, we compare the algorithm performance of five different values of initial temperature t of SA framework. Therefore, there are $3 \times 3 \times 3 \times 3 \times 3 \times 5 = 1215$ instances for parameter calibration. One-way analysis of variance technique is used to analyze the experiment results. The mean plots of SA framework with different values of t are depicted in Figure 3. The smaller RPD values, the better optimization effect.

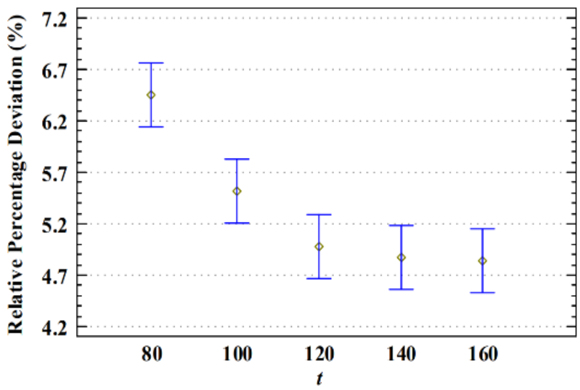

Figure 3. The mean plot of the initial temperatures with 95.0 percent Tukey HSD (Honestly Significant Difference) intervals.

From Figure 3, it can be observed that the differences are statistically significant for $t \le 120$, however, they tend to be flat for $t \ge 120$. So, when t is set to 120, the algorithm can generate a better solution with fewer iterations.

5.2. Robustness Analysis

In terms of the parameter calibration, $t = 120$ is set to the initial temperature of the SA algorithm framework. To further demonstrate the robustness of the proposed algorithm, we analyze the influence of the five instance parameters on this algorithm. The proposed algorithm is compared with the Local Search (LS) algorithm and the Simulated Annealing (SA) algorithm.

As can be seen from Figure 4, the VNS performs better than the other two algorithms. The statistical difference is significant. This is because the VNS algorithm framework can compare the optimal solutions of different neighborhoods, and the final result is closer to the global optimal solution, which can effectively avoid falling into the local optimal situation.

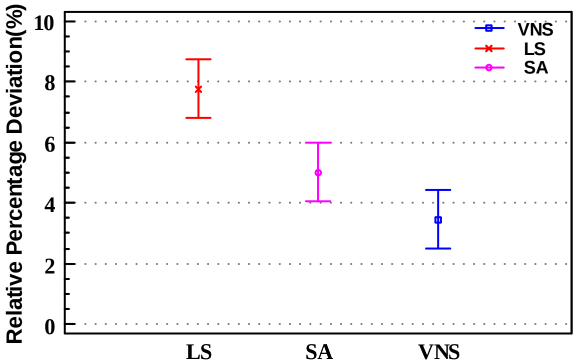

Figure 4. The mean plot of the three compared algorithms with 95.0 percent Tukey HSD intervals.

Interactions between each parameter and the compared algorithms with 95.0 percent Tukey HSD (Honestly Significant Difference) intervals are depicted in Figures 5 and 6. It can be concluded from Figures 5 and 6 that the observed differences are not statistically significant for the proposed algorithm in most cases.

Figure 5 shows that the area radius r has a great effect on the performance of proposed SA algorithm framework. Furthermore, the differences in RPD values are not statistically significant for the number of express cabinet m. This is because the radius of the region directly affects the fitness of the solution. The increase of r widens the differences between these RPD values, making it easier for the algorithm to find better solutions.

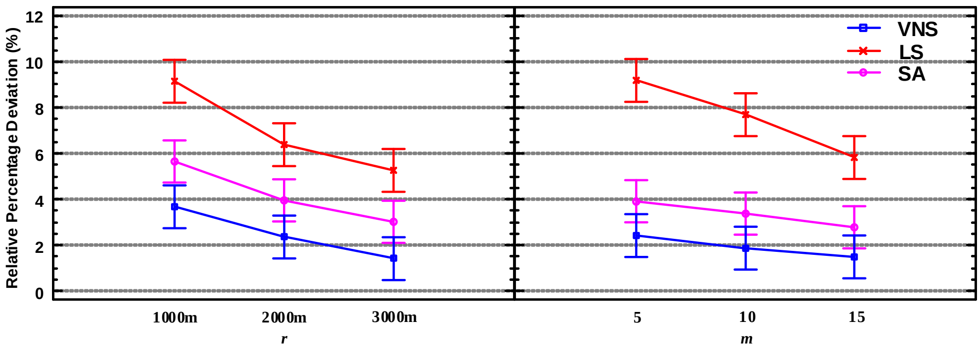

Figure 5. Interactions between area-related parameters with 95.0% Tukey HSD intervals.

