Computers & Industrial Engineering 189 (2024) 109963

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Pseudo node insertion method for synchronization in drone–truck combined operations

Sung Hoon Chung [a,*], Bhawesh Sah [b], Jinkun Lee [c]

[a] Department of Systems Science and Industrial Engineering, Binghamton University, United States of America
[b] D'amore Mckim School of Business, Northeastern University, United States of America
[c] Department of Engineering, East Carolina University, United States of America

ARTICLE INFO

Keywords:
Transportation
Drone–truck combined operations
Pseudo node insertion
Vehicle routing problem
Last-mile logistics

ABSTRACT

Drone–truck combined last-mile delivery systems present several advantages, which can be summarized in two folds: (1) enabling faster pick-up/delivery of items and (2) expanding operation areas beyond the range that could be achieved by using only one type of vehicle. Drone–truck combined operation (DTCO) solutions are based on vehicle routes and drone launch and retrieval locations, usually customer nodes. Since a perfect synchronization of both vehicles is very rare, most of the solutions obtained after designing combined delivery routes generally contain time losses, caused by the need for the first vehicle that reaches the meeting point to wait until the arrival of the other. In this paper, a pseudo node insertion method is proposed to make DTCO solutions more efficient by improving the truck–drone synchronization operations. To this end, starting from a given routing solution, pseudo nodes are created at the point where a drone and a truck can meet simultaneously. We present a detailed method to calculate the location of the pseudo nodes and provide a detailed analysis of the conditions that allow us to reduce the travel completion time for different route patterns. Furthermore, we explore a variety of scenarios to illustrate the efficiency of the pseudo node insertion method for different DTCO problem solutions. An algorithm for the pseudo node insertion method is provided, and numerical examples are presented to discuss the efficacy and efficiency of the proposed approach.

1. Introduction

The increasing attention to drone operation and drone–truck combined operation (DTCO) has been observed in the several application areas such as parcel delivery (Boysen et al., 2018; Carlsson & Song, 2017; Goodchild & Toy, 2018; Karak & Abdelghany, 2019; Murray & Chu, 2015; Murray & Raj, 2020; Poikonen & Golden, 2020; Ponza, 2016; Schermer et al., 2019b), healthcare (Kelleher, 2019; Kim et al., 2017; Scott & Scott, 2017, 2018), humanitarian logistics and disaster management (Andrei, 2017; Marin, 2016; Sandvik & Lohne, 2014), military operations (Samad et al., 2007; Xia et al., 2017), environment monitoring (Xia et al., 2019), etc. For more information on recent drone and DTCO related research, we refer interested readers to survey papers such as Otto et al. (2018), Khoufi et al. (2019), and Chung et al. (2020). In particular, the use of drones in the last-mile delivery, a labor-intensive and hence expensive logistics operation, has received substantial attention to improve its efficiency (Boyer et al., 2009). Indeed, companies such as Amazon, DHL, Google, and UPS have been preparing for a new era that will transform the current logistics system by a wide adoption of drone fleet operations. For example, Amazon has been testing a drone delivery with Amazon Prime Air Service since 2016 (Johnson, 2017). DHL has been operating an autonomous drone delivery system since 2016 (Burgess, 2016). Matternet has received an authorization for a drone logistics operation in 2017 (Ong, 2017), and UPS has tested their own drone–truck delivery system in 2017 where the truck not only delivers its own parcel but also serves as a carrier for the drone (Kastrenakes, 2017). It is worth noting that AHA, Iceland's eCommerce company, has partnered with Flytrex, a drone delivery company, and has significantly cut down its parcel delivery time by using drones as a sidekick for truck-based parcel delivery (Shu, 2017).

Drones have an advantage over trucks in a delivery speed with less constraints in route choices but have a disadvantage with a delivery capacity and range (Chung et al., 2020). The drone–truck mixed fleets have significant potential for achieving more effective and efficient last-mile delivery services if vehicle routes are carefully designed and drone launch and retrieval locations are optimally chosen. Therefore, DTCO has received a surge of interest from researchers and practitioners over the last few years and a sizable amount of research has been conducted: see, e.g., Agatz et al. (2018), Bouman et al. (2018), Ferrandez et al.

* Correspondence to: Binghamton University, 4400 Vestal Parkway East, Binghamton, NY 13902, United States of America.
E-mail addresses: schung@binghamton.edu (S.H. Chung), b.sah@northeastern.edu (B. Sah), leejin18@ecu.edu (J. Lee).

https://doi.org/10.1016/j.cie.2024.109963
Received 7 May 2023; Received in revised form 11 November 2023; Accepted 3 February 2024
Available online 6 February 2024
0360-8352/

(2016), Ha et al. (2015, 2018), Murray and Chu (2015), Murray and Raj (2020), Poikonen and Golden (2019, 2020), Ponza (2016), Savuran and Karakaya (2015), Schermer et al. (2019b), Wang et al. (2017), Yurek and Ozmutlu (2018). In particular, Murray and Chu (2015) tackle the drone–truck operation using the concept of a traveling salesman problem with drone (TSPD), propose mathematical models and methods to find optimal drone and truck routes for a parcel delivery, and name it the flying sidekick traveling salesman problem (FSTSP). Agatz et al. (2018) present an integer programming model for TSPD, and propose a heuristic method based on a local search and dynamic programming. A vehicle routing problem with drones (VRPD), introduced by Wang et al. (2017), Wang and Sheu (2019) and Poikonen et al. (2017), could be seen as an extension of TSPD in a similar fashion as a vehicle routing problem (VRP) can be viewed as an extension of TSP. VRPD (Wang & Sheu, 2019) considers a fleet of m homogeneous trucks with k drones (multiple drones per truck) where the speed of a drone is α times the speed of a truck, and drones can depart from and return to the truck at any customer locations. For a thorough review of recent DTCO research, see Chung et al. (2020).

In DTCO, a vehicle (either a drone or a truck) may need to wait for its counterpart at drone retrieval locations to work in tandem, as both vehicles do not necessarily arrive at the same time. Such wait time increases the total travel completion time and it is indeed problematic not only for DTCO but also for general vehicle routing (without drones) problems that involve multiple vehicles. Drexl (2012) provides a review of multiple-vehicle synchronization studies where synchronization classification, identification of central issues, review of application areas and methodologies are presented. DTCO concerns three synchronization constraints: (1) task synchronization: each customer must be served exactly once either by a drone or a truck, (2) operation synchronization: a drone has to meet a truck after one operation is completed before the next operation begins. Therefore, the timing of the next operation depends on the rendezvous time of two vehicles in the previous operation, and (3) movement synchronization: a drone has to move along a truck when carried by the truck.

The asynchronization issue in DTCO arises from hard constraints of allowing drone–truck rendezvous only at customer nodes or pre-defined mobile hubs/stations. This is particularly due to the fact that most TSP and VRP mathematical model formulations are based on nodes and therefore it is relatively easy to add constraints on nodes, rather than on edges or arcs. This asynchronization issue may be resolved if the launch and rendezvous points can be located on any feasible point along a truck route, which coins the term "en-route" operation. In this regard, Marinelli et al. (2017) suggest an arc-based heuristic to reduce a vehicle waiting time by considering a drone coverage range to find new en-route launch and rendezvous points. An initial node-based solution is obtained first and en-route points are calculated by their proposed heuristic algorithm at which a drone and a truck meet simultaneously. The objective is to minimize the makespan. The benefits of en-route operation are analyzed using several examples. This paper reports that 10% drone battery savings and 2 min waiting time reduction on average are achieved as compared to a non en-route operation counterpart for a 50 customer example. Schermer et al. (2019a) extend Marinelli et al. (2017) to include pre-defined en-route points, in addition to existing customer nodes, in the MILP formulation. Numerical studies with several en-route points added report an improved (reduced) makespan in general with a few exceptions. However, the authors also note that the introduction of multiple en-route points may significantly increase computational complexity. Note also that some studies, e.g., Carlsson and Song (2017), Karak and Abdelghany (2019), Wang and Sheu (2019) propose the use of mobile hubs or stations, which are not customer nodes, to launch and retrieve drones to resolve the asyncronization issue; however, such locations must be pre-determined and may require substantial resources to maintain.

In this paper, the en-route drone retrieval problem is explicitly studied using the idea of adding pseudo nodes where a drone and a truck can arrive simultaneously. A pseudo node is defined as a geographical locus, which is not a customer location nor a depot, within a road segment where a drone can depart from and rendezvous with a truck. To insert a pseudo node, we adopt and extend the concept of the drone sortie, a drone–truck combined operation unit, defined by Murray and Chu (2015). The detailed illustration of the drone sortie and the pseudo node insertion method are provided in Section 3. The contribution of this paper is summarized as follows:

1. Unlike the previous studies that rely on a simple heuristic to find pseudo nodes for en-route operations, a detailed method to calculate the location of the pseudo node is provided. This is based on the notion of drone sortie, which will be a basic building block for the entire en-route operation.
2. A detailed analysis for travel completion time saving conditions is provided to further enhance the understanding of the effect of pseudo node addition. This includes (1) the introduction of drone-waiting sortie, truck-waiting sortie, and perfect sortie, (2) the methodology to locate pseudo nodes under different combinations of such sorties, and (3) the study for maximum saving conditions.
3. Furthermore, a variety of scenarios (e.g., one-sortie case, two-consecutive-sortie case, and three-or-more-consecutive sortie case) are explored to enhance the overall efficiency of DTCO by way of inserting pseudo nodes. That is, the optimal way of inserting pseudo nodes is studied, which is the basis for the proposed pseudo node insertion heuristic.
4. Two heuristics (pseudo node insertion heuristic, and modified route and re-assign heuristic) are proposed, and numerical examples are presented to discuss efficacy and efficiency of our proposed approach.

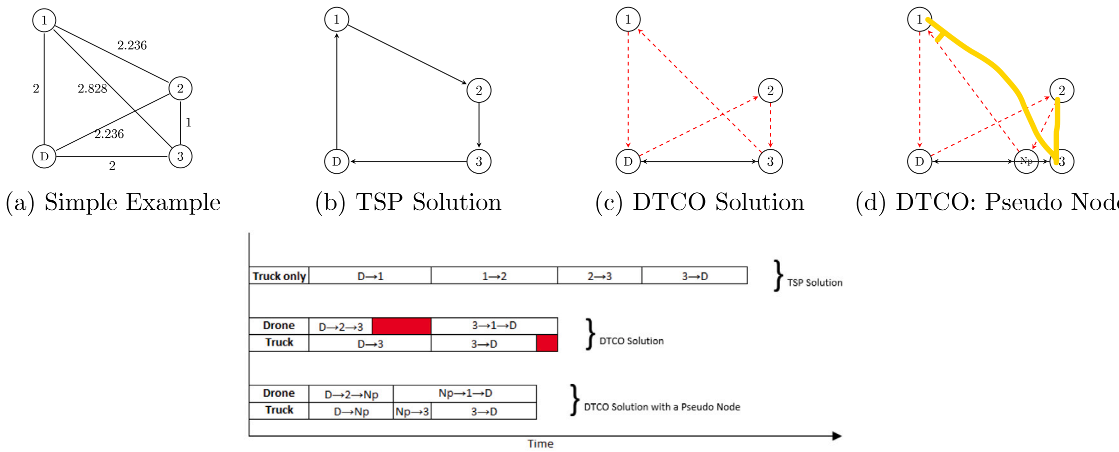
In summary, we propose a systematic en-route drone retrieval method by way of pseudo node insertion to minimize the total travel completion time in the last-mile delivery problem.

The organization of this paper is as follows. We introduce the basics of DTCO with pseudo nodes in Section 2. The detailed pseudo node insertion method is provided in Section 3, and the heuristics are presented in Section 4. Section 5 shows numerical example results and illustrates how DTCO solutions can be improved using our proposed method. Finally, Section 6 concludes the paper.

2. Illustrative example of DTCO with a pseudo node

In DTCO, some parcels are delivered by a drone while the others are delivered by a truck where the drone and the truck work in tandem. The objective of DTCO is to minimize the total travel time, travel completion time/makespan, or total cost, while satisfying delivery requirements.

