

Article

Online Delivery Problem for Hybrid Truck–Drone System with Independent and Truck-Carried Drones

Mengyuan Gou  and Haiyan Yu *

School of Economics and Management, Chongqing Jiaotong University, Nanan, Chongqing 400074, China

* Correspondence: hyyu@cqjtu.edu.cn

Abstract: Considering real-time requests and multiple truck–drone delivery modes, we propose an online delivery problem using a truck and some drones, which form a hybrid truck–drone delivery collaboration system comprising independent and truck-carried drones. Considering this problem, we focus on how to schedule the vehicles to serve real-time requests, with the objective of minimizing the time of the latest vehicle’s return to the delivery station. First, we proved the lower bound of this problem to be 1.5. Second, we designed an online re-planning algorithm and proved its competitive ratio to be 2.5. As the online re-planning algorithm invokes an offline algorithm, an offline model was established, and an offline drone priority algorithm was designed. Then, we verified the effectiveness of the offline algorithm by comparing it with the CPLEX solution, and the stability of the online re-planning algorithm with different input parameters was studied through MATLAB simulation. Finally, the minimal latest time saving was calculated by comparing the hybrid truck–drone collaboration system with a truck-only delivery system. This research provides theoretical support for addressing the hybrid truck–drone delivery problem.

Keywords: hybrid truck–drone delivery; online algorithm; competitive analysis; vehicle routing problem



Citation: Gou, M.; Yu, H. Online Delivery Problem for Hybrid Truck–Drone System with Independent and Truck-Carried Drones. *Sustainability* **2023**, *15*, 1584. <https://doi.org/10.3390/su15021584>

Academic Editor: Bin Ji

Received: 22 November 2022

Revised: 27 December 2022

Accepted: 27 December 2022

Published: 13 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Drones have recently been used in many fields due to their various capabilities and high potential, including logistics delivery. In this field, drones are combined with trucks to deliver goods, especially expensive and emergency goods. For example, during the multi-point outbreak of the coronavirus disease 2019 (COVID-19), this modality was utilized to ease the inconvenience of medical emergency delivery caused by regional control. On 29 April 2022, Jinshan Hospital of Fudan University used a drone and a truck to deliver emergency medical supplies to the Shanghai High-Tech Zone. First, the supplies were delivered to the lawn in front of an enterprise using a drone. Later, a volunteer drove to the enterprise without contact. General freight delivery is also a practical application area for this technology. Siroop has developed a pilot project for e-commerce delivery by drones using vans. In this project, the drone uses the roof of the delivery truck as a landing platform [1]. As of 18 November 2021, Zipline started working on drone delivery with Walmart in Pea Ridge, AR. Not only does this allow for delivery in one hour, it also reduces carbon emissions [2]. On 13 June 2022, Amazon announced that it would use Prime drones to deliver goods in the county of California. The Federal Aviation Administration (FAA) predicted the sales of drones for commercial purposes to grow from 277,386 in 2018 to 835,211 by 2023 [3]. Therefore, the participation of drones in last-mile delivery has become a popular trend.

Traffic control can be eased through a hybrid truck–drone collaboration delivery system, using independent and truck-carried drones. Most scholars have focused on the study of truck–drone collaboration delivery under the condition that all request information is known [4]. However, in a real scenario, an online server only knows the request after it is released, such as is the case for the Zipline or Siroop delivery services. A real-time request is a key factor in emergency delivery or instant commerce gratification. Considering

real-time requests, this problem can be considered an online delivery problem for hybrid truck–drone delivery systems with independent and truck-carried drones.

In a hybrid truck–drone collaboration delivery system, a truck carries a drone, and some drones simultaneously and independently fly between the delivery station and the customer location for delivery, as shown in Figure 1c. Parallel delivery indicates that some drones and a truck fulfill requests independently and simultaneously, as shown in Figure 1a; while synergistic delivery denotes that a truck carries a drone for simultaneous delivery, as shown in Figure 1b.

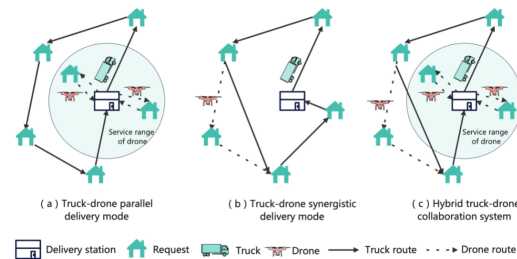


Figure 1. The truck–drone delivery modes of (a) parallel, (b) synergistic, and (c) hybrid collaboration.

Therefore, the considered problem has the characteristics of both the truck–drone collaboration delivery problem and the online multi-vehicle delivery problem. Compared with the truck–drone collaboration delivery problem, the requests are made in real time. Compared with the online multi-vehicle delivery problem, we need to consider the synergy between the truck and drone, including factors such as the drone’s limited service scope and high speed.

In analyzing the actual operation in truck–drone collaboration delivery, there are two key characteristics: (1) Multiple truck–drone delivery modes are used at the same time to improve efficiency. Thus, we study the hybrid truck–drone collaboration system in combination with parallel and synergistic delivery modes; and (2) requests are released in real-time. Thus, we study the online delivery problem for real-time requests.

To adapt to the characteristics of online requests in real delivery scenes and to complement the deficiency of current hybrid delivery system research, the online delivery problem for the hybrid truck–drone collaboration system (HTDCS) is proposed and studied.

This problem can be effectively solved using an online analysis method. First, a review of the related research is presented in Section 2. The online problem description and basic definitions are given in Section 3.1. We also study the corresponding offline problem and formulate an offline model as a supplement to the online algorithm in Section 3.2. Then, we propose an online re-planning (RP) algorithm and prove the lower bound of the online problem, in Section 4. We also prove the competitive ratio of the online RP algorithm in Section 5. Considering that the online RP algorithm invokes an offline algorithm, we detail the design of an offline drone priority algorithm and verify the effectiveness of the offline algorithm in comparison with the CPLEX solution results in Section 6. The stability of the online RP algorithm under various input parameters is studied through MATLAB simulation in Section 7. Finally, we calculate the minimal latest time saving and compare the hybrid truck–drone collaboration system with a truck-only delivery system in Section 8.

