Journal of Information Processing Vol.25 655–666 (Aug. 2017)

[DOI: 10.2197/ipsjjip.25.655]

Regular Paper

Routing of Carrier-vehicle Systems with Dedicated Last-stretch Delivery Vehicle and Fixed Carrier Route

Mohd Shahrizan bin Othman[1,2,a)] Aleksandar Shurbevski[1,b)] Yoshiyuki Karuno[3,c)]
Hiroshi Nagamochi[1,d)]

Received: November 7, 2016, Accepted: May 16, 2017

Abstract: We examine a routing problem that arises when an unmanned aerial vehicle (UAV), or *drone*, is used in the last-stretch of parcel delivery to end customers. In the scenario that we study, a delivery truck is dispatched carrying a shipment of parcels to be delivered to customers. While the truck is following a predetermined route, a drone is charged with making the last-stretch delivery of a parcel from the truck to a customer's doorstep. Given a set of customers to be served and a set of rendezvous points where the drone can meet with the truck to pick up a parcel, we ask what the quickest way is of delivering all parcels to the end customers. We model this problem as a problem of finding a special type of a path in a graph of a special structure, and show that the graph problem is NP-hard even when all edge weights are restricted to be 1 or 2. Furthermore, we identify a special instance type that can be solved optimally in polynomial time. Finally, we propose a polynomial-time approximation algorithm for the graph problem in metric graphs, and show that its approximation ratio is bounded above by 2 in restricted metric graphs.

Keywords: vehicle routing, NP-hardness, approximation algorithm, polynomial time algorithm

1. Introduction

Unmanned Aerial Vehicles (UAV-s), also known as *drones*, have recently been popular in many areas including the military, command, control and communications (3C) [6], remote sensing and scientific research [14], as well as in precision agriculture [15]. One of the areas which gather a lot of attention with announcements for using drones is parcel delivery [2], [3], [13]. Major delivery and logistics companies have already started investigating ways in which drones could improve the efficiency of their operations and widen the range of services that they could offer.

In this study, we approach a scenario in which a drone is used in tandem with a delivery truck for the *last-stretch* or *last-mile* delivery of parcels to customers' doorsteps. The truck departs from a distribution center carrying a drone and parcels for a set C of customers, and moves along a predetermined route. The last-stretch deliveries of parcels from the truck to a customer's doorstep are performed exclusively by the drone. The drone has a payload capacity of at most one parcel, and hence must return to the truck after each delivery. The drone must return to the truck at the end of the truck's predetermined route. Moreover, the drone can only rendezvous with the truck at a given set R of points along the truck's predetermined route. Notice that the order in which points of the set R are visited by the truck along its route defines a total order on the set R. While the drone is making a delivery of a parcel, the truck may either wait for it to return at the previous rendezvous point, or proceeds along its route and will intercept the drone at some future rendezvous point in the set R. Our aim is to determine a routing policy for the drone, such that all parcels are delivered in the least amount of time.

With a motivation of cooperatively routing heterogeneous vehicles with different capabilities, a similar routing problem was studied by Garone et al. [5], who named the scenario *carrier-vehicle systems*.

A closely related model to our problem appears in the work of Mathew et al. [9]. In their study, they investigate a combination of a drone and a truck to perform parcel deliveries, where last-stretch deliveries are performed solely by the drone, which delivers parcels between the truck and a customer's doorstep. The truck in turn is routed along a street network. For their original problem model, Mathew et al. [9] give a polynomial-time reduction to the Generalized TSP (GTSP) and report computational experiments using a solver for the GTSP. In addition, they also examined a special case of their problem where the drone's route alternately visits customers and points of a fixed set of depots. For this special case, they give a reduction to the TSP as well as a brute-force exponential-time algorithm, but do not comment on the computational complexity of the problem itself. Our problem models are in fact a special case of the problems investigated by Mathew et al. [9], where we assume that the truck's route is predetermined.

Another related problem is the Traveling Salesman Problem with Drone (TSP-D) in the work of Agatz et al. [1], where they work on the combination of a single truck and a single drone to

[1] Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606–8501, Japan
[2] Technical University of Malaysia Malacca, Malacca, Malaysia
[3] Faculty of Mechanical Engineering, Kyoto Institute of Technology, Kyoto 606–8585, Japan
[a)] shah@amp.i.kyoto-u.ac.jp
[b)] shurbevski@amp.i.kyoto-u.ac.jp
[c)] karuno@kit.ac.jp
[d)] nag@amp.i.kyoto-u.ac.jp

find the shortest tour in terms of time, to serve all customer locations by either the truck or the drone. They assume that both the truck and the drone travel on the road network, and the drone's speed is a constant α times higher than the truck's speed. They proposed a $(2 + \alpha)$-approximation algorithm for the TSP-D, with the minimum spanning tree heuristic for the TSP and no drone deliveries. They also allow the drone's launch and rendezvous to be at the same point, as in Mathew et al. [9]. They formulate this problem as a mixed-integer programming (MIP) model and develop several "Truck First, Drone Second" procedures, based on local search and dynamic programming.

Murray and Chu [10] introduced a similar problem of using a combination of a drone and a truck for the last-stretch deliveries, naming it the Flying Sidekick Traveling Salesman Problem (FSTSP). In their problem setting, it is assumed that delivery requests can be satisfied by either the drone or the truck, and they ask to simultaneously determine a route for both the truck and the drone, to minimize the total time it takes to complete all deliveries. They proposed a MIP formulation and a heuristic to the problem. Their heuristic is based on a "Truck First, Drone Second" idea, in which they first construct a route for the truck to perform all deliveries by solving a TSP problem over the set of customers awaiting a delivery. Next, they run a relocation procedure which iteratively checks for each vertex whether the route can be improved by having the drone instead of the truck perform the corresponding delivery.

Furthermore, Ha et al. [7] studied a variant of the TSP-D as introduced in the work of Agatz et al. [1], albeit with a formulation that excludes the possibility to have a launch and rendezvous in the same point. They called their problem the Min-Cost TSP-D. The problem is first formulated mathematically and two algorithms are proposed for the solution. The first algorithm, the Traveling Salesman Problem Local Search (TSP-LS), is inspired from the work of Murray and Chu [10], and the second one is a Greedy Randomized Adaptive Search Procedure (GRASP).

Ponza [11] gives an extensive overview of current results, and examines several heuristic solution methods, including a Simulated Annealing (SA) approach.

We examine four particular cases of the problem arising in the scenario outlined above. In the first setting, the drone immediately takes off from the truck after getting a parcel, and while the drone is making a delivery of a parcel, the truck is allowed to wait for the drone to return at the previous rendezvous point. The second problem model has a similar setting to the previous problem model. The only difference from the first setting is that the truck is not allowed to wait for the drone to return at the previous rendezvous point. We term the former the Alternating Last-Stretch Delivery Problem, or ALSDP for short. We call the later problem the No-Wait Alternating Last-Stretch Delivery Problem, or NW-ALSDP. In the next two settings, we assume that the drone may "hitch a ride" on the truck before proceeding to its next delivery. When the truck is allowed to wait for the drone to return at the previous rendezvous point while the drone is delivering a parcel, we term the problem the Last-Stretch Delivery Problem, or LSDP for short. In the last problem model, which has a similar problem setting with the LSDP, the truck is not allowed to wait for the drone to return at the previous rendezvous point. We term this problem the No-Wait Last-Stretch Delivery Problem, or NW-LSDP.

With this work, we propose graph problem models for the ALSDP, the NW-ALSDP, the LSDP, and for the NW-LSDP. Next, we show that the recognition version of the graph problems are NP-complete, and that they remain NP-hard even in metric graphs where all edge weights are restricted to be 1 or 2. Furthermore, we identify a special instance type of the NW-ALSDP and the NW-LSDP that can be solved optimally in polynomial time. Finally, we propose a polynomial-time approximation algorithm, and show that the algorithm has a factor 2 approximation guarantee in metric graphs with an additional restriction of the cost function, to be defined later.

The rest of this paper is organized as follows. Section 2 outlines the basic notation and the problem models for this paper. Section 3 establishes the NP-completeness of our problem models, while Section 4 outlines the analysis of the polynomially solvable case. Section 5 describes the approximation algorithm for our problem models, and finally, Section 6 concludes the paper.

2. Preliminaries

2.1 Notation

The set of reals (resp., nonnegative reals) is denoted by $\mathbb{R}$ (resp., $\mathbb{R}_+$).

The vertex set and the edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. For vertices $u, v \in V(G)$, we use uv to refer to an edge $e \in E(G)$ such that e is incident to u and v. We call u and v the *end vertices* of the edge uv. For a set $E' \subseteq E(G)$ of edges we write $V(E')$ for the set of all end vertices of edges in E'. A graph G is *complete* if every two vertices $u, v \in V(G)$ are adjacent. The *degree* of a vertex $u \in V(G)$ in a graph G is the number of edges $E(G)$ incident to u.

A *subgraph* G' of a graph G is a graph such that $V(G') \subseteq V(G)$, and $E(G') \subseteq E(G)$, and we write $G' \subseteq G$. A graph $G' \subseteq G$ is an *induced* subgraph of G if it holds that $E(G') = \binom{V(G')}{2} \cap E(G)$, and we also say that G' is induced by $V(G')$. Given a graph G and a set $V' \subseteq V(G)$, we write $G[V']$ for the subgraph of G induced by V'. In addition, for a subset $V'' \subseteq V(G)$, we write $G - V''$ for the graph $G[V(G) - V'']$.

