Master's Thesis of Engineering

Drone Arc Routing Problem Using Ant Colony Optimization in Snow Removal Operation

개미알고리즘을 이용한 드론의 제설 경로 최적화

February 2022

Graduate School of Engineering
Seoul National University
Civil & Environmental Engineering Major

Ji-Won Ko

Drone Arc Routing Problem Using Ant Colony Optimization in Snow Removal Operation

Dong-Kyu Kim

Submitting a master's thesis of Engineering

December 2021

Graduate School of Engineering
Seoul National University
Civil & Environmental Engineering Major

Ji-Won Ko

Confirming the master's thesis written by
Ji-Won Ko

February 2022

Chair Seung-Young Kho (Seal)

Vice Chair Chungwon Lee (Seal)

Examiner Dong-Kyu Kim (Seal)

Abstract

Drones can overcome the limitation of ground vehicles by replacing the congestion time and allowing rapid service. For sudden snowfall with climate change, a quickly deployed drone can be a flexible alternative considering the deadhead route and the labor costs. The goal of this study is to optimize a drone arc routing problem (D-ARP), servicing the required roads for snow removal. A D-ARP creates computational burden especially in large network. The D-ARP has a large search space due to its exponentially increased candidate route, arc direction decision, and continuous arc space. To reduce the search space, we developed the auxiliary transformation method in ACO algorithm and adopted the random walk method. The contribution of the work is introducing a new problem and optimization approach of D-ARP in snow removal operation and reduce its search space. The optimization results confirmed that the drone travels shorter distance compared to the truck with a reduction of 5% to 22%. Furthermore, even under the length constraint model, the drone shows 4% reduction compared to the truck. The result of the test sets demonstrated that the adopted heuristic algorithm performs well in the large size networks in reasonable time. Based on the results, introducing a drone in snow removal is expected to save the operation cost in practical terms.

Keyword : Drone arc routing problem (D-ARP), Snow removal, Ant colony optimization (ACO), Length constraint, Random walk

Student Number : 2016-25408

Table of Contents

Table & Figure Lists

Chapter 1. Introduction

1.1. Study Background

Climate change causes many problems worldwide, one of which will be a sudden snowfall. In Seoul, due to an unexpected heavy snow in 2020, the city government was in trouble as citizens were stranded on their way home from work due to the government's late response. Since then, the city's snow removal guidelines have been open to the public and are so important that they are revised every year. However, snow removal method remains classic, and the idle cost of labor and equipment results in high maintenance costs. In fact, 13% of the city's snow removal-related employees are truck drivers, and the city considers reducing the cost of equipment due to high maintenance costs (Winter snow removal plan for Seoul, 2020). A cost for trucking management was $71.78 per hour which of 33% was labor cost, 24% fuel cost, and 25% fleet management cost according to the survey from American Transportation Research Institute (ATRI, 2020). Drones are expected to reduce the labor cost and to replace the fuel cost.

Furthermore, drones can rapidly service the snow removal agents. The existing snow removal method is inefficient due to the limitations of trucks, which are ground vehicles. Four methods of snow removal are (a) plowing (b) salting (c) blowing and (d) sweeping. The main methods are plowing and salting. If the amount of snow on the road is less than 3cm, only salting is performed. If it is more than 3cm, the truck is loaded with a shovel in the front and a tank with liquid salt in the back and performs the task at the same time. However, it is most important to spreading the snow remover as soon as possible when there is not much snow. In this context, ground vehicles inevitably pass through all roads and spray deicing agents, sometimes repeating unnecessary routes. Furthermore, when only certain sections of the road are to be snow-salted, deadheading routes must be generated.

Meanwhile, the logistics delivery using drones takes attention in academic field to overcome regional and time limitations. Drones can replace congestion time or allow rapid delivery beyond the reach of the conventional vehicle. Also, the labor costs and the safety accidents would be reduced. Thanks to these advantages, many researchers investigated the drone routing mainly with node routing approach such as DDP, CVRP, VRP-D. In case of the snow operation, since it follows the segment of the road, the service targets are not nodes but arcs. Therefore, it shall be formulated as Arc Routing Problems (ARPs).

Vehicles that can be quickly deployed is important in case of unexpected snowfall situations. From this perspective, drones can be a very flexible operation option. The state-of-the-art related of snow removal assumed the use of ground vehicles such as trucks. However, to overcome the limitation of ground vehicles, we propose the idea of spreading snow removals with drones. The following situations are favorable for drones in snow removal operation.

(1) For less snowfall or additional salting in heavy snow
(2) Inspection flight for icy-spot and spreading snow remover at the same time (Inspection + Delivery)
(3) Just before rush hour (fast response and service) in boulevard
(4) For bus only lanes or sparsely populated districts

To best of our knowledge, we are the first to introduce snow routing for drones. The functions of drones are widely known as inspection or delivery. The function of drones in this paper is 'spraying/spreading', and drones or UAVs are already being used for spraying water in agriculture and firefighting as **Figure 1**.

Figure 1. Drones spreading water for agriculture

1.2. Purpose of Research

The goal of the study is to experiment a drone ARP, specifically addressing the use of a drone to traverse the required roads for snow removal (i.e., spreading salt). When a drone and a truck traverse the same network, we analyze the different results in total travel cost. Also, we consider the length constraint of drone in snow routing. Contributions of the study are:

(1) Defining a new problem of snow removal using drones and managing its constraint

(2) Reduction of search space of drone ARP by auxiliary transformation method and random walk algorithm

(3) Reflecting the actual network topology on computer experiments

Chapter 2 reviews the related works regarding ARP and its algorithms. Chapter 3 defines a problem statement and its constraint. Chapter 4 describes the algorithms applied to the problem. Chapter 5 analyzes the computational experiments. Finally, Chapter 6 presents conclusions and future research challenges.

The differences of the study from the previous literatures are as follows. First, we will introduce a new fleet to snow routing problems. Second, we will conduct an optimization study by reflecting the location of the snow removal refill container and the load capacity of the drone (i.e., length constraint). Third, this is a new problem definition that did not exist in the snow removal study yet, and accordingly, an appropriate algorithm construction is required. Fourth, we will simulate a large-scale actual network. Using real networks is important as sometimes artificially generated problems cannot reflect the properties of real networks. The different metric of truck and drone will show the difference in feasible solution clearly. This also causes a problem of computational burden, and which is dealt with in detail in Chapter 4.

Chapter 2. Literature Review

2.1. Drone Arc Routing problem

Several review papers of (Macrina et al., 2020; Corberán and Prins, 2010; Corberán et al., 2021; and Chung et al., 2020) provided good insights about drone delivery problems and the new arc routing problems. Drone arc routing problem is rare compared to its node routing so far, but very promising research field (Corberán et al., 2021). ARPs with drones have been researched in various field, such as: robotic, autonomous vehicle (Kim. M. et al., 2019), and traffic. It is interesting that the drone has different function depending on the optimization routing characteristic. For example, inspection (Easton and Burdick, 2005; Sipahioglu et al., 2010; Oh et al., 2014; Chow, 2016; Li et al., 2018; Liu et al., 2019), wide-area coverage (Dille and Singh, 2013), and disaster aid (Oruc and Kara, 2018; Singgih et al., 2018). This is different from the node routing field where drones are usually function as delivery. We focused on the spreading function of drone. To best of our knowledge, none of any related works about the spreading drone is founded even in agriculture or fireworks yet in optimization field. Thus, the study will be the first paper that define the routing optimization with a concept of the spreading drone.

Recently, Campbell et al. (2018) introduced the concept of a drone RPP. Drone RPP was defined as a fleet of drone services the lines where the deadheading cost is considered as the Euclidean distance of any two points. The authors presented a novel optimal solution by developing an algorithm that decomposes arcs into numerous segments of polygonal chain. In the following work of Campbell et al. (2021), the study optimized the length constrained k-drones up to 137 lines with branch-and-cut and metaheuristic algorithm. As Poikonen et al. (2017) noticed, the crow-fly movement and the fast speed of drones are expressed as advantage in the optimization. From this perspective, unlike the traditional postman

problems, drones are beneficial in that they fly directly between any two nodes. In addition, a few literatures considered a dynamic feature of drones in a rotating moment (Oh et al., 2014; Dille and Singh, 2013; Shivgan and Dong, 2020).

Consequently, many researchers have investigated ARPs and drone delivery problems while very few did on drone ARP so far. Developing ARP algorithms when applying drones instead of ground vehicles has quite difficulty as Corberan et al. (2021) mentioned. The polygonal chain algorithm in Campbell et al. (2018) can take advantage of the drone's characteristics, but the search area increases as the chain increases. We implement a metaheuristic algorithm to efficiently solve the drone ARP in large size networks. An Ant Colony Optimization (ACO) algorithm that is known to successfully solve combinatorial optimization problems is adopted to this drone ARP.

2.2. Snow Removal Routing Problem

Snow removal routing problem has been of interest for a long time thanks to its powerful practicality in operation routing (OR) field. In practical terms, it is not straightforward to distinguish between ARP and vehicle routing problem (VRP) for implementing many problems. In some cases, ARP with sparse demands can be defined as VRP by aggregating the demand of several sections (Corberan et al., 2021). However, snow removal problem in this paper must completely traverse the road with servicing every point of it. Thus, the problem is more appropriately defined as an arc routing problem.