2. Related Literature

In this section, we summarize the related research on hybrid truck–drone collaboration systems, scheduling problems, and online delivery problems.

In terms of theoretical research, Sung et al. [4] have examined either truck-carried drones or parallel drones collaborating with a truck delivery system. The related research has gradually increased in recent years, but most of the research has focused on a single delivery mode. However, in real scenes, multiple delivery modes can be used at the same time to improve efficiency, which needs to be further studied.

Regarding research on the HTDCS, Wang et al. [5] first proposed a collaborative delivery system combining two delivery modes, in which trucks and truck-carried drones were used to serve long-distance requests, while some trucks separately served the remaining requests. Wang and Lan [6] have proposed a hybrid truck–drone delivery (HTDD) model and studied a combination of the synergistic and parallel delivery modes for a truck and drone. On the basis of the HTDD model, Schermer et al. [7] have proposed the drone-assisted traveling salesman problem with a robot station (TSP-D-RS) model, which added a robot station. Yu et al. [8] have presented a van-based robot hybrid pickup and delivery problem considering the HTDCS case. Rave et al. [9] have added a micro-depot in HTDCS, in which drones are delivered between the request locations and the micro-depot. Kloster et al. [10] have proposed a multiple traveling salesman problem with drone stations and made general contributions to this problem. However, all of them were established for the case in which the information was complete. In addition, Sung et al. [4] have indicated that uncertainty is a meaningful factor to consider. Troudi et al. [11] have discussed some realistic factors in the drone delivery problem, such as the size of the drone fleet and battery charging. In the real scene, the server only knows the request information released at the current time but not future request information. Therefore, it is of practical significance to study the corresponding online problem and algorithm.

Regarding the research on scheduling problems, Tang et al. [12] have studied the continuous berth allocation and quay crane assignment problem. Fanjul-Peyro et al. [13] have proposed a mixed-integer linear program and a mathematical programming-based algorithm for an unrelated parallel machine scheduling problem. VRPD (vehicle routing problem with drones) can also be regarded as a scheduling problem. Wang and Sheu [14] have constructed a mixed-integer programming model, and proposed a branch-and-price algorithm to solve the VRPD. They made outstanding contributions to this field but, again, did not take into account the hybrid truck–drone collaboration system.

Regarding research on the online delivery problem, Ausiello et al. [15] have used online analysis to study the traveling salesman problem. Yu et al. [16] have studied the online nomadic traveling salesman problem, and proved its lower bound. Wen et al. [17] have studied the online traveling salesman problem with service selection and time window and proved the competitive ratio of their designed online algorithm. Jaillet and Wagner [18] have shown that the use of more predictive information could improve the competitive ratio. Ascheuer et al. [19] have proved that the competitive ratio of the plan-at-home algorithm for online dial-a-ride problem was 2.5. Dai et al. [20] have shown that the competitive ratio for online nomadic multi-vehicle scheduling problem was 3.5. Different from the normal online multi-truck delivery problem, the problem researched here also requires consideration of the synergy between drones and trucks, as well as coordination between the parallel and synergistic delivery modes.

In the following section, we formulate the problem and provide an example.

3. Problem Statement

3.1. Assumptions and Notations for the Online and Offline Delivery Problem of HTDCS

Suppose there is an emergency delivery station to serve the requests released in the area under its jurisdiction using the HTDCS. Requests involve the same goods, are released in real-time (sequentially), and cannot be rejected or split. Each request's demand is one unit. The related parameters of the used vehicles are given in Table 1, and the relevant notation and variables used in this paper are defined in Table 2.

Table 1. Parameters of delivery vehicles.

Vehicle	Number	Payload	Speed	Service Radius
Truck	1	infinite	1	infinite
Drone	$N_{drone} (N_{drone} \geq 2)$	1	$k (k > 1)$	s

Table 2. Related notation and variables.

Set	Definition
R	Set of all requests released
R'	Set of the sequence numbers in R and origin, $R' = \{0, 1, 2, \dots, n\}$
I	Set of the sequence numbers of finite request points or truck stops, $i \in I$
U	Service time set of the unserved requests for online server at time t_n , $U = \{u_1, u_2, \dots, u_n\}$
W	Service time set of the unserved requests for offline server, $W = \{w_1, w_2, \dots, w_n\}$
H	The number of used drones, $H = \{1, 2, \dots, N_{drone}\}$
σ	Request sequence of truck carrying drone B according to the online RP algorithm, $\sigma \in R$
ϑ	Service sequence of drone A for online server, $\vartheta \in R$
Parameter	Description
s	Drone service radius
k	Ratio of drone speed to truck speed
N_{drone}	Total number of drones (an integer greater than 2)
ω	Truck delivery distance for online server
δ	An arbitrarily small positive number
u_i	Service time of the i^{th} unserved request assigned to drone A for the online server at time t_n , $i \in R' / 0$
w_i	Service time of the i^{th} unserved request assigned to drone A for the offline server, $i \in R' / 0$
$r_i = (t_i, x_i, y_i)$	the i^{th} request, $r_i \in R$; t_i denotes release time, x_i denotes X coordinate, y_i denotes Y coordinate
$P_v(t_i)$	Position of the truck at time t_i
$L^*(0, o, R)$	Time taken for the offline server to serve all the requests in R starting from the origin at time 0
$d\{o, P_v(t_i)\}$	Euclidean distance between the position of the truck at time t_i and the origin
$T(o, r_n)$	Return time from the coordinate of r_n
f'_i	Time when drone B is at point i
d_{ij}	Euclidean distance between points i and j , $i, j \in R'$
v_i^{arrive}	Time when the truck arrives at point i
v_i^{depart}	Time when the truck departs from point i
T_d	Latest time when drone A returns to the origin
T_{fs}	Latest time when the truck or drone B returns to the origin
T	Latest time when either truck or drone returns to the origin, $T = \max\{T_d, T_{fs}\}$
$C_{OPT}(R)$	Optimal completion time
$C_{online}(R)$	Completion time needed by the online algorithm
Decision Variable	Description
h_i	If drone A serves request i , then $h_i = 1$; otherwise, $h_i = 0$, $i \in R$
z_{ij}	If the truck moves from point i to point j , then $z_{ij} = 1$; otherwise, $z_{ij} = 0$, $i, j \in R'$
x_{ij}	If drone B takes off from point i to serve point j , then $x_{ij} = 1$; otherwise, $x_{ij} = 0$, $i, j \in R'$
y_{jp}	If drone B lands at point p after serving point j , then $y_{jp} = 1$; otherwise, $y_{jp} = 0$, $j, p \in R'$

All related parameters used in the pseudocode in this paper are detailed in Table 3.