An *independent set* in a graph G is a set of pairwise non-adjacent vertices. A graph G is *bipartite* if the set of vertices $V(G)$ can be partitioned into two non-empty subsets $V_1, V_2 \subseteq V(G)$ such that V_1 and V_2 are independent sets. A bipartite graph G whose vertex set $V(G)$ can be partitioned into a disjoint union $V_1 \cup V_2$ is called *complete* if every two vertices $u \in V_1$ and $v \in V_2$ are adjacent.

Given a graph G, a *matching* $M \subseteq E(G)$ is a subset of edges such that each vertex in $V(G)$ is incident to at most one edge in M. A matching $M \subseteq E(G)$ is *perfect* if it holds that $V(M) = V(G)$.

A *path* is a graph $P = (\{v_1, v_2, \dots, v_p\}, \{e_1, e_2, \dots, e_{p-1}\})$ such that for $i = 1, 2, \dots, p-1$, it holds that $e_i = v_i v_{i+1}$. Such a graph P is also called a v_1, v_p-path. A path $P = (\{v_1, v_2, \dots, v_p\}, \{e_1, e_2, \dots, e_{p-1}\})$ such that $v_i \neq v_j$ for $1 \leq i \neq j \leq p$ is called a *simple* path. Given two sets $A, B \subseteq V(G)$, we call a path $P \subseteq G$ an A, B-*alternating path* if for every edge $e = uv \in E(P)$ it holds that $u \in A$, $v \in B$. Given a totally ordered set $(R, \prec)$, where for vertices $u, v \in R$, $u \preccurlyeq v$ denotes that $u \prec v$ or $u \equiv v$, we say that a v_1, v_p-path $P = (\{v_1, v_2, \ldots, v_p\}, \{e_1, e_2, \ldots, e_{p-1}\})$ *obeys the total order* $\prec$ if for any two vertices $v_i, v_j \in V(P) \cap R$, it holds that if $i < j$ then $v_i \preccurlyeq v_j$.

Given a graph G and an *edge weight* function $w : E(G) \to \mathbb{R}_+$, we say that the graph G is *weighted* by w, and write (G, w). For convenience, for any $v \in V(G)$, we define that $w(vv) = 0$. For a subset $E' \subseteq E(G)$, let $w(E')$ denote $\sum_{e \in E'} w(e)$. For brevity, for a subgraph G' of G, let $w(G')$ denote $w(E(G'))$.

A weighted graph (G, w) is called *metric* if the edge weight function w satisfies the *triangle inequality*, that is, for all $u, v, q \in V(G)$ it holds that

$$w(uv) \leq w(uq) + w(qv). \tag{1}$$

Given a total order $\prec$ over the vertex set $V(G)$, we call the weighted graph (G, w) a *line with respect to* $\prec$, if for any three vertices $u, v, q \in V(G)$ such that $u \prec v \prec q$ it holds that

$$w(uq) = w(uv) + w(vq). \tag{2}$$

2.2 Problem Models

Let C be the set of customers to which parcels need to be delivered, and let R be the set of points at which the delivery drone can rendezvous with the truck along the truck's predetermined route. Let $|R| = m$ and $|C| = n$. We can assume without loss of generality that the truck passes the points $r_1, r_2, \ldots, r_m$ of the set R in ascending order of their indices, and define a total order $\prec$ over the set R to be $r_i \prec r_j$ if and only if $i < j$. The drone has unit payload capacity, and it never visits two customers in the set C consecutively. On the other hand, between two points in R, the drone can "hitch a ride" on the truck. With this observation, we introduce the following distance functions:

- $d(u, v)$: the time it takes for the drone to travel between rendezvous point $u \in R$ and customer $v \in C$.
- $t(u, v)$: the time it takes for the truck to move from point $u \in R$ to point $v \in R$ along its predetermined route. By the assumption that points in R appear along the truck's route, we have that for any three points $u, v, q \in R$, $u \prec v \prec q$, it holds that

$$t(u, q) = t(u, v) + t(v, q). \tag{3}$$

We say that a rendezvous point and a customer point are *mutually reachable* if the drone is able to travel unobstructed between them. Further, the drone has unit payload capacity, and it never delivers two parcels consecutively without rendezvousing with the truck. An illustration of the problem scenario is depicted in **Fig. 1**.

Recall that we named the scenario where the drone is required to take off immediately carrying a parcel and only rendezvouses with the truck to pick up the next parcel to be delivered, the Alternating Last-Stretch Delivery Problem, or ALSDP for short. The drone can rendezvous with the truck at the previous point where it had picked up a parcel, or at some future rendezvous point. Let $E \subseteq \binom{R \cup C}{2} - \binom{C}{2} - \binom{R}{2}$ be the set of mutually reachable pairs. We model the ALSDP by a bipartite graph $G = (R \cup C, E)$, where it holds that $R \cap C = \emptyset$. This graph is weighted by an edge weight function $w : E \to \mathbb{R}_+$ defined to be

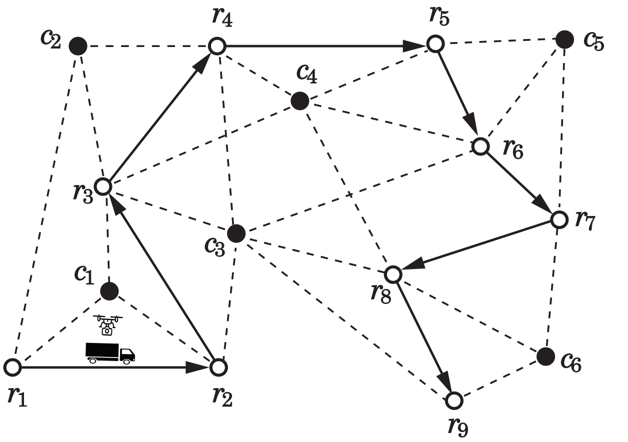


Fig. 1 An illustration of the problem scenario. The truck's predetermined route is shown by solid arrows. The points at which the delivery drone can rendezvous with the truck along the truck's predetermined route are illustrated as white circles. The customers to which parcels need to be delivered are represented by black circles. Mutually reachable pairs of a rendezvous point and a customer point are connected by dashed lines.

$$w(uv) \triangleq d(u, v), \text{ for } u \in R, v \in C, uv \in E. \tag{4}$$

Thus, we get the following problem.

The Alternating Last-Stretch Delivery Problem - ALSDP

Instance: A bipartite graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, a weight function $w : E \to \mathbb{R}_+$, and a total order $\prec$ on R. Let $|R| = m$ and let r_1 and r_m be the unique minimum and maximum elements of R with respect to $\prec$, respectively.

Feasible Solution: An R, C-alternating (not necessarily simple) r_1, r_m-path $P \subseteq G$ such that $C \subseteq V(P)$, each $c \in C$ is visited at most once, and P obeys the total order $\prec$ over R.

Objective: Minimize $w(P)$.

An illustration of the ALSDP is shown in **Fig. 2** (a), while an illustration of a feasible solution to this problem model is shown in Fig. 2 (b).

In the second problem model we examine the case when the truck is not allowed to wait for the drone at the previous rendezvous point. Instead, the truck will proceed along its route while the drone makes a delivery, and must intercept the drone at a future rendezvous point. We call this problem, the No-Wait Alternating Last-Stretch Delivery Problem, or NW-ALSDP for short. Just like the ALSDP, we model the NW-ALSDP by a bipartite graph $G = (R \cup C, E)$, where it holds that $R \cap C = \emptyset$, and $E \subseteq \binom{R \cup C}{2} - \binom{C}{2} - \binom{R}{2}$ is the set of mutually reachable pairs. This graph is weighted by an edge weight function $w : E \to \mathbb{R}_+$ defined as in Eq. (4).

The No-Wait Alternating Last-Stretch Delivery Problem - NW-ALSDP

Instance: A bipartite graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, a weight function $w : E \to \mathbb{R}_+$, and a total order $\prec$ on R. Let $|R| = m$, and let r_1 and r_m be the unique minimum and maximum elements of R with respect to $\prec$, respectively.

Feasible Solution: A simple R, C-alternating r_1, r_m-path $P \subseteq G$

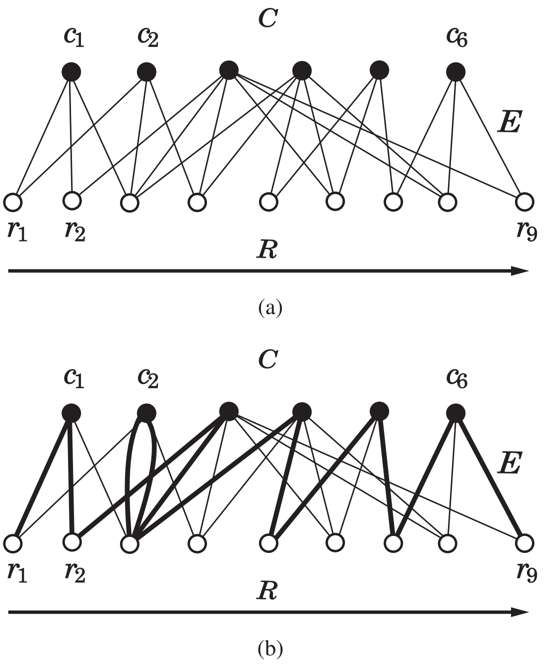


Fig. 2 (a) An instance of the ALSDP corresponding to the scenario of Fig. 1. The total order $\prec$ over the set R is expressed as an arrow under it. (b) An R, C-alternating r_1, r_m-path that obeys the total order $\prec$, visits all vertices in C exactly once, but some vertices in R are visited more than once, as a feasible path for the given instance in (a).