Figure 6 illustrates that the following parameters: the maximum weight of packages g, the total weight of packages G, and the number of UAVs h have little influence on the execution of SA algorithm framework. Furthermore, their differences in RPD values are not statistically significant. This is because these three parameters hardly affect the ability of the algorithm to search for the global optimal solution.

After comparing the robustness of LS with SA, we compared SA with Variable Neighborhood Search algorithm (VNS). From Figure 5, it can be observed that although the robustness of SA and VNS is similar, VNS optimizes the solution better especially for the following parameters: the radius of area r and the number of express cabinet m. Figure 6 shows that the robustness of the two algorithms is not different, but VNS has better search performance. The differences in RPD values are statistically significant for the following parameters: the maximum weight of packages g and the total weight of packages G. These experiments illustrate that there is little difference in robustness between algorithms SA and VNS, but the execution performance of VNS is better. This is because VSN algorithm framework can calculate the optimal solution in different neighborhoods to avoid falling into the local optimal situation, so it can frequently get the global optimal solution.

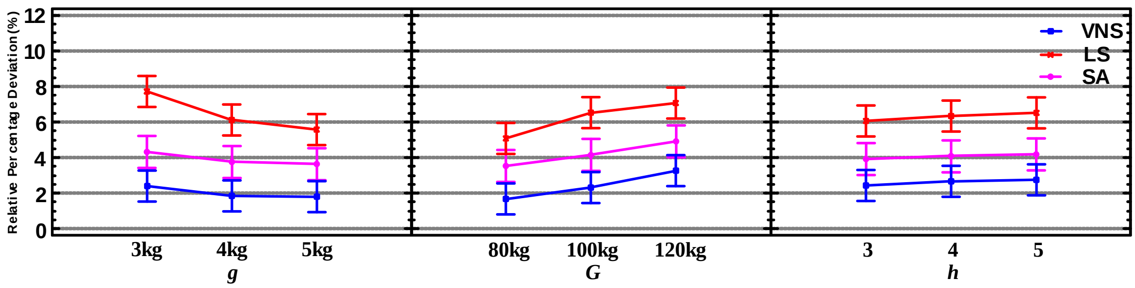

Figure 6. Interactions between g, G and h with 95.0% Tukey HSD intervals.

For each of the 243 instance combinations, five different values of initial temperature t of SA framework are chosen, i.e., in total there are 1215 instances for the algorithms to compare. Average Relative Percentage Deviations (ARPDs) on all instance combinations are shown in Table 2. This table not only shows the execution effect of different algorithms in each test instance, but it also depicts the influence of these instance parameters on different algorithms. From Table 2, it can be illustrated that VNS has the lowest ARPDs with different parameter settings and SA has the highest ARPDs. VNS performs better than the others mainly because it can escape from the local optimum by restructuring the neighborhood, and the explored solution space is larger than the other two. From the analysis of the results, it can be seen that the proposed VNS algorithm framework is robust for most parameters.

Table 2. ARPD values.

		SA	LS	VNS
r	1000 m	5.647	9.136	3.686
	2000 m	3.978	6.344	2.394
	3000 m	3.084	5.212	1.307
m	5	3.891	9.235	2.389
	10	3.376	7.618	1.915
	15	2.409	5.758	1.508
g	3 kg	4.364	7.791	2.433
	4 kg	3.862	6.179	1.884
	5 kg	3.649	5.537	1.819
G	80 kg	3.416	5.097	1.682
	100 kg	4.183	6.654	2.253
	120 kg	4.878	7.086	3.173
h	3	3.907	6.103	2.416
	4	4.084	6.313	2.589
	5	4.196	6.657	2.603
Average		3.928	6.715	2.27

In order to better show the structure of the result and system flow, the test based on a small-scale example is executed. The calculation results of the UAV flight tasks are described in detail.