To best of our knowledge, routing problems with snow removal have only considered fleets of trucks. As mentioned above, winter road maintenance includes plowing and spreading deicers, and that operational researches have different points according to which one they are focused on. A comprehensive literature about snow plowing in past decades can be found in Salazar-Aguilar et al. (2012). The authors proposed a Synchronized Arc Routing Problem which handles a synchronized fleet of trucks and minimizing the make-

spans. The concept of the problem is interesting however, the problem is valid only under condition with heavy snow and no other vehicles in the streets. Meanwhile, literatures about snow salt spreading can be found in Corberan et al. (2021). According to the paper, whether to be modeled as directed or undirected graph depends on the central barrier or one-way restriction. Since we didn't assume any of it, the experiment sets are undirected graph, which makes the algorithm complicated. The salt spreading problems normally consider a constraint about time because the salt should not be treated too early (then blown away) and too late (then iced). As a result, the problems minimize the total travel distance with time windows or divide categories considering the road's importance.

Likewise, snow salting needs fast travel in practical terms. Therefore, the idea of this paper using a fast drone as a fleet for salt spreading will be meaningful. In addition, we introduce a new concept that a fleet needs to refill the liquid salt. This is because we adopt one drone for the tour, so it has a capacity constraint. This is another challenge for formulating algorithm, and we will discuss this in Chapter 4. The idea can be applied to electric trucks with intermediate charging route if the fuel trucks are replaced by the green vehicle in the future.

2.3. The Classic ARPs and Algorithms

Most ARP studies are based on ground roadways. ARP is originated from the Königsberg bridges problem in 18th century. The 'Eulerian circuit', which is an undirected connected graph with even-degree vertices, has a tour that visits all arcs only once to complete the closed walk. However, the postman problems, CPP and RPP can visit the same arcs but aim to minimize the total travel distance. If all edges of a graph are required, they are defined as CPP. If only part of the graph is required, they are defined as RPP. Undirected and directed CPP is known to have an optimal solution in polynomial time. Edmonds and Johnson (1973) provided a matching odd-vertices algorithm for CPP. The RPP was introduced by Orloff (1974). The

‘deadhead’ in RPP is generated when a non-required or already serviced edge is visited unwillingly while searching the feasible route. RPP is known as NP-hard (Lenstra and Kan, 1976). An exact algorithm based on branch-and-bound using Lagrangian relaxation was developed by Christofides et al. (1981). However, due to the exponential algorithm and computational inefficiency, a heuristic approach has been developed (Pearn and Wu, 1995) since then.

According to Corberan et al. (2021), a general issue for solving ARPs is that which of the following modeling method is better. One is to model the problem as ARPs to represent the original structure under relatively sparse network and the other is to convert the original problem into a node routing problem by adding extra nodes and/or constraints. To explain, ARP can be solved by converting the problem into equivalent node routing formulation. Since Pearn et al. (1987), several researchers of (Laporte, 1997; Baldacci and Maniezzo, 2006; Longo et al., 2006; Foulds el al., 2015) have developed the transformation methodology with exact algorithm. However, a study in Letchford and Oukil (2009) focused on sparse graphs and criticized the transformation method. They suggested that transforming the CARP into the CVRP is not necessary because the method of Longo et al. (2006) leads to computational burden and suffer from symmetry.

Meanwhile, Lacomme et al. (2004) solved the CARP by metaheuristic, a memetic algorithm (MA). The authors used several constructive heuristics to generate initial population of MA. Santos et al. (2010) performed an improved ACO algorithm for CARP. One of the improved parts was the initial population. They generated feasible CARP routes for the initial pheromone values by comparing several heuristics: random arc selection, path-scanning with ellipse rule, and PSEU with local search. Then, a single network tours (SNTs) which is infeasible and ignore the constraint were generated. Lastly, they disaggregated the SNTs to generate a feasible route by using the method of Ulusoy (1985).

2.4. Large search space and Arc direction

Search space is important when exploring large-sized instances or when the fleets are not ground vehicles. This is because drone makes a completely connected network as we will discuss in Chapter 3. In addition, the arc direction in ARP causes special consideration compared to any TSP or VRP research. Seriously different solution arises according to the arc direction combination. Its decision procedure may cause more search space depending on how the researcher set the network concept and algorithm architecture. Since even recent related works that studied about CARP did not consider large-sized networks of drone arc routing, it was hard to find severe consideration on arc direction decision procedure.

Tirkolaee et al. (2019) presents two virtual graphs to propose ACO algorithms. This first graph corresponds to the servicing vehicles and the second graph considers their constructed routes. The second graph was the virtual graph that the node of the graph represents the required edges. Each arc $\mathrm{e(i,j)}$ consists of two directed arcs: $\mathrm{e(i,j)}$ and $\mathrm{e(j,i)}$, and they have different index of a and b. For example, $\mathrm{edge\{e(2,3), e(3,2), e(3,4), e(4,3)\} = index\{a, b, c, d\}}$.

Foulds et al. (2015) use direction vector of $a \in \{0,1\}^r$ for each required edge. Then they applied BCP algorithm to the equivalent node routing problem instance, called NR(H). It finds the least cost set of q-routes from the original arc routing problem AR(G) arising from all $a \in \{0,1\}^r$. There exists a vector $a \in \{0,1\}^r$ that makes a feasible solution to NR(H) with no greater cost than the cost of feasible solution of AR(G).

Santos et al. (2010) describes their Ant-CARP implementation details in appendix. The network was represented by two arcs: $a_k = (i,j)$ and its reverse $\overline{a_k} = (j,i)$. If one of the required arcs is serviced, then the reversed one is set to be serviced.

Ghiani et al. (2010) uses an auxiliary multigraph $G^{aux} = (V^{aux}, A^{aux})$. The first step of the ACO algorithm is constructing 'Auxiliary graph' procedure. Each required edges of original graph,

$e = (v_x, v_y)$, are correspond to a vertex $v^{aux} \in G^{aux}$. Then each vertex $(v_i^{aux}, v_j^{aux} \in V^{aux})$ are connected by arc a_{ij}^{aux}. There are twelve possible arcs in a_{ij}^{aux} considering each case of the original edges, intermediate facility, and the depot. In other words, 12 different costs of the shortest chain connecting the two required edges. Afterall, ants tour the auxiliary graph and accumulate pheromone trail.

Longo et al. (2006) transformed CARP into equivalent CVRP with $(2r + 1)$ nodes where r is number of required edges. A required edge corresponds to $S_{ij}, S_{ji}(=2r)$ and disconnected pairwise edge $r(= 1)$. They only considered the solution as feasible when S_{ij} and S_{ji} are visited in sequence and the cost was counted only when the route passes the S_{ij}, r, and S_{ji} in sequence.

To sum up, two approaches are suggested according to the related works: (a) Each arc simply generated two arcs (or index), (b) Each arc is converted to one (or more) auxiliary or virtual node(s). Disadvantage of the first approach is a large search space, and the second approach is complicated in including directional information. This study suggests a method to decrease the search space complexity and manage the direction information by defining the shortest path and adapting local improvement.

Chapter 3. Method

3.1. Problem Statement

3.1.1. General Statement

This study compares the results of optimizing the snow arc routing of a truck and a drone in the real urban networks. For this, the study assumes that a drone or a truck as a fleet of servicing edge for the given network. Problem definition is as follow:

1) Carry out the task of spreading snow remover agent on the roads using a drone or a truck.
2) The problem is under the condition that snow is accumulated less than 3cm or the plowing has already been completed.
3) There is a limit to the amount of snow remover agent that a drone can carry at once. Therefore, a constraint of refilling the snow remover at the refill point is implemented.

For drones, edges can be serviced at any point of them by crow-flying, even at the midpoint. In other words, for drone ARP, there are two possible types of edges in the network: physically existing roads (required edges) and candidate edges with Euclidean distance only for passage (deadhead). On the other hands, trucks use the physically existing roads as both required edges and deadhead. For this reason, the drone ARP network has more edges to search compared to the truck ARP. In other words, trucks and drones have the following difference in their definition of ARP.

(a) Truck CPP: problem that servicing all edges in the network

(b) Drone RPP: problem that exploring the required edges where the Euclidean flight path is added to original network as deadhead.

As a result, the original network is enhanced to a fully connected network in drone case. Note that in drone ARP, a drone should explore all edges in the fully connected network to find an optimal path. With this larger search space, an improved heuristic algorithm approach will be considered in Chapter 4.

Node routing problems, such as VRP, often ignore arcs and real shapes in the given network but this may not be a big problem. However, in ARP, it can be a serious problem because the target of the service is arcs. Under the circumstance that ground vehicles can only use the physically existing roads, it is important to implement a real road shape in the network. Therefore, edges will be provided by a real-world road network from Google Map. Also, the problem is generated as undirected network. **Figure 2** shows a simple example of real-world network based on central Paris, France.