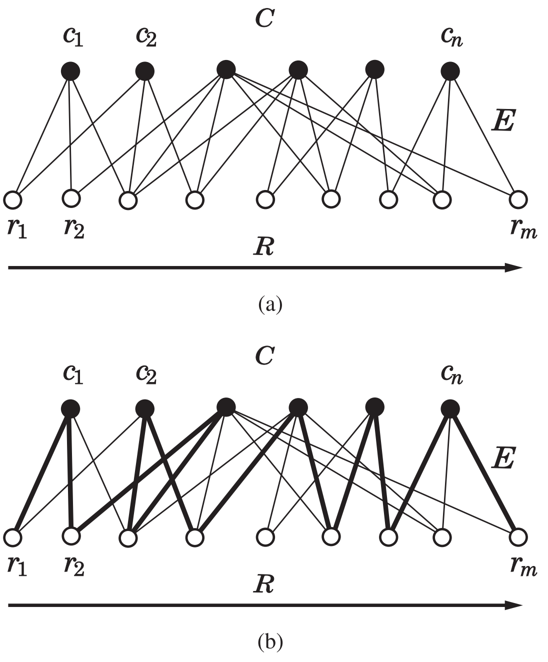


Fig. 3 (a) An instance of the NW-ALSDP. The total order $\prec$ over the set R is expressed as an arrow under it. (b) A simple R, C-alternating r_1, r_m-path that visits all vertices in C exactly once and obeys the total order $\prec$, as a feasible path for the given instance in (a).

such that $C \subseteq V(P)$, and P obeys the total order $\prec$ over R.

Objective: Minimize $w(P)$.

An illustration of the NW-ALSDP is shown in **Fig. 3** (a), while an illustration of a feasible solution to this problem model is shown in Fig. 3 (b).

We call the problem arising under the assumption that the drone may "hitch a ride" on the truck between consecutive deliveries the Last-Stretch Delivery Problem, or LSDP for short.

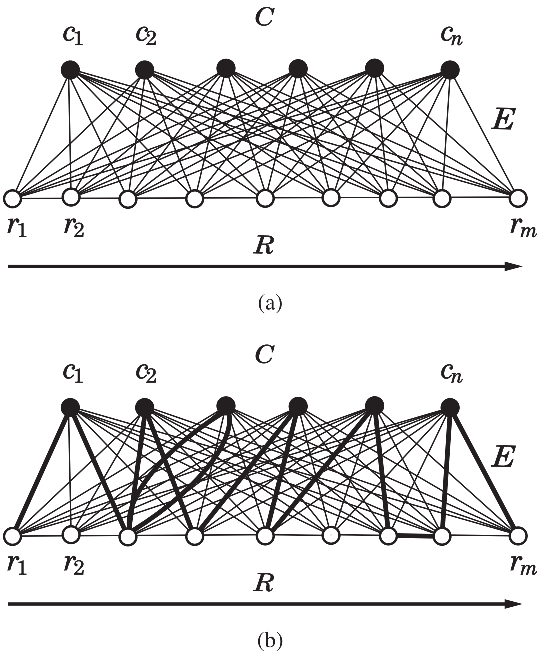


Fig. 4 (a) An instance of the LSDP. The total order $\prec$ over the set R is expressed as an arrow under it. (b) An r_1, r_m-path that obeys the total order $\prec$, visits all vertices in C exactly once, but some vertices in R are visited consecutively or multiple times, as a feasible solution to the instance in (a).

In this problem model, the truck is allowed to wait for the drone to return at the previous rendezvous point. We express the LSDP as follows. We consider a graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, it holds that $E \subseteq \binom{R \cup C}{2} - \binom{C}{2}$, and the graph $G[R]$ induced by the vertex set R is a simple path which obeys the total order $\prec$ over the set R. We define a weight function $w : E \to \mathbb{R}_+$ in this graph to be

$$w(uv) \triangleq \begin{cases} d(u,v), & \text{for } u \in R, v \in C, \\ t(u,v), & \text{for } u, v \in R. \end{cases} \tag{5}$$

The Last-Stretch Delivery Problem - LSDP

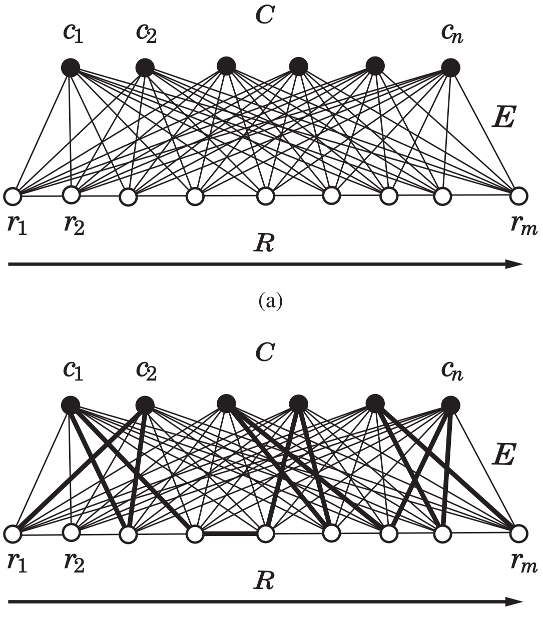
Instance: A graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$ and $E \subseteq \binom{R \cup C}{2} - \binom{C}{2}$, a weight function $w : E \to \mathbb{R}_+$, a total order $\prec$ on R, where $|R| = m$, r_1 and r_m are respectively the unique minimum and maximum elements of R with respect to $\prec$, and the graph $G[R]$ induced by the vertex set R is a simple r_1, r_m-path which obeys the total order $\prec$ over the set R.

Feasible Solution: A not necessarily simple r_1, r_m-path $P \subseteq G$ such that $C \subseteq V(P)$, each $c \in C$ is visited at most once, and P obeys the total order $\prec$ over R.

Objective: Minimize $w(P)$.

An illustration of the LSDP is shown in **Fig. 4** (a), followed by an illustration of a feasible solution to this problem in Fig. 4 (b).

In the final problem model, which has a similar problem setting with the LSDP, the truck is not allowed to wait for the drone at the previous rendezvous point. The truck will proceed along its route and only intercept the drone at a future rendezvous point in the set R. We call this problem the No-Wait Last-Stretch Delivery Problem, or NW-LSDP for short. Similar to the LSDP, in the NW-LSDP we consider a graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, it holds that $E \subseteq \binom{R \cup C}{2} - \binom{C}{2}$, and the graph $G[R]$ induced by the vertex set R is a simple path which obeys the total



(a)

(b)

Fig. 5 (a) An instance of the NW-LSDP. The total order $\prec$ over the set R is expressed as an arrow under it. (b) A simple r_1, r_m-path that obeys the total order $\prec$, visits all vertices in C exactly once, but each vertex in R is visited at most once, and some vertices in R are visited consecutively, as a feasible solution to the instance in (a).

order $\prec$ over the set R. A weight function $w : E \to \mathbb{R}_+$ in this graph is defined as in Eq. (5).

The No-Wait Last-Stretch Delivery Problem - NW-LSDP

Instance: A graph $G = (R \cup C, E)$ where it holds that $|R| > |C|$, $R \cap C = \emptyset$, $E \subseteq \binom{R\cup C}{2} - \binom{C}{2}$, a weight function $w : E \to \mathbb{R}_+$, a total order $\prec$ on R, where $|R| = m$, r_1 and r_m are respectively the unique minimum and maximum elements of R with respect to the total order $\prec$, and the graph $G[R]$ induced by the vertex set R is a simple r_1, r_m-path which obeys the total order $\prec$ over the set R.

Feasible Solution: A simple r_1, r_m-path $P \subseteq G$ such that $C \subseteq V(P)$, and P obeys the total order $\prec$ over R.

Objective: Minimize $w(P)$.

An illustration of the NW-LSDP is shown in **Fig. 5** (a), while an illustration of a feasible solution to this problem model is shown in Fig. 5 (b).

3. NP-hardness

We will first prove the NP-completeness of the recognition versions of the ALSDP and the NW-ALSDP by a reduction from the s,t-Hamiltonian Path Problem (see e.g., Garey and Johnson [4]), which is given a graph H and vertices $s, t \in V(H)$, and asks if there exists a simple s,t-path $P_H \subseteq H$ such that $V(H) \subseteq V(P_H)$.

Afterwards, we will show that the LSDP and the NW-LSDP are NP-hard, even when the given edge weight function w only takes values 1 and 2. In the recognition version, as opposed to asking for an optimal path to a given instance of the ALSDP or the NW-ALSDP, we simply ask if a feasible path exists or not. In this sense, we omit the edge weight function from the input instance, and the recognition versions of the ALSDP and the NW-ALSDP are defined as follows.

Recognition version of the ALSDP

Instance: A bipartite graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, and a total order $\prec$ on R. Let $|R| = m$, and let r_1 and r_m be the unique minimum and maximum elements of R with respect to $\prec$, respectively.

Question: Does there exist an R, C-alternating (not necessarily simple) r_1, r_m-path $P \subseteq G$ such that each vertex $c \in C$ is visited exactly once and P obeys the total order $\prec$ over R?

Recognition version of the NW-ALSDP

Instance: A bipartite graph $G = (R \cup C, E)$ such that $R \cap C = \emptyset$, and a total order $\prec$ on R. Let $|R| = m$, and let r_1 and r_m be the unique minimum and maximum elements of R with respect to $\prec$, respectively.

Question: Does there exist a simple R, C-alternating r_1, r_m-path $P \subseteq G$ such that $C \subseteq V(P)$ and P obeys the total order $\prec$ over R?

Theorem 1 The recognition versions of the ALSDP and the NW-ALSDP are NP-complete.