To illustrate how DTCO works, let us present a simple example shown in Fig. 1. Suppose that a drone and a truck are at node D (depot). A road segment is represented by an arc whose label indicates a distance between the nodes in miles. Customers are located at nodes 1, 2, and 3 to which a parcel needs to be delivered. It is assumed that the drone speed (0.5 mile per minute) is twice as fast as the truck (0.25 mile per minute), and the travel completion time is to be minimized. When only a truck is used, the optimal route is (D → 1 → 2 → 3 → D), as shown in Fig. 1(b), and it takes 28.944 min to complete the travel. The DTCO solution of this problem is given in Fig. 1(c) where the drone route is shown by dashed red lines (D → 2 → 3 → 1 → D) and the truck route (D → 3 → D) is shown by a solid black line. At node D, the drone departs from the truck to serve a customer at node 2 and the truck goes to node 3. The drone, after serving node 2 customer, flies to node 3 to meet with the truck. Note that the drone has to wait at node 3 until the truck arrives because it takes 6.472 min for the drone to go to node 3 from D via 2 but it takes 8 min for the truck to go to node 3



(a) Simple Example (b) TSP Solution (c) DTCO Solution (d) DTCO: Pseudo Node

(e) Comparison of Delivery Schedules (Red Box: Waiting Time)

Fig. 1. Illustration of DTCO with a pseudo node.

from D (ignoring the service time at node 2). In some other situations, however, a truck may have to wait until a drone arrives. Such waiting times are shown as red rectangles in Fig. 1(e). In the sequel, the drone departs again from the truck at node 3 to serve a customer at node 1, then goes to the depot to meet the truck, while the truck goes back to the depot. The DTCO solution is better than the truck only (TSP) solution for the travel completion time as shown in Fig. 1(e).

Now suppose that a pseudo node can be inserted on the arc connecting node D and 3, denoted by $\mathrm{N_p}$, as shown in Fig. 1(d), where the drone and the truck can meet simultaneously to reduce the waiting time. Such pseudo node is not a customer location but assumed to be a geographical locus where a truck can be safely stopped to retrieve the drone. In fact, a truck may not need to stop because a rendezvous may be feasible while both vehicles are moving (EasyAerial, 2019). The solution with a pseudo node is given in Fig. 1(d) where the drone route is shown by dashed red lines (D → 2 → $\mathrm{N_p}$ → 1 → D) and the truck route (D → $\mathrm{N_p}$ → 3 → D) is shown by solid black lines. It takes 16 min to complete the travel, which is a 9.38% improvement over the DTCO solution (17.656 min). Note that the total truck/drone waiting time may increase substantially as the size of problem increases. Yurek and Ozmutlu (2018) and Ha et al. (2018) show in their numerical examples that the waiting time (either by a drone or a truck) indeed increases as the size of problem increases. Therefore, the use of pseudo nodes has great potential to improve DTCO, which is verified in numerical examples in Section 5.

3. Pseudo node insertion method

In this section, we provide the detailed pseudo node insertion method to improve the DTCO solution. Most DTCO models in the literature are based on the assumption that a drone can depart from and rendezvous with a truck only at specified customer nodes. Some studies (Marinelli et al., 2017; Schermer et al., 2019a) provide the idea of en-route drone retrieval but rely on a simple heuristic. We not only relax the assumption but also provide detailed methods of adding pseudo nodes, and provide a thorough analysis of the implication of the pseudo node insertion method in DTCO.

3.1. Sortie definition

We consider a single drone and a single truck to introduce a drone sortie. Let i, j, and k be an index for nodes, $C = \{1, 2, \ldots, c\}$ be the set of all customer nodes, C' be the set of the customer nodes that are eligible to be serviced by a drone such that $C' \subseteq C$, $N_0 = \{0, 1, \ldots, c\}$ be the set of nodes from which a drone can depart where 0 represents the depot, and $N_+ = \{1, 2, \ldots, c+1\}$ be the set of nodes at which a drone can be retrieved where $c+1$ is used for notational convenience to represent the depot, i.e., note $c+1$ is the same as node 0. In addition, let τ'_{ij} and τ_{ij} be the travel time from node i to j for a drone and a truck, respectively, and e be the maximum flight time (flight range in the unit of time or distance) for a drone. The notion of a drone sortie that plays a significant role in DTCO is introduced by Murray and Chu (2015), which is illustrated in Fig. 2. A drone departs from a truck at the start node i, which can be a depot or a customer location, and goes to a drone-eligible customer node j, then rendezvouses with a truck at the end node k. In the meanwhile, a truck travels[1] from node i to node k. The set of drone sorties P can be defined as follows: $P = \{(i, j, k) : i \in N_0, j \in C', k \in N_+, i \neq j, j \neq k, k \neq i, \tau'_{ij} + \tau'_{jk} \leq e\}$

For our pseudo node insertion approach, we extend the concept of this drone sortie, which will play an integral role for the proposed method. First of all, we introduce an index s for the sortie (i, j, k) that consists of nodes i, j, and k. In addition, the drone travel time for sortie s is denoted by $T_{d_s} \equiv \tau'_{ij} + \tau'_{jk}$. Similarly, the truck travel time for sortie s is $T_{t_s} \equiv \tau_{ik}$. In our extended drone sortie notion, all sorties are divided into three groups: (1) drone waiting sortie (DWS) where $T_{d_s} < T_{t_s}$, (2) truck waiting sortie (TWS) where $T_{d_s} > T_{t_s}$, and (3) perfect sortie (PS) where $T_{d_s} = T_{t_s}$. That is, DWS is a drone sortie in which a drone needs to wait for a truck at the rendezvous location k. On the other hand, a truck has to wait for a drone at k in TWS, while both vehicles arrive at k at the same time in PS. Furthermore, we introduce W_{d_s} and W_{t_s} to represent the drone waiting time and the truck waiting time for sortie s, respectively, such that $W_{d_s} > 0$ for DWS, $W_{t_s} > 0$ for TWS, and $W_{d_s} = W_{t_s} = 0$ for PS. We also define the travel completion time for s as the latest arrival time at the sortie end node either by a drone or a truck.

With these extended notion of TWS, DWS, and PS, we discuss in the following sections the effectiveness of the pseudo node insertion

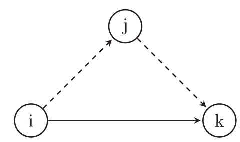


Fig. 2. Illustration for a drone sortie.

[1] There may be multiple nodes on the path connecting node i and k even though they are not shown in Fig. 2. For the study of multiple visiting drones, see Poikonen and Golden (2020).

(a) Perfect Sortie Case (b) Pseudo Node Candidates

Fig. 3. Pseudo node within PS.

method by considering three major cases: (1) one-sortie case, (2) two-consecutive-sortie case, and (3) three-or-more-consecutive-sortie case. For all such cases, the following assumptions are required to consider the location of pseudo nodes:

- A pseudo node may be located on a truck traveling arc and may not be located on a drone traveling arc.
- The speed of a drone, assumed to be constant, is α times faster than that of a truck ($\alpha > 0$). The truck speed is also assumed to be constant.
- The arc lengths in a sortie represent the travel distance of a drone and a truck, and the triangular inequality holds for each sortie.
- Only the travel time is considered, and all other miscellaneous times such as service time, battery swap time, and parcel preparation time are suppressed for the sake of simplicity; however, these can be added if needed.

3.2. One-sortie case

In this section, we focus on eliminating the waste time induced by either a drone or a truck waiting for the other vehicle at a rendezvous node considering a single drone sortie. The objective is to develop a systematic methodology for locating pseudo nodes. We attempt to gain insights out of the one-sortie case and will try to extend it to a two-sortie case, a three-or-more-sortie case, and, eventually, a general DTCO case. In the one-sortie case, we explore the following five scenarios: (1.1) pseudo node candidate located within PS, (1.2) pseudo node candidate located within TWS, (1.3) pseudo node candidate located outside TWS, (1.4) pseudo node candidate located within DWS, and (1.5) pseudo node candidate located outside DWS.

Scenario 1.1: Pseudo node candidate within PS

In this scenario, we examine if a pseudo node can be located within PS. In a sortie shown in Fig. 3(a), a truck travels from node 1 to node 2 (solid black line) while a drone departs from the truck at node 1, visits a customer at node 3, and flies to node 2 to be picked up by the truck (dashed red lines). In PS, the drone and the truck arrive at node 2 at the same time.

In this case, there is no need to insert pseudo nodes to reduce the vehicle (drone or truck) waiting time because no waiting time exists without pseudo nodes. However, as shown in Fig. 3(b), if pseudo nodes, represented by N_P, are used, the drone flying time (and also the battery usage) can be reduced (by late launch and early retrieval), while maintaining zero waiting time. In other words, these pseudo nodes can be particularly useful when the drone flying range is short.

The locations of pseudo nodes N_P can be found at the intersections of the truck arc and the drone ranges represented by the black dashed half circles centered at node 3 (Marinelli et al., 2017). The radii R_1, R_2, and R_3 shown in Fig. 3(b) are examples to illustrate a few possible drone flight ranges. That is, the half circles represent the contours of the given drone flying range. In PS, any point on the truck arc is feasible to locate a launch pseudo node (a retrieval pseudo node can be found using a circle centered at node 3) if it is within a drone flying range and there is no physical obstruction in the road.

Note that this is a graphical method. A numerical approach to find the locations of pseudo nodes is illustrated in scenario 1.3. Note also that pseudo nodes can be located outside PS (intersections of the truck route and the drone range represented by R_3 if the drone endurance allows). Note however that it may merely increase the drone flying time or distance (i.e., waste of its battery) with no benefit of reducing the waiting time.

Scenario 1.2: Pseudo node candidate within TWS

In this scenario, we consider the case where the truck arrives at node 2 before the drone, and waits to pick up the drone, i.e., the TWS case ($T_{d_s} > T_{t_s}$), as illustrated in Fig. 4(a). We examine the possibility of adding a pseudo node N_P on the truck arc between node 1 and node 2 as shown in Fig. 4(b). If this pseudo node N_P can be inserted and becomes a point that makes a simultaneous rendezvous, the truck will be able to pick up the drone at node N_P without any waiting time, and the drone will be immediately ready for the next trip. However, this scenario turns out to be infeasible. In Proposition 1, we prove that such a pseudo node does not exists if $\alpha > 1$ (A drone is faster than a truck).

Proposition 1. *If the speed of a drone is higher than that of a truck, i.e., $\alpha > 1$, a pseudo node for a synchronous rendezvous does not exist on the truck arc within TWS.*

The proof of Proposition 1 is provided in Appendix.

Scenario 1.3: Pseudo node candidate outside TWS

As we show that a pseudo node cannot be located on a truck arc within TWS if $\alpha > 1$, we explore the possibility of locating a pseudo node on a truck arc outside TWS. Fig. 5(b) illustrates the location of the pseudo node outside TWS. If such a pseudo node N_P exists, the truck does not need to wait for the drone at node 2 but moves on to its next customer node (node 4) and picks up the drone on the way. The drone will be ready for the next travel at the rendezvous point (pseudo node N_P), and the overall travel time may be reduced by eliminating the truck's waiting time, as depicted in Fig. 5(c) where the red rectangle represents the amount of truck waiting time.

Proposition 2. *If the speed of a drone is faster than that of a truck ($\alpha > 1$), a pseudo node for a synchronous rendezvous may exist on the truck arc outside TWS if a drone range allows.*

Proof. As shown in Fig. 5(b), the pseudo node may be located at the intersection of the truck arc and the drone range represented by R_3 if such intersection exists considering the drone endurance. □

The method of finding the coordinates of the pseudo node is explained in detail in Appendix.

Scenario 1.4: Pseudo node candidate within DWS

Now, we consider the case where the drone arrives at node 2 before the truck and waits to be picked up by the truck, i.e., the DWS case ($T_{d_s} < T_{t_s}$). As shown in Fig. 6(b), we explore the possibility of adding pseudo nodes N_P on the truck arc within DWS when $\alpha > 1$. If

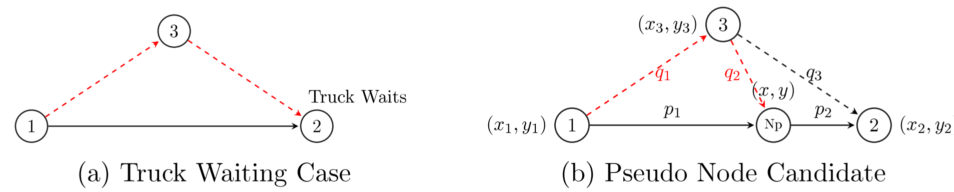


Fig. 4. Pseudo node within TWS.