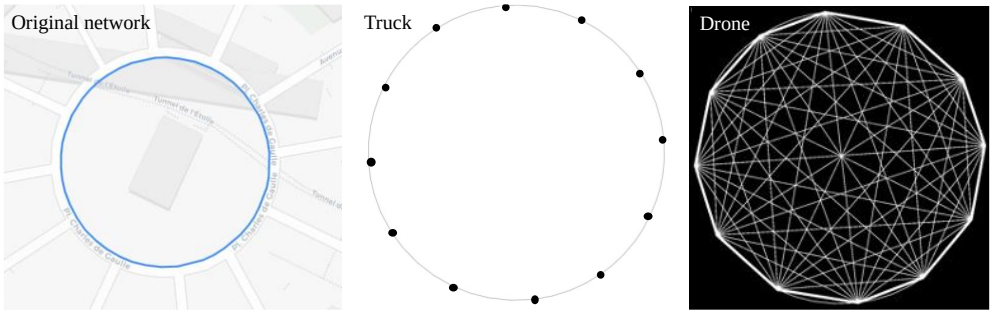

Figure 2. Networks of computer experiments

We can find that the required edges are same for both, but only drone network has fully connected network. Note that if we have n vertices and r required edges in G for truck, then $[(n^2 - n)/2)]$ edges and r required edges for drone in G.

Meanwhile, drone's speed and battery constraint are generally important in energy efficient routing. Due to these constraints, possible flight length would be limited and then k drones shall be needed. Also, the ground vehicle needs to refill gasoline and work shifts. However, since this research aims to compare the total travel cost of both vehicles under the same conditions, we do not consider these constraints currently. Future work can consider these constraints as cost dimension. Consequently, a term of cost is interchangeable with a distance in this paper.

3.1.2 Snow Remover Agent Refill

Trucks generally load snow remover in a tank about 1ton, it does not need to be refilled after leaving the depot. This constraint gives drone's flight a drawback. Therefore, to reflect the constraint that occurs in practical, we will consider a drone's snow removal agent charging model. We present a model that reflects the snow removal capacity Q to a drone. Specifically, the drone visits the nearest salt container when out of capacity Q. Then it refills and continues to fly from the spot. This is somewhat different from the Multi-Depot Vehicle Routing Problem with Inter-Depot Routes (MDVRPI). Since MDVRPI has a fixed number and location of the inter-depots, this can be composition of sub-problems where each route starts at one depot and ends at the other depot (Crevier et al., 2007). However, the problem here is different because of the following reasons: (a) It is unknow which of several containers to be used, and (b) The refill point is not a finite node, but a continuous and infinite arc point. Regarding the capacity of the snow remover, the following assumptions are used.

(1) Maximum capacity of agriculture or firefighting drone: 150L
(2) Average lane width: 3m
(3) Maximum spreading length: 7m (Cover 2 lanes at once if needed)
(4) 35L (Calcium Chloride Aqueous Solution) per 1km based on 3.5km width lane (Kim. G., 2017) = 30L/km on 3km width lane
(5) 150L drone can service max 5000m per once

With these assumptions, the capacity of the deicing agent can be simply substituted with the length constraint. As a result, a drone has max capacity Q within 5000m expressed as a limited length.

3.2. Formulation

The proposed mathematical model with length constraint is presented as Equation 1:

$$\textbf{Minimize}\, \boldsymbol{T} = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\boldsymbol{C_{e_i}} + \boldsymbol{C_{e_j}} + \boldsymbol{d}(\boldsymbol{e_i}, \boldsymbol{e_j})\right) \quad (1)$$

$Subject\ to$:

- $0 \leq Q \leq 5000$
- $C_{e_i}, C_{e_j} \geq 0$
- $\forall e_i,\ e_j \in \{required\ edges\}$
- $\forall i = 1, \ldots., n\ ,\ \forall j = 1, \ldots., n,\ i \neq j$

The objective function is to minimize the total travel cost. $\boldsymbol{C_{e_i}}$ is the cost of a required edge and $\boldsymbol{d}(\boldsymbol{e_i}, \boldsymbol{e_j})$ means the distance between the required edges. i and j is the node of the network. Q is the expression of the length constraint. This means the maximum load capacity of snow removal agent for drone. In this study, the max value is set according to the assumptions in the Chapter 3.1.2. As a result, the individual travel cost becomes the cost of the required edge plus a distance to move to the next required edge. At this point, the shortest distance of the corresponding edges is searched by a suggested algorithm. This will be discussed specifically in Chapter 4.

Chapter 4. Algorithm

4.1. Overview

The algorithm system consists of the following two-phase. First, generate single network tours (SNTs). Second, find the best refill point. This is the concept of route-first and cluster-second. To explain, we generate SNTs using metaheuristic algorithm then do random walk to find the best combination of the refill points. **Figure 3** shows the flow of algorithms.

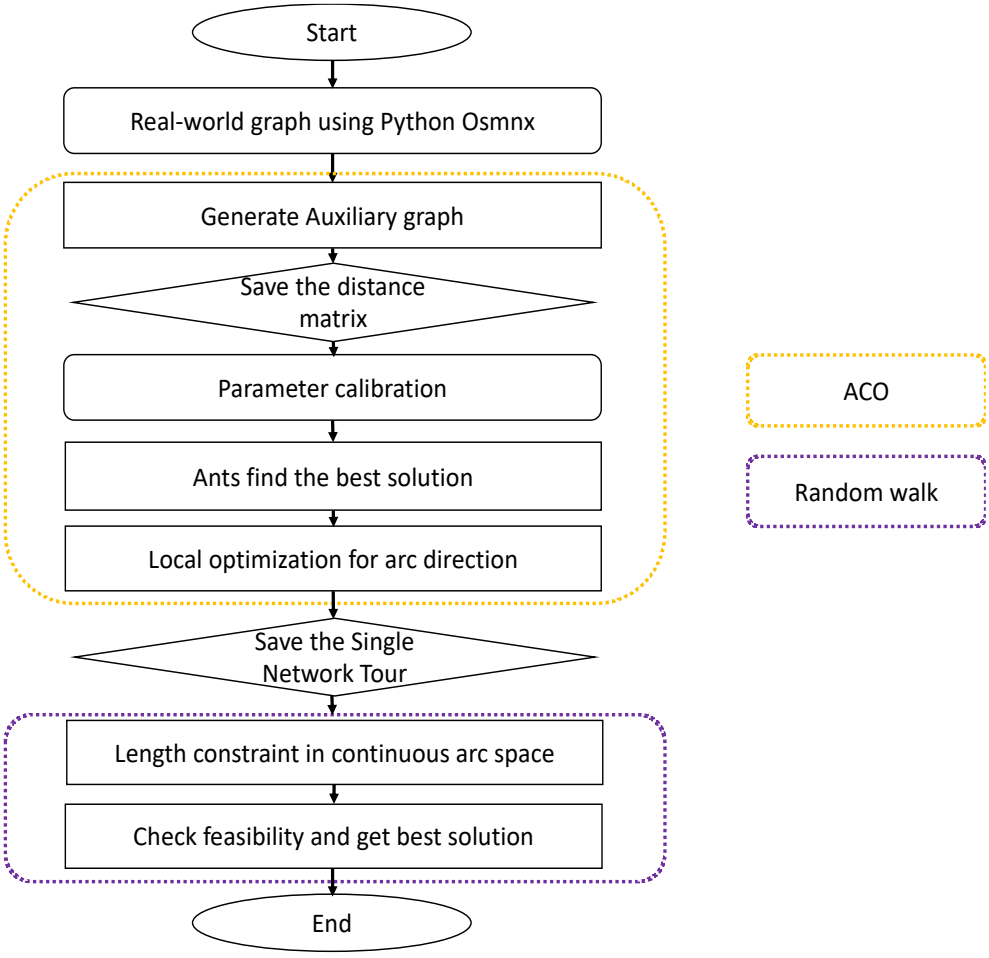

Figure 3. Algorithm flowchart

As for solving the ARPs, algorithms such as branch-and-bound based algorithms (Campbell et al., 2018; Letchford and Oukil, 2009; Corberán et al., 2007), genetic or memetic algorithm (Lacomme et al., 2004; Lacomme et al., 2006; Zhu et al., 2007), and ACO (Santos et al., 2010; Tirkolaee et al., 2019; Huang and Lin, 2014) have been proposed. In this section, a heuristic algorithm using ACO will be discussed. Especially, D-ARP has a large search space. To solve this problem, we introduce the 3-step algorithm which including the auxiliary transformation method, ACO, and local improvement algorithm.

4.2. Auxiliary Transformation method

The first mechanism of ACO in drone arc routing is to reduce the search space by reducing the size of the trail matrix and the directional decision vector. For this, the arc in the original graph is converted into pseudo node. A typical disadvantage of this approach is the number of pseudo nodes that causes computer memory requirements (Letchford et al., 2009). To relax this problem, we simply convert a required edge to a corresponding pseudo node. In other words, put one vertex M in the middle point of an edge. Foulds et al. (2015) also suggested putting a vertex in the middle of edge, but the authors added a binary vector $a \in \{0,1\}^r$ and there exists one corresponding node routing problem for each binary vector. In other words, it needs to find the best combination of the vector among all possible combinations. This approach was possible with BCP algorithm and pricing rule with dynamic programming in small-sized network. However, this algorithm architecture is not easily adapted to ACO algorithm and large-sized network.

The difference of our work is that we don't add directional decision vector on the pseudo vertex. Instead, we consider the impact of arc direction by calculating the shortest path and proceeding the local optimization. Therefore, it is easy to adapt ACO algorithm. This means matching the edge with one virtual node. This matching concept is same for both truck and drone. However, they have

different shortest path algorithm. In other words, the shortest path of deadheading flight is different between the truck and the drone. **Figure 4** shows the auxiliary transformation concept.

