Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles

Bai, Xiaoshan; Cao, Ming; Yan, Weisheng; Ge, Shuzhi Sam

Published in:
IEEE Transactions on Automation Science and Engineering

DOI:
10.1109/TASE.2019.2914113

Publication date:
2020

Citation for published version (APA):
Bai, X., Cao, M., Yan, W., & Ge, S. S. (2020). Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles. *IEEE Transactions on Automation Science and Engineering*, *17*(1), 248-260. https://doi.org/10.1109/TASE.2019.2914113

Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles

Xiaoshan Bai, Ming Cao, *Senior Member, IEEE,* Weisheng Yan, and Shuzhi Sam Ge, *Fellow, IEEE*

***Abstract*—This paper studies the precedence-constrained task assignment problem for a team of heterogeneous vehicles to deliver packages to a set of dispersed customers subject to precedence constraints that specify which customers need to be visited before which other customers. A truck and a micro drone with complementary capabilities are employed where the truck is restricted to travel in a street network and the micro drone, restricted by its loading capacity and operation range, can fly from the truck to perform the last mile package deliveries. The objective is to minimize the time to serve all the customers respecting every precedence constraint. The problem is shown to be NP-hard, and a lower bound on the optimal time to serve all the customers is constructed by using tools from graph theory. Then, integrating with a topological sorting technique, several heuristic task assignment algorithms are proposed to solve the task assignment problem. Numerical simulations show the superior performances of the proposed algorithms compared with popular genetic algorithms.**

***Note to Practitioners* — This work presents several task assignment algorithms for precedence-constrained package delivery for the team of a truck and a micro drone. The truck can carry the drone moving in a street network while the drone completes the last-mile package delivery. The paper's practical contributions are fourfold: First, the precedence constraints on the ordering of the customers to be served are considered, which enables complex logistic scheduling for customers prioritized according to their urgency or importance. Second, the package delivery optimization problem is shown to be NP-hard, which clearly shows the need for creative approximation algorithms to solve the problem. Third, the constructed lower bound on the optimal time to serve all the customers helps to clarify for practitioners the limiting performance of a feasible solution. Fourth, the proposed task assignment algorithms are efficient and can be adapted for real scenarios.**

***Index Terms*—Task assignment, precedence constraints, heterogeneous vehicles, topological sorting, heuristic algorithm.**

I. Introduction

The task assignment problem for one or multiple vehicles to visit a set of target locations has been extensively investigated in the past years due to its wide applications in logistics, terrain mapping, environmental monitoring, and disaster rescue [1]–[5]. The problem can be taken as a variant of the traveling salesman problem (TSP) or the vehicle routing problem (VRP), which are both NP-hard [6], [7]. The TSP focuses on designing one route with the minimum length for a salesman/vehicle to visit a set of dispersed customers while the VRP aims to employ multiple vehicles to efficiently deliver products/packages to a set of customers. In recent years, parcel delivery to customers is facing new challenges as e-commerce has grown vastly [8] where the benefit of using micro drones as additional support for package delivery has been identified [9]. Consequently, some leading retailers or distributors such as Amazon and DHL have planned to employ micro drones for small package deliveries. However, micro drones are subject to short operation range and small payload capacity which greatly restrict their efficiency to function in an autonomous delivery network [10]. To overcome the limitations, some investigation has been done to consider a heterogeneous team consisting of one carrier truck and one micro drone with complementary capabilities [10]–[13]. In [10], the package delivery problem for the truck and the drone has been formulated as an optimal path planning problem on a graph, and then the problem is reduced to the generalized traveling salesman problem. Murray and Chu [11] have formulated the heterogeneous parcel delivery problem as a mixed integer linear programming problem and further investigated two cases where one considers the release and recovery of the drone by using the truck while the other just uses the depot to release and recover the drone. Considering the drone's operation range and capacity constraint, Savuran and Karakaya [12] have designed a genetic algorithm (GA) to plan the route for the drone deployed on a mobile platform to visit a set of fixed targets. In [13], several worst-case analysis theorems have been investigated revealing the maximum amount of time that could be saved as a result of using trucks and drones in combination rather than employing trucks alone for delivering packages. In [11], [13], the truck itself is also allowed to deliver parcels to customers, which is different from [10].

In applications such as machine scheduling [14], [15], and vehicle routing [16]–[18], requests are often stipulated as precedence constraints. For instance, an autonomous assembly line, or a car manufacturing system, may require multiple production robots to provide service at locations in a given sequence, thus motivating the study for spatio-temporal requests [19]. In logistic scheduling, some customers/target locations can have priorities over the others to be served due to their interconnections as in the Dial-A-Ride Problem (DARP) [20] and the pickup and delivery task assignment problem [21], [22]. Pezzella *et al.* [15] developed a GA for

X. Bai and M. Cao are with the Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, The Netherlands (e-mail: (xiaoshan.bai, m.cao)@rug.nl).

W. Yan is with the School of Marine Science and Technology, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, China (e-mail: wsyan@nwpu.edu.cn).

S. S. Ge is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: samge@nus.edu.sg).

the flexible job-shop scheduling problem, in which the operations of different jobs are subject to precedence constraints (e.g. machine sequences). A mathematical programming model and a heuristic algorithm were presented in [17] for the combined vehicle routing and scheduling problem with time windows and additional temporal constraints. Cordeau *et al.* [20] conducted a review on the DARP, where the pickup and delivery requests for a set of customers need to respect the customers' origins and destinations. For the pickup and delivery problem with time windows, Ropke and Pisinger [22] designed an efficient large neighborhood search (LNS) heuristic, which consists of two processes, namely the removal process and inserting process. In the inserting process, the basic greedy heuristic inserts each of several requests, which have previously been removed in the removal process, into that vehicle's route such that the insertion causes the least increase in the value of the objective function. A dynamic programming formulation was developed in [23] for the precedence-constrained pickup and delivery problem with split loads, where all origins to be visited must be served before any destination to be visited on each route. In these cases, the precedence constraints on the visiting sequence of customers have to be respected, and the positioning of one customer in the sequence is directly affected by the customers which are required to be served earlier. Precedence constraints have been studied earlier in the so called sequence ordering problem [24], which is also referred to as the sequence problem with precedence constraints [25]. Constructing TSP tours while respecting some precedence constraints yields the precedence-constrained TSP, which was called PCTSP [26]. Considering the loading constraint of unmanned aerial vehicles and the precedence constraints on multiple visits at one target, a GA was proposed for the multi-vehicle task assignment [27]. For the TSP where a given subset of targets is required to be visited in some prescribed linear order, an algorithm guaranteeing quantifiable performances was designed in [28]. Each subset of targets with the linear visiting constraints can be treated as one single target, thus leading to the transformation of the TSP subject to the precedence constraints in [28] into the standard TSP. A topological sorting technique was integrated with a GA to solve the TSP with precedence constraints in [16]. The topological sorting technique guarantees that the planned path is feasible while the GA uses a crossover operator, mimicking the changes of the moon, to adjust the sequence for visiting the target locations. Later on, an improved GA based on topological sorting techniques was proposed in [18] to solve precedence-constrained sequencing problems. Only one chromosome is needed by the crossover operator to undergo the crossover evolution where each chromosome constructs a feasible solution to the problem.

Motivated by the existing literature just mentioned, we investigate the precedence-constrained heterogeneous delivery problem (PCHDP) for which one drone coordinates with one truck to efficiently deliver packages to a set of dispersed customers subject to precedence constraints that specify which customers need to be visited before which other customers. While one would ideally study the problem with time delivery deadlines, this problem is hard and as a result we consider a simplified version in which there are precedence constraints on the delivery order. We first investigate the feasible deployment patterns for the drone to travel from one customer to another in coordination with the truck, and then obtain the travel cost matrix specifying the feasible minimal time for the drone to fly between each pair of customers. Finally, integrating the topological sorting technique [16], [18], we design several heuristic task assignment algorithms to iteratively put the customers in an ordered manner respecting the precedence constraints. Our main contributions are as follows. Firstly, we construct a lower bound on the optimal solution by using tools from graph theory after ~~showing the~~ NP-hardness of the precedence-constrained task assignment problem. This lower bound can be used to approximately measure the quality of a solution compared with the optimal. Secondly, inspired by the two feasible deployment patterns for the drone to travel between two customers with the coordination of the truck in [10], we have exploited a different feasible deployment pattern. Lastly, two heuristic algorithms designed in the paper can obtain satisfying solutions within short computation time even when the number of customer locations is large.

The rest of this paper is organized as follows. In Section II, the formulation of the precedence-constrained package delivery for heterogeneous vehicles is given. Section III presents the problem analysis, and in Section IV several task assignment algorithms are presented. We show the simulation results in Section V and conclude the paper in Section VI.

II. Problem formulation

A. Problem setup

To rigorously formulate the problem, the definition of the arborescence of a digraph in graph theory is first introduced.

Definition 1: (arborescence [29]) An *arborescence* is a digraph with a single root from which, there is exactly one directed path to every other vertex.

Now we are ready to define the research problem PCHDP. A drone in coordination with a truck is deployed to deliver packages to a set of n dispersed customers subject to precedence constraints that specify which customers need to be visited before which other customers. Each customer receives one package to be delivered by the drone, and the truck is restricted to travel between a set of stopping/street points as vertices on a graph describing the topology of a street network. Each customer can be visited by the drone released from the truck from at least one stopping point vertex to ensure that all the customers can be served. The objective is to minimize the time when the last customer is served while satisfying every precedence constraint, and the constraints that the drone can carry only one package each time and has limited operation range. One illustration on the package delivery problem without any precedence constraint is shown in Fig. 1.