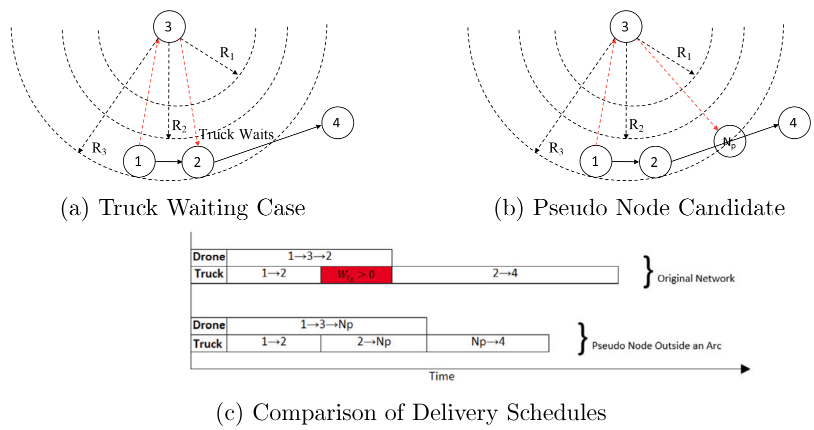


Fig. 5. Pseudo node outside TWS.

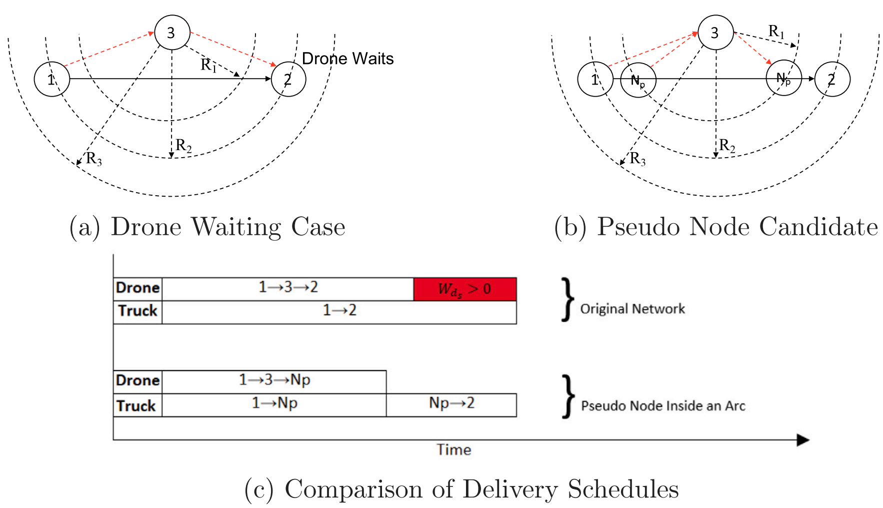


Fig. 6. Pseudo node within DWS.

these pseudo nodes can be inserted and become launch and rendezvous points with no waiting time, the drone will not only save the flying time but also be able to be picked up by the truck early (and will be ready for the next trip early, too). The travel completion time for this sortie remains the same as depicted in Fig. 6(c), because the truck travel time will stay the same. However, the overall travel completion time may be reduced by eliminating the drone's waiting time and releasing the drone for the next sortie early, especially when the next drone sortie is TWS. We explore this case in the next section. Note that the drone can depart from the truck at either node 1 or the left pseudo node (right next to node 1) in the DWS case, which can be decided considering the trade-off between the customer node launch and en-route launch.

Proposition 3. *If the speed of a drone is faster than that of a truck ($\alpha > 1$), a pseudo node for a synchronous rendezvous may exist within DWS.*

Proof. As shown in Fig. 6(b), the pseudo nodes may be located at the intersections of the truck arc and the drone range represented by R_1. As $\mathrm{R}_1 <$ the length of $(1, 3)$ or $(3, 2)$, the pseudo nodes $\mathrm{N_p}$ may exist. □

Scenario 1.5: Pseudo node candidate outside DWS

In DWS, the possibility of locating a pseudo node outside the DWS is examined. By Proposition 4, we prove that such a pseudo node does not exist.

Proposition 4. *If the speed of a drone is faster than that of a truck ($\alpha > 1$), a pseudo node for synchronous rendezvous does not exist on the truck arc outside DWS.*

Proof. In DWS, the drone waits for the truck at node 2 as shown in Fig. 6(a). Suppose that the drone already has arrived at node 2 while the truck has not. Now, if the drone departs from node 2 and moves

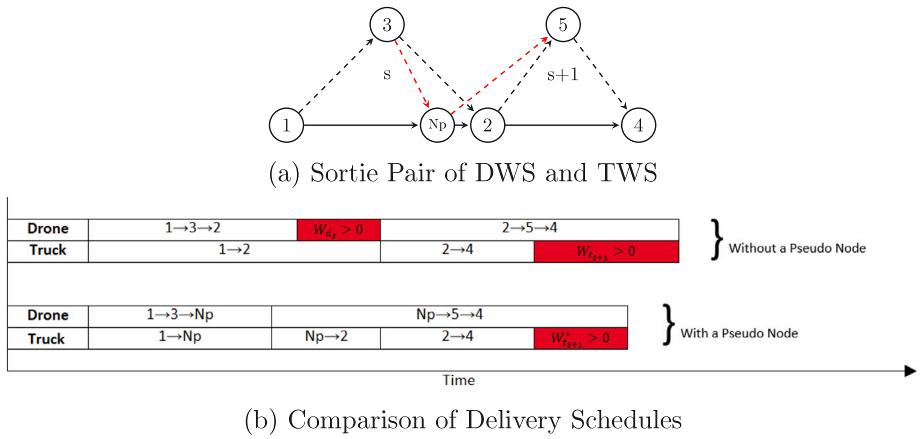


Fig. 7. Sortie pair analysis: DWS followed by TWS.

to the next truck node along the truck arc, the truck will never catch up the drone because the drone is faster than the truck. Therefore, the pseudo node does not exist outside DWS. □

In summary, a pseudo node may be located outside TWS, and within DWS and PS if $\alpha > 1$.

3.3. Two-sortie case

We extend the one-sortie case to consider two consecutive drone sorties where the benefit of adding pseudo nodes is further unveiled. From the analyses of the five scenarios in the one-sortie case, we specify the conditions under which the pseudo node approach can provide travel time savings. From the one-sortie case, the location of effective pseudo node is outside the sortie for TWS and within the sortie for DWS and PS. This suggests that a pair of TWS followed by DWS (or vice versa) is expected to be most effective to locate a pseudo node. As the PS case is the same as the DWS case in a sense that a pseudo node can be located within the sortie, we use DWS as a representative case that includes PS, unless otherwise specified, for the sake of simplicity. The two other cases (TWS followed by TWS and DWS followed by DWS) are still worth an investigation.

We define an effective sortie pair (ESP) as the combination of two sorties for which the first sortie can become PS if a pseudo node is added (that is, simultaneous rendezvous at the pseudo node), an example of which is shown in Fig. 7. The second sortie in the ESP could become either TWS, DWS, or PS, regardless of its original state. The conditions under which the second sortie becomes TWS, DWS, or PS are investigated in detail. In the ESP, even if a drone and a truck may not rendezvous simultaneously at the second sortie, the waiting time in the first sortie is eliminated, thereby providing an opportunity to reduce the total travel completion time from the original DTCO solution without pseudo nodes. The still remaining waiting time in the second sortie can be addressed by applying the proposed pseudo node insertion method again with the following sortie. We explore all two-sortie scenarios: (2.1) DWS followed by TWS, (2.2) TWS followed by DWS, (2.3) DWS followed by DWS, and (2.4) TWS followed by TWS. In the following sections, we investigate in detail the conditions under which the travel time saving can be possible for more than two successive sortie cases.

Scenario 2.1: DWS followed by TWS

Let us consider two consecutive sorties (DWS + TWS) shown in Fig. 7(a), for which we employ the sortie index s and $s+1$. The first sortie s is DWS ($T_{d_s} < T_{t_s}$) and the second sortie $s+1$ is TWS ($T_{d_{s+1}} > T_{t_{s+1}}$). Let $T(s, s+1)$ and $T^*(s, s+1)$ denote the travel completion time for the sortie pair $(s, s+1)$ before and after a pseudo node is added, respectively, which are decided by the latest arrival time at node 4 either by a drone or a truck.

Using the results from the one-sortie scenarios, we eliminate the waiting time at node 2 by inserting a pseudo node N_P within DWS where the drone and the truck arrive at the same time. This eliminates the truck waiting time at node 4. Then, the drone departs from N_P. Fig. 7(b) shows how the original solution may be improved by inserting the pseudo node.

It is clear from Fig. 7 that the first sortie will be PS when the pseudo node N_P is added on the truck arc (1,2) in the first sortie (DWS). Note that while the truck travel time remains the same regardless of the pseudo node, the drone travel time may be reduced by eliminating W_{d_s} and also considering the changes in drone travel time in s and $s+1$ caused by the pseudo node. The second sortie will become either TWS, DWS, or PS depending on the conditions that we explore in the following proposition, and the maximum possible travel completion time reduction for $(s, s+1)$ is $W_{t_{s+1}}$, which can be achieved when the second sortie becomes either DWS or PS.

Proposition 5. *In the ESP of DWS and TWS combination, the first sortie will be PS and the second sortie will become either TWS, PS, or DWS. In particular,*

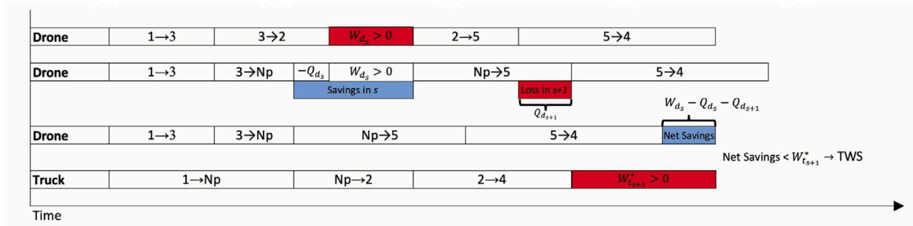
1. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < W_{t_{s+1}} \rightarrow TWS_{s+1}$
2. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} = W_{t_{s+1}} \rightarrow PS_{s+1}$
3. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > W_{t_{s+1}} \rightarrow DWS_{s+1}$

where $Q_{d_s} = \tau'_{3N_p} - \tau'_{32}$ and $Q_{d_{s+1}} = \tau'_{N_p5} - \tau'_{25}$ are the changes in drone travel time caused by the pseudo node for sorties s and $s+1$, respectively. These could also include the time required to retrieve and launch the drone en-route, instead of doing so at the original customer locations. Furthermore, TWS_{s+1}, DWS_{s+1}, and PS_{s+1} imply that the sortie $s+1$ is TWS, DWS, and PS, respectively. In addition, this ESP of DWS and TWS will result in travel completion time reduction, i.e, $T^*(s,s+1) < T(s,s+1)$ if $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0$, which can be at most $W_{t_{s+1}}$. That is, $W_{d_s} - Q_{d_s} - Q_{d_{s+1}}$ is the drone travel time saving in $(s, s+1)$ realized by adding the pseudo node. If it is less than, equal to, greater than $W_{t_{s+1}}$, truck waiting time in the second sortie, then the second sortie will be TWS, PS, and DWS, respectively.

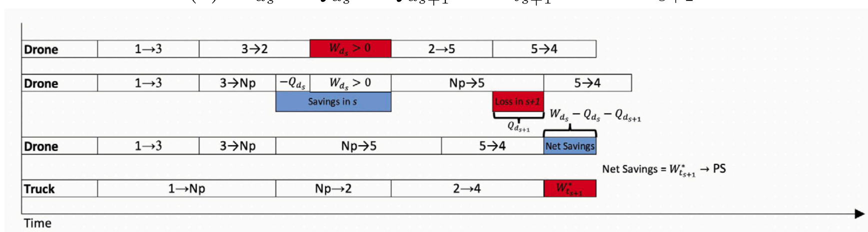
The proof of Proposition 5 is provided in Appendix.