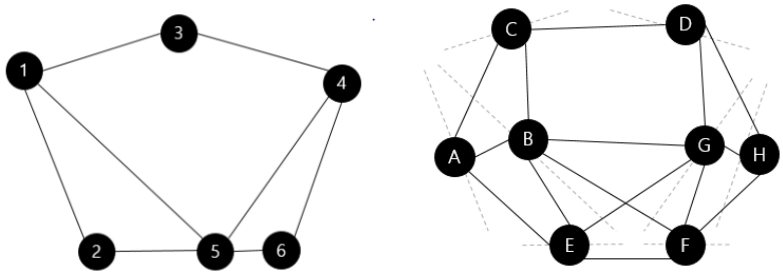

Figure 4. Auxiliary graph

For a truck, the shortest path of deadheading from edge (1,2) to edge (4,6) is calculated as A−B−G−H. Note that the distance of A−B is considered as [edge (1,2) ~ edge (1,5)] and that is a minimum of [2→1→5] or [1→1] or [1→5] or [2→5]. In case of a drone, the shortest path from edge (1,2) to edge (4,6) can be a Euclidean distance of A−H. Then it is minimum of ([1→4], [1→6], [2→4], [2→6]). The notation of equations follows **Table 1.**

TABLE 1 Notation of the Equations

$G = (n, e)$	Original graph G with node (n) and edge (e)
$G' = (N, E)$	Auxiliary graph G'
$i,\ j,\ p,\ g$	Nodes in G
$e_{ij}^R,\ e_{pq}^R$	Required edges in the original graph G
$m_{ij}^R,\ m_{pq}^R$	Middle point vertex on required edge in original graph G
n_1, n_2	Nodes of either side of required edge in original graph G
N_k^h, N_l^h	Node in pseudo graph G' (same with $m_{ij}^R,\ m_{pq}^R$)

Let, e_{ij}^R be a source edge and e_{pq}^R be a target edge. Then, the path from N_k^h to N_l^h is defined as **Equation 2** and **Equation 3**. The original edge, e^R, is converted into $N^h (=m^R)$.

$$\boldsymbol{d(N_k^h, N_l^h) = 0.5\, e_{ij}^R + \Delta(e_{ij}^R, e_{pq}^R) + 0.5\, e_{pq}^R} \qquad (2)$$

Where $d(N_k^h, N_l^h)$ means the shortest path from a pseudo node to another pseudo node and Δ refers to the distance of the shortest path of the two nodes. Consequently, the edge cost is included in the $d(N_k^h, N_l^h)$. The distance of the shortest path, Δ, is calculated as Equation 3:

$$\Delta(\boldsymbol{e_{ij}^R}, \boldsymbol{e_{pq}^R}) = \mathbf{min}[\boldsymbol{d(i,j)}, \boldsymbol{d(i,p)}, \boldsymbol{d(j,p)}, \boldsymbol{d(j,q)}] \quad (3)$$

Where, $d(i,j)$ is considered as Dijkstra algorithm for truck and Euclidean distance for drone. The shortest path matrix by this algorithm is calculated and saved as prior information of each edge. In this point, a drone has fully connected graph which gives a chance for the ants to find a shorter path (i.e., Euclidean path).

4.3. Ant Colony Optimization

ACO metaheuristic algorithm has been effectively applied to NP-hard combinatorial optimization (Dorigo and Stützle, 2004). There are several different ant algorithms, for example, Ant System (AS), Ant Colony System (ACS), Max-Min Ant System, and Rank-based system, which mainly differ in the search control method. Here, we use the AS and ACS mixed method.

A decision-making is operated by the following two parameters: (a) pheromone trail level ($\boldsymbol{\tau_{ij}}$), and (b) attractiveness ($\boldsymbol{\eta_{ij}}$). The pheromone level represents *posteriori* desirability, obtained from historical information (Rizzoli et al., 2007). Meanwhile, the attractiveness is a *priori* available heuristic value, which can be seen as the reciprocal of cost by Equation 4:

$$\boldsymbol{\eta_{ij}} = \frac{\mathbf{1}}{\boldsymbol{d(N_k^h, N_l^h)}} = \frac{\mathbf{1}}{\mathbf{0.5}\boldsymbol{e_{ij}^R} + \Delta(\boldsymbol{e_{ij}^R}, \boldsymbol{e_{pq}^R}) + \mathbf{0.5}\boldsymbol{e_{pq}^R}} \quad (4)$$

Trails, the traces of pheromones, evaporate over time to prevent excessive accumulation of the historic record. In tour

construction step, the ants (m) select the next arc (j) by the following rules. For each iteration performance, the ants expand the possible solution from the current state and choose whether to go another state by the pseudo−random−proportional action choice rule. The expressions of the following equations referred to (Dorigo and Stützle, 2004; Rizzoli et al., 2007; Dorigo et al., 1999; Dong and Xiang, 2006). The choice rule is determined by Equation 5:

$$\mathbf{j} = \begin{cases} \boldsymbol{argmax}_{g \in K_i} [\boldsymbol{\tau}_{ig}]^{\alpha} [\boldsymbol{\eta}_{ig}]^{\beta} & \textbf{(if q} \leq \boldsymbol{q_0}\textbf{)} \\ \boldsymbol{S} & \textbf{(otherwise)} \end{cases} \quad (5)$$

Parameter α and β is the weight of pheromone trail and the weight of attractiveness respectively. Parameter q is a random variable over [0,1] and q_0 is a tunable parameter over [0,1]. Then, q_0 means the importance of an exploitation versus exploration (exploitation focuses on searching deeply in the promised feasible set while exploration is a diversity of solutions). K_i belongs to the sets of feasible expansions that the ant has not yet visited. S is selected by the probabilistic rule in stochastic manner of Equation 6:

$$\boldsymbol{P_{ij}} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{h \in K_i} [\tau_{ih}]^{\alpha} [\eta_{ih}]^{\beta}} \quad (6)$$

After the ants find the paths, they update pheromone levels as follows: the pheromone trail ($\boldsymbol{\tau_{ij}}$) matrix of each edge increases by the amount of $\frac{1}{L}$ (L is an objective value). Pheromones in t iteration are updated by Equation 7:

$$\boldsymbol{\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}} \quad (7)$$

where ρ is a parameter governing the trail decay that helps to forget previous bad decisions where $0 < \rho \leq 1$. The strength of the trail fades over time, enabling the expansion of edges that have yet to be visited. The best solution (ψ) found by a global elitist ant so far, and

edges belonging to the best solution ($\forall(i^*, j^*) \in \psi$) are updated by Equation 8:

$$\boldsymbol{\tau_{i^*j^*}(t+1) = (1-\rho) \cdot \tau_{i^*j^*}(t) + \sigma \cdot \Delta\tau^{\psi}_{i^*j^*}} \quad (8)$$

where σ is intensification to the best path and $\Delta\tau^{\psi}_{i^*j^*} = 1/L^{\psi}$ (L^{ψ} is the objective value of the best solution so far).

4.3.1. Parameter Calibration

The ACO parameter is important since it can decide the efficiency of the algorithm. The influence of parameters is tested with respect to the solution quality of the middle size network ("Boston" in Table 3). The baseline of the parameter setting is $m = 100$, $q_0 = 0.1$, $\rho = 0.1$, $\sigma = 2$, $\alpha = 1$, and $\beta = 1$. To the baseline, the alternative series of m, q_0, ρ, β are examined over the following range.

- $m \in \{25, 50, 100, 200\}$
- $q_0 \in \{0, 0.1, 0.3, 0.7\}$
- $\rho \in \{0, 0.1, 0.3, 0.7\}$
- $\beta \in \{1, 3, 6, 9\}$

The remaining parameters are fixed as the baseline. The method refers to the related works of (Santos et al., 2010; Dong and Xiang, 2006). As a result, the values with $m = 100$, $q_0 = 0.1$, $\rho = 0.1$, $\sigma = 2$, $\alpha = 1$, and $\beta = 6$ was chosen as the final parameter. The examination value can be checked through Table 2.

TABLE 2 Parameter Combinations

		Best	Worst	Average			Best	Worst	Average
m= 25	Truck	**2018**	2622	2233	ρ= 0	Truck	2081	2343	2209
	Drone	**2008**	2224	2118		Drone	2018	2248	2126
m= 50	Truck	2033	2398	2213	ρ= 0.1	Truck	**2081**	2364	**2201**
	Drone	2099	2247	2162		Drone	**2018**	2480	**2142**
m= 100	Truck	2081	2364	**2201**	ρ= 0.3	Truck	2137	2332	2197
	Drone	2018	2480	**2142**		Drone	2166	2356	2262
m= 200	Truck	2087	2302	2182	ρ= 0.7	Truck	2083	2691	2296
	Drone	2083	2324	2154		Drone	2099	2625	2391

q_0= 0	Truck	2101	2324	2225	β= 1	Truck	**2081**	2364	2201
	Drone	1968	2353	2150		Drone	**2018**	2480	2142
q_0= 0.1	Truck	2081	2364	**2201**	β= 3	Truck	2081	2324	2203
	Drone	2018	2480	**2142**		Drone	2013	2190	2110
q_0 = 0.3	Truck	2081	2216	2146	β= 6	Truck	2083	2364	**2196**
	Drone	2033	2290	2166		Drone	2083	2216	**2156**
q_0= 0.7	Truck	**2033**	2468	2222	β= 9	Truck	2083	2691	2303
	Drone	**1979**	2343	2138		Drone	2018	2597	2161

Note. m: number of ants, ρ: rate of pheromone evaporates, q_0: exploitation probability, β: weighting of attractiveness

The values are adopted since they get the best global average for a truck while confirming the improved solution for a drone. It turns out that too high ρ causes the worst case in drone optimization.

