

UAM과 드론을 사용한 도시물류상의 선택적 이중구조 오리엔티어링 문제

김동균, 차형주, 정태수[†]

고려대학교 산업경영공학과

Selective Two-echelon Orienteering Problem in urban logistics using UAM and Drones

Dong Kyun Kim, Hyungjoo Cha, Taesu Cheong[†]

School of Industrial Management Engineering, Korea University

In recent days, the Urban Air Mobility (UAM) is emerging as the new transportation after drone. In this study, UAM and drones are used for an on-demand logistics service. Since UAM is considered for delivery, the consideration of vertiports and vertistops (infrastructure for UAM) is inevitable. In this problem, the UAM only visits vertiports and vertistops for the delivery. After the UAM's delivery, at vertiports drone deliver the package to the customer. At vertistops, customers visit vertistop to pick up the package. We present a mathematical model for routing problem using UAM and drones. Since this problem is a general case of Traveling Salesman Problem (TSP), it is hard to get an optimal solution if the size of problem gets bigger. Thus we develop a two-phase Variable Neighborhood Search based heuristic algorithm for large-size problems.

Keywords: Urban Air Mobility, Orienteering Problem, MILP, Variable Neighborhood Search, Electric drone

논문접수일 : 2022.10.08.

심사완료일 : 2022.12.16.

게재확정일 : 2022.12.19.

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No.2021R1F1A1059302)이며, 또한 정부(산업통상자원부)의 재원으로 한국에너지기술평가원 지원을 받아 수행된 연구 (No.20229400000090)임.

[†] Corresponding Author: tcheong@korea.ac.kr

1. 서론

수도권의 인구 집중으로 인한 차량의 증가는 오랫동안 도심의 심각한 교통체증의 원인이 되어 왔다. 이러한 도시집중화로 인한 교통 혼잡으로 많은 사회간접비용이 발생하고 있다. 한국교통연구원의 연구에 따르면 교통혼잡비용은 2010년 28.5조 원에서 2015년 33.3조 원까지 지속적으로 증가하였으며 2016년, 2017년 교통혼잡비용은 각각 36.3조 원, 38.7조 원으로 꾸준히 증가하는 모습을 보였다 (Cheon et al., 2020). 교통 혼잡으로 인한 문제는 COVID-19로 인한 비대면 배송 수요의 증가와 증가한 1인 가구로 인한 배송 건수 증가와 맞물려 배송업계에도 큰 문제가 되고 있다. 하지만 기술의 발달로 항공 운송수단 운용 가능성이 높아지며 이러한 문제를 해결할 수 있는 Urban Air Mobility (UAM)가 떠오르고 있다. UAM은 운송수단이라기 보다는 하나의 시스템을 의미한다. 현재의 도심에 수직이착륙이 가능한 비행체 (electric Vertical Take Off and Landing, eVTOL)가 새로운 이동수단으로 추가되어 기존의 운송수단과 협력하며, 원활한 이동을 가능하게 하는 시스템이다. 또한, 도심에서 수직이착륙이 가능한 개인용 비행체 (Personal Air Vehicle, PAV)로 이동하는 공중 교통 체계를 의미하기도 한다. UAM에 주로 사용되는 공중 기반의 이동수단인 eVTOL은 보통 사람이 탈 수 있을 정도로 큰 드론의 형태이다. 기존 수직이착륙이 가능한 공중 이동수단인 헬리콥터를 이용할 수도 있겠지만, 헬리콥터는 소음과 관련된 문제가 많아 고려되지 않고 있다 (Jeong et al., 2021).

UAM이 적용되기 위해서는 실제로 사용될 eVTOL과 기반 시설이 필수적으로 고려되어야 한다. 드론의 기술적인 발전에 힘입어 사람이 탑승할 수 있는 큰 드론이 UAM에 활용될 eVTOL로 고려되고 있다. 해외의 경우 대표적으로 UAM의 도입과 이를 위한 eVTOL의 개발에 앞장서고 있는 기업은 미국의 Uber 사이다. 국내에서는 현대모터그룹에서 미국의 Uber 사와 협업하여 2028년 UAM 상용화를 목표로 하고 있다. 최근에는 ‘CES2020’에서 UAM 사업의 첫 비전으로 콘셉트 모델인 ‘S-A1’을 제시하는 등 UAM의 도입을 준비하며 이를 보다 구체화하고 있다 (Hyundai, 2022). 아래 Figure 1은 현대모터그룹에서 제시한 콘셉트 모델 ‘S-A1’의 사진이다.