Note that Q_{d_s} and $Q_{d_{s+1}}$ only include respective drone travel time changes in s and $s+1$ in the proof but it can be easily modified, if needed, to include the time required to receive and launch the drone en-route, instead of at the customer location. This could be an important issue because it may impact the overall makespan significantly.

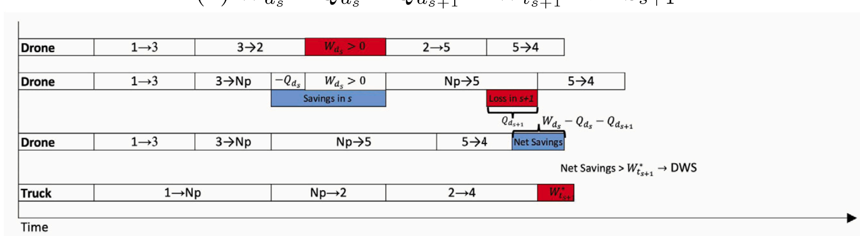
It is verified from the proof of Proposition 5 that the ESP of DWS and TWS combination can be reduced to either one-sortie with the waiting time case (either DWS or TWS) or no-waiting-time (two PSs) case. Fig. 8 illustrates the three conditions when the second sortie becomes TWS, PS, or DWS. In Figs. 8(a), 8(b), and 8(c), the first drone row shows



(a) $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < W_{t_{s+1}} \rightarrow \mathrm{TWS}_{s+1}$



(b) $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} = W_{t_{s+1}} \rightarrow \mathrm{PS}_{s+1}$



(c) $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > W_{t_{s+1}} \rightarrow \mathrm{DWS}_{s+1}$

Fig. 8. Sortie pair: DWS followed by TWS.

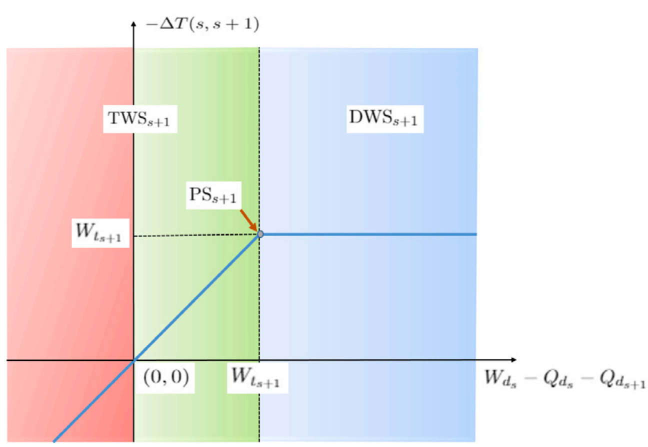


Fig. 9. Travel time saving conditions in DWS+TWS.

the drone travel time without the pseudo node. The second drone row shows how drone travel time can be calculated if the pseudo node is inserted (blue: travel time decrease, red: travel time increase). Here, $-Q_{d_s}$ shows the drone flying time saving due to shorter flying distance; W_{d_s} represent the saving in waiting time (no waiting is needed); and $-Q_{d_{s+1}}$ implies the increased flying time in $s+1$ due to the pseudo node. The third drone row shows the final drone travel time after the pseudo node in inserted. The truck row shows the truck travel time.

Fig. 9 shows how the travel time saving conditions can be analyzed where the x-axis represents the drone travel time saving ($W_{d_s} - Q_{d_s} - Q_{d_{s+1}}$) and the y-axis represents the travel completion time saving in $(s, s+1)$, i.e, $-\Delta T(s, s+1)$. That is, the blue line shows how the travel completion time saving changes for the sortie pair $(s, s+1)$, as the drone travel time saving changes. If $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < 0$ (the red area in Fig. 9), then the travel completion time saving in $(s, s+1)$ is negative (increase in travel completion time). If it is between 0 and $W_{t_{s+1}}$ (green area), the travel completion time saving in $(s, s+1)$ increases as $W_{d_s} - Q_{d_s} - Q_{d_{s+1}}$ increases. If it is greater than or equal to $W_{t_{s+1}}$ (blue area), then the travel completion time saving is fixed as $W_{t_{s+1}}$.

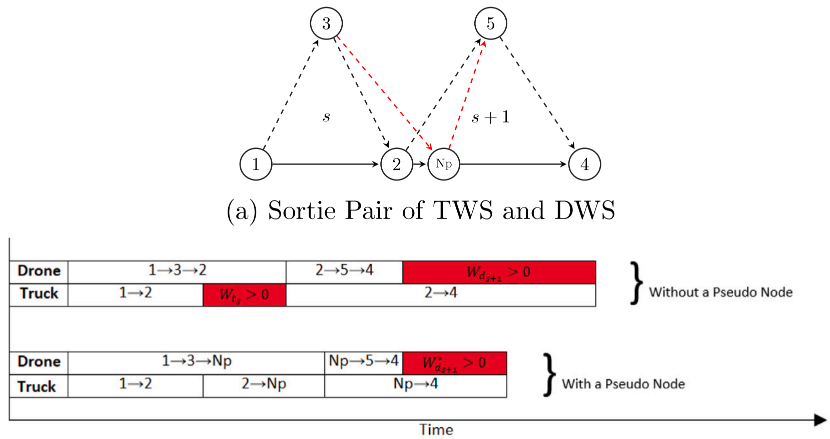
Scenario 2.2: TWS followed by DWS

Similar to scenario 2.1, we consider the two-sortie case here but the order is switched. That is, we consider two consecutive sorties s and $s+1$: TWS ($T_{d_s} > T_{t_s}$) followed by DWS ($T_{d_{s+1}} < T_{t_{s+1}}$) as shown in Fig. 10(a).

Utilizing the one-sortie scenario results, we eliminate the waiting time at the first sortie inserting a pseudo node $\mathrm{N_P}$ on the truck arc (2,4), which is outside the sortie s. By way of a pseudo node, the truck waiting time W_{t_s} is eliminated and the first sortie becomes PS. Fig. 10(b) illustrates how the solution may be improved by inserting the pseudo node. It is clear from Fig. 10 that the first sortie will be PS when the pseudo node $\mathrm{N_P}$ is added on the truck arc (2,4) in the second sortie. Unlike scenario 2.1 where the truck travel time remains the same regardless of the pseudo node, both drone travel time and truck travel time are subject to change. The second sortie will become either TWS, DWS, or PS depending on the conditions that we explore in the following proposition, and the maximum possible travel completion time reduction for $(s, s+1)$ is W_{t_s}, which can be achieved when the second sortie becomes either TWS or PS.

Proposition 6. *In the ESP of TWS and DWS combination, the first sortie will be PS and the second sortie will become either DWS, PS, or TWS. In particular,*

1. $W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} > W_{t_s} \rightarrow DWS_{s+1}$



(a) Sortie Pair of TWS and DWS

(b) Comparison of Delivery Schedules

Fig. 10. Sortie pair analysis: TWS followed by DWS.

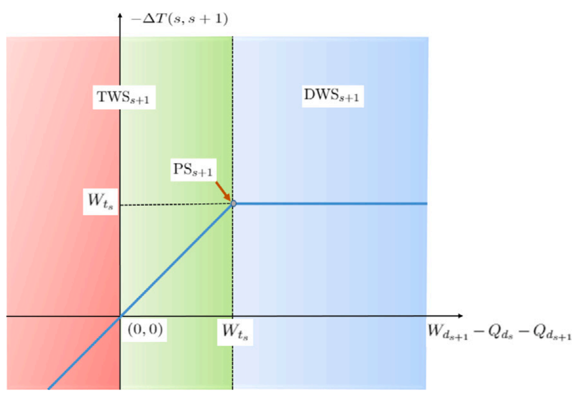


Fig. 11. Travel time saving conditions in TWS+DWS.

2. $W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} = W_{t_s} \rightarrow PS_{s+1}$
3. $W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} < W_{t_s} \rightarrow TWS_{s+1}$

where $Q_{d_s} = \tau'_{3N_p} - \tau'_{32}$ *and* $Q_{d_{s+1}} = \tau'_{N_p5} - \tau'_{25}$ *are the changes in drone travel time caused by the pseudo node for sorties* s *and* $s+1$*, respectively. In addition, this ESP of TWS and DWS will result in travel completion time reduction, i.e,* $T^*(s, s+1) < T(s, s+1)$ *if* $W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} > 0$*, which can be at most* W_{t_s}*.*

Proof. The proof can be done in the same fashion as the proof for Proposition 5. □

Fig. 11 shows how the travel time saving conditions can be analyzed, which seems to be very similar to Fig. 9. The blue line shows how the travel completion time saving increases as the drone travel time saving increases. Note that the maximum saving is W_{t_s}, which is not $W_{t_{s+1}}$ in scenario 2.1.

Scenario 2.3: DWS followed by DWS

Now, let us consider the two consecutive sorties: DWS ($T_{d_s} < T_{t_s}$) followed by DWS ($T_{d_{s+1}} < T_{t_{s+1}}$), shown in Fig. 12. Same as in the first two scenarios, the first sortie can become PS if a pseudo node $\mathrm{N_P}$ is added on the truck arc (1,2) as shown in Fig. 12. Intuitively, the second sortie is expected to be DWS because the drone waiting time W_{d_s} is removed in the first sortie, which will usually result in the drone travel completion time saving in the sortie pair $(s, s+1)$ while the truck travel time remains the same as illustrated in Fig. 12. However, the drone travel time does not always decrease; rather, it will increase if $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < 0$, i.e., if the time saving by removing the drone waiting time in s is smaller than the summation of the increased drone retrieval and re-launch time en-route and the drone travel time changes in s and $s+1$ caused by the addition of the pseudo node in s. Indeed, the second sortie can be either TWS, DWS, or PS depending on the conditions that we explore in the following proposition:

Proposition 7. *In the ESP of DWS and DWS combination, the first sortie will be PS and the second sortie will become either TWS, PS, or DWS. In particular,*

1. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > -W_{d_{s+1}} \rightarrow DWS_{s+1}$
2. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} = -W_{d_{s+1}} \rightarrow PS_{s+1}$
3. $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < -W_{d_{s+1}} \rightarrow TWS_{s+1}$

where $Q_{d_s} = \tau'_{3N_p} - \tau'_{32}$ *and* $Q_{d_{s+1}} = \tau'_{N_p5} - \tau'_{25}$*. In addition, this ESP of DWS and DWS does not reduce the travel completion time for this sortie pair* $(s, s+1)$*, i.e., the truck travel time remains the same. The drone travel completion time is reduced, i.e.,* $W^*_{d_{s+1}} > W_{d_{s+1}}$ *if* $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0$*.*

Proof. The proof can be done in the same fashion as the proof for Proposition 5. □

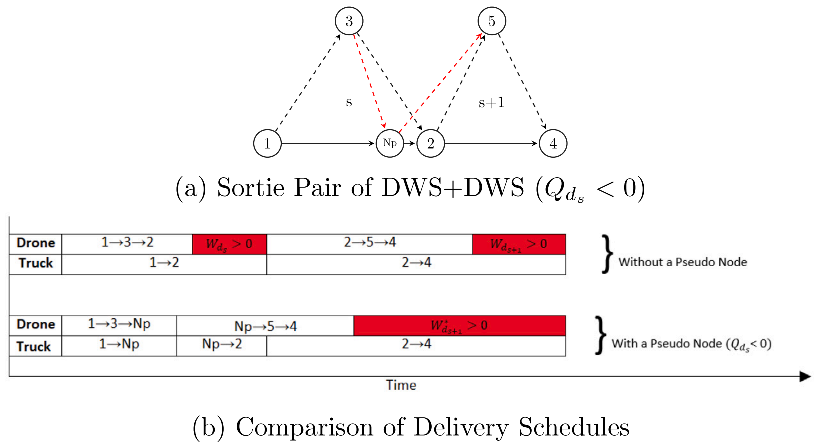
In the DWS+DWS case, there is no travel completion time saving possible in $(s, s+1)$, as the truck travel time remains the same regardless of the pseudo node addition. However, the drone travel completion time can be reduced if $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0$, which may provide an opportunity to reduce the travel completion time in the following sorties.