4.4. Post Processing for Arc Direction Decision

Local search processes are often used in metaheuristics to improve a current solution. The procedures are typically done with a neighborhood search such as 2−opt and insertions (Santos et al., 2010). In this part, the best solution is observed to improve its directional information. In the post process, N^h of the auxiliary graph is reconstructed to the original required edges $e_{ij}^R(n_1, n_2)$, with the decision of entering direction (i.e., whether $n_1 \rightarrow n_2$ or $n_2 \rightarrow n_1$) by a greedy search. After this local optimize procedure is determined and the best solution comes out, the cost of the objective function is recalculated as the following Equation 9:

$$\textbf{Total distance} = \boldsymbol{e_{12}^R} + \Delta\left(\boldsymbol{e_{12}^R}, \boldsymbol{e_{23}^R}\right) + \boldsymbol{e_{23}^R} + \ldots \quad (9)$$

The procedure conducts a greedy search algorithm with survey of each element in the path. It performs the local optimization by comparing which of the connecting nodes of the edge to entry and exit. This is a procedure to find any reversed arc that provides an improved result. We do a local optimization process with this method until finding a better solution.

4.5. Length Constraint and Random Walk

In this section, we consider the salt refill route of drone. We generated a single network tour (SNT) from algorithms with ACO. Thus, we get a 'total length of SNT' which now we call as T. Then, calculate the least refill number N (round down of T/Q to integer). This second phase has the following steps:

(1) Divide the SNT into N sets of arcs. This generates initial seed for each arc set.
(2) Generate refill points by random walk and get candidate combinations
(3) Examine the feasibility of each route and take feasible route scenarios
(4) Calculate the exact coordinate of the feasible refill point by using an interpolation method
(5) Get a shortest path between the refill point and any refill containers
(6) Calculate the total travel distance and take the best route scenario

A random walk concept is a sequence of discrete steps of fixed length in random process such as Brownian motion phenomenon. A random walk is the stochastic process formed by successive formation of independent, identically distributed random variables (Gregory F. L. and V. Limic, 2010). According to Spitzer, F. (2001), the transition function $P(x,y)$ requires the properties of $0 \leq P(x,y) = P(0,y-x)$, and $\sum_{x \in R} P(0,x) = 1$, then, a random walk can be defined as a function $P(x,y)$.

In our work, the random walk in one dimension and the walk series $S_{i,t}$ is defined as Equation 10. The random point variable x follows the normal distribution of $x \sim \mathrm{N}(\mu, \sigma^2)$. A ith random walk of t iteration time and p is a tunable parameter. We set the $\mu = 0$ and $\sigma = 0.5$ and $p = 10$. $S_{i,0}$ are the initial seeds. $S_{i,t+1}$ is initialized when the interval is over $1.2Q$.

$$S_{i,t+1} = S_{i,t} + x \times \frac{Q}{p} \qquad (10)$$

Subject to :

- $\left| S_{i,t+1} - S_{j,t+1} \right| \leq 1.2Q, \ \forall i,j$
- $S_{i,0} = i \times Q$

The reason we do random walk here is as follows. A stochastic manner is another option, but it easily generates infeasible solutions when sampling the candidate routes. For example, when the total length of SNT is 100 and the Q is 25, then $\{0, 25, 50, 75, 100\}$ is the best solution, but the stochastic manner can choose $\{0, 1, 3, 5, 100\}$ unfortunately. Therefore, we set the initial seed value (a known feasible solution) and do a random walk to get as many sets as possible. Now we call the refill point as 's' and the location of container as 'c', where s is from 1 to N. Since the total distance need the shortest path between the refill point and then the Equation 9 is changed to **Equation 11**:

$$\text{Total distance} = \sum_{i,j,p,q}[e_{ij}^R + \Delta(e_{ij}^R, e_{pq}^R) + e_{pq}^R] + \sum_{k=1}^{N} \Delta(s_k, c_k) \quad (11)$$
$$\{i, j, p, q\} \in G(n),\ i \neq j,\ p \neq q,\ e_{ij}^R \neq e_{pq}^R$$

Figure 5 shows a network with refill constraint. The first phase algorithm creates an array of arcs and nodes in SNT. A set of arc consists of its (node, node). From this, the total length of the SNT is calculated to be $23{,}000\text{m}$. So, the drone must have $\frac{\mathrm{T}}{\mathrm{Q}} = \frac{23{,}000}{5{,}000} = 4.6$, i.e., at least 4 refill points. The baseline for these 4 refill points is to visit every $5{,}000\text{m}$ of course, as shown by the white star in the figure. However, given the location of the salt container, the optimal refill point is the location of the yellow star. We realized that the refill positions were adjusted according to the salt container location. At the same time, this is a feasible route, since the distance between each yellow star does not exceed $5{,}000\text{m}$. Here, the position $(\mathrm{x}', \mathrm{y}')$ of the yellow star is obtained by interpolation method. For example, we know the coordinates of node2 and node5 of the two closest nodes $(1,2)$ and $(5,7)$. Also, since the distance from $(1,2)$ to $(\mathrm{x}', \mathrm{y}')$ and the distance from $(\mathrm{x}', \mathrm{y}')$ to $(5,7)$ are known, it can be obtained simply by interpolation.

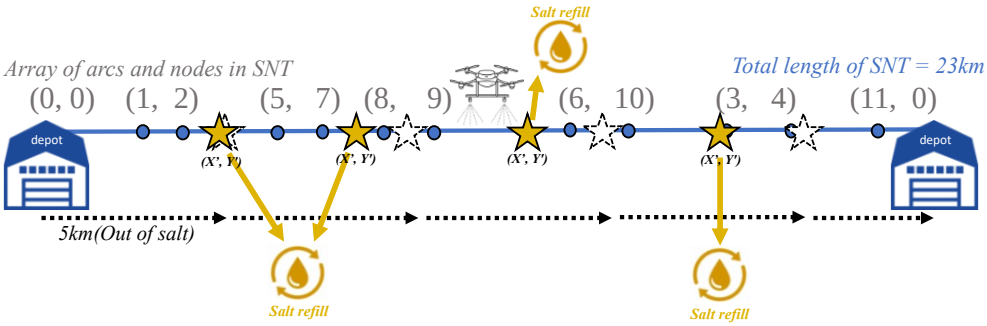

Figure 5. Toy network of refill constraint

The important point is that while we perform the drone arc routing, possible candidates for refill points are continuous. As a simple case, we can set a stop point every 5,000m, but this is not an optimized result. Because of the fixed position of the salt container, there may be a better stop point for the objective function. Therefore, it is necessary to optimize the refill point considering the location of the salt container, which is possible at any point in the arc (remember that the drone can escape from any point of the arc).

On the other hand, a problem arises if a stop point is assumed at only a node to handle a continuous search space. First, because the length of any arc in the network can exceed the capacity, and if so, it is infeasible in any cases. Second, there is inefficiency. This is because the drone tends to refill at the previous node if it notices that the salt will be exhausted at the next arc even though there is enough salt remaining. Therefore, the refill point can increase as $N+\alpha$ and this causes increase in objective function value. As a result, the best method is to create a refill point in a random walk with initial seed while managing a continuous space, and then check the feasibility. We provide a better solution by searching for any improved route. To sum up, we can extract as many route candidates as we want through a random walk, then filter all feasible routes and find the best solution with the minimum total distance.

Chapter 5. Results

In this section, the route optimization result is demonstrated with the representative topologies. Also, we evaluate the efficiency of using drones in snow removal through various instances of real urban networks. A comparison of truck ARP and drone ARP in terms of total travel cost and flight pattern is suggested.

5.1. Application in Toy networks

Figure 6 of star network and **Figure 7** of tree network present the different flight pattern in detail. Both vehicles may start from any point but must return to the starting point. As a result, in the first case (Figure 6), the truck must visit the center node for five times while the drone just needs three times. It is noticeable that the drone brings the much shorter path at the longest edge. This is because the drone directly travels from the end of the edge to another edge while fully minimizing the deadhead. This feature greatly reduces the deadhead of the longest edge in the graph.

The second (Figure 7) is a tree network where any two vertices are connected by one path. A ground vehicle goes back and forth at every edge when move to other edges. The inefficient movement generates the deadhead as much as the edge cost. Thus, a truck tends to choose the nearby edge for the next destination. In comparison, a drone can escape the tree edge and fly to the others directly. Given this fact, the drone may choose the next edges relatively randomly. Thus, drones are not bound by the neighboring edge and we call this feature as 'non-hierarchy'.

As for the numerical result, the drone in the first case costs for 869 while the truck costs for 950, thus decreasing about 9% of the total travel cost. In addition, the total travel cost of the truck is 1,902 while the drone has 1,476 in the second case. This results in 22% of reduction of travel cost compared to the truck. These results indicate that the drones' traveling pattern has a direct impact on minimizing the deadheading cost.