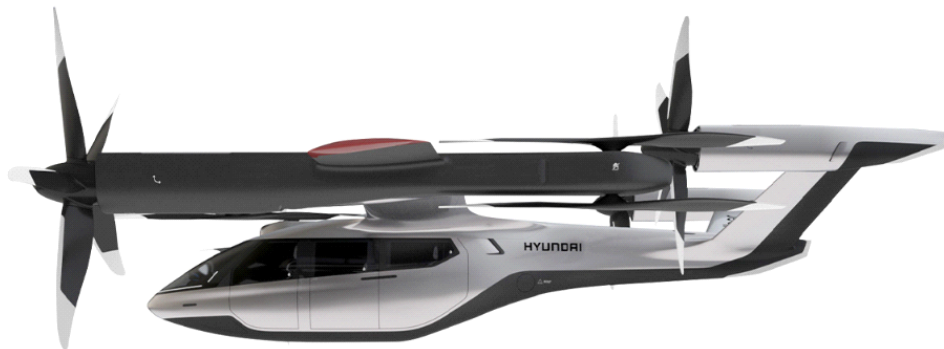


Figure 1. Hyundai Motor Group's UAM vision concept, 'S-A1'

다음으로 UAM의 도입을 위해서는 기반 시설이 필요하다. Fadhil(2018)에서는 이러한 기반 시설을 vertiport (혹은 skyport)와 vertistop으로 구분하였다. Vertiport와 Vertistop은 두 가지 모두 UAM의 수직 이착륙을 위한 station의 개념이라는 점은 동일하다. 둘의 가장 큰 차이점은 vertistop의 경우 UAM의 이착륙과 승객의 승차하만 가능하지만, vertiport는 UAM의 이착륙과 승객의 승차하와 더불어, UAM의 충전, 다른 모빌리티로의 환승 등이 가능한 multi-hub의 역할을 할 수 있다는 점이 가장 큰 차이점이다. Vertiport와 Vertistop 이 둘은 그 역할 및 수행 가능한 기능에도 차이가 있지만, 크기와 규모에도 차이가 있으므로 이를 구분하여 고려하는 것이 중요하다.

UAM의 운영을 위해 기반 시설이 필수적이기 때문에 UAM 단독으로 물품의 배송을 진행하기에는 어려움이 따른다. 따라서 본 연구에서는 UAM과 함께 드론을 사용하여 UAM과 드론의 협업 배송모형을 제안한다. UAM은 배송이 가능한 여러 개의 배송지 (delivery center)를 모두 방문하지 않고 이를 선택적으로 방문하여 중 선택된 임의의 배송지를 방문하며 배송을 진행하고 출발지로 돌아온다. 이때 배송지는 UAM의 특성을 고려하여 vertiport와 vertistop으로 나누어진다. 각 vertiport와 vertistop으로 물품의 배송이 완료되면 배송지의 종류에 따라 서로 다른 방식으로 고객에게 배송을 시작한다. Vertiport의 경우 multi-hub의 역할을 할 수 있어 다른 모빌리티의 운용이 용이하기에 드론을 활용하여 배정된 고객에게 배송을 진행한다. 반면, vertistop의 경우 UAM의 이착륙과 승객의 승하차 정도만 가능한 시설이기 때문에 다른 모빌리티의 운용, 접근 등이 어렵다. 이에 다른 모빌리티를 활용한 배송이 적합하지 않다. 대신 해당 vertistop에 존재하는 보관함에 물품을 보관하고, 해당 vertistop에 배정된 고객이 직접 방문하여 물품을 가져가는 형태로 배송이 진행된다. 본 논문에는 이러한 형태의 배송 모델인 Two-echelon Orienteering Problem with UAM을 제안한다.