Figure 7 shows the path scheduling results of three isomorphic UAVs in a small-scale scenario. For this type of UAV, the upper limit of load is 5 kg, the average flight speed is 48 km/h, and the total flying time of single charging is 0.67 h. In this experiment, the coordinates (abscissa/km, ordinate/km) of each express cabinet are as follows: s_0(1.604, 1.720), s_1(1.163, 3.585), s_2(1.220, 2.153), s_3(2.467, 3.623), s_4(0.519, 3.264), s_5(0.714, 2.742).

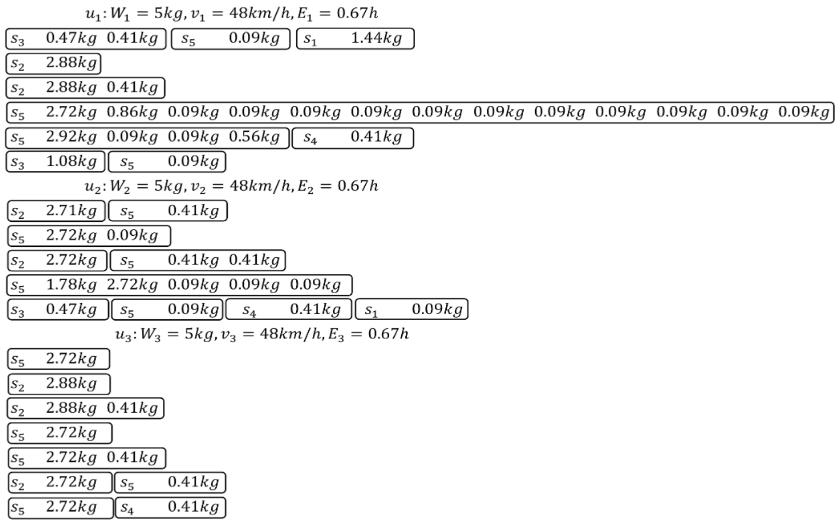

Figure 7. Small scale experimental results of UAV path planning.

Figure 8 is the path planning diagram of UAV_1 based on the map data and the calculation results shown in Figure 7. UAV_1 has six missions in total. The first one: it starts from s_0, passes s_3, s_5 and s_1 in turn, and then returns to s_0. The second one and the third one are both missions between s_0 and s_2. The fourth one is shuttle flights between s_0 and s_5. The fifth one is flight from s_0, passes s_5 and s_4 in turn, and then returns to s_0. The sixth one is from s_0, passes s_3 and s_5 in turn, and then returns to s_0. Among them, the reason that the same destination is distributed twice is that packages delivered exceed the upper load limit of the UAV.

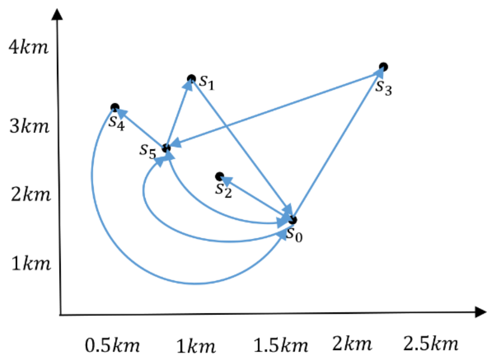

Figure 8. Routes of UAV_1.

Figure 9 is the path planning diagram of UAV_2 based on the map data and the calculation results shown in Figure 7. As shown in Figure 9, UAV_2 has seven missions in total. The first and the third missions both start from s_0, pass s_2 and s_5 in turn, and then return to s_0. The second one and the fourth one's missions are between s_0 and s_5. The fifth one is flight from s_0, passes s_3, s_5, s_4, and s_1 in turn, and then returns to s_0.

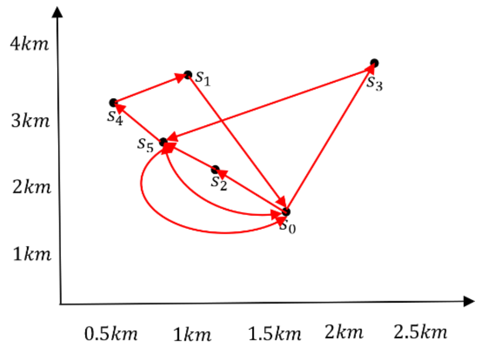

Figure 9. Routes of UAV_2.

Figure 10 is the path planning diagram of UAV_3 based on the map data and the calculation results shown in Figure 7. As shown in Figure 10, UAV_3 has seven missions in total. The first, fourth, and fifth missions are flights between s_0, and s_5. The second and third missions are shuttle flights between s_0 and s_2. The sixth one is flight from s_0, passes s_2, s_5, and then returns to s_0. The seventh one is flight from s_0, passes s_5, s_4, and then returns to s_0.