Table 3. Parameters used in the pseudocode.

Parameter	Definition
R	Set of requests
$ R $	Total number of elements in R
r_v	The v^{th} request in R
$coordinate(v)$	The v^{th} request's coordinate
s	Service radius of the drone
$newcenter$	Set of cluster center coordinates after k -means clustering
a	Cell set of $newcenter$
c_i	The i^{th} element in the set $newcenter$
M_u	Coordinate according to the mean value of the u^{th} element in a
DF	Set of the squares of differences between r_v and M_u
$Ssemeanx$	Set of sequence numbers according to $ R $
$Ssemeany$	Set of mean values according to DF
SY	Set of requests served by synergistic delivery mode
D_A	Set of requests served by drone A
D_B	Set of requests served by drone B
D_T	Set of requests served by truck
T_f	The latest time when the truck or drone B returns to the origin
T_d	The latest time when drone A returns to the origin
T	The latest time when either the truck or drone returns to the origin, $T = \max\{T_d, T_f\}$

It is assumed that some of the drones, such as the parallel drones (drone A), provide services between the delivery station and the request locations and that the truck carries another drone (drone B) to serve the requests. The truck can only carry one drone. All the drone parameters are the same. We assume that the drones can be charged at the delivery station and truck and that the drones are fully charged when they start to deliver. The truck can still serve along the path after the drone takes off. Drone B can land at the request and truck location when it takes off. Once the truck and drone B start to deliver, they do not need to return to the delivery station to pick up goods and only need to return to the delivery station after serving all the requests. We study how to arrange the vehicles to optimally serve real-time requests, with the objective of minimizing the time of the latest vehicle returning to the delivery station.

In the example shown in Figure 2, there are some drones and one truck that serve existing requests. When two new requests are released at the same time, the original request allocation and vehicle routing should be re-planned, as shown in Figure 2b. In Figure 2b, request R6 will be delivered by drone B, and the delivery path of the truck is changed. Through optimization, the efficiency and reasonableness of the delivery system can be improved.

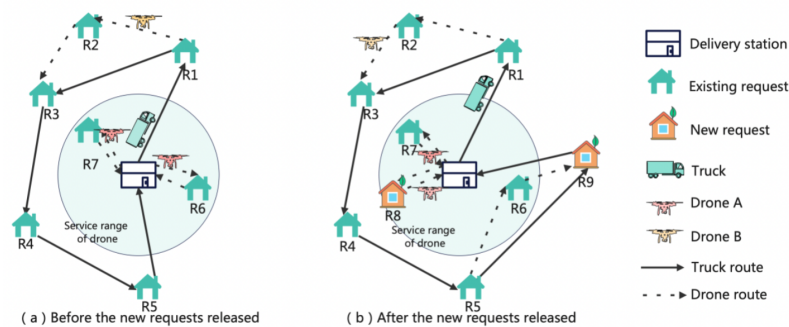


Figure 2. Example of the online delivery problem for HTDCS of the (a) before new requests are released and (b) after new requests are released.

Competitive analysis, considering the worst-case scenario, is a popular way to measure the performance of an online algorithm; it involves a game process between an online server and an offline server. During the game, the offline server releases requests, which will benefit the offline server as much as possible and not the online server. Thus, the offline server knows all the requests at time 0, while the online server does not; meanwhile, the offline server can take actions at time 0, while the online server cannot. For the same input sequence θ , the online server adopts the β algorithm to serve θ , and the solution is $C_{online}(\theta)$. The offline server adopts the optimal algorithm to serve the same input sequence θ , and the solution is $C_{OPT}(\theta)$. For every input sequence θ , there are constants ρ and γ such that $C_{online}(\theta) \leq \rho * C_{OPT}(\theta) + \gamma$ holds; thus, the competitive ratio of the online algorithm β is ρ . For any online algorithm, if it is not possible to get less than the competitive ratio, ρ_0 , this ratio is called the lower bound of the problem. The closer the value of ρ to 1, the better the performance of the algorithm.

3.2. Offline Model Formulation

We also study the corresponding offline problem as a supplement to the online algorithm. We built a model based on a symmetric network assuming that 0 is the delivery station sequence number. The related notations are given in Table 2.

$$\min T = \max \{ T_d, T_{fs} \} \quad (1)$$

$$St. T_d \geq t_i + \sum_{j \geq i} h_j * 2d_{0j}/k, \forall i \in R', \quad (2)$$

$$T_{fs} \geq v_i^{depart} + d_{0i}, \forall i \in R', \quad (3)$$

$$H \leq N_{drone}, \quad (4)$$

$$\sum_{j \in R'} x_{ji} + \sum_{j \in R'} z_{ij} + h_i = 1, \forall i \in R', \quad (5)$$

$$\sum_{j \in R'} z_{0j} = \sum_{i \in R'} z_{i0} = 1, \quad (6)$$

$$u_i - u_j + nz_{ij} \leq n - 1, 1 \leq i \neq j \leq n, \quad (7)$$

$$\sum_{i \in R', i \neq j} z_{ij} = \sum_{p \in R', p \neq j} z_{jp}, \forall j \in R', \quad (8)$$

$$\sum_{j \in R', j \neq i} x_{ij} \leq \sum_{h \in R', h \neq i} z_{hi}, \forall i \in R', \quad (9)$$

$$\sum_{j \in R', j \neq p} y_{jp} \leq \sum_{l \in R', l \neq p} z_{lp}, \forall p \in R', \quad (10)$$

$$\sum_{i \in R', i \neq j} x_{ij} = \sum_{p \in R', p \neq j} y_{jp}, \forall j \in R', \quad (11)$$

$$v_i^{arrive} \leq f_j' + M(1 - x_{ij}) - d_{ij}/k, \forall i, j \in R', \quad (12)$$