Fig. 13 illustrates how the travel time saving in $(s, s+1)$ changes as the drone travel time changes (blue line). The green line shows the drone travel time saving in $(s, s+1)$, which could potentially impact the overall makespan because the drone will be ready for the next sortie operation early if $-\Delta W_d(s, s+1) > 0$.

Scenario 2.4: TWS followed by TWS

Last but not least, let us consider the TWS ($T_{d_s} > T_{t_s}$) followed by TWS scenario, which is illustrated in Fig. 14. It is clear that the first sortie will be PS when the pseudo node $\mathrm{N_P}$ is added on the truck arc (2,4) in the second sortie (TWS). The maximum travel completion time reduction for $(s, s+1)$ achieved by adding the pseudo node is $W_{t_s} + W_{t_{s+1}}$. The second sortie will become either TWS, DWS, or PS depending on the conditions that we explore in the following proposition:

Proposition 8. *In the ESP of TWS and TWS combination, the first sortie will be PS. In addition, the second sortie will become either TWS, PS, or DWS. In particular,*



(a) Sortie Pair of DWS+DWS ($Q_{d_s} < 0$)

(b) Comparison of Delivery Schedules

Fig. 12. Sortie pair analysis: DWS followed by DWS.

Table 1
Summary of Two-sortie Cases.

Scenario	Pseudo node	2nd Sortie	2nd Sortie condition	Completion time saving	Max saving
DWS+TWS	s	TWS, PS, DWS	$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} (<,=,>) W_{t_{s+1}}$	$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0$	$W_{t_{s+1}}$
TWS+DWS	$s+1$	TWS, PS, DWS	$W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} (<,=,>) W_{t_s}$	$W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}} > 0$	W_{t_s}
DWS+DWS	s	TWS, PS, DWS	$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} (<,=,>) - W_{d_{s+1}}$	$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0^*$	NA
TWS+TWS	$s+1$	TWS, PS, DWS	$-Q_{d_s} - Q_{d_{s+1}} (<,=,>) W_{t_s} + W_{t_{s+1}}$	$-Q_{d_s} - Q_{d_{s+1}} > 0$	$W_{t_s} + W_{t_{s+1}}$

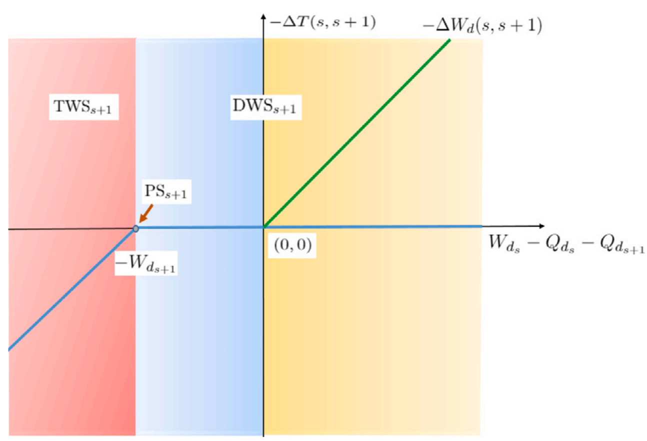


Fig. 13. Travel time saving conditions in DWS+DWS.

1. $-Q_{d_s} - Q_{d_{s+1}} > W_{t_s} + W_{t_{s+1}} \rightarrow DWS_{s+1}$
2. $-Q_{d_s} - Q_{d_{s+1}} = W_{t_s} + W_{t_{s+1}} \rightarrow PS_{s+1}$
3. $-Q_{d_s} - Q_{d_{s+1}} < W_{t_s} + W_{t_{s+1}} \rightarrow TWS_{s+1}$

where $Q_{d_s} = \tau'_{3N_p} - \tau'_{32}$ *and* $Q_{d_{s+1}} = \tau'_{N_p5} - \tau'_{25}$ *are the differences in the drone travel time for sorties* s *and* $s+1$ *due to* N_p*, respectively. In addition, this ESP of TWS and TWS will result in travel completion time reduction, i.e,* $T(s, s+1) > T^*(s, s+1)$ *if* $-Q_{d_s} - Q_{d_{s+1}} > 0$*, which can be at most* $W_{t_s} + W_{t_{s+1}}$*.* $T^*(s, s+1)$ *is the travel completion time for ESP after the pseudo node is added,*

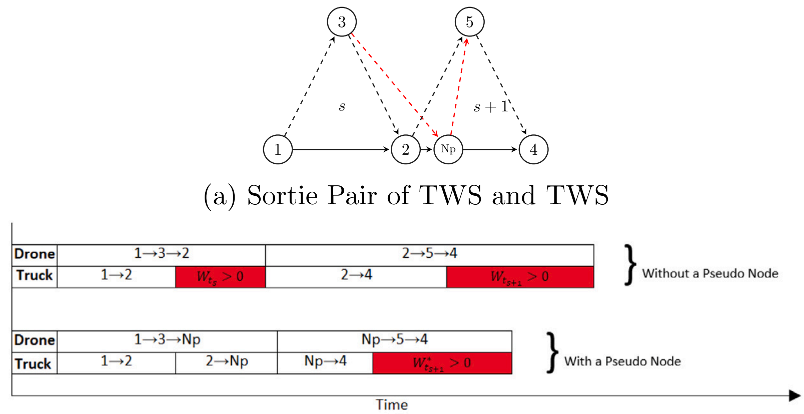
Proof. The proof can be done in the same fashion as the proof for Proposition 5. □

Note that the drone travel completion time saving condition is different to the previous cases because it does not contain the drone waiting time in s or $s+1$, i.e., W_{d_s} or $W_{d_{s+1}}$. Fig. 15 illustrates how the travel completion time saving changes as the drone travel time saving changes (blue line).

Table 1 summarizes the two-sortie case analyses. As expected DWS+TWS and TWS+DWS cases have high potential for travel completion time savings. The DWS+DWS combination cannot reduce any travel completion time for $(s, s+1)$ but the drone may be ready early for the next operation. Therefore, there can be substantial benefit if TWS follows DWS+DWS. The TWS+TWS case reduces the truck travel time and, therefore, the travel completion time for $(s, s+1)$ will also decrease if the drone travel time decreases, which can occur if $Q_{d_s} - Q_{d_{s+1}} < 0$.

3.4. Three-sortie case

Turning our attention to the three-sortie case, there are total 8 cases as shown in Table 2, where the blue box and red box represent drone waiting time and truck waiting time, respectively. There are two choices: considering the former two sorties $(s, s+1)$ first or the latter two sorties $(s+1, s+2)$ first. Fig. 16 shows how this three-sortie case can be reduced. If the former two sorties s and $s+1$ are considered first, the first sortie s becomes PS while the second sortie $s+1$ becomes either DWS, TWS, or PS, as shown in Figs. 16(b) and 16(c). Therefore, the three-sortie case is reduced to either two-sortie (PS+DWS+TWS, PS+TWS+DWS, etc.) or even one-sortie case (PS+PS+DWS or PS+PS+TWS), excluding the PS in the count. When the latter two sorties $s+1$ and $s+2$ are considered first, the second sortie $s+1$ becomes PS, thereby dividing the three-sortie case into two one-sortie cases (DWS+PS+TWS, TWS+PS+DWS, etc.), as shown in Figs. 16(d) and 16(e). This is a worse condition than considering the former two sorties $(s, s+1)$ first because making the second sortie PS eliminates the opportunity of reducing the waiting time of the first sortie. Therefore, it can be concluded that it has substantial advantage to apply a pseudo node to make a PS from the beginning, i.e., consider $(s, s+1)$ first rather than $(s+1, s+2)$, in the three-sortie case. As long as pseudo nodes are successively applied, a reverse direction from the last is also possible. This principle can be extended to the more-than-three-sortie cases. Therefore, once the DTCO solution without pseudo nodes is obtained, the natural strategy to apply the pseudo node insertion algorithm, which will be discussed in detail in Section 4, is to add pseudo nodes in the sortie pairs successively from the beginning or from the end. Table 2 summarizes the 8 cases including the saving conditions and associated maximum possible savings.



(a) Sortie Pair of TWS and TWS

(b) Comparison of Delivery Schedules

Fig. 14. Sortie pair analysis: TWS followed by TWS.

Table 2
Summary of Three–sortie Scenarios.

Scenario	Figure	Saving condition	Max saving
DDD		$\{W_{d_s} - Q_{d_s} - Q_{d_{s+1}}, W_{d_{s+1}} - Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	NA
DDT		$\{W_{d_s} - Q_{d_s} - Q_{d_{s+1}}, W_{d_{s+1}} - Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_{s+2}}$
DTD		$\{W_{d_s} - Q_{d_s} - Q_{d_{s+1}}, W_{d_{s+2}} - Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_{s+1}}$
DTT		$\{W_{d_s} - Q_{d_s} - Q_{d_{s+1}}, -Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_{s+1}} + W_{t_{s+2}}$
TDD		$W_{d_{s+1}} - Q_{d_{s+1}} - Q_{d_{s+2}} > 0$	W_{t_s}
TDT		$\{W_{d_{s+1}} - Q_{d_s} - Q_{d_{s+1}}, W_{d_s+1} - Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_s} + W_{t_{s+2}}$
TTD		$\{-Q_{d_s} - Q_{d_{s+1}}, W_{d_{s+2}} - Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_s} + W_{t_{s+1}}$
TTT		$\{-Q_{d_s} - Q_{d_{s+1}}, -Q_{d_{s+1}} - Q_{d_{s+2}}\} > 0$	$W_{t_s} + W_{t_{s+1}} + W_{t_{s+2}}$

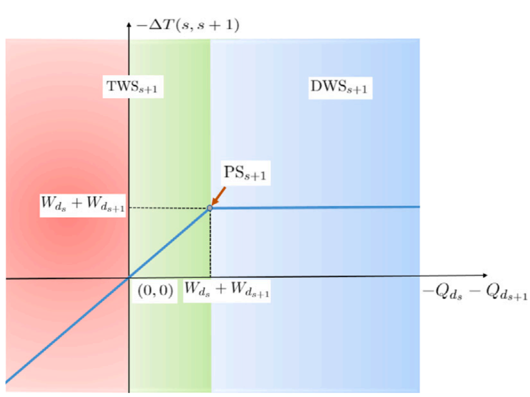


Fig. 15. TWS+TWS results.

4. Heuristic approaches

Based on the analysis for one-sortie, two-sortie, and three-or-more-sortie cases presented above, two heuristic approaches are proposed in this section. The first is called the pseudo node insertion heuristic and the second is called modified route and re-assign heuristic.

4.1. Pseudo node insertion heuristic

A pseudo node insertion heuristic examines the possible travel completion saving or drone travel time saving successively. Algorithm 1 can be applied after the DTCO solution (with no pseudo nodes) is obtained (shown in line 2). This initial DTCO solution can be obtained by an exact algorithm (for small instances) or by heuristic methods: see, e.g., Gonzalez-R et al. (2020), Murray and Chu (2015). The DTCO

```
Data: Set of all nodes in the network N, set of all customers C, set of customers that can be
      served by drone C′, set of all sorties P
Result: New routes for drone and truck to achieve maximum travel completion time saving
1  Initialize
2  Solve the original DTCO problem without pseudo nodes
3  Identify all the sorties to be DWS, TWS, or PS
4  s = 1
5  for Sortie pair (s, s + 1) do
6      if s= DWS or PS then
7          Place the pseudo node within the sortie
8          if Travel completion time saving for (s, s + 1) >0 or drone travel time saving >0 then
9              Make the pseudo node permanent
10             Update drone and truck routes and sortie information
11         end
12     else
13         Place the pseudo node outside the sortie
14         if Travel completion time saving for (s, s + 1) >0 then
15             Make the pseudo node permanent
16             Update drone and truck routes and sortie information
17         end
18     end
19     s = s + 1
20 end
```
Algorithm 1: Pseudo Node Insertion Algorithm

problem is already NP-hard and, therefore, it cannot be solved using commercial solvers such as Gurobi for medium or large-scale problems. Therefore, the line 2 in this algorithm requires some heuristic methods. However, to focus on the pseudo node insertion method, it is assumed that a DTCO solver is readily available. Once the DTCO solution is obtained, all sorties are identified as DWS, TWS, or PS. Then, pseudo nodes are inserted successively if there are a travel completion time saving or a drone travel time saving (lines 5–19).