Remark 1: The motivation for minimizing the time when the last customer is served is that the total service time

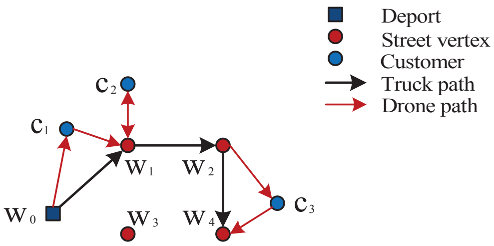

Fig. 1. One illustration on the heterogeneous package delivery problem with one drone coordinating with one truck to deliver parcels to three dispersed customers.

when the drone reaches the last customer is considered more important than the time when the truck and the drone return to the depot to increase the customer satisfaction index.

B. Formulation as an optimization problem

Let $\mathcal{C} = \{c_1, \ldots, c_n\}$ denote the set of indices of the customer locations, and the indices of the stopping point vertices are denoted by $\mathcal{W} = \{w_1, \cdots, w_m\}$. Let w_0 denote the depot, a special stopping point vertex, where the heterogeneous vehicle team is initially located. For each $i, k \in \mathcal{I}$ where $\mathcal{I} = \mathcal{W} \cup \{w_0\}$ and $j \in \mathcal{C}$, let the binary decision variable $\sigma_{ijk} = 1$ if and only if it is planned that the drone serves customer j by directly flying from stopping point vertex i and then flying to stopping point vertex k, and the binary variable $y_{ik} = 1$ if and only if it is planned that the truck directly travels from stopping point vertex i to stopping point vertex k. The position of each stopping point vertex, $i \in \mathcal{I}$, is denoted by $p(i)$. Let $d(i, j)$ denote the Euclidean distance between vertices i and j, and the binary variable $p_j^r = 1$ if one requires customer r to be visited before customer j, and $p_j^r = 0$ if there is no such a requirement. As shown in Fig. 2 (a), the digraph $\mathcal{G}^p = (V^p, E^p)$ consists of a subset of customer vertices in $\mathcal{C}$ and a set of directed edges E^p showing the precedence constraints among the vertices. It can be easily checked that the problem has feasible solutions only if no direct cycles exist in $\mathcal{G}^p$, i.e. $\mathcal{G}^p$ is acyclic. It is assumed that the drone flies with a constant speed v_d under the maximum fly distance L, and the truck travels with a constant speed v_t under no travel range constraint.

The variable t_j is employed to represent the time when customer $j, j \in \mathcal{C}$, is served, and $P(t)$ is the truck's position at time t. Then, the problem is to minimize the time for visiting all the customer locations

$$f = \max_{j \in \mathcal{C}} t_j, \tag{1}$$

subject to

$$\sum_{i,k \in \mathcal{I}} \sigma_{ijk} = 1, \quad \forall j \in \mathcal{C}; \tag{2}$$

$$\sigma_{ijk} - y_{ik} = 0, \quad \forall i, k \in \mathcal{I}, \forall j \in \mathcal{C}; \tag{3}$$

$$\sigma_{ijk}, y_{ik} \in \{0, 1\}, \quad \forall i, k \in \mathcal{I}, \forall j \in \mathcal{C}; \tag{4}$$

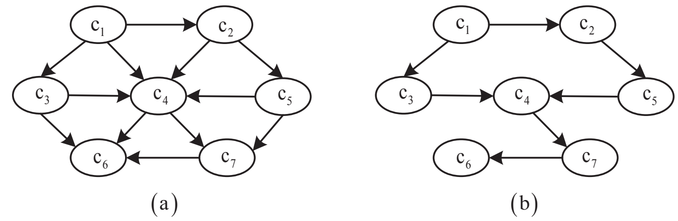

Fig. 2. The digraph $\mathcal{G}^p = (V^p, E^p)$ shows precedence constraints on serving the customer (target) vertices (a) Digraph in [16] and (b) Simplified digraph.

$$(P(t_j - \frac{d(i,j)}{v_d}) - p(i))\sigma_{ijk} = 0, \quad \forall i, k \in \mathcal{I}, \forall j \in \mathcal{C}; \tag{5}$$

$$(d(i,j) + d(j,k))\sigma_{ijk} \leq L, \quad \forall i, k \in \mathcal{I}, \forall j \in \mathcal{C}; \tag{6}$$

$$(t_r - t_j)p_j^r \leq 0, \quad \forall r, j \in \mathcal{C}. \tag{7}$$

Constraint (2) ensures that each customer is served; (3) guarantees the drone to be recharged by the truck after serving each customer; (5) makes sure the drone's path is feasible through coordinating with the truck, namely given $\sigma_{ijk} = 1$ the time that the drone can be released from the truck to serve customer j is the moment when the truck is at the stopping point vertex i; (6) ensures the drone's fly distance is within its capability; (7) guarantees the precedence constraints on visiting the customers are satisfied.

Remark 2: *In the problem formulation, it is assumed that the drone flies with a constant speed v_d constrained by the maximum fly distance L, and the truck travels with a constant speed v_t without any travel range constraint. The practicalities of the implementation on using trucks and drones under the assumptions for package delivery have been discussed in detail in [10], [11], [30], [31].*

After formulating the task assignment problem as a constrained minimization problem, we present in the following section the analysis of the optimization problem.

III. Problem Analysis

A. Proof of NP-hardness

We can simplify the digraph $\mathcal{G}^p = (V^p, E^p)$, specifying the precedence constraints on visiting the customer locations, whenever the following two conditions hold at the same time: (i) one customer vertex c_i has one edge directly pointing at another customer vertex c_j and (ii) c_i has multiple directed paths to c_j. An example is shown in Fig. 2 (a), where customer c_1 has three independent directed paths to c_4 as $c_1 \to c_3 \to c_4$, $c_1 \to c_2 \to c_4$ and $c_1 \to c_4$. Since c_2 is required to be visited before c_4 and c_1 is required to be visited before c_2, the precedence constraint from c_1 to c_4 becomes redundant. Then, after deleting some redundant precedence constraints, the digraph shown in Fig. 2 (a) can be simplified to Fig. 2 (b). It can be easily checked that the problem has feasible solutions only if no direct cycles exist in $\mathcal{G}^p$. Thus, we make this standing assumption for the rest of the paper.

Assumption 1: *$\mathcal{G}^p$ is acyclic.*

Now consider the undirected graph $\mathcal{G} = (V, E, D)$ consisting of a vertex set $V = \mathcal{C} \cup \mathcal{I}$, an edge set $E = E_{\mathcal{C}} \cup E_{\mathcal{I}}$, and a cost matrix D that saves the weight of each edge in E. $E_{\mathcal{I}}$ is a fully connected edge set containing the edge (w_i, w_k) for every pair of stopping point vertices $w_i, w_k \in \mathcal{I}$. $E_{\mathcal{C}}$ contains pairs of flight edges (w_i, c_j) and (c_j, w_k) for all $c_j \in \mathcal{C}$ and $w_i, w_k \in \mathcal{I}$ where the drone can fly from vertex w_i to serve customer c_j and then return to w_k. These w_i are called the *viable deployment vertices* for customer vertex c_j from which the drone can be released to reach c_j, and then the drone can return at lease one stopping point vertex w_k. The set V_j is employed to contain all such w_i for customer c_j. Let $D = (d(i,j))_{(m+n+1)\times(m+n+1)}, i, j \in V$, where $d(i,j)$ specifies the distance between the two vertices i and j associated with the edge (i,j).

In graph theory, the open TSP (OTSP) involves determining a Hamiltonian path with the minimal length connecting in sequence each vertex exactly once in a directed or undirected graph, and the TSP determines a Hamiltonian cycle with the minimal length that is a cycle. Determining whether such cycles and paths exist in graphs is the NP-complete Hamiltonian path problem [32, 199-200]. The requirement for the traveling salesman to return to the starting city does not change the computational complexity of the problem. So the OTSP is NP-hard as well [33]. Now we show that the optimization problem PCHDP is also NP-hard.

Theorem 1: There exists a set of non-empty precedence constraints for which the precedence-constrained task assignment problem PCHDP is NP-hard.

Proof 1: To prove the NP-hardness of the PCHDP, it suffices to show that: (i) every instance of the OTSP can be reduced to an instance of the PCHDP in polynomial time and (ii) an optimal solution to the PCHDP leads to an optimal OTSP solution. Let $\mathcal{G}' = (V', E', D')$ be an input to the OTSP, where $V' = \{v_0, v_1, \ldots, v_{n-1}\}$ contains the $n-1$ dispersed cities to be visited and the depot v_0 where the traveling salesman is initially located. To prove (i), $\mathcal{G}'$ is transformed into an input $\mathcal{G} = (V, E, D)$ and $\mathcal{G}^p$ of the PCHDP as shown in Fig. 3, which requires polynomial time.

We construct the PCHDP where each customer $c_i \in V, i \in \{1, \ldots, n\}$ corresponds to the vertex v_{i-1} in V', where c_i has exactly one unique viable deployment stopping point vertex w_i. A starting vertex (depot) w_0, where the truck and the drone start the delivery task, is added to V. For each edge $(v_i, v_j) \in E'$ with the weight $d'(v_i, v_j) \in D'$, add into E directed edges from customer vertex c_{i+1} to c_{j+1} as (c_{i+1}, w_{i+1}), (w_{i+1}, w_{j+1}) and (w_{j+1}, c_{j+1}) with the weight $d(w_{i+1}, w_{j+1}) = d'(v_i, v_j)$ while $d(c_{i+1}, w_{i+1}) = 0$ and $d(w_{j+1}, c_{j+1}) = 0$. In addition, add a bidirectional edge from w_0 to w_1 with $d(w_0, w_1) = 0$. Let the truck in the PCHDP have the same travel speed as that of the traveling salesman in OTSP. Finally, to construct $\mathcal{G}^p$, let customer vertex c_1 have precedence constraints on the other customers as the directed edges with arrows shown in Fig. 3 (c), and there are no other precedence constraints among the customers. Thus, the transformation from $\mathcal{G}'$ to $\mathcal{G}$ and $\mathcal{G}^p$ is constructed, and we have obtained the inputs to the PCHDP.