Proof. It is obvious that the recognition version of both the ALSDP and the NW-ALSDP are in NP, as when given a path P as a witness, it is straightforward to verify in polynomial time whether the path P is feasible for the ALSDP or the NW-ALSDP. Therefore, we proceed to show that they are also NP-hard, by a reduction from the s,t-Hamiltonian Path Problem. We will first focus on the recognition version of the ALSDP, and later give a note that the same reduction is valid for the recognition version of the NW-ALSDP as well.

Let $(H = (V, A), s, t \in V)$ be an instance of the s,t-Hamiltonian Path Problem, i.e., a simple undirected graph with a vertex set V, terminals $s, t \in V$, and an edge set A. Let $|V| = n$, and $|A| = p$, as illustrated in **Fig. 6** (a). We construct a recognition instance $I = (G = (R \cup C, E), \prec)$ of the ALSDP as follows. First, set $C = V$. Fix an arbitrary total order $a_1, a_2, \dots, a_p$ of the edges in A, and for each edge $a_i = uv \in A$ construct $n-1$ vertices $r_i^1, r_i^2, \dots, r_i^{n-1}$. Set R to be

$$R \triangleq \{r_i^j \mid i = 1, 2, \dots, p,\ j = 1, 2, \dots, n-1\} \cup \{r_0\} \cup \{r_{p+1}\},$$

and E to be

$$\begin{aligned} E \triangleq\ & \{ur_i^j \mid u \in V,\ a_i = uv \in A,\ j = 1, 2, \dots, n-1\} \\ & \cup \{r_0 s, t r_{p+1}\}. \end{aligned}$$

Finally, a total order $\prec$ over R can be introduced without any loss of generality as follows.

(i) $r_0 \prec r$, for all $r \in R - \{r_0\}$,

(ii) $r \prec r_{p+1}$, for all $r \in R - \{r_{p+1}\}$,

(iii) $r_i^j \prec r_k^l$, for $i, k = 1, 2, \dots, p$, and $j, l = 1, 2, \dots, n-1$, if and only if

  (a) $j < l$, or

  (b) $j = l$ and $i < k$.

The above operations can be seen as adding a degree-2 vertex on each edge $a \in A$ as shown in Fig. 6 (b), and then, taking $n-1$ copies of each of these new degree-2 vertices together with the edges incident to them. The newly added degree-2 vertices form the set R. Finally, we add to the set R a vertex r_0 adjacent to the vertex s in the set C and a vertex r_{p+1} adjacent to the vertex t in

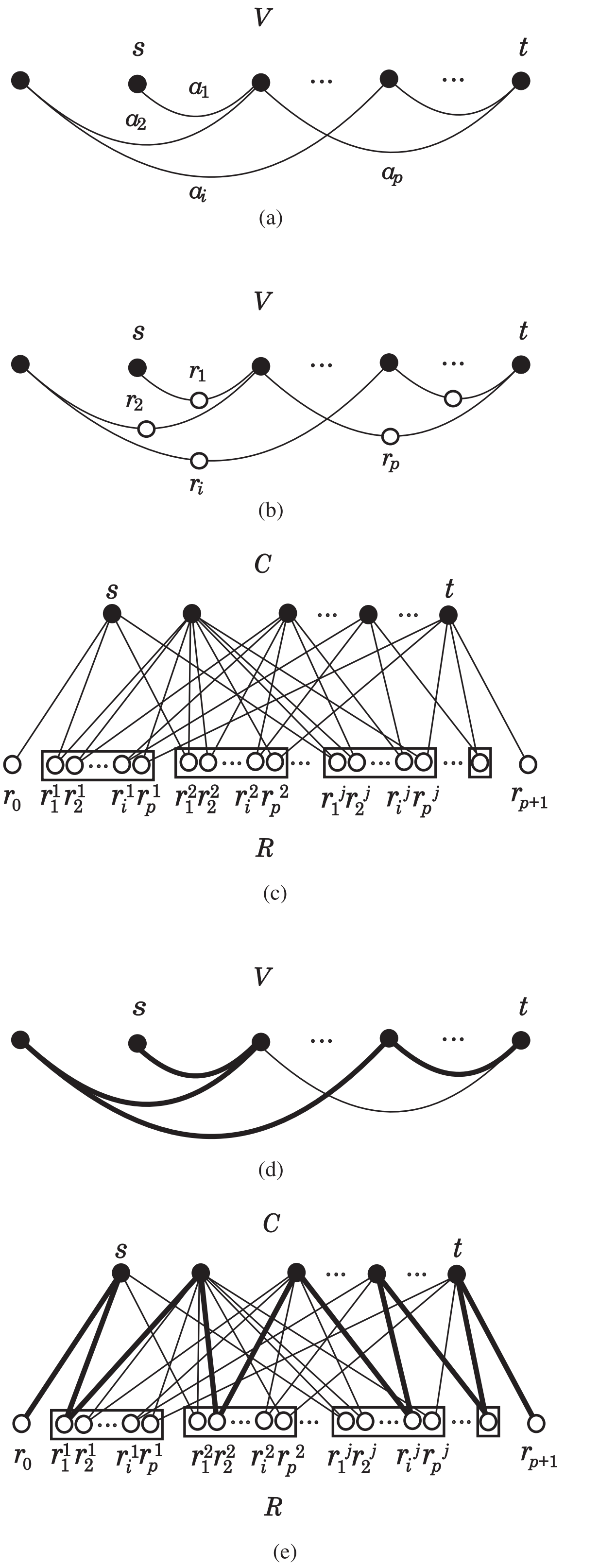


Fig. 6 A reduction from the s,t-Hamiltonian Path Problem to a recognition instance $I = (G = (R \cup C, E), \prec)$ of the ALSDP. (a) An arbitrary instance of the s,t-Hamiltonian Path Problem, $(H = (V, A), s, t \in V)$. (b) Add a degree-2 vertex r_i for each edge $a \in A$. (c) Add a start vertex r_0 adjacent to the vertex s in the set C, and a terminal vertex r_{p+1} adjacent to the vertex t in the set C, and arrange the vertices in R in a total order. (d) An s,t-Hamiltonian path in H. (e) A feasible path in the transformed instance.

the set C, and arrange the vertices in R in the defined total order, as shown in Fig. 6 (c).

We need to show that the constructed recognition instance $I = (G = (R \cup C, E), \prec)$ of the ALSDP has a feasible r_0, r_{p+1}-path P_{ALSDP} if and only if the graph H admits an s,t-Hamiltonian path, such as in Fig. 6 (d).

First, we demonstrate the "if" direction by showing how an s,t-Hamiltonian path P_H in H can be used to construct a feasible r_0, r_{p+1}-path P_{ALSDP} in G. We start by numbering the vertices in the vertex set V in the order in which they appear in P_H, such that $P_H = v_1, v_2, \ldots, v_n$, where $s = v_1$ and $t = v_n$ holds. Furthermore, assume without loss of generality that the edges of P_H are $a_1 = v_1v_2, a_2 = v_2v_3, \ldots, a_{n-1} = v_{n-1}v_n$. By the construction of the graph G, for all a_i, $i = 1, 2, \ldots, n-1$, and $j = 1, 2, \ldots, n-1$, all edges $v_i r_i^j$ and $v_{i+1} r_i^j$, are present in E. Then, we can construct a feasible r_0, r_{p+1}-path P_{ALSDP} in G such that

$$\begin{aligned} V(P_{\mathrm{ALSDP}}) &= \{r_0, r_{p+1}\} \cup \{v_i \mid i = 1, 2, \ldots, n\} \\ &\cup \{r_i^i \mid i = 1, 2, \ldots, n-1\} \end{aligned}$$

and

$$\begin{aligned} E(P_{\mathrm{ALSDP}}) &= \{r_0 s, t r_{p+1}\} \cup \{v_i r_i^i \mid i = 1, 2, \ldots, n-1\} \\ &\cup \{r_i^i v_{i+1} \mid i = 1, 2, \ldots, n-1\}, \end{aligned}$$

as shown in Fig. 6 (e).

Next, to show the opposite direction of the claim, we show that a feasible r_0, r_{p+1}-path P_{ALSDP} in the graph G uniquely determines an s,t-Hamiltonian path P_H in H. This follows from the observation that each vertex $r_i^j \in R$ has degree 2, and its neighbors, $u, v \in C$, are exactly the end vertices of the edge $a_i = uv \in A$. Therefore, since each $v \in C$ appears exactly once in P_{ALSDP}, each edge $a_i \in A$ can be chosen at most once, giving an s,t-Hamiltonian path in H.

Notice that each vertex $r_i^j \in R$ has degree 2, and therefore there does not exist a non-simple r_0, r_{p+1}-path that is feasible to the ALSDP. Therefore, the recognition version of the NW-ALSDP is also NP-hard. □

Following, we will argue the NP-hardness of the LSDP and NW-LSDP.

Theorem 2 Both the LSDP and the NW-LSDP are NP-hard even in complete bipartite graphs where all edge weights are restricted to be 1 or 2.

Proof. We will focus on showing the NP-hardness of the LSDP, and later note that this proof implies NP-hardness of the NW-LSDP as well. By the result of Theorem 1 that the recognition version of the ALSDP is itself NP-complete, we show a reduction from the recognition version of the ALSDP to an instance of the LSDP with edge weights restricted to be 1 or 2.