The pseudo node insertion algorithm is straightforward and easy to apply, yet it works quite well. However, it relies on the DTCO solution, which implies that the solution methodology must be in two stages (DTCO solution, and then the pseudo node insertion).

In Fig. 17, we show how the pseudo node insertion method (Algorithm 1) can be used to improve the DTCO solution. In Fig. 17(a), a

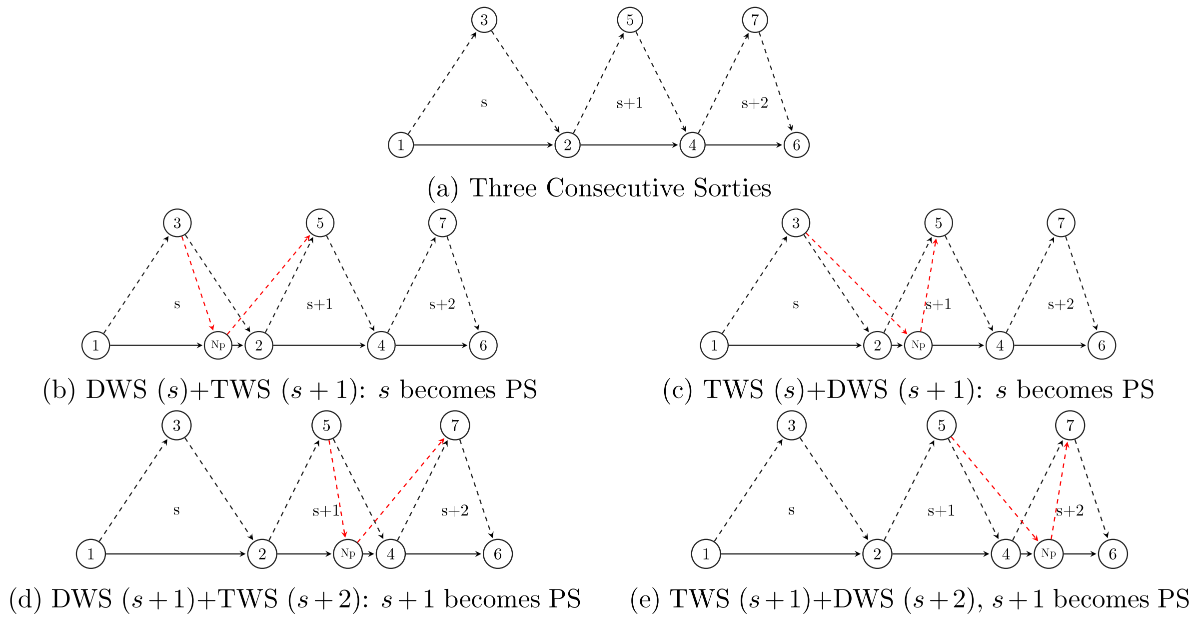


Fig. 16. Analysis of three-sortie cases.

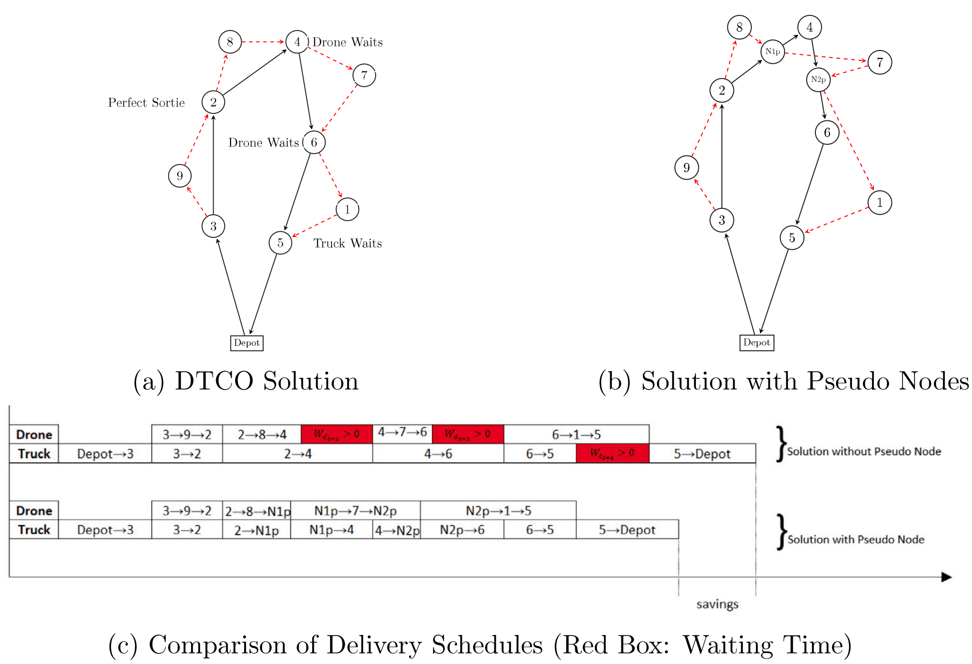


Fig. 17. Illustration of saving achieved using the pseudo method.

DTCO solution is shown, which is composed of four consecutive sorties: PS (3 – 9 – 2), DWS (2 – 8 – 4), DWS (4 – 7 – 6), and TWS (6 – 1 – 5). Due to the presence of a consecutive DWS, DWS, and TWS, we are able to create two pseudo nodes N_{1p} and N_{2p} on the arc between nodes 2 and 4, and nodes 4 and 6, respectively, as shown in Fig. 17(b). The travel completion time saving is shown in Fig. 17(c). Note that a pseudo node is not inserted in PS in this example, because it may not help to reduce the drone travel time. However, it may be inserted if beneficial (e.g., saving drone battery, etc.).

4.2. Modified route and re-assign heuristic

We extend the DTCO heuristic (route and re-assign) proposed by Murray and Chu (2015) to find a near-optimal DTCO solution while the pseudo node insertion is considered at the same time. The heuristic, shown in Algorithm 2, starts with finding a truck only route by solving a TSP problem, which requires a TSP solver. Then, it assigns a drone eligible customer to see if there can be travel time savings and, if yes, creates a drone sortie. Up to this point, this is the same as the previous approach. Then, the algorithm checks if there is another sortie before and after the current one and also calls the $PseudoInsertion$ function, which is part of the pseudo node insertion algorithm (line 6–18 of Algorithm 1).

In Algorithm 2, subroutines $calcSaving$, $calcCostTruck$, $calcCostDrone$, and $performUpdate$ are used, which are borrowed from Murray and Chu (2015). The details of such subroutines are suppressed in this paper for the sake of simplicity. Rather, their roles will be introduced briefly as Algorithm 2 is explained. $solveTSP(C)$

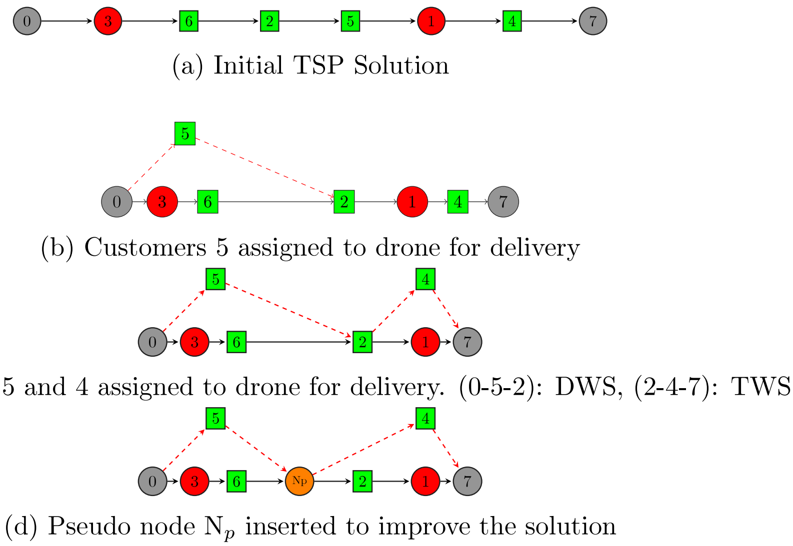


Fig. 18. Modified route and re-assign algorithm example.

```
   Result: Pseudocode for the main modified DTCO algorithm
1  Initialize:
2  Cprime = C'  % Set of drone eligible customers
3  [truckroute, t] = solveTSP(C)  % C: Set of all customers
4  truckSubRoutes = truckRoute
5  maxsaving = 0
6  repeat
7      for j ∈ Cprime do
8          Call the calcSaving(j, t) function.
9      end
10     for subroute ∈ truckSubRoutes do
11         if drone associated with this subroute then
12             Call the calcCostTruck(j, t, subroute) function.
13         else
14             Call the calcCostDrone(j, t, subroute) function.
15         end
16     end
17     if maxSaving >0 then
18         Call the performUpdate function.
19         if sortie is generated then
20             Identify the current sortie s to be DWS, TWS, or PS
21             For sorties s − 1 (if exists) and s, Call function PseudoInsertion
22             For sorties s and s + 1 (if exists), Call function PseudoInsertion
23         end
24     end
25     Reset maxSaving = 0
26 until Stop
```

Algorithm 2: DTCO Heuristic, modified from Murray and Chu (2015)

is a subroutine that takes as input the location of customers C and returns the sequence of nodes visited by a truck (given as *truckroute* array) and respective arrival times (given as t array); i.e., it basically solves the TSP problem. We assume that the TSP solver is given; any TSP heuristics such as nearest neighbor, insertion algorithm, and tabu search can be applied. The ordered vector *truckSubRoutes* contains the TSP route initially, which is broken down into *subroutes* as some nodes are assigned to the drone. Next, the algorithm calls the *calcSaving* subroutine as shown in line 8 to determine a saving associated with removing each customer from *Cprime* and serving it using the drone. An example, shown in Fig. 18, is used to illustrate this, where gray circles represent a depot, red circles are truck nodes, and the green rectangles are drone eligible customers. If node 5 is removed from the *truckroute*, then the saving associated with this move is given as $\tau_{2,5} + \tau_{5,1} - \tau_{2,1}$ as the sequence $2 \rightarrow 5 \rightarrow 1$ is replaced with $2 \rightarrow 1$. There may be a case when the drone eligible customer may be placed between the launch and rendezvous nodes, i.e., within a drone sortie. For example, node 6 in Fig. 18(c). In this case, the *saving* could be negative if removing customer j in the truck route results in truck waiting at the rendezvous node.

Then, if a drone is associated with the current truck subroute being considered, *calcCostTruck* is called; otherwise *calcCostDrone* is called. A cost of inserting j customer in a different position of a truck route and a drone route, respectively, is calculated in these functions. Based on this cost calculations and associated savings, it is decided how customer j can be served. For example, depending on such calculations, customer 4 in Fig. 18(b) can either be kept at it's current spot, be moved to the first *trucksubroute*, or it could be served using the drone in the second *trucksubroute* as shown in Fig. 18(c).

After a drone is assigned to a customer, the previous and next sorties are investigated for the possibility of pseudo node insertion to improve the saving as shown in lines 19–23 of Algorithm 2. For example in Fig. 18(c), when customer 4 is assigned to the drone in sortie (2–4–7), the previous sortie (0–5–2) is investigated for a possible pseudo node insertion. Suppose that (0–5–2) is DWS whereas (2–4–7) is TWS, so a pseudo node N_p is inserted in the truck arc between node 6 and node 2, eliminating the drone waiting time at node 2.

5. Numerical examples

To test the proposed algorithms, the benchmark problems from Bouman et al. (2015) are used, whose results are shown in Tables 3 and 4. Instances are randomly picked up from Bouman et al. (2015), and 42.3%, 34.6%, and 23.1% of instances are from uniform, doublecenter, and singlecenter problem types, respectively, where Euclidean distances are used between nodes. In all the examples, the drone speed is set to 2 speed unit while the truck speed is set to 1 speed unit, i.e., $\alpha = 2$. All problems are coded using Python 2.7.15 (Windows 10, 12 GB RAM, 4th Generation Intel Quad Core i7-4712HQ), and the computing time for all examples is reasonable (less than 1 min at most per example).