$$v_i^{depart} \geq f_j' + M(1 - y_{ji}) + d_{ji}/k, \forall i, j \in R', \quad (13)$$

$$y_{jp} \leq M * \max\{0, s - d_{jp}\}, \forall j, p \in R', p \neq j, \quad (14)$$

$$x_{ij} \leq M * \max\{0, s - d_{ij}\}, \forall i, j \in R', j \neq i, \quad (15)$$

$$v_i^{depart} \geq t_i, \forall i \in R', \quad (16)$$

$$x_{ij}, y_{jp}, z_{ij}, h_i \in \{0, 1\}, \forall i, j, p \in R'. \quad (17)$$

The objective function (1) seeks to obtain the minimum latest time T when the vehicle returns to the origin. Constraint (2) expresses the minimum latest time when drone A

returns to the origin. Constraint (3) indicates the minimum latest time when the truck or drone B returns to the origin. Constraint (4) ensures that the number of used drones is less than or equal to the total number of drones. Constraint (5) ensures that each request can only be served once. Constraint (6) describes that the truck must return to the origin after departing from it. Constraint (7) is used to eliminate sub-tours for each truck. Constraint (8) ensures the flow conservation of delivery trucks. Constraints (9) and (10) ensure that, if drone B takes off from point i and lands at point p , then the truck must pass through points i, p . Constraint (11) indicates that drone B takes off from point i , serves point j , and lands at point p . Constraint (12) specifies that the truck must have arrived at point i if drone B takes off from the point i . Constraint (13) ensures that the truck is at point i when drone B lands at point i . Constraints (14) and (15) ensure that the flying distance of drone B shall not exceed its service scope. Constraint (16) guarantees that requests are served after release. Finally, Constraint (17) specifies related decision variable definitions.

4. Algorithm and the Lower Bound of the Online Problem

In this section, we detail the design of an online RP algorithm based on the plan-at-home idea for the problem. We also prove the lower bound of the problem for any online algorithm. The related notation is given in Tables 2 and 3. Compared with the classical plan-at-home algorithm [15], the online RP algorithm does not include the origin into the former truck request set when a new request is released, which can decrease the minimum latest time for the truck carrying drone B. Suppose that there exists a corresponding offline algorithm that can obtain an optimal solution for any input.

Online RP algorithm: The online server allocates all current requests according to the offline algorithm when a new request is released in order to form the new service sequences for the vehicles. If drone A is not at the delivery station, the nearest drone A to the delivery station will fly back to prepare for serving, even if it is on the way to serve other requests. If there are unserved requests in the new sequence of the truck and/or drone B, the truck and drone B will continue or start to serve from the current position. The pseudocode for the online RP algorithm is given in Algorithm 1:

Algorithm 1 Pseudocode for the online RP algorithm.

Input: $R, |R|$

Output: T

```

1: for  $|R| = 1 : n$  do
2:   Identify all current requests and related information, according to  $R$  and  $|R|$ 
3:   Invoke the offline algorithm to allocate all current requests and obtain  $SY$  and  $D_A$ 
4:   for each request in  $D_A$  and  $SY$  do
5:     if the request is served then
6:       remove it from the set of  $D_A$  or  $SY$ 
7:     end if
8:   end for
9:   if there is any unserved request in  $D_A$  then
10:    if there is a drone A located at the delivery station then
11:      This drone A starts to serve the unserved request
12:    else
13:      Drone A which is closest to the delivery station will return to the delivery station and start to serve
unserved requests
14:    end if
15:  end if
16:  if there is any unserved request in  $SY$  then
17:    Truck and drone B will start to serve unserved requests from their current position
18:  end if
19:  Record the latest time when drone A returns to the delivery station as  $T_d$ 
20:  Record the latest time when truck or drone B returns to the delivery station as  $T_{fs}$ 
21: end for
22: Calculate the latest time  $T$  when either truck or drone returns to the delivery station, according to
 $T = \max\{T_d, T_{fs}\}$ 

```

Next, we prove the lower bound of the problem for any online algorithm (related notation are provided in Table 2).

Theorem 1. *The lower bound for the online delivery problem of a hybrid truck–drone collaboration system with independent and truck-carried drones is 1.5.*

Proof of Theorem 1. Take a straight line as an example without loss of generality. We assume that there is a farthest request $(t_i, |y|, 0)$, where $|y| > s$. The request $r_1 = (s/k - \delta, s, 0)$ is released at time $s/k - \delta$, where δ is an arbitrarily small positive number. As r_1 is located within the service scope of the drone, the online server can use either a drone or a truck for service. Let t denote the time when the online server starts serving request r_1 . Consider the following two cases:

Case 1 ($t < s/k$). If the online truck is located at $-\epsilon$ at time y ($\epsilon \geq 0, |y| \geq \epsilon$), the offline server releases request $r_2 = (y, y, 0)$ ($y > s/k$). For the same reason, if the online truck is located at ϵ at time y , the offline server will release request $r_2 = (y, -y, 0)$. As $|y| > s$, the request is beyond the service scope of the drone. Therefore, the online server must use a truck or a truck carrying a drone to serve request r_2 . For the offline server, if the request is $r_2 = (y, y, 0)$, drones cannot serve the request, and the truck moves along the positive half-axis from time 0. If the request is $r_2 = (y, -y, 0)$, the truck starts to serve along the negative half-axis, and a drone moves along the positive half-axis from time 0. We have $C_{online} \geq \max\{3y + \epsilon, (3s)/k - \delta\} = 3y + \epsilon$ and $C_{OPT} = 2y$; therefore, we obtain $\rho = C_{online}/C_{OPT} \geq (3y + \epsilon)/(2y) \geq (3y)/(2y)$. Thus, in case 1, $\rho \geq 1.5$.

Case 2 ($t \geq s/k$). No new requests are released. The offline drone moves along the positive half-axis from time 0. The online drone has to move after it is released. Thus, $C_{OPT} = 2s/k$ and $C_{online} \geq 3s/k - \delta$, and we obtain $\rho = C_{online}/C_{OPT} \geq (3s/k - \delta)/(2s/k)$. Thus, in case 2, when $\delta \rightarrow 0, \rho \geq 1.5$. \square

Therefore, the lower bound of the competitive ratio for the problem on the line is 1.5, and the proof is completed.