제안 모델은 UAM의 이동시간과 실제로 고객이 물품을 받는 시간 모두를 최소화하는 것을 목표로 한다. 각 고객은 배송을 받기를 원하는 시간이 있다. 만약 이 시간에 맞춰서 배송하지 못한다면 고객이 물품을 받은 시간과 원하는 배송시간의 차이만큼의 페널티가 주어지게 된다. 결국 제안 모델에서는 UAM의 이동시간, 실제로 고객이 물품을 받는 시간과 고객이 원하는 시간의 차이, 고객에게 물품 배송이 완료되는 시간 모두를 최소화하는 것을 목적으로 하게 된다.

본 논문에서는 제안 모델에 대한 혼합정수계획법 기반의 수리최적화모형을 제안하고 이와 더불어 Variable Neighborhood Search (VNS) 기반의 2단계의 메타-휴리스틱 알고리즘을 제안한다. 기존 연구들의 대부분은 지상 전기차와 전기 드론의 협업 배송의 형태이다. 차량의 경우 도로를 따라 움직이기 때문에 이동 경로에 대한 제약이 있고 신호대기 시간이나 교통 체증에 빈번하게 노출된다. 반면 UAM은 이동경로가 비교적 자유로우며 신호대기 시간이나 교통 체증의 영향을 받지 않는다. 또한, 일반적인 차량보다 훨씬 빠른 속도로 이동이 가능하기 때문에 본 연구에서 사용되는 UAM이 보다 빠르고 효율적인 배송을 진행할 수 있을 것으로 기대된다.

본 연구의 주요 기여점은 다음과 같다.

첫째, UAM과 전기 드론의 협업 배송을 진행하는 모델인 제안 모델을 처음 제시한다. 제안 모델은 UAM이 방문하는 배송지로 UAM 운용을 위한 기반 시설인 vertiport와 vertistop을 다르게 고려하며, 배송지의 종류에 따라 고객까지의 배송이 다르게 이루어진다.

둘째, 제안 모델에서는 배송을 진행할 때 UAM의 이동 시간과, 각 고객이 배송 받기를 원하는 시간과 실제 배송 받은 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 경로를 찾는다. 이를 통해 배송 시간과 더불어 고객이 느끼는 서비스의 만족도까지 고려하는 모델을 제안한다.

셋째, 제안 모델을 풀기 위한 혼합정수계획법 기반의 수리최적화모형과 동시에 VNS 기반의 2 단계 휴리스틱 알고리즘을 제안한다.

본 논문은 다음과 같은 순서로 구성되어 있다. 2장에서는 드론이 사용된 협업 배송과 UAM에 대한 연구의 흐름을 알아본다. 3장에서는 제안 모델에 대한 정의와 이 문제를 해결하기 위한 혼합정수계획법 기반의 수리최적화모형, VNS 기반의 2단계 휴리스틱 알고리즘을 알아본다. 4장에서는 실험을 통해 수리최적화모형과 휴리스틱 알고리즘의 성능을 평가한다. 마지막으로 5장에서는 본 논문의 결론을 제시한다.

2. 관련 연구

UAM의 개념이 처음 등장한 이후, 관련 연구는 꾸준히 진행되고 있다. 국내에서는 Choi et al.(2021)이 UAM의 국내 도입을 위해 국제표준을 고려한 vertiport의 설치기준을 마련하였다. 또한, Jeong et al.(2021)은 K-Means Algorithm을 활용하여 vertiport의 위치를 선정하고, Aviation Environmental Design Tool (AEDT)을 사용해 소음을 분석하여 COEX부터 김포공항까지의 최저소음 경로를 찾는 연구를 진행하였다. 해외에서도 다양한 연구가 진행되었는데 Wu and Zhang(2021)은 UAM과 다양한 이동수단의 연계를 고려한 수리최적화모형을 제안하였다. 이동수단별로 각기 다른 이동시간과 비용을 고려하였으며, vertiport에 진입 가능한지 여부도 구분하였다. Bennaceur et al.(2022)은 승객을 중심으로 접근하였다. UAM을 운영할 때, 승객들의 대기 시간을 최소화할 수 있는 방법을 수리최적화모형과 함께 제안하였으며, 경로생성과 재충전 방안을 동시에 제안해 운영비 감소와 배터리 수명 증가의 효과도 얻었다. Shin et al.(2022)의 연구는 차량과 UAM의 연계와 UAM의 공중 충돌위험을 고려한 수리최적화모형과 유전 알고리즘 (Genetic Algorithm) 기반의 휴리스틱을 함께 제시하였다. 대부분의 UAM을 활용한 경로 문제는 고객들의 운송을 가정하기 때문에 UAM과 함께 활용되는 이동수단은 대부분 차량이나 대중교통이었다. 하지만 본 연구에서는 UAM을 활용한 화물의 배송을 고려하며 이에 따라 기존 연구에서는 함께 활용되지 않았던 드론을 사용한다.