First, 26 small examples whose sizes are from 6 to 9 customers are solved, and the total travel completion time results are summarized in Table 3. The second column shows the results without the use of pseudo nodes, which are obtained using the MILP formulation of Murray and Chu (2015) and a Gurobi solver. Based on these no-pseudo-node solutions, the pseudo node insertion algorithm (Algorithm 1) is implemented, whose results are shown in column 3. The results with pseudo nodes are better than the DTCO solutions with no pseudo nodes in all cases (1.99%–7.15% improvements in travel completion time savings, 3.53% on average). The percent improvements are shown in column 4. Note that as the exact optimal solutions can be obtained for the base DTCO problems (without the pseudo nodes) using a commercial solver, no heuristic method is used to obtain such base DTCO solutions.

Table 3
Travel completion time analysis for small size problems.

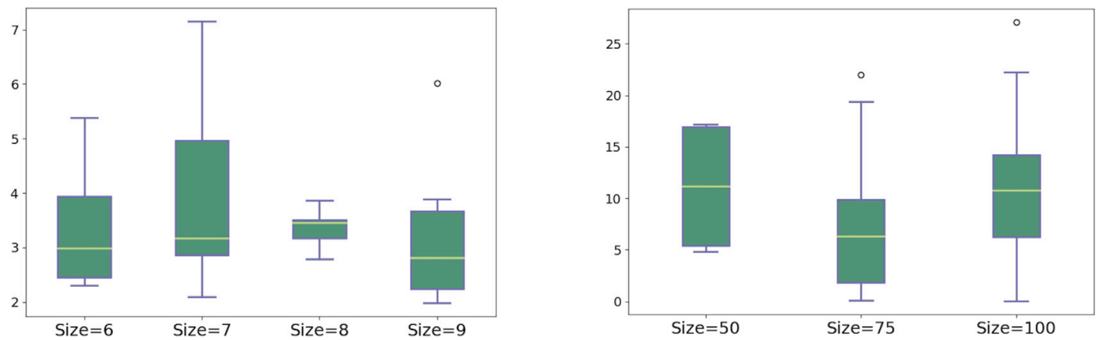
Instance	No pseudo node	With pseudo node (Algorithm 1)	Improvement (Gap)	Size
D16	398.89	378.52	5.38%	6
S2	175.56	171.25	2.46%	6
S3	354.83	344.53	2.99%	6
S4	263.03	252.67	3.94%	6
S5	152.75	149.23	2.31%	6
S6	429.18	416.64	3.01%	7
S7	194.71	180.79	7.15%	7
S8	116.24	112.56	3.17%	7
S9	478.90	456.23	4.97%	7
S10	420.19	411.55	2.10%	7
S11	193.33	186.53	3.52%	7
S12	232.02	225.96	2.62%	7
S13	177.10	172.04	2.86%	7
S14	180.81	171.43	5.19%	7
S15	177.88	171.69	3.49%	8
S16	220.11	212.39	3.51%	8
S17	472.12	459.29	2.79%	8
S18	468.81	453.27	3.43%	8
S19	215.67	207.35	3.86%	8
S20	214.90	208.29	3.08%	8
S21	203.27	199.23	1.99%	9
S22	575.87	563.98	2.11%	9
S23	438.02	425.36	2.98%	9
S24	214.02	201.14	6.02%	9
S25	221.20	215.36	2.64%	9
S26	218.33	209.85	3.89%	9
		Avg.	3.52%	

Table 4
Travel completion time analysis of large size problems.

Instance	No pseudo node	With pseudo node (Algorithm 2)	Improvement (Gap)	Size
L1	1251.37	1071.11	16.83%	50
L2	975.37	923.8	5.58%	50
L3	1248.59	1191.04	4.83%	50
L4	1262.92	1077.75	17.18%	50
L5	1428.98	1222.05	16.93%	75
L6	1444.95	1342.99	7.59%	75
L7	1528.53	1280.55	19.37%	75
L8	1796.47	1690.19	6.29%	75
L9	1853.65	1519.42	22.00%	75
L10	1458.28	1324.36	10.11%	75
L11	1277.42	1266.82	0.84%	75
L12	1678.82	1569.56	6.96%	75
L13	1164.84	1141.73	2.02%	75
L14	720.75	716.71	0.56%	75
L15	678.14	667.17	1.64%	75
L16	654.6	639.46	2.37%	75
L17	732.81	732.01	0.11%	75
L18	702.89	686.75	2.35%	75
L19	654.33	596.7	9.66%	75
L20	1791.19	1616.95	10.78%	100
L21	1639.9	1534.55	6.87%	100
L22	1535.53	1444.84	6.28%	100
L23	1998.22	1749.61	14.21%	100
L24	1987.3	1625.52	22.26%	100
L25	1862.97	1641.26	13.51%	100
L26	2058.89	1620.32	27.07%	100
L27	1658.4	1574.22	5.35%	100
L28	1545.81	1544.68	0.07%	100
		Avg.	9.27%	

To test the effectiveness of pseudo nodes on large size problems, 28 examples whose sizes are between 50 to 100 customers are chosen. The results are summarized in Table 4. Since these problems cannot be solved using commercial solvers such as Gurobi, the route-and-reassign algorithm is used to obtain the base DTCO solutions with no pseudo node. The updated route-and-reassign algorithm (Algorithm 2) is then implemented to obtain solutions with pseudo nodes, and compared with the base solution with no pseudo node. The no-pseudo-node results are shown in column 2 while the with-pseudo-node solutions obtained by Algorithm 2 are shown in column 3. The pseudo node results are better (0.07%–27.07% improvements in travel completion time savings). The percent improvements are shown in column 4. Based on these results, the pseudo node insertion method proves to be effective.

All the results are summarized using box plots in Fig. 19. For small size problems (6 to 9 nodes), the pseudo node methods improve the no-pseudo-node solution by 2.5%–4% in most cases (median values are 2.9%–3.3%). For larger problems, the median values of percentage



(a) % Improvement (small instances) (b) % Improvement (large instances)

Fig. 19. Box plots for results.

improvement are 11.21% (50 node cases), 6.29% (75 node cases), and 10.78% (100 node cases). In general, the pseudo node insertion methods is more effective in larger problems because of more opportunities to save travel completion time.

6. Conclusion and future work

In this paper, a pseudo node insertion method is presented to consider a synchronization issue between a drone and a truck in DTCO. In particular, a detailed method of locating pseudo nodes is provided with the notion of DWS, TWS, and PS. In addition, the effectiveness of added pseudo nodes is analyzed using one-sortie, two-sortie, and three-or-more-sortie cases, for which travel completion time saving conditions are presented. These decomposed examples that consider not only the size (one, two, or three sorties) but also type (DWS, TWS, and PS) of sorties provide a basis for proposing the pseudo node insertion algorithm. In particular, two heuristic algorithms are tested using numerical examples. Overall, the pseudo node insertion method proves effective in reducing the total travel completion time, thereby making the DTCO more efficient and effective.

Extending the DTCO problem to consider multiple drones and multiple trucks with the possibility of drone sharing among trucks can be one of the possible future research directions as the use of multiple vehicles is of particular interest (Murray & Raj, 2020; Tu et al., 2018). If multiple drones and multiple trucks are considered, the proposed method may end up generating too many pseudo nodes. An extensive study of how to decide the optimal number and locations of pseudo nodes will be required. There is also a possibility to relate such pseudo nodes to drone hubs at which no synchronization is required. At the drone hubs, a drone battery can be recharged or swapped, and drones can be picked up by any truck visiting the hub. Multiple pseudo nodes located close to each other may be a good candidate for such a hub location. This will also bring scheduling and capacity planning problems, which is an interesting and worthwhile topic to be investigated.

Other natural extension would be to develop better heuristic methods to enhance the computational tractability, which will actively adopt modern heuristic, meta-heuristic, and mathematical programming methods, tailored for DTCO. This will be particularly important for large-scale complex DTCO problems that may consider the topics mentioned above such as multiple vehicles, hub locations, scheduling and capacity planning.

CRediT authorship contribution statement

Sung Hoon Chung: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original and revised manuscript, Supervision, Funding acquisition, Project administration. **Bhawesh Sah:** Conceptualization, Methodology, Formal analysis, Visualization, Validation. **Jinkun Lee:** Conceptualization, Methodology.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is partly supported by the first author's National Science Foundation, United States of America grant, award number 2032460.

Appendix

Proof of Proposition 1.

Proof. Let us consider the case where the truck waits at node 2. In Fig. 4(b), the arc labels denote distances and node labels denote coordinates. The speeds of drone and truck are denoted by v_d and v_t, respectively, and their speed ratio is defined as $\alpha = v_d/v_t > 1$. From the nonnegative distance condition, $p_i,\ q_j > 0,\ i = 1,2, j = 1,2,3$. Since the drone and the truck are assumed to meet simultaneously at the pseudo node $\mathrm{N_P}$, $\tau'_{13} + \tau'_{3N_P} = \tau_{1N_P}$, which leads to:

$$q_1 + q_2 = \alpha p_1 \tag{1}$$

From the triangle inequality with $\triangle\, \mathrm{N_P}23$:

$$q_3 < q_2 + p_2 \tag{2}$$

Were it not for the pseudo node, the truck waits for the drone at node 2. That is, $T_{t_s} < T_{d_s}$, where $T_{t_s} = (p_1+p_2)/v_t$ and $T_{t_d} = (q_1+q_3)/v_d$, from which we obtain:

$$\begin{aligned} \alpha(p_1 + p_2) &< q_1 + q_3 \\ &< q_1 + q_2 + p_2 \quad (\because (2)\ q_3 < q_2 + p_2) \\ &< \alpha p_1 + p_2 \quad (\because (1)\ q_1 + q_2 = \alpha p_1) \end{aligned} \tag{3}$$

which leads to $(\alpha - 1)p_2 < 0$. As $p_2 > 0$, this contradicts the condition of $\alpha > 1$. This implies that a pseudo node where a drone and a truck meet simultaneously on the truck arc within TWS does not exist. □

Proof of Proposition 5.

Proof. In the two-sortie case, the total travel completion time before a pseudo node is added can be calculated as follows:

$$T(s, s+1) = \tau'_{13} + \tau'_{32} + W_{d_s} + \tau'_{25} + \tau'_{54} = \tau_{12} + \tau_{24} + W_{t_{s+1}} \tag{4}$$

which is clear from Fig. 7(b). In addition, the total travel time of drone and truck, after a pseudo node is added, can be calculated as follows:

$$T^*_d(s, s+1) = \tau'_{13} + \tau'_{3\mathrm{N_p}} + \tau'_{\mathrm{N_p}5} + \tau'_{54} \qquad \text{(drone travel time)} \tag{5}$$

$$T_t^*(s,s+1) = \tau_{1\mathrm{N_p}} + \tau_{\mathrm{N_p}2} + \tau_{24} \qquad \text{(truck travel time)} \tag{6}$$

where $T_d^*(s,s+1)$ and $T_t^*(s,s+1)$ denote the travel time of drone and truck, respectively, excluding any waiting time for the sortie pair $(s,s+1)$ after the pseudo node is added. It is clear that the second sortie will be TWS if $T_d^*(s,s+1) > T_t^*(s,s+1)$, PS if $T_d^*(s,s+1) = T_t^*(s,s+1)$, and DWS if $T_d^*(s,s+1) < T_t^*(s,s+1)$. From the TWS condition,

$$T_d^*(s,s+1)-T_t^*(s,s+1) > 0 \rightarrow \tau'_{13}+\tau'_{3\mathrm{N_p}}+\tau'_{\mathrm{N_p}5}+\tau'_{54}-\tau_{1\mathrm{N_p}}-\tau_{\mathrm{N_p}2}-\tau_{24} > 0 \tag{7}$$

Substituting

$$\tau'_{3\mathrm{N_p}} = \tau'_{32} + Q_{d_s}, \;\; \tau'_{\mathrm{N_p}5} = \tau'_{25} + Q_{d_{s+1}}, \text{ and } \tau_{1\mathrm{N_p}} + \tau_{\mathrm{N_p}2} = \tau_{12}$$

into (7) will lead to:

$$\tau'_{13} + \tau'_{32} + Q_{d_s} + \tau'_{25} + Q_{d_{s+1}} + \tau'_{54} - \tau_{12} - \tau_{24} > 0 \tag{8}$$

Combining (4) and (8) gives:

$$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} < W_{t_{s+1}} \tag{9}$$

The PS and DWS cases can be proven in the same fashion. The travel completion time can be reduced if the following condition is satisfied:

$$T_d^*(s,s+1) < T(s,s+1) \rightarrow \tau'_{13}+\tau'_{3\mathrm{N_p}}+\tau'_{\mathrm{N_p}5}+\tau'_{54} < \tau'_{13}+\tau'_{32}+W_{d_s}+\tau'_{25}+\tau'_{54} \tag{10}$$

which leads to:

$$W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0 \tag{11}$$

Therefore, the ESP of DWS and TWS will experience a travel completion time decrease if $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > 0$, which can be at most $W_{t_{s+1}}$. Note that the second sortie will become DWS if $W_{d_s} - Q_{d_s} - Q_{d_{s+1}} > W_{t_{s+1}}$, which will produce no more travel completion time saving for this sortie pair $(s,s+1)$. However, when the second sortie becomes DWS, there may be an opportunity to reduce the travel time further if it is considered with the following (the third) sortie in the entire network. □

Coordinates of the Pseudo Nodes:

Let us consider how to find the intersection coordinates to place a pseudo node. In Fig. 21, the node labels represent the coordinates.