To prove (ii), after the transformation of $\mathcal{G}'$ to $\mathcal{G}$ and

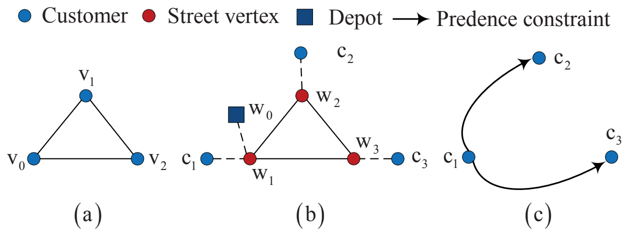

Fig. 3. A transformation from the OTSP on graph $\mathcal{G}'$ to the PCHDP on graph $\mathcal{G}$ and $\mathcal{G}^p$ where (a) $\mathcal{G}' = (V', E', D')$, (b) the $\mathcal{G} = (V, E, D)$ with $D(w_{i+1}, w_{j+1}) = D'(v_i, v_j), \forall v_i, v_j \in V'$, $D(w_0, w_1) = 0$, $D(c_i, w_i) = 0$ and $D(w_i, c_i) = 0, \forall w_i \in V$, and (c) $\mathcal{G}^p$.

$\mathcal{G}^p$, it is straightforward to see that an optimal solution to PCHDP is also optimal to the OTSP based on the edge weights of the graph $\mathcal{G}$ and the precedence constraints among the customers, shown in Fig. 3 (b) and Fig. 3 (c). From Fig. 3 (b) and Fig. 3 (c), a PCHDP solution is of the form $P_d = (w_0, w_1, c_1, w_1, \cdots, w_n, c_n, w_n)$ and $P_t = (w_0, w_1, \cdots, w_n)$ where P_d is the path of the drone and P_t is the path of the truck. The PCHDP solution can be employed to generate an OTSP path of the form $P' = (v_0, \cdots, v_{n-1})$ by either extracting the ordered stopping point vertices $(w_0, w_1, \cdots, w_n)$ from P_d as the truck needs to visit every w_i to serve the corresponding c_i or directly from P_t. If $E_{P'}$ contains the optimal sequence of edges in P', then $E_{P_t} = E_{P'}$. Since $d(w_0, w_1) = 0$, $d(c_i, w_i) = 0$ and $d(w_i, c_i) = 0$, it is straightforward to check that the shortest time for the drone to serve all the customers satisfies $\sum_{e \in E_{P_d}} d(e) = \sum_{e \in E_{P'}} d'(e)$. Thus, the proof is complete. $\square$

Remark 3: If every customer location only has one precedence constraint requiring it to be visited either before or after another customer location, the resulting problem is a variant of the single-vehicle Dial-a-Ride Problem to design one vehicle route to serve a set of customers at the required destinations [20].

B. A lower bound on the optimal solution

It can be costly to solve the PCHDP optimally due to the NP-hardness of the problem. As a consequence, it is natural to design heuristic algorithms to find sub-optimal solutions. Then, one issue arises on how to evaluate the performances of the sub-optimal solution as the optimal is typically unknown. In this section, a lower bound on the minimal time for the vehicles to serve all the customers while satisfying every precedence constraint is constructed through obtaining a min-cost arborescence (MCA) of a weighted digraph $\mathcal{G}^d = (V^d, E^d, D^d)$ by the Edmonds' algorithm [34]. The sum of all the edge weights of the MCA is minimal among all the arborescences of $\mathcal{G}^d$. The vertex set $V^d = \mathcal{C} \cup \{w_0\}$ consists of the indices of the n customer vertices and the depot. The edge set E^d, deduced from the digraph $\mathcal{G}^p$, contains every directed edge orienting from vertex c_r to c_j for all $c_r, c_j \in V^d$ if vertex c_r can be

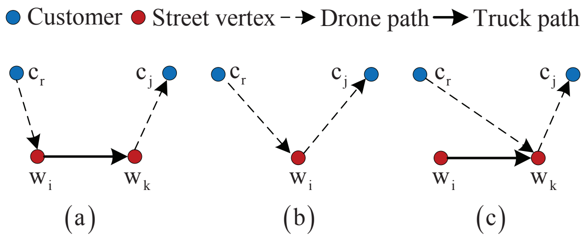

Fig. 4. Three potential micro deployment patterns for the drone to serve two customers c_r and c_j successively by the usage of stopping point vertices w_i and w_k: (a) Pattern α, (b) Pattern β and (c) Pattern γ.

visited before vertex c_j. The matrix D^d contains the feasible minimal time for the drone to fly from an arbitrary vertex c_r to every c_j, which is obtained as follows.

For every customer vertex c_r, let the set V_r contain the indices of the viable deployment stopping point vertices from which the drone is viable to be released to serve c_r. Extending the two feasible deployment patterns in [10], in Fig. 4 we show three potential micro time deployment patterns for the drone located at customer vertex c_r with the remaining fly distance L_{c_r} to serve customers c_j when the truck is located at stopping point vertex w_i. The corresponding travel times of the drone in the three micro deployment patterns are computed respectively as:

$$\begin{aligned} t_\alpha(c_r,c_j) &= \min_{\substack{L_{c_r}\geq d(c_r,w_i),\\ \forall w_i\in\mathcal{W},\forall w_k\in V_j}} \{\frac{d(c_r,w_i)}{v_d}+\frac{d(w_i,w_k)}{v_t} \\ &\quad +\frac{d(w_k,c_j)}{v_d}\}; \qquad (8)\\ t_\beta(c_r,c_j) &= \min_{\substack{L_{c_r}\geq d(c_r,w_i),\\ \forall w_i\in V_j}} \frac{d(c_r,w_i)+d(w_i,c_j)}{v_d}; \qquad (9)\\ t_\gamma(c_r,c_j) &= \min_{\substack{L_{c_r}\geq d(c_r,w_k),\\ \forall w_k\in V_j}} \{\max\{\frac{d(c_r,w_k)}{v_d},\frac{d(w_i,w_k)}{v_t}\} \\ &\quad +\frac{d(w_k,c_j)}{v_d}\}. \qquad (10) \end{aligned}$$

Let $L_{c_r} = L - \min_{\forall w_h\in V_r} d(w_h,c_r)$, then the shortest time for the drone located at the customer vertex c_r to serve c_j is

$$t^\star(c_r,c_j) = \min\{t_\alpha(c_r,c_j),t_\beta(c_r,c_j),t_\gamma(c_r,c_j)\}, \qquad (11)$$

which is the weight $D^d(c_r,c_j)$ for the directed edge $(c_r,c_j)\in E^d$.

We give an example on formulating the directed E^d and the associated D^d based on the directed $\mathcal{G}^p$ shown in Fig. 2 (b) and the corresponding weighted undirected $\mathcal{G}$. A vertex set S_r is used to save the indices of the customer vertices before which $c_r, \forall c_r\in\mathcal{C}$, can be served. S_r is achieved in a backward manner. Firstly, let $S_r=\mathcal{C}\setminus\{c_r\}, \forall c_r\notin V^p$. Then, in Fig. 2 (b), S_6 is first calculated as $S_6=\cup_{c_r\in\mathcal{C}\setminus V^p} c_r$. Afterwards, $S_r=\cup_{p_j'^r=1}(S_j\cup\{c_j\})$ where $p_j'^r=1$ if the customer vertex c_r has an edge directly pointing at customer vertex c_j in the simplified $\mathcal{G}^p$ as c_1 and c_2 in Fig. 2 (b). Thus, $S_7=S_6\cup\{c_6\}$. Then, S_i can be obtained iteratively for every $c_i\in V^d$. Finally, an edge $E^d(c_r,c_j)$ exists connecting vertex c_r and every $c_j\in S_r$, and the corresponding $D^d(c_r,c_j)=t^\star(c_r,c_j)$ saves the shortest feasible time for the drone to travel from c_r to c_j as shown in (11); for the other cases $D^d(c_r,c_j)=\infty$. Let f_a be the sum of all the edge weights of an MCA of $\mathcal{G}^d$, and f_o be the optimal for the objective function shown in (1). Now the property of the optimal solution is investigated.

Proposition 1: It holds that $f_a\leq f_o$.

Proof 2: According to Definition 1, an optimal path for the drone to serve all the customers while respecting all the precedence constraints on visiting the customers is an arborescence of the weighted digraph $\mathcal{G}^d$. As f_a is the sum of all the edge weights of an MCA of $\mathcal{G}^d$, it is straightforward to check that $f_a\leq f_o$.

□

Having done the theoretical analysis, we construct several heuristic algorithms in the next section.

IV. Task assignment algorithms

In this section, we first introduce one topological sorting technique, based on which we propose three task assignment algorithms.