Figure 6. (a) Truck route optimization in star shape

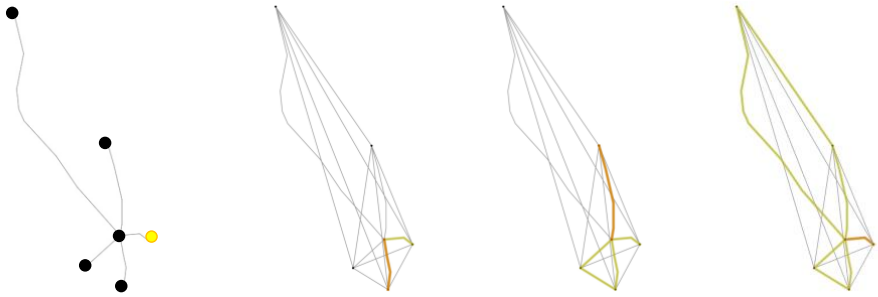

Figure 6. (b) Drone route optimization in star shape

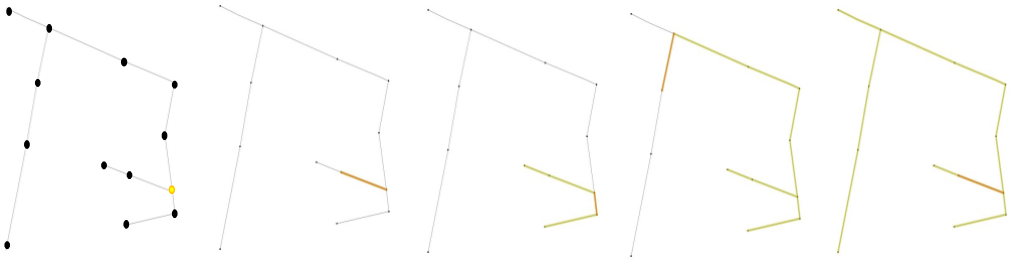

Figure 7. (a) Truck route optimization in tree shape

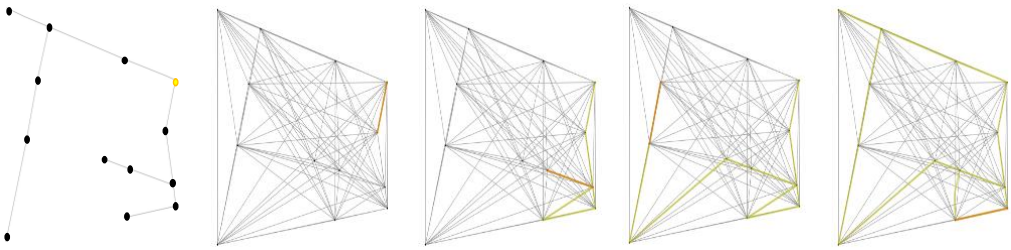

Figure 7. (b) Drone route optimization in tree shape

5.2. Application in Real-world Networks

The efficiency of snow removal using a drone compared to a truck is assessed based on real-world networks with different demand levels and network topologies each other. Instances should not be too artificial or too homogeneous, while covering the wide range of characteristics found in real applications. Thus, several large-sized real networks are explored. The size of these network varies from 1.6km^2 to 4km^2. Each instances have different road topologies, for example, Manhattan has a rectangular shape while central Paris has a circular shape. The results of optimization are presented in Table 3.

TABLE 3 Optimization Results

Network Case	Vehicle Type	#Num. (N, E, E^R)	Minimum Total Travel Distance	Median Total Travel Distance	Improvement
Boston	T	35, 45, 45	4444	4813	6.77%
	D	35, 595, 45	4143	4408	
New York	T	124, 213, 213	55423	57993	7.38%
	D	124, 7626, 213	51332	53716	
Paris	T	185, 268, 268	62591	66156	7.56%
	D	185, 17020, 268	57857	60124	
Incheon	T	266, 331, 331	319418	347322	22.03%
	D	266, 35345, 331	249041	257628	
Vancouver	T	274, 438, 438	170613	177953	8.47%
	D	274, 37401, 438	156155	163149	
Washington D.C.	T	336, 561, 561	163086	165897	7.05%
	D	336, 56280, 561	151596	155286	
Seoul	T	340, 507, 507	134493	144686	5.18%
	D	340, 57630, 507	127528	129293	

Since the heuristic has a random nature, each test set was conducted for 10 times. The results are reported with the minimum total travel cost and the median value. Both median value and minimum value of drone were generally lower than the truck. The improvement is checked as Equation 12 for comparison. The improvement of drone's route is measured based on the truck's result.

$$\boldsymbol{Improvement} = \frac{\textit{Min total travel distance of truck} - \textit{Min total travel distance of drone}}{\textit{Min total travel distance of truck}} \quad (12)$$

Consequently, a drone optimized the path with reduction of 5.18% to 22.03% among the instances. Especially, ‘Incheon’ case shows outstanding result of 22.03% reduction. In ‘Incheon’ case, the network topology was different from the others. In detail, the roads consisted of long arcs with less nodes, and the middle of the network was empty like donuts. This is because Incheon International Airport made a giant hole in the network as shown in **Figure 8**.

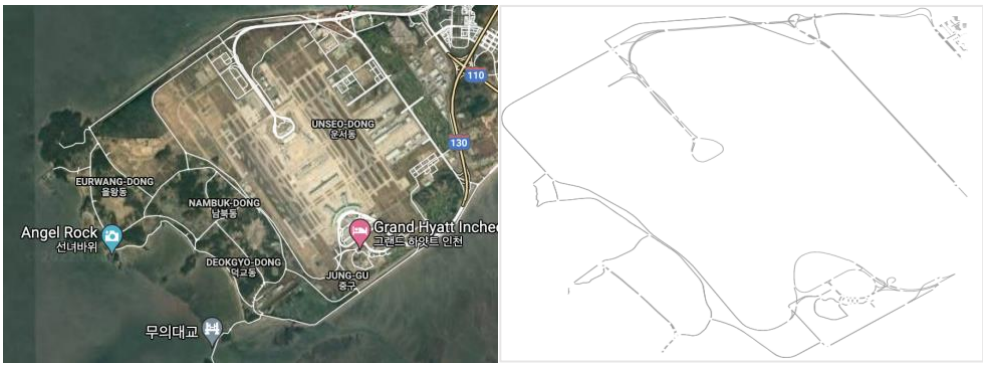

Figure 8. Topology of ‘Incheon’ case

This probably generated the longer deadhead to the truck but consequently this topology was beneficial for the drone. Based on this result, adoption of drone is expected to reduce the snow removal operation time and save costs in Incheon International Airport. It is recommended to use drones in urgent snow removal operation in such unique topology condition in practical.

The overall results present that a drone generally provide a shorter distance compared to a truck by minimizing the deadhead effectively. Based on the results, it is conformed that introducing a drone in snow removal reduces the operational cost and effective for fast travel. In addition, we can observe the drone’s flight pattern of non-hierarchy again. In many cases in the experiment sets, we confirmed that the drone's flight is not limited to prioritizing the neighboring arcs. This non-hierarchical feature is useful when adopting a drone synchronization with a ground vehicle.

5.3. Application of the refill constraint

In this part, we experiment a case study in Seoul with the refill constraint model. We targeted a district of the city where the roads are complicated. It was in Gangnam−Gu, Seoul, with a radius of 10km. The number of candidate salt containers in the area is 6.

As a result of the first phase, the total length of SNT, T is 16,623m. Therefore, the minimum refill number, N is calculated as $\frac{16623}{5000} = 3.32$, so 3 refill points is required. Therefore, after setting 3 initial random walks in the second phase, candidate routes were extracted through random walks. **Figure 9** shows the result of random walk.

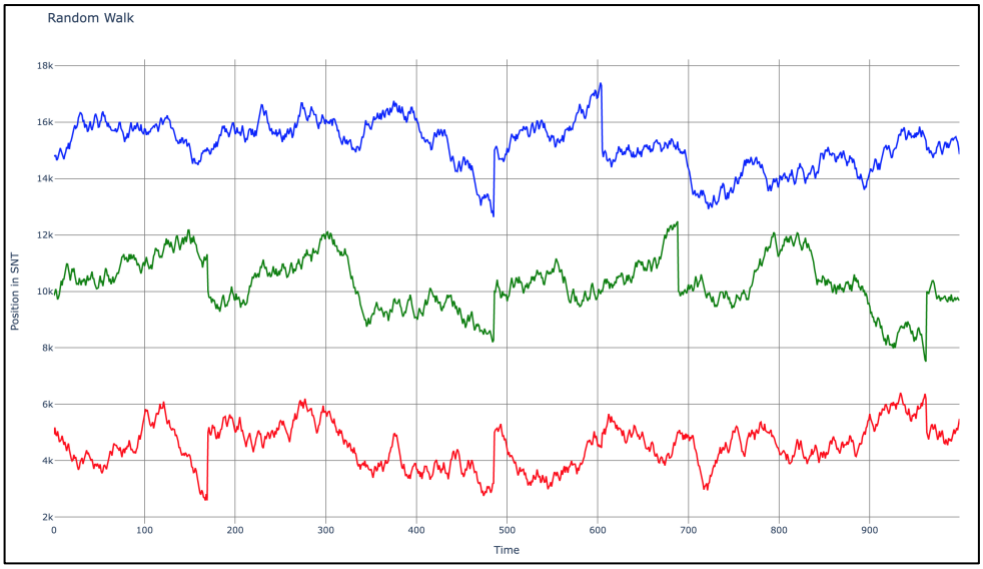

Figure 9. Result of random walk

Horizontal axis in the figure means total iteration time that means the random walk generated 1,000 route combinations. Vertical axis in the figure means the position in the SNT. In other word, there are 3 initial positions from the initial seed and the routes can move their positions as random walk.