Let $I_{\mathrm{ALSDP}} = (G_{\mathrm{ALSDP}} = (R \cup C, E_{\mathrm{ALSDP}}), \prec)$ be an arbitrary instance of the recognition version of the ALSDP. Let $|R| = m$, $|C| = n$, and r_1 and r_m respectively denote the unique minimum and maximum elements in R with respect to the total order $\prec$. We construct an instance $I_{\mathrm{LSDP}} = (G_{\mathrm{LSDP}} = (R \cup C, E_{\mathrm{LSDP}}), w, \prec)$ as follows. First, let

$$E_{\mathrm{LSDP}} = \{uv \mid u \in R, v \in C\} \cup \{r_i r_{i+1} \mid i = 1, 2, \ldots, n-1\},$$

and define the edge weight function $w : E_{\mathrm{LSDP}} \to \{1, 2\}$ to be

$$w(e) \triangleq \begin{cases} 1 & \text{if } e \in E_{\mathrm{ALSDP}}, \\ 2 & \text{otherwise.} \end{cases}$$

Notice that the number of edges in P_{LSDP} is $2n$. Then, the instance I_{ALSDP} has a feasible path, i.e., is a "yes" instance, if and only if the instance I_{LSDP} has a feasible path P_{LSDP} of cost $w(P_{\mathrm{LSDP}}) = 2n$, that is, for each $e \in E(P_{\mathrm{LSDP}})$ it holds that $w(e) = 1$.

Notice that a non-simple r_0, r_{p+1}-path in the transformed instance as above cannot have a length of $2n$ or less, and the requirement for a simple path of the NW-LSDP can be taken without loss of generality. Therefore, the NW-LSDP is also NP-hard. □

4. Polynomially Solvable Case of the NW-ALSDP and the NW-LSDP

In this section we describe how an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| = |C| + 1$ can be solved in polynomial time by reducing it to a minimum cost bipartite matching instance. Note that in this case, the NW-LSDP is equivalent to the NW-ALSDP, in the sense that in both problems a feasible path is R, C-alternating. Then, in the remainder of this section, we will talk only about the NW-ALSDP, but the same conclusions still hold for the NW-LSDP as well.

Observation 1 In any feasible path P, if a vertex $v \in C$ is visited immediately after $r_i \in R$, then $r_{i+1} \in R$ is visited immediately after v.

Observation 1 gives us the following procedure for constructing a solution to an instance $I = (G = (R \cup C, E), w, \prec)$ with $|R| = |C| + 1$ and $R \cap C = \emptyset$ by a transformation to minimum cost bipartite matching, given in Procedure Alternating Path I.

```
Procedure Alternating Path I
Input: An NW-ALSDP instance $I = (G = (R \cup C, E), w, \prec)$ where
  $|R| = m$, $|C| = n$, $m = n + 1$, $R \cap C = \emptyset$, and $G$ is a complete bipartite graph.
  Vertices in $R$ are numbered as $r_1, r_2, \ldots, r_m$ with respect to the total order $\prec$.
Output: A simple $R, C$-alternating $r_1, r_m$-path $P \subseteq G$ that obeys
  the total order $\prec$ over $R$, and $C \subseteq V(P)$.
Let $\widetilde{G} := G - \{r_m\}$;
Let an edge weight function $\widetilde{w} : E(\widetilde{G}) \to \mathbb{R}_+$ be
    $\widetilde{w}(r_i v) := w(r_i v) + w(v r_{i+1}), \quad \forall r_i \in R - \{r_m\},\ v \in C;$
Compute a minimum cost perfect matching $M$ in $(\widetilde{G}, \widetilde{w})$;
Let $\mu : R - \{r_m\} \to C$ be the bijection defined by the matching $M$ such that
  for any $r_i \in R - \{r_m\}$ and $v \in C$, it holds that $v = \mu(r_i)$ if and only if $r_i v \in M$;
Let $P$ be a simple path such that
    $P = (V(G), \{r_i \mu(r_i), \mu(r_i) r_{i+1} \mid i = 1, 2, \ldots, n\});$
return $P$.
```

Lemma 1 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| = |C| + 1$, Procedure Alternating Path I can be implemented to run in $O(|C|^3)$ time.

Proof. The procedure to find a minimum cost bipartite matching M, which takes $O(|C|^3)$ time [8], dominates the runtime, from which the claim follows. □

Observation 2 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| = |C|+1$, let $(\widetilde{G}, \widetilde{w})$ be a transformed bipartite graph as in Procedure Alternating Path I, M be a minimum cost perfect matching in $(\widetilde{G}, \widetilde{w})$, and P be an output of Procedure Alternating Path I given the instance I as input. Assume without loss of generality that vertices $c_1, c_2, \ldots, c_n \in C$ appear in P in that order. Also, assume that $M = \{r_i c_i \mid i = 1, 2, \ldots, n\}$. Then, for $w(P)$ it holds that

$$\begin{aligned} w(P) &= \sum_{1 \le i \le n} \Big(w(r_i c_i) + w(c_i r_{i+1}) \Big) \\ &= \sum_{1 \le i \le n} \widetilde{w}(r_i c_i) = \widetilde{w}(M). \end{aligned} \tag{6}$$

Theorem 3 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| = |C| + 1$, let P^* be an optimal path for I. Let $(\widetilde{G}, \widetilde{w})$ be a transformed bipartite graph as in Procedure Alternating Path I, and let M be a minimum cost perfect matching in $(\widetilde{G}, \widetilde{w})$. Then, it holds that

$$w(P^*) = \widetilde{w}(M). \tag{7}$$

Proof. By Procedure Alternating Path I and Observation 2, we know how to construct a feasible path P such that $w(P) = \widetilde{w}(M)$, and therefore it holds that

$$w(P^*) \le \widetilde{w}(M).$$

Next, we show the claim that

$$\widetilde{w}(M) \le w(P^*),$$

which will prove the claim of the theorem. To derive a contradiction, assume that $\widetilde{w}(M) > w(P^*)$. Let $|C| = n$, and $r_1, r_2, \ldots, r_{n+1}$ be numbered according to the total order $\prec$. Without loss of generality assume that $P^* = (V(G), \{r_i c_i \mid i = 1, 2, \ldots, n\} \cup \{c_i r_{i+1} \mid i = 1, 2, \ldots, n\})$. Then, from P^* we can obtain a perfect bipartite matching M' in $\widetilde{G}$, as $M' = \{r_i c_i \mid i = 1, 2, \ldots, n\}$. By the construction of $(\widetilde{G}, \widetilde{w})$, we have that

$$\widetilde{w}(M') = \sum_{1 \le i \le n} \widetilde{w}(r_i c_i) = \sum_{1 \le i \le n} \Big(w(r_i c_i) + w(c_i r_{i+1}) \Big) = w(P^*).$$

Then, $w(P^*) < \widetilde{w}(M)$ would imply that $\widetilde{w}(M') < \widetilde{w}(M)$, which contradicts the initial assumption that M is a minimum cost perfect matching in $(\widetilde{G}, \widetilde{w})$, and this completes the proof. □

Corollary 1 Both the NW-ALSDP and the NW-LSDP with $|R| = |C| + 1$ can be solved in polynomial time.

5. Approximation Algorithms

As a result of Theorem 1, it is NP-complete to decide whether there is a feasible solution or not for the ALSDP and the NW-ALSDP. In this section, we limit our exposition to special types of instances $I = (G = (R \cup C, E), w, \prec)$ such that the graph G includes the complete bipartite graph with bipartition $R \cup C$ as a subgraph. To claim constant factor approximation ratio for the ALSDP and the LSDP, we further assume that the weighted graph (G, w) is metric and moreover, that the graph $G[R]$ induced by the set R is a line with respect to the total order $\prec$.

5.1 Approximation algorithm for the ALSDP and the LSDP

First, we propose a heuristic for the ALSDP, summarized as Procedure Alternating Path II. Next, we will show that the proposed procedure is a 2-approximation algorithm in problem settings with a metric edge weight. Then, we show that the procedure produces a path that is feasible to both the ALSDP and the LSDP, given a graph $G = (R \cup C, E)$ such that $|R| \geq |C| + 1$, $\binom{R\cup C}{2} - \binom{C}{2} - \binom{R}{2} \subseteq E$, and a total order $\prec$ on R.

```
PROCEDURE ALTERNATING PATH II
Input : A weighted complete bipartite graph $(G, w)$ with bipartition $R \cup C$ and a total order $\prec$ over the vertex set $R$. Assume that $|R| = m$, $|C| = n$, and $r_1, r_2, \ldots, r_m \in R$ are numbered with respect to the total order $\prec$.
Output: An $R, C$-alternating $r_1, r_m$-path $P$ that obeys the total order $\prec$ and $C \subseteq V(P)$.
Let $F \subseteq E$ be a minimum cost subset of edges such that $C \subseteq V(F)$;    /* as shown in Fig. 7 (a) */
Let $R' := R \cap V(F)$;
Let $\varphi : C \to R'$ be a mapping such that for all $v \in C$, it holds that $u = \varphi(v)$ iff $uv \in F$;
Let $c_1, c_2, \ldots, c_n \in C$ be numbered such that if $i < j$, then it holds that $\varphi(c_i) \preccurlyeq \varphi(c_j)$;
Let $P \subseteq G$ be a path such that
    $V(P) = R' \cup C;$
    $E(P) = \{r_1c_1, c_nr_m\}$
        $\cup \{c_i\varphi(c_i) \mid i = 1, 2, \ldots, n-1\}$
        $\cup \{\varphi(c_i)c_{i+1} \mid i = 1, 2, \ldots, n-1\};$
    /* as shown in Fig. 7 (b) */
return $P$.
```

Lemma 2 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the ALSDP such that $|R| = m$ and $|C| = n$, Procedure Alternating Path II can be implemented to run in $O(mn)$ time.