Theorem 1 shows that the lower bound for the competitive ratio of the time needed by any online algorithm to the optimal time needed by an offline algorithm is no less than 1.5 for the considered problem.

5. Competitive Analysis

In this section, we measure the performance of the online RP algorithm and prove the upper bound of the algorithm to be 2.5. We study the characteristics of the problem and the online RP algorithm (see Lemmas 1 and 2) in order to obtain the competitive ratio for the online RP algorithm (see Theorem 2).

Lemma 1. *For any $r_n \in R$, the inequality $L^*(t_n, P_v(t_n), \sigma) \leq L^*(0, o, \sigma) + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$ holds.*

Proof of Lemma 1. When the new request $r_n = (t_n, x_n, y_n)$ is released, the service time for the truck carrying drone B is $L^*(t_n, P_v(t_n), \sigma)$. $d\{o, P_v(t_i)\}$ denotes the Euclidean distance between the truck's position and the origin at time t_i . The truck's farthest position is the farthest request position, from which it is easy to determine that $d\{o, P_v(t_n)\} \leq \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$. The total time when the truck carrying drone B serves the request sequence from its current position is no greater than the total time when it returns to the origin and then goes to serve the same request sequence. Thus, $L^*(t_n, P_v(t_n), \sigma) \leq L^*(t_n + d\{o, P_v(t_n)\}, o, \sigma) + d\{o, P_v(t_n)\}$. According to $d\{o, P_v(t_n)\} \leq \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$,

we have $L^*(t_n, P_v(t_n), \sigma) \leq L^*\left(t_n + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}, o, \sigma\right) + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$. The later the online server gives the order to leave, the more request information the online server possesses, improving the allocation result obtained by the online server. As

such, $L^* \left(t_n + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}, o, \sigma \right) \leq L^*(0, o, \sigma)$. Thus, we have $L^*(t_n, P_v(t_n), \sigma) \leq L^*(0, o, \sigma) + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$. The proof is completed. \square

Lemma 2. For any $r_n \in R$ satisfying $d_{0n} \leq S$, the inequality $C_{OPT} \geq \max\{L^*(0, o, R), t_n + d_{0n}/k\}$ holds.

Proof of Lemma 2. When $d_{0n} \leq S$, the new request is added to the request sequence of drone A, the truck, or drone B. The request cannot be served until it is released and so, for the last service request $r_n = (t_n, x_n, y_n)$, the inequality $C_{OPT} \geq t_n + T(o, r_n)$ holds. For any $r_n \in R$, we suppose that the truck delivery distance is ω . When the truck carries a drone to serve, the delivery distance of drone B is $d_{0n} - \omega$, and the speed of the drone is $k (k > 1)$. Therefore, the inequality $T(o, r_n) = \omega + \{d_{0n} - \omega\}/k \geq d_{0n}/k$ holds. Due to the possibility that all requests may be delivered by drone A only, $\omega = 0$ and $C_{OPT} \geq t_n + d_{0n}/k$. Thus, we have $C_{OPT} \geq \max\{L^*(0, o, R), t_n + d_{0n}/k\}$. The proof is completed. \square

In the following analysis, we focus on the worst case, as the competitive ratio is the upper bound of the online RP algorithm. Furthermore, we consider the metric space to be the real line.

Theorem 2. For the online delivery problem, the competitive ratio of the proposed online RP algorithm is 2.5.

Proof of Theorem 2. Suppose that a new request $r_n = (t_n, x_n, y_n)$ is released. The worst case is that drone A, the truck, and drone B all have unserved requests at time t_n . For the online server, we suppose that $U = \{u_1, u_2, \dots, u_n\}$ is the service duration set of unserved requests for drone A at time $t_n - \delta$, where δ is an arbitrarily small positive number. For the offline server, we assume that $W = \{w_1, w_2, \dots, w_n\}$ is a service duration set of unserved requests for drone A. As the offline algorithm is invoked, for the same input, the offline and online servers have the same arrangement at time t_n , such that $U \subseteq W$. At this time, we have $T_d \leq t_n + \sum_{r_i \in \emptyset} u_i + \{2d_{0n}\}/k$, $u_i \in U$. According to Lemma 2, we know that $t_n + d_{0n}/k \leq C_{OPT}(R)$. As $C_{OPT}(R) \geq \sum_{i=1}^n w_i \geq \sum_{r_i \in \emptyset} u_i$, $d_{0n}/k \leq \frac{1}{2}C_{OPT}(R)$, there exists $T_d \leq 2.5C_{OPT}(R)$. At this time, we know that $T_{fs} \leq t_n + L^*(t_n, P_v(t_n), \sigma)$. According to Lemma 1, we know that $T_{fs} \leq t_n + L^*(0, o, \sigma) + \max_{i \in \sigma} \{d\{o, P_v(t_i)\}\}$. As $\max_{i \in \sigma} \{d\{o, P_v(t_i)\}\} \leq \frac{1}{2}C_{OPT}(R)$, we can obtain $T_{fs} \leq 2.5C_{OPT}(R)$. Thus, we have $C_{online}(R) \leq 2.5C_{OPT}(R)$. \square

In summary, the upper bound of the online RP algorithm for the problem is 2.5, and the proof is completed.

We also compared our research with the existing literature, as detailed in Table 4. We obtained a smaller gap between the lower bound and competitive ratio compared with Dai et al. [20]. Due to the different assumptions, our results can be seen as complementary to the studies of Dai et al. [20] and Ausiello et al. [15].

Table 4. Comparison of competitive ratio results.

Research	Difference	Lower Bound	Competitive Ratio
Ausiello et al. [15]	Homing Single vehicle No time window Traveling salesman problem	$2 - \epsilon, \forall \epsilon \geq 0$	2
Ascheuer et al. [19]	Homing Single vehicle No time window Dial-a-ride problem	2	2.5
Dai et al. [20]	Nomadic Multiple vehicles Time window Vehicle routing problem	2	3.5
This work	Homing Truck and drones No time window Vehicle routing problem	1.5	2.5

6. Offline Algorithm and Simulation Study

Notice that the online RP algorithm invokes an offline algorithm, while the corresponding offline problem is NP-hard. Therefore, we designed an offline drone priority algorithm for the offline model described in Section 3.2, and analyzed its time complexity and verified its effectiveness.