To check the feasibility of these routes, we examined each array of arcs whether they fulfill the capacity constraint. The number

of feasible routes were 35 out of 1,000. After this, each feasible route's total travel distance is calculated by Equation 11.

As a result, the route with the smallest cost could be selected. Referring to **Figure 10**, the green dot is node, and the red dot is the location of the salt container, and the blue dot is the refill point derived from the experimental results. **Table 4** shows the results of the case study of Gangnam district with refill constraint model.

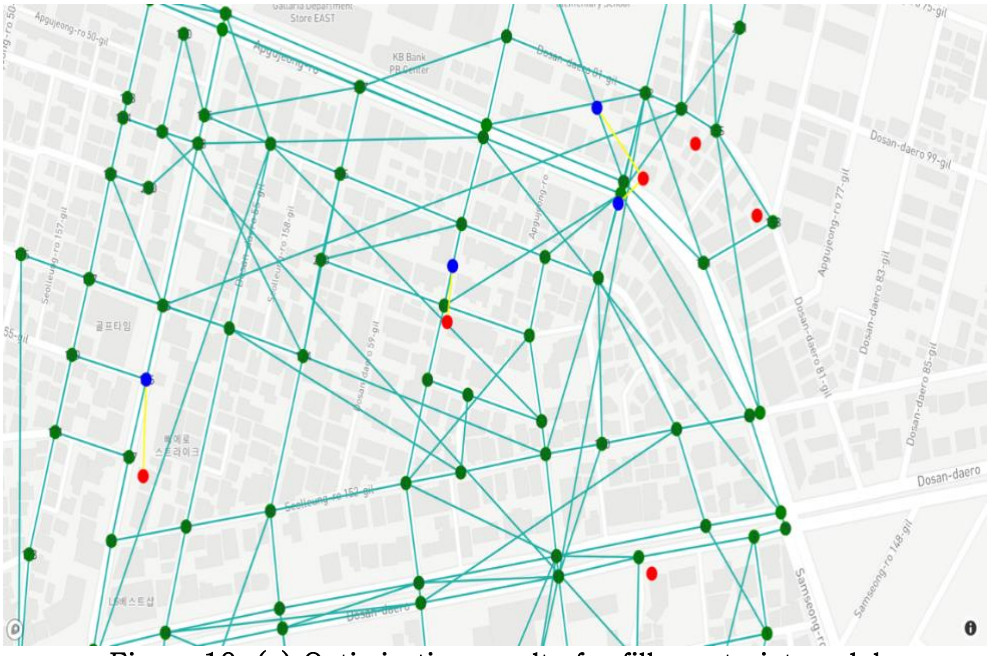

Figure 10. (a) Optimization result of refill constraint model

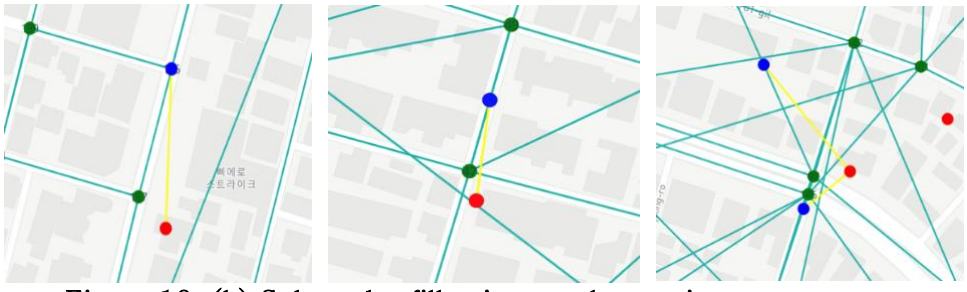

Figure 10. (b) Selected refill points on the continuous arc space

TABLE 4 Optimization Results of Gangnam district with refill constraint

Vehicle	Node	Required Edge	Edge	SNT length	Required edge distance	Deadhead distance	Total length with refill constraint
Drone	73	105	2628	16623	8865	7758	16925
Truck	73	105	105	17676	8865	8811	17676

The results from the Table 4 were also measured by Equation 12. It shows that the total length of SNT of drone was 6% shorter than the truck. A row of 'required edge distance' and 'deadhead distance' in the table means that the summation of required edges and summation of deadhead route as the optimization result respectively. Of course, the 'required edge distance' is fixed cost base on the graph, whereas 'deadhead distance' is variable cost depending on the optimization result. To sum up, a drone path minimized the deadhead distance after the optimization. In this regard, the drone makes 12% reduction in terms of deadhead compared to the truck. In case of refill constraint, drone has the total travel distance of 16,925 while truck has 17,676. As a result, drone has 4% reduction in terms of total travel distance. Consequently, it was found that the drone route is more efficient than a truck even under the refill conditions.

Chapter 6. Conclusion

This study conducted D-ARP that a drone travels the required edges for snow removal while minimizing the deadhead distance. A metaheuristic algorithm ACO was adopted to find the SNTs in the routing problem. The auxiliary transformation method was developed to reduce the search space and ease the algorithmic design without performance loss in ACO. Also, the post process improved the direction of arcs with local optimization. For the snow refill constraint, the problem was formulated as a length constrained model. Random walk method was adopted to implement the continuous search space of arcs.

The efficiency of snow removal using a drone is assessed based on real-world networks with different demand levels and network topologies each other. The results found that the drone travels a shorter distance compared to the truck with a reduction of approximately 5% to 22% among the instances. It was observed that the drone's flight was not limited to prioritizing the neighboring arcs. Furthermore, under the length constraint with refill point in the case study, the drone shows 4% reduction compared to the truck's travel. This means that the drone has more efficient route than a truck considering its constraint. Finally, the result of the test sets demonstrated that the modified algorithm performs well in the large size networks in reasonable time.

For the future work, a synchronized model of trucks and drones for snow removal considering the road priority would be interesting work. Also, implementing a post disaster routing problem that drones follow the roads to inspect disaster victims and deliver aid immediately would be helpful in disaster management.

References

1. Macrina, G., L. D. P. Pugliese, F. Guerriero, and G. Laporte. Drone-aided Routing: A Literature Review. Transportation Research. Part C, Emerging Technologies, 2020. 120: 102762.
2. Corberán, Á., and C. Prins. Recent Results on Arc Routing Problems: An Annotated Bibliography. Networks, 2010. 56(1): 50–69.
3. Corberán, Á., R. Eglese, G. Hasle, I. Plana, and J. M. Sanchis. Arc Routing Problems: A Review of the Past, Present, and Future. Networks, 2021. 77(1): 88–115.
4. Kim, M., J. Ahn and J. Park, Global Planning Method for Visiting Roads with Parking Spaces in Priority using Rural Postman Problem. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019. doi: 10.1109/ITSC.2019.8917318.
5. Campbell, J. F., Á. Corberán, I. Plana, and J. M. Sanchis. Drone Arc Routing Problems. Networks, 2018. 72(4): 543-59.
6. Edmonds, J., and E. L. Johnson. Matching, Euler Tours and the Chinese Postman. Mathematical Programming, 1973. 5(1): 88-124.
7. Orloff, C. S. A Fundamental Problem in Vehicle Routing. Networks, 1974. 4(1): 35–64.
8. Lenstra, J. K., and A. H. G. R. Kan. On General Routing Problems. Networks, 1976. 6(3): 273-80.
9. Christofides, N., V. Campos, A. Corberán, and E. Mota. An algorithm for the rural postman problem. Imperial College Report IC-OR, 1981.
10. Pearn, W.L., and T.C. Wu. Algorithms for the Rural Postman Problem. Computers & Operations Research, 1995. 22(8): 819-28.
11. Pearn, WL., A. Assad, and Golden, B. L. Transforming Arc Routing into Node Routing Problems. Computers & Operations Research, 1987. 14(4): 285-288.
12. Laporte, G. Modeling and Solving Several Classes of Arc Routing Problems as Traveling Salesman Problems. Computers & Operations Research, 1997, 24(11): 1057–1061.
13. Baldacci, R., and V. Maniezzo. Exact methods based on node-routing formulations for undirected arc-routing problems. Networks, 2006. 47(1): 52–60.