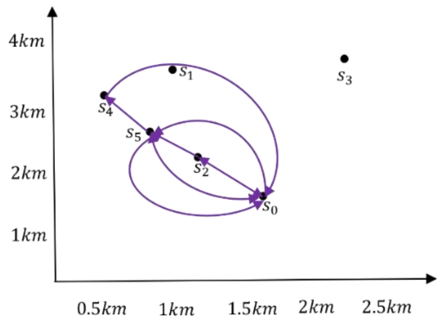

Figure 10. Routes of UAV_3.

Since the considered problem is biobjective, we apply a linear weighted sum (LWS) method to evaluate the solutions on both the total completion time and the satisfaction perspectives, as defined as Equation (14). ω and $1-\omega$ $(\omega \in (0,1))$ are the weight according to the importance of the total completion time and the satisfaction, respectively. Therefore, the smaller the LWS, the better the solution will be.

$$\mathrm{LWS} = \frac{A - A_{min}}{A_{min}} \cdot \omega + \frac{B - B_{min}}{B_{min}} (1 - \omega) \tag{14}$$

As can be seen from Figure 11, compared with no difference shown in (c), when the weight of customer satisfaction A is higher and the weight of total completion time B is lower, the Relative Percentage Deviation (RPD) value of Local Search (LS) algorithm is larger and the RPD value of variable neighborhood search (VNS) algorithm is smaller shown in (e). Conversely, the RPD value of LS algorithm is smaller and the RPD value of VNS algorithm is larger, shown in (a). So, LS algorithm is more suitable to solve the problem of high demand for customer satisfaction, VNS algorithm is more suitable to solve the problem of high demand for total completion time.

From Figure 11 it is observed that when the parameter ω changes, the result of SA algorithm is the most stable and there is almost no fluctuation. The result of VSN algorithm is relatively stable with small fluctuation, while the result of LS algorithm has large fluctuation, which is sensitive to the parameter, so its stability is poor. The experimental results demonstrate that VNS is feasible and effective for the considered problem.

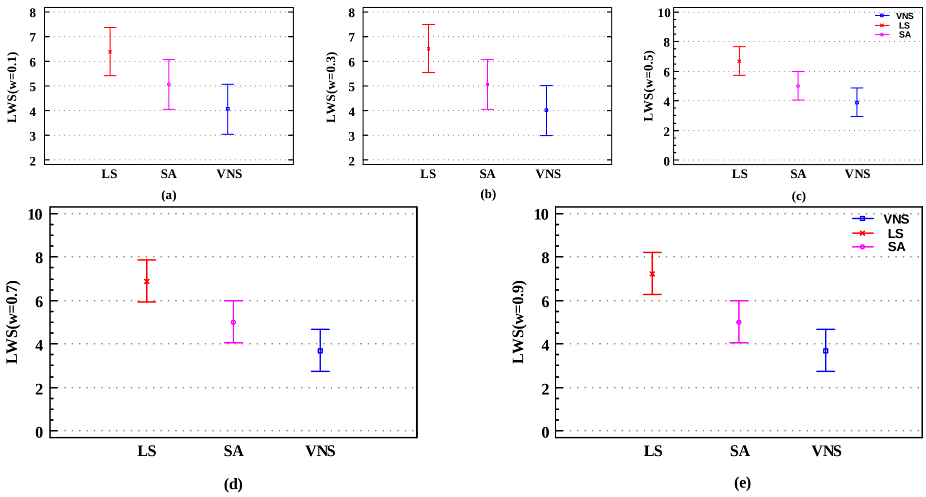

Figure 11. The result of linear weighted sum. (**a**) LWS (w = 0.1), (**b**) LWS (w = 0.3), (**c**) LWS (w = 0.5), (**d**) LWS (w = 0.7), (**e**) LWS (w = 0.9).

Table 3 shows the average operation times for the three specifications of radius of area r. Figure 12 shows the convergence curves of the three algorithms LS, SA, and VNS simultaneously. The horizontal coordinate is the run time and the vertical coordinate is the ARPD value. From Figure 12 it is observed that LS has the fastest convergence, SA is second, and VNS has the slowest convergence, but VNS outputs the best results. Since it can escape from the local optimum by changing the neighborhood, it is possible to find a better solution compared with LS and SA. However, constant changes of neighborhood increase its cost of time.