6.1. Offline Algorithm

In this section, we describe a heuristic algorithm to solve the corresponding offline problem.

Offline drone priority algorithm: Drone A serves all requests within the service scope. For requests beyond the service scope, through the elbow method, we obtain the optimal number of clustering centers and clusters through the k -means clustering method. Drone B is preferentially assigned to serve, compared with the truck. The remaining requests beyond the service scope of drone B and clustering centers are assigned to the truck. The truck serves these requests according to the request release time sequence.

We provide the pseudocode of the whole offline drone priority algorithm, in three parts, in Appendix A. The related parameters involved in the pseudocode are described in Table 3.

First, we give pseudocode for calculating the optimal clustering center number by the elbow method in order to prepare for the k -means clustering method, the pseudocode of which is shown in Algorithm A1. Then, we cluster these requests using the k -means clustering method and arrange requests, the pseudocode of which is shown in Algorithm A2. Finally, we provide the pseudocode of the main procedure based on the above two algorithms, as shown in Algorithm A3.

6.2. Performance Analysis for Offline Drone Priority Algorithm

Furthermore, we analyzed the performance of the offline drone priority algorithm in terms of time complexity and compared it with the CPLEX solver through small-scale examples.

Time complexity analysis: The time complexity of the offline drone priority algorithm is $O(n^3)$. Assuming the number of requests is n , the time complexity of this algorithm is mainly determined according to Algorithms A1 and A2. In Algorithm A1, the time complexity of the k -means algorithm is $O(n^2 + n)$, the time complexity of calculating the SSE is $O(n^2)$, and the time complexity of other codes is constant, such that the time complexity of this part is $O(n^3)$. In Algorithm A2, the time complexity is $O(k \cdot n + 2n)$,

where $k \leq n$. Therefore, according to the analyses of Algorithms A1 and A2, the time complexity of the drone priority algorithm is $O(n^3)$.

Aiming at the same input, the effectiveness of the offline drone priority algorithm was analyzed by comparison with the CPLEX solution results. The initial settings were 1 for the number of trucks, 2 for the number of drones, 5 for the drone service radius, and 50 for the upper bound of the release time. We generated new instances randomly, considering these initial settings. In every instance, half of the requests were outside the service scope of drone A. The comparison results are given in Table 5.

We found that the offline algorithm took a shorter time, and the maximal gap was 3.56%, which is within an acceptable range. When the number of requests reached 100, the CPLEX duration exceeded the default time limit. In contrast, we verified that the offline drone priority algorithm still worked when the number of requests increased to 200.

Table 5. Comparison of offline drone priority algorithm and CPLEX solution results.

Request Number	Algorithm Result	Time (s)	CPLEX Result	Time (s)	Gap
30	60.6301	3.241	60.6301	88.799	0.00%
32	55.0623	3.757	54.6569	108.515	0.74%
34	62.8062	4.402	60.8310	120.073	3.24%
36	62.0416	5.062	62.0416	135.207	0.00%
38	57.6711	5.750	57.6023	142.103	0.11%
40	60.8489	6.601	60.4403	156.527	0.67%
42	58.5440	7.445	58.5440	201.089	0.00%
44	61.8062	8.375	61.8062	238.308	0.00%
46	58.8102	9.501	58.2195	334.681	1.01%
48	58.2195	11.033	56.4721	397.740	3.09%
50	58.0554	12.252	56.0554	428.395	3.56%
100	58.5440	25.375	—	7201.442	—
200	54.4721	142.546	—	—	—

7. Online Algorithm Simulation Analysis

The coordinates and release times of requests were randomly generated, and the ratio of online RP algorithm results divided by offline drone priority algorithm results was calculated in order to measure the stability of the online RP algorithm. All computational work was conducted on a Macbook Air desktop with an Apple M1 processor and 8 GB RAM, using MATLAB 2018a software.

Based on a symmetric network, we used the Euclidean distance as a calculation standard. The initial settings were as follows: Number of requests, 20; drone to truck speed ratio, 2; ratio of network radius to drone service radius, 2; and upper bound of release time, 50. For the same input, the online RP algorithm and offline drone priority algorithm were independently tested 50 times, and the results were averaged, as shown in Table 6. As the online RP algorithm competitive ratio obtained in Section 5 was based on the worst case, and the input parameters do not necessarily reflect the worst case, it is reasonable that the actual ratio obtained was less than the competitive ratio.

The higher the ratio, the worse the result. According to the simulation, we came to the following conclusions:

- (1) With changes in all kinds of parameters, the ratios were far below the competitive ratio of the online RP algorithm. We concluded that the online algorithm is effective and steady in realistic scenarios.
- (2) We observed some correlations between the parameters and the performance of the online RP algorithm. In particular, the performance of the online RP algorithm was negatively correlated with the number of requests and positively correlated with the upper bound of the release time.
- (3) In real scenarios, we advise not to select the drone type with flight radius close to half of the network radius; instead, it is better to select a drone with a larger flight radius.

In addition, better performance can be obtained in scenarios with fewer request points and more concentrated request generation time.

Table 6. Simulation results.

Parameter	Value	MRPAR ¹	MDPAR ²	Ratio
Request number	20	84.951446	62.117312	1.377449153
	40	161.918002	107.029284	1.532552400
	60	238.211708	156.728344	1.563810532
	80	319.624538	206.740620	1.565059287
	100	389.227470	237.722238	1.652084789
Drone to truck speed ratio	1.25	86.614150	64.196994	1.356576443
	1.5	86.075648	62.864068	1.376177388
	2	87.834754	64.728844	1.371517126
	2.25	85.221996	62.725790	1.364440938
Upper bound of the release time	50	86.396640	60.639298	1.426099299
	100	106.519612	103.960546	1.025561585
	150	151.800832	149.332184	1.016672049
Ratio of network radius to drone service radius	1	57.757144	52.006772	1.112815111
	2	88.207810	62.945100	1.407709065
	3	118.065572	91.626774	1.296150275
	4	154.846792	127.343768	1.227915656

¹ Mean of online RP algorithm results; ² Mean of offline drone priority algorithm results.