A. *Topology sorting technique*

Through topological sorting, all the feasible paths in a directed acyclic graph can be obtained [35]. In [16], [18], the precedence-constrained TSP is solved by employing a topological sorting technique which iteratively sorts the customer vertices in $\mathcal{G}^{p'}$ without any predecessor in each iteration. Initially, let $\mathcal{G}^{p'}=\mathcal{G}^p$. The customer vertices without any predecessor are called *viable customer vertices*, which can be inserted into the TSP path behind the customer vertices that are the predecessors of them. Then, to choose which customer vertex among the viable customers to be inserted into the TSP path is determined by the task assignment principles, which will de presented later in Section IV. B. Once inserting a viable customer vertex, to update $\mathcal{G}^{p'}$, the customer vertex and the precedence constraints corresponding to the edges leaving the customer in the current $\mathcal{G}^{p'}$ are deleted. We give an example to show how to construct a TSP path to visit all the customers from the representation scheme by considering the precedence constraints shown in Fig. 2 (b). In the digraph, the first customer vertex sorted to be inserted into the TSP path is c_1, since c_1 is the only customer in $\mathcal{G}^{p'}$ without any predecessor. Then, c_1 is stored in the TSP path, and at the same time c_1 and the edges (c_1,c_3), (c_1,c_2) originating from c_1 are deleted from $\mathcal{G}^{p'}$. In the next iteration, c_2 and c_3 are viable customer vertices in the resulting $\mathcal{G}^{p'}$, which can then be inserted into the path after c_1. The process continues until all the customer vertices in $\mathcal{G}^p$ are inserted.

B. *Task assignment algorithms*

Using the topological sorting technique in [16], [18], the precedence-constrained package delivery task assignment problem can be solved by first iteratively inserting a viable customer vertex into the drone's path based on the travel cost matrix D^d, and then planing the path for the truck to ensure the drone's path is feasible. That is to say the truck needs to release the drone at one stopping point vertex to serve each customer, and then recharges the drone at one stopping point vertex after the customer is served as the drone's capacity of carrying packages is only one.

Let $\mathcal{R}_l$ save the indices of the ordered customers already inserted into the drone's path after iteration l, and the customer set $\mathcal{T}^A_{\mathcal{R}_l}$ contain the indices of those customer vertices that have no predecessor in $\mathcal{G}^{p'}$ and have not been inserted after iteration l. Let $\mathcal{T}^j_{\mathcal{R}_l} = \{c_r \in \mathcal{R}_l : (c_r, c_j) \in E^p\}$ for each $c_j \in \mathcal{T}^A_{\mathcal{R}_l}$, and the set $\mathcal{T}^j_a$ save the ordered locations in $\mathcal{R}_l$ in which the customer c_j is viable to be inserted while satisfying every precedence constraint on visiting c_j. Obviously, c_j can only be inserted into $\mathcal{R}_l$ behind all the customer vertices in $\mathcal{T}^j_{\mathcal{R}_l}$. If $\mathcal{T}^j_{\mathcal{R}_l} = \emptyset$, c_j is viable to be inserted at any position of $\mathcal{R}_l$. For the task assignment algorithms, $\mathcal{R}_l$ is initialized as $\{w_0\}$ where the drone and the truck are initially located.

1) Nearest inserting algorithm: The first heuristic algorithm is the nearest inserting algorithm (NIA) which puts the customer vertices in a sequence based on the time for the drone to travel from the customer vertices already inserted into the path to the one that can be inserted.

In iteration $l+1$, NIA finds the customer $c_{j^\star} \in \mathcal{T}^A_{\mathcal{R}_l}$ to be inserted and the associated inserting position $q^\star + 1$ as

$$(q^\star, c_{j^\star}) = \underset{q \in \mathcal{T}^j_a, c_j \in \mathcal{T}^A_{\mathcal{R}_l}}{\operatorname{argmin}} \; t^\star(\mathcal{R}_l(q), c_j), \tag{12}$$

where $\mathcal{T}^j_a = \{p, ..., |\mathcal{R}_l|\}$; $p = \max_{c_r \in \mathcal{T}^j_{\mathcal{R}_l}} find(\mathcal{R}_l = c_r)$ finds the farthest position to the end of $\mathcal{R}_l$ after which customer c_j is viable to be inserted; and $\mathcal{R}_l(q)$ is the qth ordered customer on the path $\mathcal{R}_l$. The operator $find(\mathcal{R}_l = c_r)$ finds the location in $\mathcal{R}_l$ where the customer vertex c_r is located. Then, the path $\mathcal{R}_l$ is updated to

$$\mathcal{R}_{l+1} = \begin{cases} \{\mathcal{R}_l, c_{j^\star}\}, \text{ if } q^\star = |\mathcal{R}_l|, \\ \{\mathcal{R}_l(1:q^\star), c_{j^\star}, \mathcal{R}_l(q^\star+1:|\mathcal{R}_l|)\}, \text{ otherwise,} \end{cases} \tag{13}$$

where $|\mathcal{R}_l|$ is the size of $\mathcal{R}_l$ and $\mathcal{R}_l(1:q^\star)$ saves the ordered customer vertices located between the first and the $q^\star$th positions of $\mathcal{R}_l$.

After the insertion of $c_{j^\star}$, we delete all the edges implying the precedence constraints initiating from $c_{j^\star}$ and the vertex $c_{j^\star}$ to update $\mathcal{G}^{p'}$. Then, the topological sorting technique is used to update $\mathcal{T}^A_{R_{l+1}}$ which saves the indices of viable customer vertices after iteration $l+1$. The inserting procedures (12) and (13) continue until all the customer vertices in $\mathcal{C}$ are inserted into the drone's path.

Now we show an example on how NIA works as follows. Assume that the current drone path for visiting the customers subject to the precedence constraints shown in Fig. 2 (b) is $\mathcal{R}_l = \{c_1, c_2, c_5\}$. Then, the viable customer set is $\mathcal{T}^A_{\mathcal{R}_l} = \{c_3\}$ as c_3 is the only customer without any predecessor after deleting the customer vertices already in $\mathcal{R}_l$ and the corresponding edges from $\mathcal{G}^{p'}$. Since c_1 is the only customer that is required to be visited before c_3, c_3 is viable to be inserted at any place behind c_1. Assume that $\mathcal{R}_l = \{c_1, c_3, c_2, c_5\}$ after the insertion of c_3. Then, the next viable customer c_4 can only be inserted after c_5 as c_5 is required to be visited before c_4, where $q = 4$ according to (12). One feasible drone path is $\mathcal{R}_l = \{c_1, c_3, c_2, c_5, c_4, c_7, c_6\}$ after iteratively using the topological sorting technique and the inserting procedure.

2) Minimum marginal-cost algorithm: The second task assignment algorithm is the minimum marginal-cost algorithm (MMA), which determines the next customer to be inserted and the corresponding inserting position based on the marginal travel time incurred by inserting the customer. The marginal travel time incurred by inserting customer c_j at the qth position of $\mathcal{R}_l$ is approximated as

$$t(\mathcal{R}_l \oplus_q c_j) - t(\mathcal{R}_l) = \begin{cases} t^\star(\mathcal{R}_l(q-1), c_j), \text{ if } q = |\mathcal{R}_l| + 1, \\ t^\star(\mathcal{R}_l(q-1), c_j) + t^\star(c_j, \mathcal{R}_l(q)) \\ -t^\star(\mathcal{R}_l(q-1), \mathcal{R}_l(q)), \text{ otherwise,} \end{cases} \tag{14}$$

where the operation $\mathcal{R}_l \oplus_q c_j$ inserts customer c_j at the qth position of $\mathcal{R}_l$. Target c_j is inserted to the end of $\mathcal{R}_l$ if $q = |\mathcal{R}_l| + 1$, and $t(\mathcal{R}_l)$ denotes the total travel time for the drone to deliver packages to all the customers in $\mathcal{R}_l$. The incurred marginal time can be approximated by the usage of the the travel cost matrix D^d shown in (11).

MMA determines the customer $c_{j^\star} \in \mathcal{T}^A_{\mathcal{R}_l}$ to be inserted into the path $\mathcal{R}_l$ and the associated inserting position $q^\star$ in iteration $l+1$ as

$$(q^\star, c_{j^\star}) = \underset{p+1 \le q \le |\mathcal{R}_l|+1, c_j \in \mathcal{T}^A_{\mathcal{R}_l}}{\operatorname{argmin}} \{t(\mathcal{R}_l \oplus_q c_j) - t(\mathcal{R}_l)\}, \tag{15}$$

where $p = \max_{c_r \in \mathcal{T}^j_{\mathcal{R}_l}} find(\mathcal{R}_l = c_r)$. Then, the path $\mathcal{R}_l$ is updated to

$$\mathcal{R}_{l+1} = \mathcal{R}_l \oplus_{q^\star} c_{j^\star}. \tag{16}$$

After the insertion of $c_{j^\star}$, all the edges implying the precedence constraints initiating from $c_{j^\star}$ and the vertex $c_{j^\star}$ are deleted to update $\mathcal{G}^{p'}$. The customer inserting process continues until all the customers are inserted into the path.

Remark 4: The MMA and the basic greedy heuristic used in the LNS [22] both insert a feasible customer/request into the vehicle's route with the least incurred increase in the value of the objective where the greedy heuristic is given a number of partial routes and a number of request to insert. If building the solution from scratch (an empty tour) and considering the precedence constraints when inserting each customer/request, the greedy heuristic will be the same as the MMA.