드론을 협업 배송에 고려하는 대표적인 연구는 Murray and Chu(2015)의 단일 차량과 드론의 협업배송 문제이다. 해당 연구에서는 flying sidekick traveling salesman problem (FSTSP)과 parallel drone scheduling traveling salesman problem (PDSTSP)을 제안한다. FSTSP는 단일 차량을 활용하는 경로 문제인 Traveling Salesman Problem (TSP)에 드론을 보조 배송 수단으로 활용한다. 차량은 드론을 탑재하고 배송을 진행하여 특정 노드에서 드론이 차량을 떠나 배송을 수행한 뒤 차량을 떠난 노드가 아닌 다른 노드에서 차량과 합류하는 형태의 배송 방식을 취한다. PDSTSP에서는 차량은 드론이 배송하지 않는 노드들에 대해 배송을 진행하고 드론은 출발지에서 일정 거리 이내의 노드에 대해 배송을 진행한다. Kim and Moon(2018)의 연구는 이러한 PDSTSP의 일반적인 형태인 traveling salesman problem with a drone station (TSP-DS)을 제안한다. 해당 연구에서는 드론이 출발지에서 배송을 진행하는 것이 아니라 특정 노드에 설치한 드론 스테이션에서 배송을 진행한다. 차량이 드론이 배송하지 않는 노드들의 배송을 진행하며 드론 스테이션을 방문하면 드론 스테이션에서 드론이 배송을 시작한다. 차량은 드론 스테이션 방문 후 대기하지 않고 이후 다른 노드들의 배송을 진행하며 출발지로 돌아오는 형태이다. 본 연구는 PDSTSP나 TSP-DS에서 차량의 역할을 UAM이 맡는 형식으로 볼 수 있지만, 해당 연구에서는 그 동안 고려하지 않았던 중간 노드를 고려한다는 점에서 차이점이 존재한다.

지금까지 살펴본 연구들은 모두 TSP의 활용 형태로 주어진 모든 노드를 방문하는 경로를 생성하며, 협업 배송을 진행하는 경우에는 서로 다른 이동수단이 하나의 노드에서 동시에 만나서 이동하는 것을 고려한다. 하지만 본 연구는 주어진 모든 노드를 방문할 필요가 없다. 주어진 노드를 모두 방문하지 않아도 되는 경로를 생성하는 문제는 Location Routing Problem (LRP) 혹은 Orienteering Problem (OP)으로 볼 수 있다. 따라서 본 연구는 LRP, OP로 확장이 가능하다. LRP는 facility location problem (FLR)을 푸는 것을 주된 목적으로 하며 이를 위해 동시에 TSP 혹은 vehicle routing problem (VRP)을 풀게 된다 (Nagy and Salhi., 2007). FLR에서 각 시설에 할당된 고객들이 서비스를 받을 때 시설에서 경로를 이루며 고객에게 서비스를 제공한다면 LRP, 한 명의 고객에게 서비스를 제공하고 다시 시설로 돌아간 뒤 다음 고객에게 서비스를 제공하게 된다면 이는 location-allocation problem이 된다 (Hassanzadeh et al., 2009). 반면 OP의 경우 주어진 시간 혹은 거리에 맞는 경로를 형성하기 위해 노드를 선택하여 방문하게 되며 중간 노드로 고려되는 시설, 혹은 서비스의 제공 방식에 대한 제약은 존재하지 않는다. 본 연구에서는 고객이 경로를 이루어 서비스를 받지 않고 드론을 통해 왕복으로 서비스를 받거나 직접 vertistop을 방문하게 된다. 따라서 다양한 방식으로 서비스의 제공이 가능한, 보다 일반적인 경우를 가정하기 위해 OP 방식을 고려하여 문제를 해결하려고 한다.