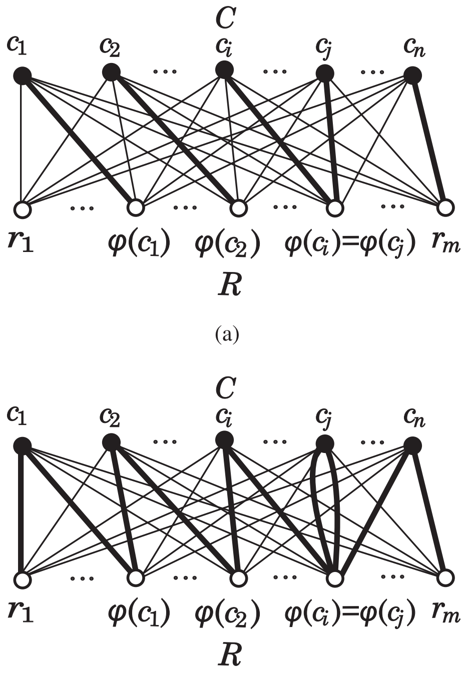
Proof. The procedure to compute a minimum weight edge set F can be done in time $O(mn)$. Then, a path P can be constructed in $O(n)$ time. The procedure for computing the edge set F dominates the runtime, from which the claim follows. □

Before proceeding with analyzing the approximation ratio of Procedure Alternating Path II, we give two technical lemmas which are useful in the analysis.

Lemma 3 Let (G, w) be a weighted graph, $R, C \subseteq V(G)$ be two vertex sets such that $R \cap C = \emptyset$, and $\prec$ be a total order on R. Let $P_R \subseteq G[R]$ be the unique simple r_1, r_m-path visiting all and only the vertices $r_1, r_2, \ldots, r_m$ in the set R obeying the total order $\prec$. Let P^* be an optimal path in G such that $C \subseteq V(P^*)$ and P^* obeys the total order $\prec$ over R. If w satisfies the triangle inequality, and in addition $G[R]$ is a line with respect to the total order $\prec$, then it holds that

$$w(P_R) \leq w(P^*). \tag{8}$$

Proof. Let us assume without loss of generality that vertices $c_1, c_2, \ldots, c_n \in C$ appear in P^* in that order, and that for all $k = 1, 2, \ldots, n$, the vertices r_k and r_{k+1} appear immediately before and following c_k in the path P^*, respectively. For $w(P^*)$ we get



(a)

(b)

Fig. 7 Procedure Alternating Path II. (a) A minimum weight edge set F such that $C \subseteq V(F)$. (b) An R, C-alternating r_1, r_m-path P obtained in Procedure Alternating Path II.

$$w(P^*) = \sum_{1 \leq k < n} (w(r_kc_k) + w(c_kr_{k+1})).$$

Notice that by the triangle inequality, for any two vertices, $r_i, r_j \in R$ such that $r_i \prec r_j$, and for all $c \in C$, it holds that

$$w(r_ir_j) \leq w(r_ic) + w(r_jc). \tag{9}$$

By repeatedly using the fact that $G[R]$ is a line with respect to the total order $\prec$, it holds that

$$w(r_ir_j) = \sum_{i \leq \ell < j} w(r_\ell r_{\ell+1}),$$

and therefore, for the cost

$$w(P_R) = \sum_{1 \leq i < m} w(r_ir_{i+1})$$

it always holds that

$$w(P_R) \leq w(P^*)$$

as required. □

Lemma 4 Let (G, w) be a weighted graph, $R, C \subseteq V(G)$ be two vertex sets such that $R \cap C = \emptyset$, and $\prec$ be a total order on R. Let $r_1, r_2, \ldots, r_m \in R$ be numbered according to the total order $\prec$. If w satisfies the triangle inequality, then for any three vertices $r_i, r_j \in R$, $c \in C$, with $r_i \preccurlyeq r_j$ (possibly $r_i = r_j$), it holds that

$$w(r_ic) \leq w(r_jc) + \sum_{i \leq \ell < j} w(r_\ell r_{\ell+1}), \tag{10}$$

and

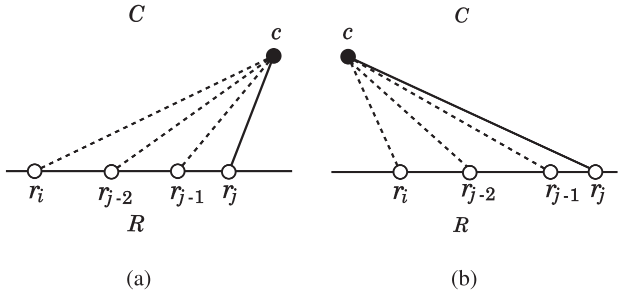


Fig. 8 Three vertices $r_i, r_j \in R$, and $c \in C$ as in Lemma 4.

$$w(r_jc) \le w(r_ic) + \sum_{i\le\ell<j} w(r_\ell r_{\ell+1}). \tag{11}$$

Proof. Let us observe three vertices, $r_i, r_j \in R$ and $c \in C$, such that it holds that $r_i \preccurlyeq r_j$ as shown in **Fig. 8** (a). Notice that by the triangle inequality, it holds that

$$w(r_{j-1}c) \le w(r_{j-1}r_j) + w(r_jc). \tag{12}$$

We can repeat the observation for r_{j-2} and the edge r_{j-1}, and in general for $r_{j-k}c$, until $j - k = i$. In total, we get

$$w(r_ic) \le w(r_jc) + \sum_{i\le\ell<j} w(r_\ell r_{\ell+1}). \tag{13}$$

Similarly, for three vertices $r_i, r_j \in R$ and $c \in C$, such that it holds that $r_i \preccurlyeq r_j$, as shown in Fig. 8 (b), we get

$$w(r_jc) \le w(r_ic) + \sum_{i\le\ell<j} w(r_\ell r_{\ell+1}), \tag{14}$$

as required. □

Theorem 4 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the ALSDP such that $R \cap C = \emptyset$, let P^* be an optimal path for I. If w satisfies the triangle inequality, and in addition $G[R]$ is a line with respect to the total order $\prec$, then for a path P computed by Procedure Alternating Path II given the instance I as an input, it holds that

$$w(P) \le 2w(P^*). \tag{15}$$

Proof. Notice that the degree of every vertex $c \in C$ in the path P^* is 2, and $E(P^*)$ contains two disjoint sets of edges F_1 and F_2 such that it holds that $C \subseteq V(F_1)$ and $C \subseteq V(F_2)$, and

$$w(P^*) = w(F_1) + w(F_2). \tag{16}$$

By this observation, for the edge set F calculated in Procedure Alternating Path II it holds that

$$w(F) \le \min\{w(F_1), w(F_2)\} \le \frac{1}{2}w(P^*). \tag{17}$$

Recall that in Line 4 of Procedure Alternating Path II, the vertices $c_1, c_2, \dots, c_n \in C$ are numbered such that if $i < j$, then it holds that $\varphi(c_i) \preccurlyeq \varphi(c_j)$. We can see that for the path P returned by Procedure Alternating Path II it holds that

$$\begin{aligned} w(P) \;=\; & w(r_1c_1) + \sum_{1\le i<n} w(c_i\varphi(c_i)) + \sum_{1\le i<n} w(\varphi(c_i)c_{i+1}) \\ & +w(c_nr_m). \end{aligned} \tag{18}$$

Let $P_R \subseteq G[R]$ be the unique simple r_1, r_m-path visiting all and only the vertices $r_1, r_2, \dots, r_m$ in the set R obeying the total order $\prec$ and $V(P_R) = R$, as in Lemma 3. By using Eqs. (10) and (11) from Lemma 4, for the values of $w(r_1c_1)$, $w(\varphi(c_i)c_{i+1})$, and $w(c_nr_m)$ in Eq. (18), for $w(P)$ we get

$$\begin{aligned} w(P) \;\le\; & w(\varphi(c_1)c_1) + \sum_{r_1\preccurlyeq r_\ell\prec\varphi(c_1)} w(r_\ell r_{\ell+1}) + \sum_{1\le i<n} w(c_i\varphi(c_i)) \\ & + \sum_{1\le i<n}\left(w(c_{i+1}\varphi(c_{i+1})) + \sum_{\varphi(c_i)\preccurlyeq r_\ell\prec\varphi(c_{i+1})} w(r_\ell r_{\ell+1})\right) \\ & +w(\varphi(c_n)c_n) + \sum_{\varphi(c_n)\preccurlyeq r_\ell\prec r_m} w(r_\ell r_{\ell+1}) \\ =\; & \sum_{r_1\preccurlyeq r_\ell\prec r_m} w(r_\ell r_{\ell+1}) + 2\sum_{1\le i\le n} w(c_i\varphi(c_i)) \\ =\; & w(P_R) + 2w(F). \end{aligned} \tag{19}$$

Finally, by Eqs. (17) and (19), and Lemma 3, it holds that

$$w(P) \le 2w(P^*),$$

as required. □

According to Theorem 4, given an instance I of the ALSDP that satisfies the necessary conditions, Procedure Alternating Path II will deliver a path with weight at most twice optimal, and is therefore a 2-approximation algorithm for the ALSDP. We proceed by investigating the performance of Procedure Alternating Path II as a solution method to the LSDP.