3) Second-order minimum marginal-cost algorithm: The customer ordering strategy (15) shows that the customers already ordered in the iteration l directly affect the customer to be inserted in the next iteration as well as the inserting position. To accommodate the future customer to be inserted, we propose the second-order minimum marginal-cost algorithm (SMMA) which calculates the cost incurred by inserting a customer both considering the current iteration and the future iteration. For SMMA, the customer $c_{j_1^\star}$ to be inserted and its inserting position $q_1^\star$ in iteration $l+1$ satisfy

$$c_{j_1^\star} = \operatorname*{argmin}_{\substack{p_2+1\le q_2\le|\mathcal{R}_l|+2,\\ c_{j_1}\in\mathcal{T}^A_{\mathcal{R}_l}, c_{j_2}\in\mathcal{T}^A_{\mathcal{R}_l}\setminus\{c_{j_1}\}}} \{t((\mathcal{R}_l\oplus_{q_1^\star} c_{j_1})\oplus_{q_2} c_{j_2}) - t(\mathcal{R}_l)\}, \tag{17}$$

where $q_1^\star = \operatorname{argmin}_{p_1+1\le q_1\le|\mathcal{R}_l|+1}\{t(\mathcal{R}_l\oplus_{q_1} c_{j_1}) - t(\mathcal{R}_l)\}$, $p_1 = \max_{c_r\in\mathcal{T}^{j_1}_{\mathcal{R}_l}} find(\mathcal{R}_l = c_r)$, and $p_2 = \max_{c_r\in\mathcal{T}^{j_2}_{\mathcal{R}_l\oplus_{q_1^\star}c_{j_1}}} find(\mathcal{R}_l\oplus_{q_1^\star} c_{j_1} = c_r)$. Then, in iteration $l+1$ the path $\mathcal{R}_l$ is updated to

$$\mathcal{R}_{l+1} = \mathcal{R}_l \oplus_{q_1^\star} c_{j_1^\star}. \tag{18}$$

After the insertion of $c_{j_1^\star}$, the vertex $c_{j_1^\star}$ and all the edges implying the precedence constraints initiating from $c_{j_1^\star}$ are deleted to update $\mathcal{G}^{p'}$. The customer inserting process continues until all the customers are inserted into the drone's path.

It should be noted that the stopping point vertices used for formulating the minimal travel time for the drone to fly between each two customers might not be feasible to construct a truck path coordinating with the drone to serve all the customers due to the truck's limited travel speed. To show more detail, let the stopping point vertex be w_r where the drone is released to serve customer c_r. Then, the feasible shortest time $t(c_r, c_j)$ for the drone to successively visit customer c_j after visiting c_r is obtained by equations from (8) to (11) where the drone's remaining fly distance L_{c_r} is $L - d(w_r, c_r)$, thus leading to $t^\star(c_r, c_j) \le t(c_r, c_j)$ in (11) as $L - d(w_r, c_r) \le L - \min_{\forall w_h\in V_r} d(w_h, c_r)$. That is because a longer remaining fly distance L_{c_r} would enable the drone to have more choices on choosing stopping point vertices w_i and w_k to serve customers c_r and c_j as shown in Fig. 4. Thus, after achieving the drone's path for visiting the customers based on the algorithms NIA, MMA and SMMA, $L_{c_r} = L - d(w_r, c_r)$ is used to calculate the feasible travel time for the drone released from stopping point vertex w_r to fly directly from customer c_r to customer c_j while considering the truck movement.

C. Computational Complexity

In this section, we analyze the computational complexity of running NIA, MMA and SMMA. The three algorithms iteratively insert a customer vertex to the drone's path whose length is $|\mathcal{R}_l| = l$ after the lth iteration of the inserting operation. The computational complexity of NIA is determined by (12) where finding p requires $|\mathcal{T}^j_{\mathcal{R}_l}||\mathcal{R}_l|$ basic operations in the $(l+1)$th iteration of the assignment. Thus, to find $q^\star$ and $c_{j^\star}$ in (12), $|\mathcal{T}^j_a||\mathcal{T}^A_{\mathcal{R}_l}||\mathcal{T}^j_{\mathcal{R}_l}||\mathcal{R}_l|$ basic operations are needed in the $(l+1)$th iteration, where $|\mathcal{T}^j_a| \le |\mathcal{R}_l|$, $|\mathcal{T}^A_{\mathcal{R}_l}| \le n - |\mathcal{R}_l|$, and $|\mathcal{T}^j_{\mathcal{R}_l}| \le |\mathcal{R}_l|$. As a consequence, at most $l^3(n-l)$ basic operations are required in the $(l+1)$th iteration. Taking the sum for l to change from 1 to n, we get the computational complexity of NIA $\sum_{l=1}^{l=n} l^3(n-l)$, resulting in $O(n^5)$. Here, a function $f(x)$ is said to be $O(g(x))$ if there are constants c and x' such that $f(x) \le cg(x)$ for all $x \ge x'$. Similar to NIA, the computational complexity of MMA is determined by (15) where at most $2|\mathcal{T}^j_a||\mathcal{T}^A_{\mathcal{R}_l}||\mathcal{T}^j_{\mathcal{R}_l}||\mathcal{R}_l|$ basic operations are required in the $(l+1)$th iteration. Thus, the computational complexity of MMA is also $O(n^5)$.

The computational complexity of SMMA is determined by (17) where finding p_1 requires $|\mathcal{T}^{j_1}_{\mathcal{R}_l}||\mathcal{R}_l|$ basic operations, finding $q_1^\star$ requires at most $2|\mathcal{R}_l|$ basic operations, and finding p_2 requires $|\mathcal{T}^{j_2}_{\mathcal{R}_l\oplus_{q_1^\star}c_{j_1}}||\mathcal{R}_l\oplus_{q_1^\star} c_{j_1}|$ basic operations in the $(l+1)$th iteration of the assignment. Thus, at most $2|\mathcal{R}_l||\mathcal{T}^A_{\mathcal{R}_l}|(|\mathcal{T}^A_{\mathcal{R}_l}|-1)(|\mathcal{T}^{j_1}_{\mathcal{R}_l}||\mathcal{R}_l|+2|\mathcal{R}_l|+|\mathcal{T}^{j_2}_{\mathcal{R}_l\oplus_{q_1^\star}c_{j_1}}||\mathcal{R}_l\oplus_{q_1^\star} c_{j_1}|)$ basic operations are required in the $(l+1)$th iteration, which is at most $2l(2l^2+4l+1)(n-l)^2$. Then, we know the computational complexity of SMMA is $\sum_{l=1}^{l=n} 2l(2l^2+4l+1)(n-l)^2$, which is $O(n^6)$.

Through integrating with the topology sorting technique, the task assignment algorithms NIA, MMA, and SMMA can generate feasible paths for the drone and the truck to visit all the customers while satisfying every precedence constraint. The NIA, MMA, and SMMA can be adjusted to solve task assignment problems such as the TSP and the VRP, where the position on the vehicles' current paths used to insert a target in each iteration can be anywhere if no precedence constraints apply.

Now we have presented all the theoretical results of this paper. In the following section, we carry out simulation studies.

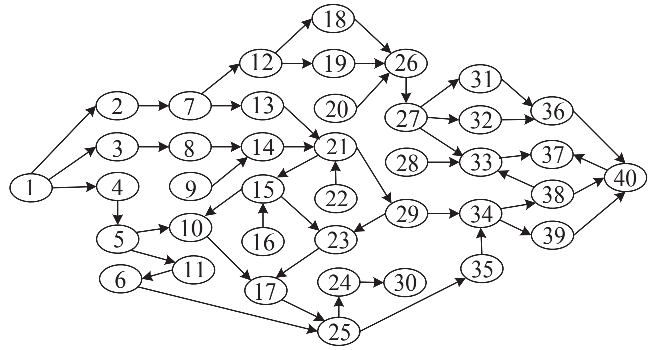

Fig. 5. The digraph $\mathcal{G}^p$ contains the precedence constraints on serving 40 customers.

V. Simulations

Numerical simulations are carried out to test the proposed algorithms in comparison with the GAs [16], [18] for the precedence-constrained assignment problems. The genetic parameters for the compared GAs, named GA02 and GA to distinguish them, are set according to [16] and [18] respectively as follows. For GA02, the maximum generation number is 500; the population size is 150; the crossover rate is 0.5 and the mutation rate is 0.2 [16]. For GA, the maximum generation number is 2000; the population size is 20; the crossover rate is 0.5 and the mutation rate is 0.05 [18]. All the experiments have been performed on an Intel Core $i5-4590$ CPU 3.30 GHz with 8 GB RAM, and the algorithms are compiled by Matlab under Windows 7. Apart

TABLE I. The solution quality q of the algorithms (A) for the traveling salesman problem with precedence constraints with 40 target locations subject to the precedence constraints shown in Fig. 5 under different instances (I).

I \ A	GA02	GA	NIA	MMA	SMMA
1	2.6761	2.4081	2.9080	2.3119	2.3119
2	3.2416	2.9354	3.0226	2.6937	2.7655
3	2.9583	2.6377	3.0937	2.4118	2.4005
4	3.1600	2.9423	3.5400	2.7504	2.6762
5	3.4219	3.0991	3.6018	3.1078	2.9766
6	3.8582	3.2057	3.5416	3.3479	3.1549
7	2.9699	2.6514	2.5874	2.7137	2.4528
8	2.8946	2.5995	2.7540	2.5036	2.4847
9	3.6385	3.2609	3.3647	2.8736	3.2579
10	3.4780	3.0181	3.5016	2.8095	2.7796

TABLE II. The corresponding computation time (s) for the algorithms (A) to get the solution to the traveling salesman problem with precedence constraints with 40 target locations subject to the precedence constraints shown in Fig. 5 under different instances (I).

I \ A	GA02	GA	NIA	MMA	SMMA
1	65.132	26.402	0.067	0.097	0.2018
2	65.174	25.570	0.055	0.056	0.2429
3	65.247	25.448	0.050	0.055	0.2611
4	64.051	26.467	0.050	0.055	0.2499
5	64.005	26.305	0.051	0.054	0.2936
6	63.895	25.980	0.050	0.053	0.2670
7	64.851	25.945	0.051	0.053	0.2112
8	64.761	25.906	0.051	0.051	0.2245
9	64.849	26.031	0.051	0.055	0.2399
10	64.508	25.885	0.050	0.054	0.2317

from evaluating the travel time f shown in (1), the solution quality of each algorithm is also quantified by

$$q = \frac{f}{f_a}, \tag{19}$$

where f_a is the sum of all the edge weights of an MCA of the weighted directed customer-vehicle graph $\mathcal{G}^d$. Since $f_a \leq f_o$, from Proposition 1 where f_o is the maximal travel time of an optimal solution, the value of the ratio q closer to 1 means a better performance of the solution.