As we already know a pseudo node cannot be located on the truck arc (1,2) when $\alpha > 1$, let us try to place the pseudo node candidate on the truck arc (2,4). In order for drone and truck to meet at the pseudo node $\mathrm{N_P}$ at the same time, the truck travel time ($1 \rightarrow 2 \rightarrow \mathrm{N_P}$) must be same as the drone travel time ($1 \rightarrow 3 \rightarrow \mathrm{N_P}$), which leads to:

$$\alpha = \frac{\sqrt{(x-x_3)^2+(y-y_3)^2}+\sqrt{(x_3-x_1)^2+(y_3-y_1)^2}}{\sqrt{(x-x_2)^2+(y-y_2)^2}+\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}} \tag{12}$$

In addition, the following equation can be derived from the fact that the pseudo node is located on the arc (2,4):

$$y - y_2 = \frac{y_4-y_2}{x_4-x_2}(x-x_2), \quad \text{where} \quad x_2 < x < x_4, \; y_2 < y < y_4 \tag{13}$$

There are two Eqs. (12)–(13) with two unknowns, x and y. By solving the system of equations, we may obtain $\{(x,y) : \text{ feasible solution of the system of equations (12)–(13)}\}$.

Note that the above method is based on the simple triangle abstraction of a drone sortie. The coordinates of pseudo node in the real road networks can be numerically found by computing the travel time of drone and truck along the arc length where the pseudo node is expected to be located. This is important because an analytic solution for the system of irrational functions may not be readily obtainable. This numerical approach applies regardless the truck route is a straight line or a curve, and the coordinate of pseudo node can be found given that the geometric information of the curved road is provided. For example, a curved truck route can be approximated with a few number of points by a Lagrange polynomial where the origin and destination

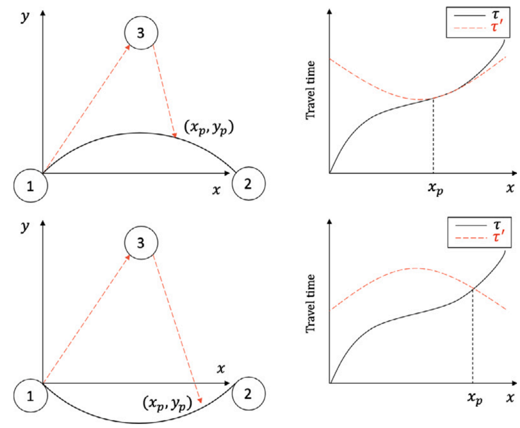


Fig. 20. Pseudo node location by comparing travel times in curved routes.

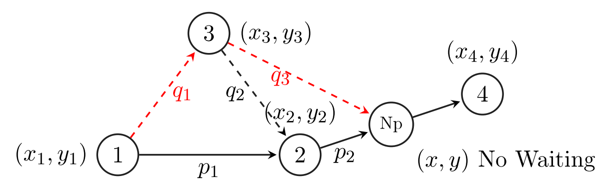


Fig. 21. Pseudo node candidate location.

nodes are located in the horizontal axis as shown in Fig. 20. Given a set of $k+1$ data points $(x_0,y_0),\ldots,(x_j,y_j)\cdots,(x_k,y_k)$ where no two $x'_j s$ are the same, the Lagrange polynomial $L(x)$ can be given as follows:

$$L(x) = \sum_{j=0}^{k} y_j l_j(x) \tag{14}$$

$$l_j(x) = \prod_{0 \le m \le k, m \ne j} \frac{x-x_m}{x_j-x_m}, \quad 0 \le j \le k \tag{15}$$

where $l_j(x)$ is the Lagrange basis polynomials. Note that $L(x)$ is the linear combination of $l_j(x)$'s with ordinates of data points y_j ($j = 1,\ldots,k$) as weights. The arc length, $s_a(x_p)$, along the truck arc from the origin to a possible pseudo node location $\mathrm{N_P}(x_p, y_p)$ can be computed by:

$$s_a(x_p) = \int_0^{x_p} \sqrt{1+\left(\frac{dL(x)}{dx}\right)^2}\, dx \tag{16}$$

The truck travel time is the truck arc length $s_a(x_p)$ divided by the truck speed v_t. The drone travel time is computed as the summation of travel times from the origin (node 1) to a drone node (node 3), τ'_{13}, and from the drone node (node 3) to the trace point (x_p, y_p) along the truck arc, $\tau'_{3,(x_p,y_p)}$. The location of pseudo node $\mathrm{N_P}$ is then the x and y coordinates of the trace point along the truck arc where the drone and truck travel time curves intersect. Fig. 20 illustrates the way a pseudo node location is computed along the curved routes, where τ represents the truck travel time while $\tau' = \tau'_{13} + \tau'_{1,(x_p,y_p)}$ denotes the drone travel time.

References

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, *52*(4), 965–981.

Andrei, M. (2017). UB engineer sees drones as more than eyes in the sky for disaster response. http://www.buffalo.edu/ubnow/stories/2017/07/drones-disaster-response.html. (Online accessed 30 March 2019).

Bouman, P., Agatz, N., & Schmidt, M. (2015). *Instances for the Tsp with Drone*. Zenodo, http://dx.doi.org/10.5281/zenodo.22245.

Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, *72*(4), 528–542.

Boyer, K. K., Prud'homme, A. M., & Chung, W. (2009). The last mile challenge: evaluating the effects of customer density and delivery window patterns. *Journal of Business Logistics*, *30*(1), 185–201.

Boysen, N., Briskorn, D., Fedtke, S., & Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, *72*(4), 506–527.

Burgess, M. (2016). DHL's delivery drone can make drops quicker than a car. http://www.wired.co.uk/article/dhl-drone-delivery-germany/. (Online accessed 19 July 2018).

Carlsson, J. G., & Song, S. (2017). Coordinated logistics with a truck and a drone. *Management Science*, *64*(9), 3971–4470.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, Article 105004.

Drexl, M. (2012). Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. *Transportation Science*, *46*(3), 297–316.

EasyAerial (2019). Drone-In-A-Box - Landing in moving vehicle. https://www.youtube.com/watch?v=rIwbq8VeO1M. (Online accessed 19 Februrary 2020).

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., & Rich, R. (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management*, *9*(2), 374–388.

Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., & Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C (Emerging Technologies)*, *114*, 657–680.

Goodchild, A., & Toy, J. (2018). Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry. *Transportation Research Part D: Transport and Environment*, *61*, 58–67.

Ha, Q. M., Deville, Y., Pham, Q. D., & Ha, M. H. (2015). Heuristic methods for the traveling salesman problem with drone. arXiv preprint arXiv:1509.08764.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C (Emerging Technologies)*, *86*, 597–621.

Johnson, L. (2017). 9 things you need to know about the Amazon Prime Air drone delivery service. http://www.digitalspy.com/tech/feature/a820748/amazon-prime-air-drone-delivery-service/. (Online accessed 18 July 2018).

Karak, A., & Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C (Emerging Technologies)*, *102*, 427–449.

Kastrenakes, J. (2017). UPS has a delivery truck that can launch a drone. http://www.theverge.com/2017/2/21/14691062/ups-drone-delivery-truck-test-completed-video/. (Online accessed 19 July 2018).

Kelleher, K. (2019). UPS, Matternet delivering medical samples in North Carolina using drones. *Association for Unmanned Vehicle Systems International*, https://www.auvsi.org/industry-news/ups-matternet-delivering-medical-samples-north-carolina-using-drones. (Online accessed 12 June 2019).

Khoufi, I., Laouiti, A., & Adjih, C. (2019). A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones*, *3*(3), 66.

Kim, S. J., Lim, G. J., Cho, J., & Côté, M. J. (2017). Drone-aided healthcare services for patients with chronic diseases in rural areas. *Journal of Intelligent and Robotic Systems*, *88*(1), 163–180.

Marin, L. (2016). The humanitarian drone and the borders: Unveiling the rationales underlying the deployment of drones in border surveillance. In *The future of drone use* (pp. 115–132). Springer.

Marinelli, M., Caggiani, L., Ottomanelli, M., & Dell'Orco, M. (2017). En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems*, *12*(4), 253–261.

Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C (Emerging Technologies)*, *54*, 86–109.

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C (Emerging Technologies)*, *110*, 368–398.

Ong, T. (2017). The first autonomous drone delivery network will fly above Switzerland starting next month. https://www.theverge.com/2017/9/20/16325084/matternet-autonomous-drone-network-switzerland/. (Online accessed 19 July 2018).

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, *72*(4), 411–458.

Poikonen, S., & Golden, B. (2019). Multi-visit drone routing problem. *Computers & Operations Research*, Article 104802.

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers & Operations Research*, *113*, Article 104802.

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, *70*(1), 34–43.

Ponza, A. (2016). *Optimization of drone-assisted parcel delivery* (Master's thesis), Italy: Università degli Studi di Padova.

Samad, T., Bay, J. S., & Godbole, D. (2007). Network-centric systems for military operations in urban terrain: The role of UAVs. *Proceedings of the IEEE*, *95*(1), 92–107.

Sandvik, K. B., & Lohne, K. (2014). The rise of the humanitarian drone: giving content to an emerging concept. *Millennium*, *43*(1), 145–164.

Savuran, H., & Karakaya, M. (2015). Route optimization method for unmanned air vehicle launched from a carrier. *Lecture Notes on Software Engineering*, *3*(4), 279–284.

Schermer, D., Moeini, M., & Wendt, O. (2019a). A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, *109*, 134–158.

Schermer, D., Moeini, M., & Wendt, O. (2019b). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C (Emerging Technologies)*, *106*, 166–204.

Scott, J., & Scott, C. (2017). Drone delivery models for healthcare. In *Proceedings of the 50th hawaii international conference on system sciences* (pp. 3297–3304).

Scott, J. E., & Scott, C. H. (2018). Models for drone delivery of medications and other healthcare items. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, *13*(3), 20–34.

Shu, C. (2017). Flytrex launches an autonomous on-demand drone delivery service in Iceland's capital. http://techcrunch.com/2017/08/23/flytrex-launches-an-autonomous-on-demand-drone-delivery-service-in-icelands-capital/. (Online accessed 20 July 2018).

Tu, P. A., Dat, N. T., & Dung, P. Q. (2018). Traveling salesman problem with multiple drones. In *Proceedings of the ninth international symposium on information and communication technology* (pp. 46–53).

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. *Optimization Letters*, *11*(4), 679–697.

Wang, Z., & Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation Research, Part B (Methodological)*, *122*, 350–364.

Xia, Y., Batta, R., & Nagi, R. (2017). Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment. *Operations Research*, *65*(3), 674–692.

Xia, J., Wang, K., & Wang, S. (2019). Drone scheduling to monitor vessels in emission control areas. *Transportation Research, Part B (Methodological)*, *119*, 174–196.

Yurek, E. E., & Ozmutlu, H. C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C (Emerging Technologies)*, *91*, 249–262.