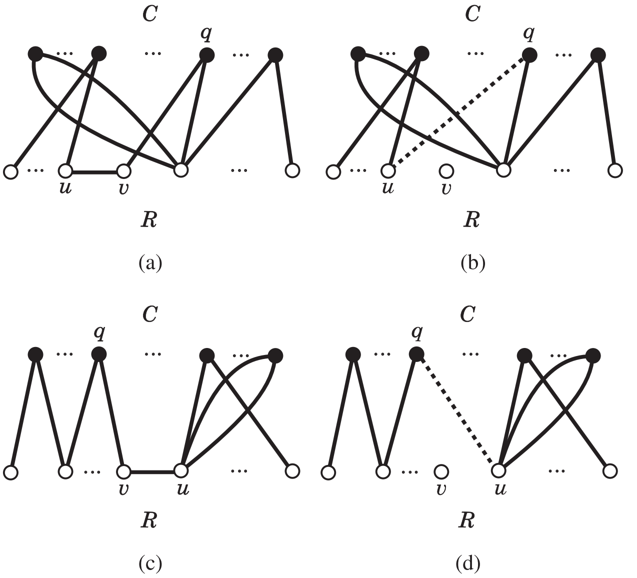


Fig. 9 (a) An optimal path P^* with three vertices $u, v \in R$, $q \in C$, appearing consecutively in the order u, v, q. (b) A path P' obtained from P^* by shortcutting the vertex v. (c) An optimal path P^* with three vertices $u, v \in R$, $q \in C$, appearing consecutively in the order q, v, u. (d) A path P' obtained from P^* by shortcutting the vertex v.

Lemma 5 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the LSDP, if the edge weight function w satisfies the triangle inequality, then there exists an R, C-alternating optimal path P in G.

Proof. Recall that $|R| = m$ and $|C| = n$, and the vertices in R are numbered $r_1, r_2, \dots, r_m$ according to the total order $\prec$. Let P^* be an optimal path for the instance I. If there exist three vertices u, v, q, such that $u, v \in R$, $q \in C$ and they appear consecutively in the path P^* in the order u, v, q or q, v, u, then we can obtain a path P' from P^* by replacing the edges uv and vq by a single edge uq, as shown in **Fig. 9**. For the cost $w(P')$ of the path P' we get

$$w(P') = w(P^*) - w(uv) - w(vq) + w(uq).$$

Then, from the triangle inequality we know that it holds that

$$w(uq) \leq w(uv) + w(vq),$$

and therefore

$$w(P') \leq w(P^*).$$

We can repeat the above shortcutting procedure until no more such triplets u, v, q of vertices exist. The resulting path from these shortcuts is R, C-alternating. □

From Lemma 5, we know that if the edge weight function w satisfies the triangle inequality, then there always exists an optimal path such that vertices in R and C appear alternately. As a consequence, the result of Theorem 4 leads to the following claim.

Corollary 2 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the LSDP such that $R \cap C = \emptyset$, let P^* be an optimal path for I. If w satisfies the triangle inequality, and in addition $G[R]$ is a line with respect to the total order $\prec$, then for a path P computed by Procedure Alternating Path II given the instance I as an input, it holds that

$$w(P) \leq 2w(P^*). \tag{20}$$

5.2 Approximation Algorithm for the No-wait Problem Models

First, we propose a heuristic for the NW-ALSDP based on minimum cost bipartite matching. Our proposed procedure is summarized as Procedure Alternating Path III. In fact, given a graph $G = (R \cup C, E)$ such that $|R| \geq |C| + 1$, $\binom{R\cup C}{2} - \binom{C}{2} - \binom{R}{2} \subseteq E$, and a total order $\prec$ on R, the procedure produces a path that is feasible to both the NW-ALSDP and the NW-LSDP. Afterwards, we will show that in problem settings with a metric edge weight, restricted such that $G[R]$ is a line with respect to the total order $\prec$, the proposed procedure is a 2-approximation algorithm.

```
Procedure Alternating Path III
Input: A weighted complete bipartite graph $(G, w)$ with bipartition $R \cup C$, where $|R| > |C|$ and $R \cap C = \emptyset$, and a total order $\prec$ over the vertex set $R$. Assume that $|R| = m$, $|C| = n$, and $r_1, r_2, \ldots, r_m \in R$ are numbered with respect to the total order $\prec$.
Output: A simple $R, C$-alternating $r_1, r_m$-path $P$ that obeys the total order $\prec$ and $C \subseteq V(P)$.
1:  Compute a minimum cost matching $M_1$ in $(G - \{r_1\}, w)$ such
    that $C \subseteq V(M_1)$ ;              /* as shown in Fig. 10 (a) */
2:  Compute a minimum cost matching $M_m$ in $(G - \{r_m\}, w)$ such
    that $C \subseteq V(M_m)$ ;              /* as shown in Fig. 10 (b) */
    /* Because it holds that $|R| > |C|$, such matchings exist */
3:  Let $M = \mathrm{argmin}\{w(M_1), w(M_m)\}$;
4:  Let $R' := R \cap V(M)$ be the set of vertices in $R$ that are
    matched by $M$;
5:  Let $\mu : C \to R'$ be a bijection such that $u = \mu(v)$ holds iff
    $uv \in M$;
6:  Let $c_1, c_2, \ldots, c_n \in C$ be numbered such that if $i < j$,
    then it holds that $\mu(c_i) \preccurlyeq \mu(c_j)$;
7:  if $w(M_1) \leq w(M_m)$ then
8:      Let $P \subseteq G$ be a path such that
            $V(P) = R' \cup C;$
            $E(P) = \{r_1 c_1, c_n r_m\}$
                    $\cup \{c_i \mu(c_i) \mid i = 1, 2, \ldots, n-1\}$
                    $\cup \{\mu(c_i) c_{i+1} \mid i = 1, 2, \ldots, n-1\}$
        /* as shown in Fig. 10 (c) */
9:  else /* $w(M_m) < w(M_1)$ */
10:     Let $P \subseteq G$ be a path such that
            $V(P) = R' \cup C;$
            $E(P) = \{r_1 c_1, c_n r_m\}$
                    $\cup \{c_i \mu(c_{i+1}) \mid i = 1, 2, \ldots, n-1\}$
                    $\cup \{c_i \mu(c_i) \mid i = 2, 3, \ldots, n\}$
        /* as shown in Fig. 10 (d) */
11: end if;
12: return $P$.
```

Lemma 6 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| = m$ and $|C| = n$, Procedure Alternating Path III can be implemented to run in $O(nm(n + \log m))$ time.

Proof. A minimum cost bipartite matching $M \subseteq \binom{R\cup C}{2} - \binom{C}{2} - \binom{R}{2}$ such that $C \subseteq V(M)$ can be computed via a max-flow algorithm (see, e.g., Korte and Vygen [8]) which requires $O(n)$ calls to a shortest path algorithm, which in turn can be implemented to run in time of $O(mn + m \log m)$. Thus, the minimum cost matchings M_1 and M_m in Line 1 and Line 2 can be computed in $O(nm(n + \log m))$ time. Then, a simple r_1, r_m-path P can be constructed in $O(n)$ time. The procedure to find the minimum cost bipartite matchings dominates the runtime, from which the claim follows. □

Theorem 5 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-ALSDP such that $|R| > |C|$, and $R \cap C = \emptyset$, let P^* be an optimal path for I. If w satisfies the triangle inequality, and in addition $G[R]$ is a line with respect to the total order $\prec$, then for a simple path P computed by Procedure Alternating Path III given the instance I as an input, it holds that

$$w(P) \leq 2w(P^*). \tag{21}$$

Proof. Let us assume without loss of generality that

$$\begin{aligned} P^* = (&\{r_1, c_1, r_2, c_2, \ldots, r_i, c_i, r_{i+1}, \ldots, c_n, r_m\}, \\ &\{r_1c_1, c_1r_2, \ldots, c_nr_m\}). \end{aligned}$$

Notice that $E(P^*)$ contains two disjoint matchings

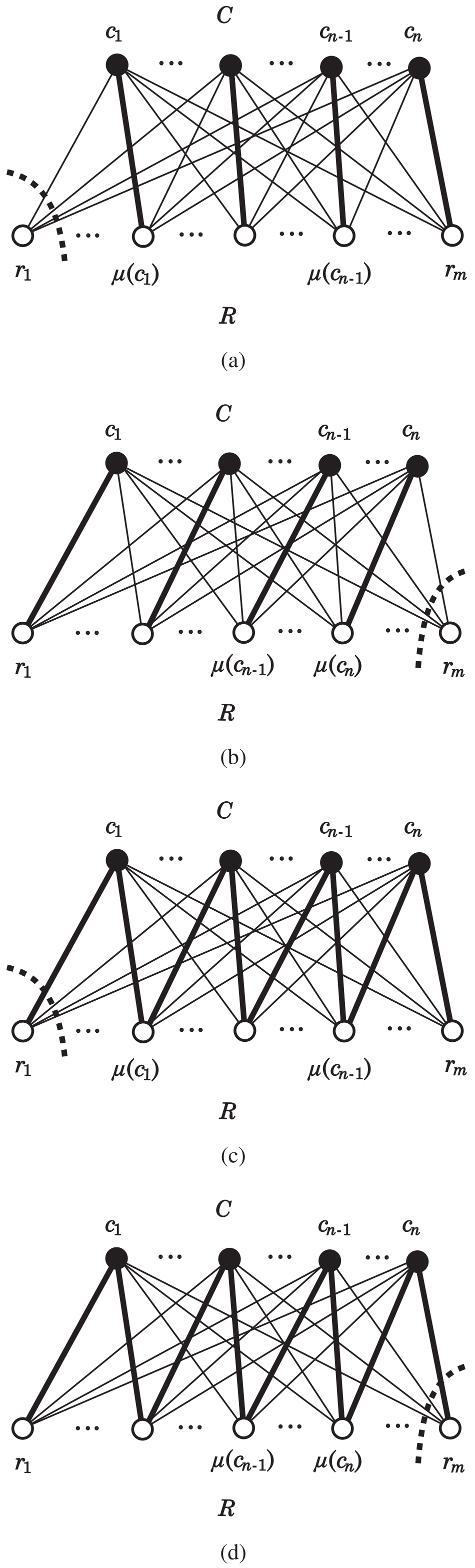
$$M' = \{r_1c_1, r_2c_2, \ldots, r_ic_i, \ldots, r_nc_n\}$$

and

$$M'' = \{c_1r_2, c_2r_3, \ldots, c_ir_{i+1}, \ldots, c_nr_m\},$$

such that it holds that $C \subseteq V(M')$ and $C \subseteq V(M'')$, and

$$w(P^*) = w(M') + w(M''). \tag{22}$$



(a)

(b)

(c)

(d)

Fig. 10 Procedure Alternating Path III. (a) A minimum cost bipartite matching M_1 in $G - \{r_1\}$ such that $C \subseteq V(M_1)$. (b) A minimum cost bipartite matching M_m in $G - \{r_m\}$ such that $C \subseteq V(M_m)$. (c) An R, C-alternating r_1, r_m-path P obtained if $w(M_1) \leq w(M_m)$. (d) An R, C-alternating r_1, r_m-path P obtained if $w(M_m) < w(M_1)$.