The algorithms are first tested on the traveling salesman problem with precedence constraints where 40 target locations are subject to the precedence constraints shown in Fig. 5 which is simplified from Fig. 11 in [16]. Ten instances of the initial positions of the targets and the vehicle are randomly generated in a square area with the edge length 10^3m. For each instance, 20 trials of the GAs are performed to eliminate their randomness. The q of the proposed algorithms and the average q of the GAs on each instance, and the corresponding average computation time are shown in Table I and Table II respectively. First, GA ~~betters~~ than GA02 since its q values of every instance shown in Table I are smaller than that of GA02. Second, GA is better than NIA as most of its q are smaller than that of NIA, and so does MMA to NIA. Finally, SMMA is the best algorithm among all the algorithms as it achieves the smallest q for most of the instances in Table I. Table II shows that the

TABLE III. The average travel time (s) on 10 scenarios, resulting from the algorithms (A), for the vehicles to deliver packages to 40 customer locations subject to the precedence constraints shown in Fig. 5 under different operation ranges L of the drone and $v_d/v_t = 1$.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	15134	14628	14490	14192	14258	14243
NIA	16580	15973	15897	15793	15786	15759
MMA	14818	14207	13950	13883	13864	13862
SMMA	14596	14016	13725	13633	13619	13611

TABLE IV. The corresponding average solution quality q of the algorithms (A) on 10 scenarios for employing the vehicles to deliver packages to 40 customer locations subject to the precedence constraints shown in Fig. 5 under different operation ranges L of the drone and $v_d/v_t = 1$.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	2.4552	2.4462	2.4227	2.3721	2.3821	2.3795
NIA	2.6909	2.6738	2.6617	2.6442	2.6432	2.6385
MMA	2.4049	2.3726	2.3304	2.3195	2.3164	2.3160
SMMA	2.3682	2.3393	2.2911	2.2764	2.2742	2.2728

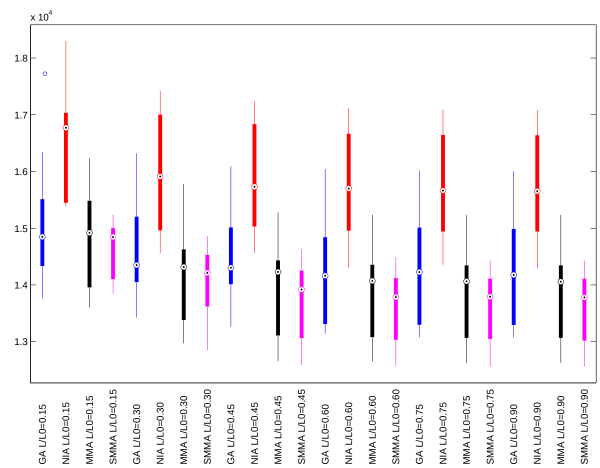

Fig. 6. Box plots of the travel times (s) on 10 scenarios, resulting from the algorithms, for the vehicles to deliver packages to 40 customer locations subject to the precedence constraints shown in Fig. 5 under different operation ranges L of the drone and $v_d/v_t = 1$.

computation time of SMMA is a little bit ~~larger~~ than those of the MMA and NIA. However, the computation time is relatively short compared with those of the GAs.

The designed algorithms are then tested on the task assignment problem PCHDP where 40 customer vertices are subject to the constraints shown in Fig. 5. Due to the better performance of GA over GA02, GA is used to compare the performance of the proposed algorithms. Ten scenarios of the initial positions of the customers are randomly generated in a square area with length $L_0 = 10^3$m, and there are 121 stopping point vertices evenly distributed in the area. The

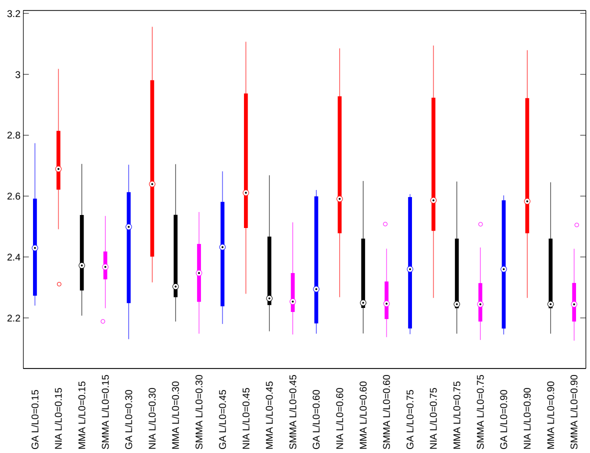

Fig. 7. Box plots of the corresponding solution quality q of the algorithms on 10 scenarios for employing the vehicles to deliver packages to 40 customer locations subject to the precedence constraints shown in Fig. 5 under different operation ranges L of the drone and $v_d/v_t = 1$.

TABLE V. The average travel time (s) on 10 scenarios, resulting from the algorithms (A), for the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	48059	46052	46074	45089	44822	45048
NIA	22733	20644	20778	20730	20732	20724
MMA	18665	17373	17320	17308	17307	17304
SMMA	18188	17182	17104	17095	17099	17097

TABLE VI. The corresponding average solution quality q of the algorithms (A) on 10 scenarios for employing the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	4.1413	3.9769	3.9763	3.8930	3.8714	3.8883
NIA	1.9590	1.7815	1.7938	1.7897	1.7899	1.7893
MMA	1.6068	1.4989	1.4944	1.4934	1.4933	1.4931
SMMA	1.5675	1.4834	1.4767	1.4760	1.4764	1.4762

drone and the truck initially located at the origin are first assumed to travel with the unit speed. For each scenario, we test the algorithms' performances when increasing the drone's maximum fly distance L. The average travel time to deliver packages to all the customers and the corresponding average q of the algorithms on the scenarios are shown in Table III and Table IV respectively. For the drone to deliver packages to all the customers while satisfying the precedence constraints, Table III first shows that the average travel time resulting from each algorithm decreases with the increase of the drone's operation range L. It is reasonable as the drone generally has more viable deployment stopping point vertices for serving each customer when increasing its operation range, which leads to more efficient paths for the drone to travel between two customers with the cooperation of the truck. The average q of the algorithms shown in Table IV always has the same changing trend as the average travel time when increasing the drone's maximum fly distance. This might be due to the smaller difference between the real travel time $t(c_r, c_j)$ and the $t^\star(c_r, c_j)$ shown in (11) when increasing the drone's operation range. For every instance shown in Table IV, GA has the smaller q compared with NIA, but it has the largest q compared with MMA and SMMA, which verifies the satisfying performance of MMA and SMMA. Table IV also shows that SMMA has the better performance than MMA. That is because SMMA employs the predictive strategy shown in (17) to determine the serving sequence of each customer. The box plots of the travel times for the vehicles to deliver packages to the 40 customers and the box plots of the q of the algorithms on the ten scenarios are shown in Fig. 6 and Fig. 7, respectively. First, the box plots denoting the performance of GA and NIA shown in Fig. 6 and Fig. 7 are comparatively taller than that of MMA and SMMA where SMMA has the lowest box plots, which shows the better performance of MMA and SMMA as those illustrated in Table III and Table IV. Second, when the drone's operation range L increases, the box plots of the travel times resulting from the algorithms shown in Fig. 6 generally downgrade, which shows that the travel time resulting from each algorithm generally decreases with the increase of the drone's operation range L. Third, Fig. 7 shows that the box plots of the corresponding solution quality q of MMA and SMMA do not vary much, and are comparatively shorter compared with those of the other algorithms, which suggests that MMA and SMMA are more robust than the other algorithms.

To further test the performance of the algorithms, simulation experiments on the problem with 120 customers where every customer has only one precedence constraint requiring it to be served either before or after another customer as in the Dial-a-Ride Problem [20]. For the simulation, 10 scenarios of the customers' initial locations and destinations are randomly generated in the same square area with the same number of stopping point vertices. The drone and the truck are also assumed to travel with the unit speed. For each scenario, we test the algorithms' performances when increasing the drone's operation range L. The average time to serve all the customers and the corresponding average q of the algorithms on the scenarios are shown in Table V and Table VI respectively. For every algorithm, Table V first shows that the average travel time for the drone to deliver packages to all the customers generally decreases when increasing the drone's operation range, which is the same compared with Table III. Table V also shows that the three proposed algorithms have a better performance compared

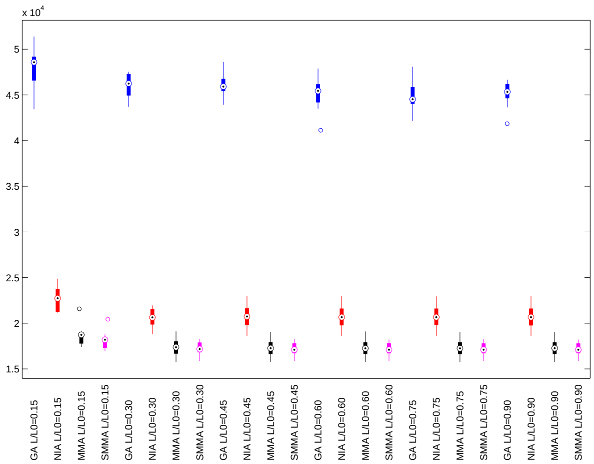

Fig. 8. Box plots of the travel times (s) on 10 scenarios, resulting from the algorithms, for the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.0$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