Table 3. Average operation time (s).

r	*LS*	*SA*	*VNS*
1000	7.63	19.72	50.26
2000	14.82	38.13	98.57
3000	28.74	74.05	189.41

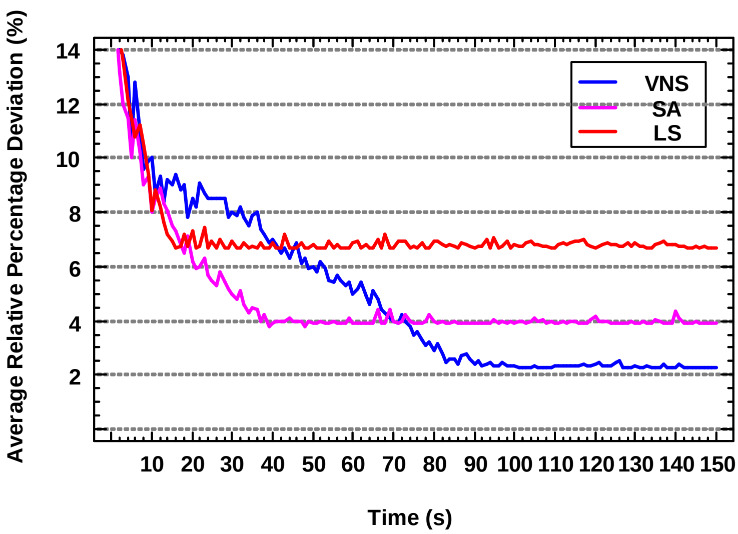

Figure 12. The time convergence curve of the algorithms.

Based on the above experimental results and analysis, we determine that VNS performs better than the other two which are the most commonly used heuristics for optimization problems, although it converges the slowest. VNS is easy to implement and it can balance very well between diversity and convergence. We believe that there are many other metaheuristics which are suitable for solving the problem and may have better performance. Based on our preparatory work, GA and PSO converge slowly and their performances on TSP are good for small-size testing instances. If the computation time is not limited, we believe they can find better solutions than those of VNS in probability significance. NSGA-II is another potentially competitive algorithm which may be very suitable for the considered problem, since NSGA-II converges faster than GA and PSO and has promising performance on multiobjective optimization problems. We will further investigate these algorithms to improve our work on the considered problem based on these assumptions.

6. Conclusions

The delivery problem of logistics UAV in urban environment is considered to be a scheduling problem. We first present the automatic delivery system for logistics UAVs to improve the distribution efficiency and reduce costs. The Variable Neighborhood Search algorithm framework is proposed to address the scheduling problem of the system. We use the special method of two-stage optimization to solve the optimization problem with customer satisfaction and total completion time. The experimental results show that the proposed VNS is robust and has better performance. For future work, more multivariate heuristic algorithms will be incorporated, edge computing scenarios that are more realistic will be considered, more practical UAV volume models and energy consumption models will be applied, and the information interaction, multi-UAVs cooperation, and optimized operation mode in the distribution process will be further considered. There are also many other potential algorithms that may be more suitable for solving the considered problem. We will further investigate these algorithms to improve our work on the considered problem.

Author Contributions: Conceptualization and Methodology: Y.L., X.Y., H.H., J.Z., and M.W.; Validation: J.Z., H.H., and M.W.; Investigation: Y.L., X.Y., and J.Z.; Writing–original draft preparation: Y.L. and X.Y.; Writing–review and editing: Y.L., X.Y., J.Z. and H.H.; Supervision: J.Z., H.H., and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is sponsored by the National Natural Science Foundations of China (Grant Nos. 61672297, 61872196), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 18KJB520039), the Key Research and Development Program of Jiangsu Province (Social Development Program, No.BE2017742), and the National Science Foundation for Post-doctoral Scientists of China (Grant No. 2018M640510).