8. Comparative Study of HTDCS and Traditional Truck-Only Delivery System

In this section, we compare the minimum latest time between the HTDCS and a traditional truck-only delivery system using an offline server.

Regarding the input information, we used partial actual data. Workhouse has used a Horsefly drone collaborating with a Workhorse electric truck to deliver packages in January 2021 [21]. Each drone's capacity was 10 pounds, their maximum speed was 46 miles per hour, and the maximum flight time was 25 min. The slow speed of drones is due to FAA regulations. In fact, Wingcopter has designed a delivery drone that can reach speeds of 150 miles per hour [22]. The FAA rules for commercial drones are expected to tend to loosen up gradually [23]. In this paper, we assume that the FAA regulations will be relaxed in the future, so the speed of each drone was set to 100 miles per hour. Each electric truck's capacity is 5000 pounds, and speed is 68 miles per hour, and the travel range is 150 miles. Based on the actual data, we assumed the network radius to be 24 miles ($2\pi r = 150$) and the flight radius to be 9.6 miles ($(25/60) \times 46/2$). Then, we set the ratio of the network radius to the drone service radius as 2.5 and the drone to truck speed ratio as 1.5 (100/68).

Some input information was generated randomly. We assumed that each request's demand was 10 pounds while the release times and locations were randomly generated. We assume that the number of trucks was 1 and the number of drones was 2. The upper bound of the release time was 100. In every instance, half of the requests were outside the service scope of drone A. The minimum latest time was calculated by the offline drone priority algorithm. For the same input, we compared the minimum latest time with or without drones, as detailed in Table 7.

From the table, we can see that the gap was zero when the request number was 20. Then, the gap widens until the request number reaches 60. When the request number is greater than 60, the gap fluctuates between -30% and -40% . It is obvious that the truck–drone collaboration delivery system can greatly lower the latest time saving compared with the truck-only delivery system. For emergency and instant deliveries, improving the minimum latest time saving can improve customer satisfaction and attract more customers.

Table 7. Comparison of minimum latest time.

Request Number	HTDCS	Truck-Only Delivery System	Gap (%)
20	101.1598	101.1598	0
40	103.3815	143.5081	−27.96
60	183.2385	264.1488	−30.63
80	186.0301	285.4602	−34.83
100	258.0078	392.4112	−34.25
120	333.8981	502.4427	−33.55
140	362.0530	517.5884	−30.05
160	357.7037	577.5909	−38.07
180	442.0229	694.5653	−36.36
200	489.0484	780.1884	−37.32

9. Conclusions

To meet emergency delivery or commerce instant gratification needs, real-time delivery is a key factor. In a real scenario, an online server only knows a request after it is released. Considering the real-time request, in this paper, we propose an online delivery problem for a hybrid truck–drone collaboration system, including independent and truck-carried drones. Considering real-time requests to the hybrid truck–drone collaboration delivery system, the lower bound for this problem was proven to be 1.5 by competitive analysis, while the worst-case ratio of the online RP algorithm was shown to be 2.5. We also compared our proposed design with existing approaches in the literature, and we obtained a smaller gap between the lower bound and competitive ratio. As the online algorithm is based on the plan-at-home idea, an offline drone priority algorithm was also designed. We constructed a corresponding offline model and designed a heuristic algorithm. Through comparison with CPLEX solution results, we found that the proposed offline drone priority algorithm was faster, and the gap was within a reasonable range. By adjusting input parameters, we verified the stability of the online RP algorithm and provided some pertinent conclusions. Finally, we used partial actual data to verify the minimum latest time saving obtained by the HTDCS, compared with a truck-only delivery system.

This research provides theoretical support for online delivery using an HTDCS. This better corresponds to realistic scenarios compared with existing approaches. Furthermore, our results can be seen as providing a complementary view to online delivery problems, which is necessary for the development of better online algorithms with lower competitive ratios. In future work, we intend to design a new online algorithm and compare the associated competitive ratios. Furthermore, the pickup and delivery problem under the HTDCS is still interesting, which should make full use of the vehicle capacity.

Author Contributions: M.G.: Conceptualization, Methodology, Software, Validation, Visualization, Formal analysis, Resources, Investigation, Writing—original draft, Writing—review and editing. H.Y.: Methodology, Formal analysis, Writing—original draft, Writing—review and editing, Supervision, Funding acquisition. All the authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the anonymous referees for the constructive comments. This research was supported by the National Natural Science Foundation of China (No. 71702016), Humanities and Social Science Research Project of Education Commission (No. 21YJC630159, 21YJC630138), Humanities and Social Science Research Project of Chongqing Education Commission (No. 22SKJD092).

Data Availability Statement: The data sets analyzed in the current study were randomly generated by setting some parameters. All parameters used in this study are included in this published article. The resulting data sets are available from the corresponding author on reasonable request.

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

Appendix A. Pseudocode of Algorithm

Algorithm A1 Pseudocode for calculating the optimal clustering center number.

Input: $|R|, S_{\text{semeanx}}$

Output: k

```

1: for  $i = 1 : n$  do
2:   Generate request coordinates, according to  $|R|$ , then generate a distance matrix  $D'$ , according to these
   coordinates
3:   Calculate  $\text{newcenter}$  according to the value of  $i$  and  $D'$  by  $k$ -means clustering method
4:    $a = \text{cell}(\text{size}(\text{newcenter}, 1), 1)$ 
5:   for  $u = 1 : \text{size}(\text{newcenter}, 1)$  do
6:     for  $v = 1 : n$  do
7:       if  $r_v \in c_u$  then  $a\{u\} = [a\{u\}; \text{coordinate}(v)]$ 
8:     endif
9:   endfor
10:  endfor
11:  for  $u = 1 : \text{size}(\text{newcenter}, 1)$  do
12:    Calculate the mean value of the  $u^{\text{th}}$  element in the set  $a$ , record it as  $M_u$ 
13:    for  $v = 1 : n$  do
14:      if  $r_v \in a\{u\}$  then
15:        Calculate corresponding coordinate difference between  $r_v$  and  $M_u$ , record its square as the  $u^{\text{th}}$ 
        element in  $DF$ 
16:      endif
17:      Sum over all the values in the set  $DF$ 
18:    endfor
19:  endfor
20:   $S_{\text{semany}}(i) = \text{mean}(DF)$ 
21: endfor
22: Generate discrete points, according to  $S_{\text{semany}}$  as the Y coordinate and  $S_{\text{semeanx}}$  as the X coordinate
23: Obtain the derivative of these discrete points by Taylor formula, set a threshold, and record the inflection
    point as  $k$ 

```

Algorithm A2 Pseudocode for clustering and arranging requests.