본 연구는 Hong et al.(2019)의 연구를 확장 및 발전시킨 연구이다. 해당 연구는 차량을 활용한 on-demand 물류 서비스를 LRP로 풀어내었다. 본 연구에서는 해당 문제를 OP 방식으로 접근한다. OP는 Golden et al.(1987)에 의해 처음 소개되었으며, 앞서 설명하였던 것과 같이 주어진 모든 노드를 방문하는 경로가 아니라 주어진 노드의 일부만을 선택적으로 방문하여 제한 시간 혹은 거리를 넘기지 않는 경로를 찾는 것이 목적이다. 이러한 OP는 방문할 노드를 선택하는 문제인 Knapsack Problem과 선택된 노드들을 방문하는 경로를 생성하는 TSP로 나누어질 수 있다 (Gunawan et al., 2016). OP는 기존 하나의 경로를 찾는 문제에서 여러 개의 경로를 찾는 문제로 변경하여 Team Orienteering Problem (TOP)으로 확장 될 수 있다. Labadie et al.(2012)은 노드별로 특정 시간대에 서비스가 시작되게 하는 제약을 추가하여 OP를 OP with Time Windows (OPTW)로 확장하였다. Verbeeck et al.(2014)은 두 노드 사이의 이동시간이 첫 번째 노드의 출발 시간에 의존적인 문제를 Time Dependent OP (TDOP)로 정의하였다. 이후 TDOP는 노드간 이동시간의 불확실성을 고려한 Ilhan et al.(2008)의 OP with Stochastic Profits (OPSP)로 확장되었다. 노드간 이동시간이 아니라 노드에 부여되는 이익 혹은 점수에 차별점을 부여한 Generalized OP (GOP)도 제안되었다(Geem et al., 2005). 해당 연구에서는 각 노드에 하나의 점수가 부여되는 것이 아니라 특성에 따라 점수의 집합이 부여되고 선택된 노드의 점수 집합 전체의 합이 목적함수가 되는 차이점이 있다. 노드가 아니라 아크를 방문하는 형태로 OP를 고려한 Verbeeck et al.(2014)의 Arc OP (AOP), TOP의 발전 형태로 각 노드에 수요와 점수 값이 함께 부여되는 Archetti et al.(2009)의 Capacitated TOP (CTOP) 등 활발한 연구가 진행되고 있다.

노드간의 이동시간이나 노드에 부여되는 속성, 점수의 변화에 대한 발전이 대부분이었던 기존 연구와는 다르게 본 연구에서는 중간 노드의 추가를 고려한다. 또한, 앞서 언급하였던 Hong et al.(2019)의 연구와는 다르게 배송지의 종류가 vertiport와 vertistop으로 나누어지며, 배송지의 종류에 따라 고객까지의 배송을 다르게 진행한다.

3. 문제 정의

본 장에서는 제안 모델에 대한 정의와 혼합정수계획법 기반의 수리최적화모형을 제시하고, 이후 VNS 기반의 2단계 휴리스틱 알고리즘을 제시한다. 제안 모델은 UAM과 드론의 협업 배송모형이다. UAM은 배송 시작지점을 출발하여 물품을 보관할 수 있는 배송지에 방문하여 물품을 배송하고, 다시 배송 시작지점으로 돌아온다. 이때 UAM은 방문 가능한 배송지를 모두 방문할 필요가 없으며, 배송지를 선택적으로 방문하여 보다 효율적인 경로를 찾는다. 배송지는 UAM의 특성을 고려하여 vertiport와 vertistop으로 나누어지며, 물품을 저장할 수 있는 용량과 UAM의 배송 이후 고객에게 직접 물품을 배송하는 방식에 차이가 있다. UAM으로부터 물품을 받은 배송지가 vertiport라면 한 대의 드론을 활용하여 고객에게 직접 배송을 진행한다. 만약 vertistop이라면 고객들이 직접 vertistop으로 방문하여 보관되어있는 물품을 수령한다.

제안 모델의 해를 도출하는 순서는 다음과 같다. 먼저, 고객의 수요량과 배송지의 최대 수용량을 고려하여 배송지에 고객을 배정한다. 모든 고객이 끝나면 UAM은 고객이 배정된 배송지를 따라 경로를 형성하며 배송을 진행한다. 이후, 각 배송지는 특성에 맞게 고객들에게 물품을 제공한다.