14. Longo, H., M. P. De Aragão, and E. Uchoa. Solving Capacitated Arc Routing Problems Using a Transformation to the CVRP. Computers & Operations Research, 2006. 33(6): 1823-1837.
15. Foulds, L., H. Longo, and J. Martins. A Compact Transformation of Arc Routing Problems into Node Routing Problems. Annals of Operations Research, 2015. 226(1): 177-200.
16. Letchford, A. N., and A. Oukil. Exploiting Sparsity in Pricing Routines for the Capacitated Arc Routing Problem. Computers & Operations Research, 2009. 36(7): 2320-2327.
17. Chung, S. H., B. Sah, and J. Lee. Optimization for Drone and Drone-truck Combined Operations: A Review of the State of the Art and Future Directions. Computers & Operations Research, 2020. 123: 105004.
18. Easton, K., and J. Burdick. Coverage algorithm for multi-robot boundary inspection. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. 2005: 727-734.
19. Sipahioglu, A., G. Kirlik, O. Parlaktuna, and A. Yazici. Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach. Robotics and Autonomous Systems, 2010. 58(5): 529-538.
20. Oh, H., S. Kim, A. Tsourdos, and B. A. White. Coordinated Road-network Search Route Planning by a Team of UAVs. International Journal of Systems Science, 2014. 45(5): 825-840.
21. Chow, J. Y. J. Dynamic UAV-based traffic monitoring as a stochastic arc-inventory routing policy, International Journal of Transportation Science and Technology, 2016. 5(3): 167–185.
22. Li, M., L. Zhen, S. Wang, W. Lv, and X. Qu. Unmanned aerial vehicle scheduling problem for traffic monitoring. Computers & Industrial Engineering, 2018. 122: 15-23.
23. Liu, Y., J. Shi, Z. Liu, J. Huang, and T. Zhou. Two-layer routing for high-voltage powerline inspection by cooperated ground vehicle and drone. Energies (Basel), 2019. 12(7): 1385.
24. Dille, M., and S. Singh, Efficient aerial coverage search in road networks. AIAA Guidance, Navigation, and Control (GNC) Conference, 2013. https://doi.org/10.2514/6.2013-5094.
25. Oruc, B. E., and B. Y. Kara. Post-disaster Assessment Routing Problem. Transportation Research. Part B, Methodological, 2018. 116: 76-102.
26. Singgih, I. K., J. Lee, and BI. Kim. Node and edge surveillance using drones considering required camera altitude and battery charging,

Advances in Production Management Systems. Smart Manufacturing for Industry 4.0., 2018. 536: 172−180.

27. Ulusoy, G. The Fleet Size and Mix Problem for Capacitated Arc Routing. European Journal of Operational Research, 1985. 22(3): 329–337.
28. Campbell, J. F., Á. Corberán, I. Plana, J. M. Sanchis, and P. Segura. Solving the Length Constrained K−drones Rural Postman Problem. European Journal of Operational Research, 2021. 292(1): 60−72.
29. Murray, C. C., and A. G. Chu. The Flying Sidekick Traveling Salesman Problem: Optimization of Drone−Assisted Parcel Delivery. Transportation Research. Part C, Emerging Technologies, 2015. 54: 86–109.
30. Poikonen, S., X. Wang, B. Golden. The vehicle routing problem with drones: Extended models and connections. Networks, 2017. 70(1): 34–43.
31. Shivgan, R., and Z. Dong, Energy−Efficient Drone Coverage Path Planning using Genetic Algorithm. 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), 2020. doi: 10.1109/HPSR48589.2020.9098989.
32. Corberán, A., I. Plana, and J. M. Sanchis. A Branch & Cut Algorithm for the Windy General Routing Problem and Special Cases. Networks, 2007. 49(4): 245−57.
33. Lacomme, P., C. Prins, and W. Ramdane−Cherif. Competitive Memetic Algorithms for Arc Routing Problems. Annals of operations research, 2004. 131(1): 159–185.
34. Lacomme, P., C. Prins, and M. Sevaux. A Genetic Algorithm for a Bi−objective Capacitated Arc Routing Problem. Computers & Operations Research, 2006. 33(12): 3473−3493.
35. Zhu, Z., X. Li, Y. Yang, X. Deng, M. Xia, Z. Xie, and J. Liu. A Hybrid Genetic Algorithm for the Multiple Depot Capacitated Arc Routing Problem. 2007 IEEE International Conference on Automation and Logistics, 2007. doi: 10.1109/ICAL.2007.4338951.
36. Santos, L., J. Coutinho−Rodrigues, and J. R. Current. An improved ant colony optimization based algorithm for the capacitated arc routing problem. Transportation research. Part B, methodological, 2010. 44(2): 246–266.
37. Tirkolaee, E. B., M. Alinaghian, A. A. R. Hosseinabadi, M. B. Sasi, and A. K. Sangaiah. An Improved Ant Colony Optimization for the Multi−trip

Capacitated Arc Routing Problem. Computers & Electrical Engineering, 2019. 77: 457-470.

38. Huang, SH., and TH. Lin. Using Ant Colony Optimization to Solve Periodic Arc Routing Problem with Refill Points. Journal of Industrial and Production Engineering, 2014. 31(7): 441-451.
39. Dorigo, M., and T. Stützle. Ant Colony Optimization Algorithms for the Traveling Salesman Problem. In Ant Colony Optimization, MIT Press, 2004. p. 65-119.
40. Rizzoli, A. E., R. Montemanni, E. Lucibello, and L. M. Gambardella. Ant Colony Optimization for Real-world Vehicle Routing Problems. Swarm Intelligence, 2007. 1(2): 135-151.
41. Dorigo, M., G. Di Caro, and L. M. Gambardella. Ant Algorithms for Discrete Optimization. Artificial life, 1999. 5(2): 137–172.
42. Dong, L. W., and C. T. Xiang. Ant Colony Optimization for VRP and Mail Delivery Problems. 2006 4th IEEE International Conference on Industrial Informatics, 2006. doi: 10.1109/INDIN.2006.275779.
43. Salazar-Aguila, M. A., A. Langevin, and G. Laporte, Synchronized arc routing for snow plowing operations, Computers & Operations Research, 2012, 39(7): 1432-1440, https://doi.org/10.1016/j.cor.2011.08.014.
44. Crevier, B., J. F. Cordeau, and G. Laporte, The multi-depot vehicle routing problem with inter-depot routes, European Journal of Operational Research, 2007, 176(2): 756-773.
45. Kim, G. Y., The establishment of tailored snow removal correspondence system considering regional characteristics, Seoul Ministry of Public Administration and Security, 2017. p. 232.
46. Winter snow removal management plan for safety from snow, Seoul metropolitan government of safety, 2020. https://opengov.seoul.go.kr.
47. DJI drones, https://www.dji.com/kr/.
48. An Analysis of the Operational Costs of Trucking, American Transportation Research Institute, 2020. https://truckingresearch.org/.
49. Spitzer, F. Principles of Random Walk, New York: Springer-Verlag, 1976. p. 1-3.
50. Gregory F. L. and V. Limic, Random walk: A modern introduction, Cambridge University Press, 2010.

국문초록

드론은 혼잡시간대를 대체하고 빠른 서비스를 가능하게 함으로써 지상차량의 한계를 극복할 수 있다. 최근 기후변화에 따른 갑작스런 강설의 경우에, 드론과 같이 빠르게 투입할 수 있는 서비스는 운행 경로와 노동비용을 고려했을 때도 유연한 운영 옵션이 될 수 있다. 본 연구의 목적은 드론 아크 라우팅(D-ARP)을 최적화하는 것이며, 이는 제설에 필요한 도로를 서비스하는 경로를 탐색하는 것이다. 드론 아크 라우팅은 특히 큰 네트워크에서 컴퓨터 부하를 생성한다. 다시 말해 D-ARP 는 큰 검색공간을 필요로 하며, 이는 기하급수적으로 증가하는 후보 경로 및 호의 방향 결정 그리고 연속적인 호의 공간으로부터 기인한다. 검색공간을 줄이기 위해, 우리는 개미알고리즘에 보조변환방법을 적용하는 방안을 도입하였으며 또한 랜덤워크 기법을 채택하였다. 본 연구의 기여는 제설 운영에 있어 D-ARP 라는 새로운 문제를 설정하고 최적화 접근법을 도입하였으며 검색공간을 최소화한 것이다. 최적화 결과, 드론은 지상트럭에 비해 약 5% ~ 22%의 경로 비용 감소를 보였다. 나아가 길이 제약 모델에서도 드론은 4%의 비용 감소를 보였다. 또한 실험결과는 적용한 휴리스틱 알고리즘이 큰 네트워크에서도 합리적 시간 내에 최적해를 찾음을 입증하였다. 이러한 결과를 바탕으로, 드론을 제설에 도입하는 것은 미래에 제설 운영 비용을 실질적으로 감소시킬 것으로 기대된다.

주요어 : 드론 아크 라우팅 (D-ARP), 제설, 개미알고리즘 (ACO), 길이제약, 랜덤워크

학 번 : 2016-25408

Acknowledgment

Thank you to everyone who helped me return to graduate school and complete my master's program. First of all, I would like to thank my mentors, Professor Dong-Kyu Kim and Professor Seung-Young Kho, who gave me the opportunity to start studying again and guided me. They gave good lectures and taught a lot even in a difficult environment due to the covid-19. Also, thank you for giving me the courage to start studying again, team professor Lee Kyu-Jin and team leader Hwang Byeong-Yong, giving me the chance to participate in the class. And, my dear family, my parents, my sister, and Dong-ju, thank you for always supporting me. I would like to thank all the seniors and juniors in the Transportation Lab who helped me a lot in my graduate school life, and I wish you all the best. Lastly, I will continue to strive to be a person who can act with good motives in the future. Thank you.