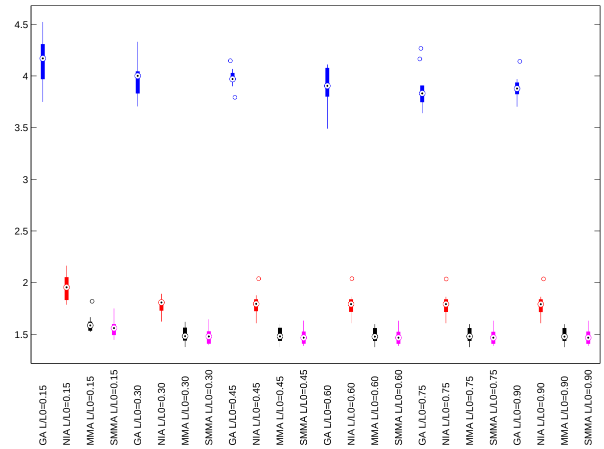

Fig. 9. Box plots of the corresponding solution quality q of the algorithms on 10 scenarios for employing the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.0$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

with the GA where their average q values are within twice of the optimal as shown in Table VI. That again verifies the satisfying performance of MMA and SMMA. The box plots of the travel times for the vehicles to deliver packages to the 120 customers and the box plots of the q of the algorithms on the ten scenarios are shown in Fig. 8 and Fig. 9, respectively. First, the box plots denoting the performance of GA shown in Fig. 8 and Fig. 9 are comparatively taller than that of NIA, MMA and SMMA where SMMA has the lowest box plots, which shows the better performance of NIA, MMA and SMMA as those illustrated in Table V and Table VI. Second, when the drone's operation range L increases, the box plots of the travel times resulting from the algorithms shown in Fig. 8 generally downgrade, which is the same as shown in Fig. 6. Third, Fig. 9 shows that the box plots of the corresponding solution quality q of NIA, MMA and SMMA do not vary much, and are comparatively shorter compared with those of GA, which suggests that NIA, MMA and SMMA are more robust than GA for these experiments with a large problem size.

Finally, for the same environment setup for the 120 customers, we investigate the algorithms' performances when increasing both the drone's fly speed v_d and maximum fly distance L. The truck is assumed to travel with the unit speed as $v_t = 1$ while the drone's speed is increase to $v_d/v_t = 1.4$. The average time to deliver packages to all the customers and the corresponding average q of the algorithms on the 10 scenarios are shown in Table VII and Table VIII respectively. For each algorithm, the average travel time shown in Table VII also decreases with the increase of the drone's operation range L. Table VII also shows that NIA, MMA and SMMA have better performance compared with GA, and Table VIII

TABLE VII. The average travel time (s) on 10 scenarios, resulting from the algorithms (A), for the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.4$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	47335	41759	38672	36115	35315	34946
NIA	21017	17678	17199	16999	16946	16863
MMA	16497	14763	14464	14381	14359	14344
SMMA	16400	14495	14116	14065	14066	14053

TABLE VIII. The corresponding average solution quality q of the algorithms (A) on 10 scenarios for employing the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.4$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	5.5826	4.9611	4.6161	4.3278	4.2482	4.2150
NIA	2.4792	2.0988	2.0518	2.0346	2.0360	2.0320
MMA	1.9444	1.7527	1.7256	1.7210	1.7251	1.7285
SMMA	1.9314	1.7207	1.6840	1.6832	1.6858	1.6934

shows that the average q values of MMA and SMMA are still within twice of the optimal, which verifies the superior performance of MMA and SMMA. Table V and Table VII show that the drone's average travel time to serve all the customers decreases more rapidly when increasing the drone's speed in comparison with increasing its maximum fly distance. That is because the time for the drone to travel

TABLE IX. The corresponding average computation time (s) for the algorithms to plan the paths for the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.4$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

A \ L/L_0	0.15	0.30	0.45	0.60	0.75	0.90
GA	128.88	128.85	128.81	128.50	130.62	131.75
NIA	0.27	0.31	0.31	0.30	0.31	0.32
MMA	0.85	0.84	0.85	0.86	0.87	0.85
SMMA	37.63	37.55	37.36	37.63	37.42	37.60

between two customers decreases when increasing its speed, which directly leads to the decrease of the total time to serve all the customers. However, the drone's speed cannot be increased too much to ensure safety when delivering the parcels in the city. Meanwhile, increasing the drone's operation range might not necessarily decrease its total travel time as the drone needs to return to the truck to be recharged with parcel after serving each customer. In Table IX, we also show the corresponding average computation time for the algorithms to achieve the solutions shown in Table VII. Table IX shows that the average computation time of GA are far larger than those of NIA, MMA and SMMA where the SMMA is most time-consuming among the proposed algorithms as indicated in Section IV-C. Concluding from the above analysis, MMA and SMMA are more efficient than NIA and GA in every instance while SMMA performs better than MMA. However, SMMA needs more computation time than MMA. Then, it is suggested to use MMA for planning the routes for the vehicles online as it can achieve the satisfying solution under short computation time while to use SMMA if more computation time is allowed as it offers a better solution. The box plots of the travel times for the vehicles to deliver packages to the 120 customers and the box plots of the q of the algorithms on the ten scenarios are shown in Fig. 10 and Fig. 11, which show the same changing trend as illustrated in Fig. 8 and Fig. 9.

Fig. 12 and Fig. 13 present a realistic package delivery scenario on a Google street map of a residential neighbourhood in Groningen, The Netherlands. The neighborhood considered is a residential area outside the busy city center, but not too far away from the city's high-speed ring road. So it is an ideal test area for the possible coordination between the truck and drone. For each of the drone's landing on a delivery point or each of its loading of one package on the truck, we assume a landing time of 30s is needed. We assume the speeds of the drone and the truck are $v_d = 30$ km/hr and $v_t = 40$ km/hr, respectively. Fig. 12 shows the paths for the truck and the drone to visit 20 delivery points where the visiting of the target locations respects the precedence constraints shown in Fig. 5, and $L = 250$m. The total time for the drone and the truck to visit all the delivery points is 1849.6s, which is approximately a reduction of 10% in the time to serve all the customers by using a single delivery truck. Fig. 13 shows the paths for the truck and the drone to visit the 20 delivery points where the drone's maximum flight range is increased to $L = 300$m. The total time for the drone and the truck to visit all the delivery points is 1775.9s. It is worth mentioning that the time to deliver packages to the 20 customers decreases in the two scenarios even though the sequences for serving the customers are the same, which is due to the increase of the drone's operation range.

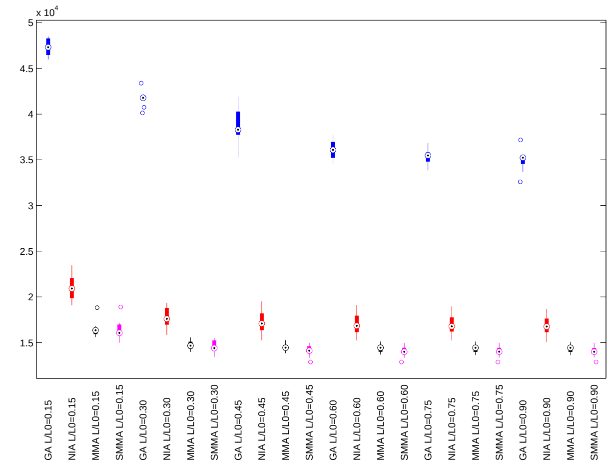

Fig. 10. Box plots of the travel times (s) on 10 scenarios, resulting from the algorithms, for the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.4$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

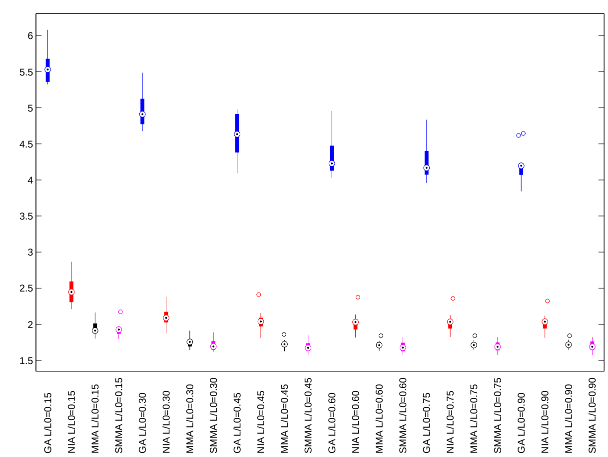

Fig. 11. Box plots of the corresponding solution quality q of the algorithms on 10 scenarios for employing the vehicles to deliver packages to 120 customer locations under different operation ranges L of the drone and $v_d/v_t = 1.4$, where every customer has only one precedence constraint requiring it to be served either before or after another customer.

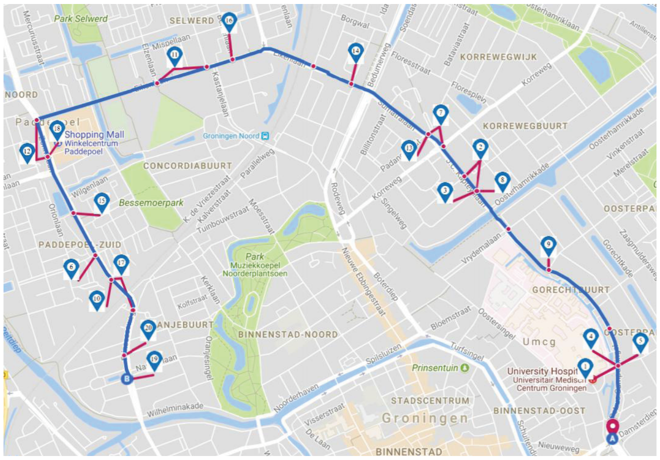

Fig. 12. The paths resulting from SMMA for the truck and the drone to deliver packages to 20 customers subject to the precedence constraints shown in Fig. 5 where $L = 250$m and $v_d = 30$ km/hr and $v_t = 40$ km/hr. The total travel time of the drone is 1849.6s.