Input: D, k, s

Output: D_A, D_B, D_T

```

1: for  $i = 1 : n$  do
2:   if  $D(1, i) \geq s$  then  $S_Y \leftarrow r_i$ 
3:   else  $D_A \leftarrow r_i$ 
4:   endif
5: endfor
6: Calculate  $\text{newcenter}$  according to the value of  $k$  and  $D$  by  $k$ -means clustering method
7: Find the request nearest to the center as the truck parking node
8: for  $i = 1 : n$  do
9:   for  $j = 1 : k$  do
10:    if  $r_i \in c_j$  then
11:      Calculate the distance between each request and the parking node
12:      if  $\text{distance} > s$  then  $D_T \leftarrow r_i$ 
13:      else  $D_B \leftarrow r_i$ 
14:    endif
15:  endfor
16: endfor
17: endfor
18: for  $i = 1 : n$  do
19:   if  $r_i \notin D_A \cup D_B$  then  $D_T \leftarrow r_i$ 
20:   endif
21: endfor

```

Algorithm A3 Pseudocode for offline drone priority algorithm.

Input: $R, s, |R|$

Output: $D_A, D_B, D_T, T_f, T_d, T$

```

1: Generate distance matrix  $D$  according to the set  $R$ 
2: Generate  $S_{\text{semeanx}}$  according to  $|R|$ 
3: Invoke Algorithm A1
4: Invoke Algorithm A2
5: Calculate  $T_f, T_d, T$  according to  $D_A, D_B, D_T$ 

```

References

1. Salama, M.R.; Srinivas, S. Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102788. [CrossRef]
2. Walmart. Available online: <https://corporate.walmart.com/newsroom/2020/09/14/walmart-and-zipline-team-up-to-bring-first-of-its-kind-drone-delivery-service-to-the-united-states> (accessed on 26 December 2022).
3. Interdrone. Available online: <https://interdrone.com/news/the-faa-predicts-commercial-drone-market-could-triple-by-2023/> (accessed on 26 December 2022).
4. Sung, H.C.; Sah, B.; Lee, J. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Comput. Oper. Res.* **2020**, *123*, 105004–105033.
5. Wang, D.; Hu, P.; Du, J.; Hu, M. Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. *IEEE Internet Things* **2019**, *6*, 10483–10495. [CrossRef]
6. Wang, C.; Lan, H. An Expressway Based TSP Model for Vehicle Delivery Service Coordinated with Truck + UAV. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 307–311.
7. Schermer, D.; Moeini, M.; Wendt, O. The Drone-Assisted Traveling Salesman Problem with Robot Stations. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020; HICSS: Waikoloa, HI, USA, 2020; pp. 1308–1317.
8. Yu, S.; Puchinger, J.; Sun, S. Van-based robot hybrid pickup and delivery routing problem. *Eur. J. Oper. Res.* **2022**, *298*, 894–914. [CrossRef]
9. Rave, A.; Fontaine, P.; Kuhn, H. Drone location and vehicle fleet planning with trucks and aerial drones. *Eur. J. Oper. Res.* **2022**, *accepted*. [CrossRef]
10. Kloster, K.; Moeini, M.; Vigo, D.; Wendt, O. The multiple traveling salesman problem in presence of drone- and robot-supported packet stations. *Eur. J. Oper. Res.* **2023**, *305*, 630–643. [CrossRef]
11. Troudi, A.; Addouche, S.A.; Dellagi, S.; Mhammedi, A. Sizing of the drone delivery fleet considering energy autonomy. *Sustainability* **2018**, *10*, 3344. [CrossRef]
12. Tang, M.; Ji, B.; Fang, X.; Yu, S.S. Discretization-strategy-based solution for berth allocation and quay crane assignment problem. *J. Mar. Sci.* **2022**, *10*, 495. [CrossRef]
13. Fanjul-Peyro, L.; Ruiz, R.; Perea, F. Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. *Comput. Oper. Res.* **2019**, *101*, 173–182. [CrossRef]
14. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2019**, *122*, 350–364. [CrossRef]
15. Ausiello, G.; Feuerstein, E.; Leonardi, S.; Stougie, L.; Talamo, M. Algorithms for the on-Line travelling salesman. *Algorithmica* **2001**, *29*, 560–581. [CrossRef]
16. Yu, W.; Liu, Z.; Bao, X. Optimal deterministic algorithms for some variants of online quota traveling salesman problem. *Eur. J. Oper. Res.* **2014**, *238*, 735–740. [CrossRef]
17. Wen, X.; Xu, Y.; Zhang, H. Online traveling salesman problem with deadlines and service flexibility. *J. Comb. Optim.* **2015**, *30*, 545–562. [CrossRef]
18. Jaillet, P.; Wagner, M.R. Online routing problems: Value of advanced information as improved competitive ratio. *Transp. Sci.* **2006**, *40*, 133–258. [CrossRef]
19. Ascheuer, N.; Krumke, S.O.; Rambau, J. Online Dial-a-Ride Problems: Minimizing the Completion Time. In Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, 15–17 February 2001; Springer: Paris, France, 2000; pp. 639–650.
20. Dai, M.; Xu, Y.; Dong, Y.; Du, Y. A competitive analysis for the open online trucks scheduling problem with time window. *Syst. Engine* **2006**, *4*, 93–96. (In Chinese)
21. Workhorse. Available online: <https://workhorse.com/w750/> (accessed on 26 December 2022).
22. Techtimes. Available online: <https://www.techtimes.com/articles/248356/20200326/are-delivery-drones-coming-ups-is-getting-closer-to-making-it-possible.htm> (accessed on 26 December 2022).
23. Natlawreview. Available online: <https://www.natlawreview.com/article/5-trends-to-watch-2022-drones> (accessed on 26 December 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.