제안 모델에서 고객들은 물품을 받을 때 선호하는 시간이 정해져 있다. 제안 모델은 고객들의 선호 시간 이전에 배송을 완료하는 것을 목표로 하지만 이를 실패한 경우에는 불가능해라고 판단하는 대신 일정량의 페널티를 받게 된다. 이를 통해 제안 모델은 UAM의 이동 거리, 고객이 배송을 원하는 시간과 실제 배송을 받은 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 경로를 찾을 수 있다. 아래 Figure 2는 UAM의 배송지의 선택적 방문에 대한 예시를, Figure 3은 vertiport와 vertistop에서의 다른 물품 배송 방식을 그림으로 나타낸다.

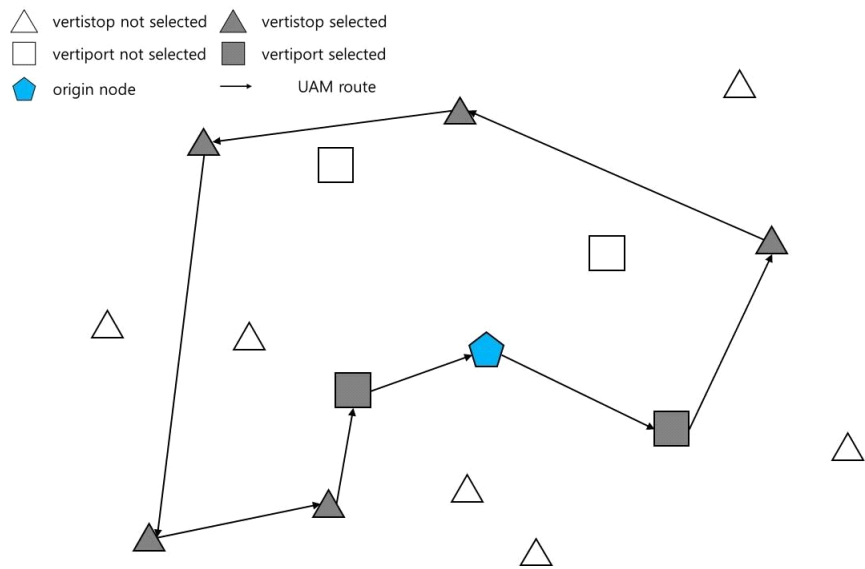
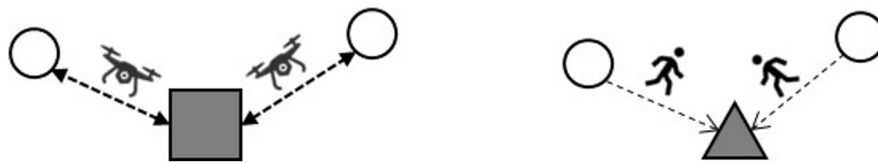


Figure 2. A schematic diagram of the UAM visiting the delivery centers selectively



(a) A drone delivers parcels to customers (b) Customers visiting the vertistop sequentially at vertiport to retrieve their parcels

Figure 3. Different delivery methods of vertiport and vertistop

3.1 집합과 파라미터

본 논문에서 제시하는 수리최적화모형에 사용되는 집합과 파라미터의 정의는 아래와 같다.

Table 1. Sets and Parameters

인덱스 집합 및 상수	
C	고객 인덱스의 집합
P	vertiport 인덱스의 집합
S	vertistop 인덱스의 집합
D	모든 배송지 인덱스의 집합 ($P \cup S$)
0	UAM 출발 지점의 인덱스 (디포, depot)
d_{jk}	배송지 j 와 k 사이의 거리 ($j, k \in D, j \neq k$)
l_{ij}	고객 i 와 배송지 j 사이의 거리 ($i \in C, j \in D$)
C_j	배송지 j 를 사용하는데 드는 비용 ($j \in D$)
p_i	고객 i 의 선호 배송 시간 ($i \in C$)
$demand_i$	고객 i 의 수요량 ($i \in C$)
R_{\max}	허용 가능한 고객과 배송지 사이의 최대 거리
v	UAM의 속도
v^d	Drone의 속도
v^p	고객이 걸어서 vertistop을 방문할 때의 속도
α	UAM 이동 거리에 대한 비용 가중치
β	고객의 선호 배송 시간과 실제 배송 시간의 차이에 대한 비용 가중치
γ	배송지 사용 비용에 대한 가중치
N_j	배송지 j 의 최대 수용량 ($j \in D$)