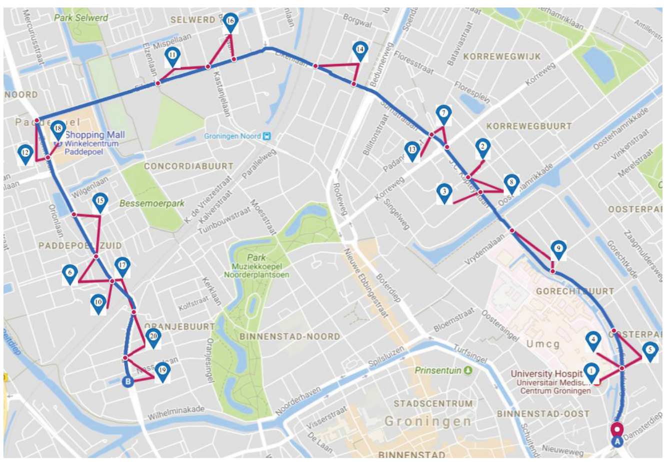

Fig. 13. The paths resulting from SMMA for the truck and the drone to deliver packages to 20 customers subject to the precedence constraints shown in Fig. 5 where $L = 300$m and $v_d = 30$ km/hr and $v_t = 40$ km/hr. The total travel time of the drone is 1775.9s.

VI. Conclusion

In this paper, we have investigated the precedence-constrained package delivery problem where one drone coordinates with one truck to efficiently deliver parcels to a set of customers while satisfying the precedence constraints on visiting the customers. The problem has been shown to be NP-hard and a lower bound on the minimum time to serve all the customers has been found. Integrated with a topological sorting technique, we have designed several heuristic task assignment algorithms. Numerical experiments have shown that the proposed algorithms can quickly obtain satisfying solutions to the precedence-constrained task assignment problem compared with the existing genetic algorithm. The proposed algorithms will be extended to the precedence-constrained package delivery with one truck coordinating with multiple drones. Another research direction is to investigate the precedence-constrained package delivery with one truck coordinating with one drone where the drone can deliver multiple packages in one run.

References

[1] C.-H. Fua and S. S. Ge, "Cobos: Cooperative backoff adaptive scheme for multirobot task allocation," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1168–1178, 2005.

[2] S. L. Smith and F. Bullo, "Monotonic target assignment for robotic networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2042–2057, 2009.

[3] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 485–497, 2010.

[4] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 327–341, 2015.

[5] X. Bai, W. Yan, S. S. Ge, and M. Cao, "An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field," *Information Sciences*, vol. 453, pp. 227–238, 2018.

[6] E. L. Lawler, J. K. Lenstra, A. R. Kan, D. B. Shmoys *et al.*, *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley New York, 1985, vol. 3.

[7] J. K. Lenstra and A. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[8] E. Morganti, S. Seidel, C. Blanquart, L. Dablanc, and B. Lenz, "The impact of e-commerce on final deliveries: alternative parcel delivery services in france and germany," *Transportation Research Procedia*, vol. 4, pp. 178–190, 2014.

[9] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, p. 460, 2015.

[10] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[11] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[12] H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air vehicle deployed on a moving carrier," *Soft Computing*, vol. 20, no. 7, pp. 2905–2920, 2016.

[13] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optimization Letters*, vol. 11, no. 4, pp. 679–697, 2017.

[14] V. S. Gordon, C. N. Potts, V. A. Strusevich, and J. D. Whitehead, "Single machine scheduling models with deterioration and learning: handling precedence constraints via priority generation," *Journal of Scheduling*, vol. 11, no. 5, p. 357, 2008.

[15] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3202–3212, 2008.

[16] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, 2002.

[17] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European journal of operational research*, vol. 191, no. 1, pp. 19–31, 2008.

[18] Y. Yun and C. Moon, "Genetic algorithm approach for precedence-constrained sequencing problems," *Journal of Intelligent Manufacturing*, vol. 22, no. 3, pp. 379–388, 2011.

[19] S. Chopra and M. Egerstedt, "Spatio-temporal multi-robot routing," *Automatica*, vol. 60, pp. 173–181, 2015.

[20] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of operations research*, vol. 153, no. 1, pp. 29–46, 2007.

[21] M. W. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transportation science*, vol. 29, no. 1, pp. 17–29, 1995.

[22] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.

[23] M. Nowak, M. Hewitt, and C. C. White III, "Precedence constrained pickup and delivery with split loads," *International Journal of Logistics Research and Applications*, vol. 15, no. 1, pp. 1–14, 2012.

[24] L. F. Escudero, "An inexact algorithm for the sequential ordering problem," *European Journal of Operational Research*, vol. 37, no. 2, pp. 236–249, 1988.

[25] G. Steiner, "On the complexity of dynamic programming for sequencing problems with precedence constraints," *Annals of Operations Research*, vol. 26, no. 1, pp. 103–123, 1990.

[26] M. Kubo and H. Kasugai, "The precedence constrained traveling salesman problem," *Journal of the Operations Research Society of Japan*, vol. 34, no. 2, pp. 152–172, 1991.

[27] T. Shima, S. J. Rasmussen, A. G. Sparks, and K. M. Passino, "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms," *Computers & Operations Research*, vol. 33, no. 11, pp. 3252–3269, 2006.

[28] H.-J. Böckenhauer, T. Mömke, and M. Steinová, "Improved approximations for tsp with simple precedence constraints," *Journal of Discrete Algorithms*, vol. 21, pp. 32–40, 2013.

[29] S. G. Williamson, *Combinatorics for computer science*. Courier Corporation, 1985.

[30] S. Mourelo Ferrandez, T. Harbison, T. Webwer, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *Journal of Industrial Engineering and Management*, vol. 9, no. 2, pp. 374–388, 2016.

[31] S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks*, vol. 70, no. 1, pp. 34–43, 2017.

[32] M. R. Garey and D. S. Johnson, *Computers and intractability*. WH Freeman New York, 2002, vol. 29.

[33] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *Foundations of Computer Science, 1976., 17th Annual Symposium on*. IEEE, 1976, pp. 216–227.

[34] J. Edmonds, "Optimum branchings," *Mathematics and the Decision Sciences, Part*, vol. 1, pp. 335–345, 1968.

[35] M. A. Weiss, *Data Structures and Algorithm Analysis in C*. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1993.

Xiaoshan Bai has been a PhD student with the Engineering and Technology Institute (ENTEG) at the University of Groningen, the Netherlands, since September 2015. He received the Bachelor's degree in Electrical Engineering and Automation from Xi'an Shiyou University, Xi'an, China, in 2012. From September 2012 to January 2015, he was a master student of the Northwestern Polytechnical University (NWPU), Xi'an. He was a research fellow in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from January to July 2015. His main research interest is in multi-vehicle task assignment, and vehicle path planning.

Ming Cao is currently a Professor of Systems and Control with the Engineering and Technology Institute (ENTEG) at the University of Groningen, the Netherlands, where he started as a tenure-track Assistant Professor in 2008. He received the Bachelor's degree in 1999 and the Master's degree in 2002 from Tsinghua University, Beijing, China, and the Ph.D. in 2007 from Yale University, New Haven, CT, USA, all in Electrical Engineering. From September 2007 to August 2008, he was a postdoctoral research associate with the Department of Mechanical and Aerospace Engineering at Princeton University, Princeton, NJ, USA. He worked as a research intern during the summer of 2006 with the Mathematical Sciences Department at the IBM T. J. Watson Research Center, NY, USA. He is the 2017 and inaugural recipient of the Manfred Thoma Medal from the International Federation of Automatic Control (IFAC) and the 2016 recipient of the European Control Award sponsored by the European Control Association (EUCA). He is an Associate Editor of IEEE Transactions on Automatic Control, IEEE Transactions on Circuits and Systems and Systems and Control Letters. He is also a ~~member~~ of the IFAC Technical Committee on ~~Networked~~ Systems. His main research interest is in autonomous agents and multi-agent systems, mobile sensor networks and complex networks.

Weisheng Yan received the B.Sc., M.Sc. and Ph.D. degrees in mechanical engineering and automatic control from Northwestern Polytechnical University (NWPU), Xi'an, China, in 1990, 1993 and 1999, respectively. Since 1993, he has been with the Department of Mechanical Engineering and Automation, School of Marine Science and Technology, NWPU, Xi'an, China, as Lecturer (1993), Associate Professor (1999) and Full Professor (2003). From 2003 to 2013, he served as the Dean of the Department of Mechanical Engineering and Automation. Since 2013, he has been serving as the Associate Dean of the School. His research interests are mainly focused on advanced control theory and control of underwater vehicles.

Shuzhi Sam Ge received the B.Sc. from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986, and Ph.D. from the Imperial College, London, UK, in 1993. He is the Founding Director of the Social Robotics Laboratory, Interactive Digital Media Institute, National University of Singapore. He is a Professor in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He is also the Director of the Center for Robotics, University of Electronic Science and Technology of China, Chengdu, China. He has authored or co-authored seven books and more than 300 international journal and conference papers. His current research interests include social robotics, multimedia fusion, medical robots, and intelligent systems. Dr. Ge is the Editor-in-Chief of the International Journal of Social Robotics. He has served/been serving as an Associate Editor for a number of flagship journals. He also serves as an Editor of the Taylor & Francis Automation and Control Engineering Series. He also served as the Vice President of Technical Activities, 2009C2010, and the Vice President for Membership Activities, 2011C2012, IEEE Control Systems Society. He is a fellow of IEEE, IFAC and IET.