Moreover, notice that it holds that $r_m \notin V(M')$ and $r_1 \notin V(M'')$. By this observation, for the minimum cost matchings M_1 and M_m computed by Procedure Alternating Path III, it holds that

$$w(M_1) \leq w(M'')$$

and

$$w(M_m) \leq w(M').$$

Based on these observations, it holds that

$$\min\{w(M_1), w(M_m)\} \leq \min\{w(M'), w(M'')\} \leq \frac{1}{2}w(P^*). \quad (23)$$

Next, we will show the analysis for the case when $w(M_1) \leq w(M_m)$. The analysis of the other case, when $w(M_m) < w(M_1)$, is similar. We can see that for the simple path P returned by Procedure Alternating Path III it holds that

$$\begin{aligned} w(P) \ =\ & w(r_1c_1) + \sum_{i\leq i<n} w(c_i\mu(c_i)) + \sum_{i\leq i<n} w(\mu(c_i)c_{i+1}) \\ & + w(c_nr_m). \end{aligned} \quad (24)$$

Let $P_R \subseteq G[R]$ be the unique simple r_1, r_m-path visiting all and only the vertices $r_1, r_2, \ldots, r_m$ in the set R obeying the total order $\prec$ and $V(P_R) = R$, as in Lemma 3. By using Eqs. (10) and (11) from Lemma 4, for the values of $w(r_1c_1)$, $w(\mu(c_i)c_{i+1})$, and $w(c_nr_m)$ in Eq. (24), for $w(P)$ we get

$$\begin{aligned} w(P) \ \leq\ & w(\mu(c_1)c_1) + \sum_{r_1 \preceq r_\ell \prec \mu(c_1)} w(r_\ell r_{\ell+1}) + \sum_{1\leq i<n} w(c_i\mu(c_i)) \\ & + \sum_{1\leq i<n} \left(w(c_{i+1}\mu(c_{i+1})) + \sum_{\mu(c_i) \preceq r_\ell \prec \mu(c_{i+1})} w(r_\ell r_{\ell+1}) \right) \\ & + w(\mu(c_n)c_n) + \sum_{\mu(c_n) \preceq r_\ell \prec r_m} w(r_\ell r_{\ell+1}) \\ =\ & \sum_{r_1 \preceq r_\ell \prec r_m} w(r_\ell r_{\ell+1}) + 2\sum_{1\leq i\leq n} w(c_i\mu(c_i)) \\ =\ & w(P_R) + 2w(M). \end{aligned} \quad (25)$$

Finally, by the assumption that $w(M_1) \leq w(M_m)$ and Lemma 3, it holds that

$$w(P) \leq 2w(P^*),$$

as required. □

By Theorem 5, given an instance I of the NW-ALSDP that satisfies the necessary conditions, Procedure Alternating Path III will deliver a path with weight at most twice optimal, and is therefore a 2-approximation algorithm for the NW-ALSDP. From Lemma 5, we know that if the edge weight function w satisfies the triangle inequality, then there always exists an optimal path such that vertices in R and C appear alternately. As a consequence, the result of Theorem 5 leads to the following claim.

Corollary 3 For an instance $I = (G = (R \cup C, E), w, \prec)$ of the NW-LSDP such that $|R| > |C|$, and $R \cap C = \emptyset$, let P^* be an optimal path for I. If w satisfies the triangle inequality, and in addition $G[R]$ is a line with respect to the total order $\prec$, then for a simple path P computed by Procedure Alternating Path III given the instance I as an input, it holds that

$$w(P) \leq 2w(P^*). \quad (26)$$

6. Conclusion

In this work, we have investigated a scenario in which a drone is used in tandem with a delivery truck for the last-stretch delivery of parcels to customers' doorsteps. We introduced four models: the ALSDP, the NW-ALSDP, the LSDP and the NW-LSDP as problems of finding a minimum-cost path of a special structure in a graph. We showed that all of the graph problems are NP-hard, even when the weights of all edges are restricted to be 1 or 2. We identified a polynomially solvable instance type of the NW-ALSDP and the NW-LSDP with $|R| = |C| + 1$. Further, we proposed a polynomial-time approximation algorithm for the graph problem, and showed that its approximation ratio is bounded above by 2 in metric graphs.

As future work, it remains to investigate whether an approximation algorithm with an approximation ratio better than 2 exists or not. Further, it would be interesting to analyze some extensions of these routing problems, possibly in metric settings but with edge weight bias to account for additional transportation effort exerted by the drone when delivering a parcel [12], as well as to examine a combinatorial optimization based model for a routing problem including both the delivery truck and the drone [9], [10].

Acknowledgments The authors are thankful to the anonymous reviewers for their valuable suggestions on this paper.

References

[1] Agatz, N., Bouman, P. and Schmidt, M.: Optimization approaches for the traveling salesman problem with drone, *ERIM Report Series Reference*, No.ERS-2015-011-LIS (2015).

[2] D'Andrea, R.: Guest editorial, Can drones deliver? *IEEE Trans. Automation Science and Engineering*, Vol.11, No.3, pp.647–648 (2014).

[3] Gallacher, D.: Drones to manage the urban environment: Risks, rewards, alternatives, *Journal of Unmanned Vehicle Systems*, Vol.4, No.1, pp.1–10 (2016).

[4] Garey, M.R. and Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company (1979).

[5] Garone, E., Naldi, B. and Casavola, A.: Traveling-Salesman Problem for a Class of Carrier-Vehicle Systems, *Journal of Guidance, Control, and Dynamics*, Vol.34, No.4, pp.1272–1275 (2011).

[6] Gupta, S.G., Ghonge, M.M., Jawandhiya, P.: Review of unmanned aircraft system (UAS), *International Journal of Advanced Research in Computer Engineering & Technology* (*IJARCET*), Vol.2, No.4, pp.1646–1658 (2013).

[7] Ha, Q.M., Deville, Y., Pham, Q.D. and Hà, M.H.: On the min-cost traveling salesman problem with drone.arXiv preprint, arXiv:1512.01503 (2015).

[8] Korte, B. and Vygen, J.: *Combinatorial Optimization*: *Theory and Algorithms*, Springer Science & Business Media (2006).

[9] Mathew, N., Smith, S.L., Waslander, S.L.: Planning paths for package delivery in heterogeneous multirobot teams, *IEEE Trans. Automation Science and Engineering*, Vol.12, No.4, pp.1298–1308 (2015).

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

[11] Ponza, A.: Optimization of drone-assisted parcel delivery, Master's thesis, University of Padova (2016).

[12] Shurbevski, A., Nagamochi, H. and Karuno, Y.: Approximating the bipartite TSP and its biased generalization, *Proceedings of International Workshop on Algorithms and Computation WALCOM 2014*, pp.56–67, Springer (2014).

[13] Van Woensel, L., Archer, G., Panades-Estrutch, L. and Vrscaj, D.: Ten technologies which could change our lives: Potential impacts and policy implications, *European Parliamentary Research Service In-Depth Analysis*, PE 527.417 (2015).

[14] Watts, A.C., Ambrosia, V.G. and Hinkley, E.A.: Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use, *Remote Sensing*, Vol.4, No.6, pp.1671–1692 (2012).

[15] Zhang, C., Kovacs, J.M.: The application of small unmanned aerial systems for precision agriculture: A review, *Precision Agriculture*, Vol.13, No.6, pp.693–712 (2012).

Mohd Shahrizan bin Othman was born in 1981. He received his M.Sc. degree from the University of Science, Malaysia in 2004. Currently he is a Ph.D. candidate at the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University. His research interests include Combinatorial Optimization, Discrete Optimization, Scheduling, etc.

Aleksandar Shurbevski was born in 1986. He received his M.Sc. degree from Kyoto Institute of Technology in 2012, and a Ph.D. degree in Informatics from Kyoto University in 2014. As of 2014 he is an Assistant Professor at the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, and his research interests include Combinatorial Optimization, Graph Algorithms, etc. He is a member of IEEE and JSOR.

Yoshiyuki Karuno was born in 1966. He received his M.E. degree from Kyoto Institute of Technology in 1992 and Ph.D. degree in Informatics from Kyoto University in 2000. He is an Associate Professor of the Faculty of Mechanical Engineering at Kyoto Institute of Technology. His research interests include load balancing, routing, scheduling, etc., which are related to production and material handling systems.

Hiroshi Nagamochi was born in 1960. He received his D.E. degree from Kyoto University in 1988. He became a Professor in the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University in 2004 after serving for Toyohashi University of Technology. His research interests include Discrete Optimization, Graph Algorithms, Scheduling, Packing Problems, Chemical Graphs, etc.