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

References

1. Samira, H.; Evsen, Y.; Raheeb, M. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *17*, 2624–2661.
2. Panetta, K. Gartner's Top 10 Strategic Technology Trends for 2019. Available online: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019/ (accessed on 15 October 2018).
3. Yao, D.; Kun, Y.; Kezhi, W.; Guopeng, Z.; Yizhe, Z.; Dongwei, C. Joint Resources and Workflow Scheduling in UAV-Enabled Wirelessly-Powered MEC for IoT Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10187–10200.
4. Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
5. Chao, W.; Dong, M.; Fu, Z.; John, W. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Comput. Ind. Eng.* **2015**, *83*, 111–122.
6. Jiahai, W.; Taiyao, W.; Qingfu, Z. A Two-Stage Multiobjective Evolutionary Algorithm for Multiobjective Multidepot Vehicle Routing Problem with Time Windows. *IEEE Trans. Cybern.* **2019**, *49*, 2467–2478.
7. Jiangyun, L.; Xiaobo, J.; Chaonan, T. Modeling and simulation of VRP in wartime using NSGA II. In Proceedings of the Chinese Control and Decision Conference, Nanchang, China, 3–5 June 2012; pp. 997–1001.
8. Wang, G.-S.; Yu, Y.-X. An Improved Ant Colony Algorithm for VRP Problem. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jinan, China, 2–4 April 2010; pp. 129–133.

9. Hosseinabadi, A.A.R.; Alavipour, F.; Shamshirbnd, S.; Balas, V.E. A Novel Meta-Heuristic Combinatory Method for Solving Capacitated Vehicle Location-Routing Problem with Hard Time Windows. *Adv. Intell. Syst. Comput.* **2016**, *454*, 707–728. [CrossRef]
10. Li, Z. Simulation Model on Emergency Rescue VRP Based on Improved Ant Colony Optimization. In Proceedings of the 2nd International Conference on Information Science and Engineering, Guangzhou, China, 3–5 December 2010; pp. 1499–1502.
11. Zheng, J. A Vehicle Routing Problem Model With Multiple Fuzzy Windows Based on Time-Varying Traffic Flow. *IEEE Access* **2020**, *8*, 39439–39444. [CrossRef]
12. 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]
13. Jonghoe, K.; James, R.M. On the concerted design and scheduling of multiple resources for persistent UAV operations. *J. Intell. Robot. Syst.* **2014**, *74*, 479–498.
14. Jia, Z.; Xiaoke, Y.; Lingyu, Y.; Gongzhang, S. Modeling for UAV resource scheduling under mission synchronization. *J. Syst. Eng. Electron.* **2010**, *21*, 821–826.
15. Denis, B.; Konstantin, A.; Georgy, V.; Petr, S.; Natalia, A. Real-time task scheduling for multi-agent control system of UAV's group based on network-centric technology. In Proceedings of the 2016 International Conference on Control, Decision and Information Technologies, Saint Julian's, Malta, 6–8 April 2016; pp. 378–381.
16. Ahmadian, N.; Lim, G.J.; Torabbeigi, M.; Kim, S.J. Collision-Free Multi-UAV Flight Scheduling for Power Network Damage Assessment. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 794–798.
17. Ahmed, S.; Chowdhury, M.Z.; Jang, Y.M. Energy-Efficient UAV-to-User Scheduling to Maximize Throughput in Wireless Networks. *IEEE Access* **2020**, *8*, 1. [CrossRef]
18. Xu, C.; Yin, C.; Han, W.; Wang, D. Two-UAV trajectory planning for cooperative target locating based on airborne visual tracking platform. *Electron. Lett.* **2020**, *56*, 301–303. [CrossRef]
19. Ghazzai, H. Optimal Sequential and Parallel UAV Scheduling for Multi-Event Applications. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 25–28 May 2018; pp. 1–6.
20. Yixuan, L.; Xiaoxiang, Y.; Jie, Z.; Haiping, H.; Min, W. Multi-objective Scheduling of Logistics UAVs Based on Simulated Annealing. In Proceedings of the 2019 International Conference on Parallel Architectures, Algorithms and Programming, Guangzhou, China, 12–14 December 2019; pp. 287–298.
21. MladenoviĆ, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
22. Ma, Y.; Li, Z.; Yan, F.; Feng, C. A hybrid priority-based genetic algorithm for simultaneous pickup and delivery problems in reverse logistics with time windows and multiple decision-makers. *Soft Comput.* **2019**, *23*, 6697–6714. [CrossRef]
23. Jiang, X.; Zhou, Q.; Ye, Y. Method of task assignment for UAV based on particle swarm optimization in logistics. In Proceedings of the International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, New York, NY, USA, 25–27 March 2017; pp. 113–117.