Table 2. Decision variables

결정변수	
x_{ij}	고객 i 의 제품이 배송지 j 로 배송되면 1 아니면 0의 값을 가지는 이진변수 ($i \in C, j \in D$)
y_j	배송지 j 가 사용되면 1 아니면 0의 값을 가지는 이진변수 ($j \in D$)
z_{jk}	UAM이 배송지 j 방문 직후 배송지 k 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ($j, k \in D$)
t_j	UAM이 배송지 j 에 도착하는 시간을 나타내는 연속변수 ($j \in D$)
g_i	고객 i 의 선호 배송 시간과 실제 배송 시간의 차이를 나타내는 연속변수 ($i \in C$)
t_{ij}	g_i 를 계산하기 위한 보조 결정변수인 연속변수 ($i \in C, j \in D$)
t'_{ij}	고객 i 를 방문하기 전 드론이 vertiport j 에 도착한 시간을 나타내는 연속변수 ($i \in C, j \in P$)
w_{ij}^k	Vertiport k 에 있는 드론이 고객 i 를 방문한 다음 고객 j 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ($i, j \in C, k \in P, i \neq j$)
w'_{ij}	드론이 고객 i 를 방문하고 vertiport j 를 방문하거나, vertiport j 를 방문하고 고객 i 를 방문하면 1 아니면 0의 값을 가지는 이진변수 ($i, j \in C \cup P$)
u_j	Depot을 시작으로 계산한 UAM의 배송지 j 의 방문순서를 나타내는 정수변수 ($u_j \in Z^+$)

3.2 수리최적화모형

아래는 본 논문에서 제안하는 제안 모델의 수리최적화모형이다.

$$\text{Minimize } \alpha \sum_{j \in D} \sum_{\substack{k \in D \\ k \neq j}} d_{jk} z_{jk} + \beta \sum_{i \in C} g_i + \gamma \sum_{j \in D} C_j y_j \quad (1)$$

Subject to

$$\text{s.t. } \sum_{j \in D} x_{ij} = 1, \quad \forall i \in C \quad (2)$$

$$x_{ij} \leq y_j, \quad \forall i \in C, j \in D \quad (3)$$

$$\sum_{i \in C} x_{ij} \geq y_j, \quad \forall j \in D \quad (4)$$

$$z_{jk} \leq \frac{y_j + y_k}{2}, \quad \forall j, k \in D, j \neq k \quad (5)$$

$$y_j \geq z_{kj} + z_{jk}, \quad \forall j, k \in D, j \neq k \quad (6)$$

$$\sum_{k \in D} z_{kj} = \sum_{k \in D} z_{jk} = y_j, \quad \forall j \in D \quad (7)$$

$$\sum_{j \in D} z_{0j} = \sum_{j \in D} z_{j0} = 1 \quad (8)$$

$$t_0 = 0 \quad (9)$$

$$t_j + \frac{d_{jk}}{v} \leq t_k + M(1 - z_{jk}), \quad \forall j, k \in D, j \neq k \quad (10)$$

$$t_{ij} \geq t_j - M(1 - x_{ij}) + \frac{l_{jk}}{v^p} x_{ij}, \quad \forall i \in C, j \in S \quad (11)$$

$$t_{ij} \geq t'_{ij} - M(1 - x_{ij}) + \frac{l_{jk}}{v^d} x_{ij}, \quad \forall i \in C, j \in P \quad (12)$$

$$t'_{ij} \geq t_j - M(1 - w'_{ji}), \quad \forall i \in C, j \in P \quad (13)$$

$$t'_{jk} \geq t'_{ik} + 2 \frac{l_{ik}}{v^d} w_{ij}^k - M(1 - w_{ij}^k), \quad \forall i, j \in C, k \in P, i \neq j \quad (14)$$

$$g_i \geq t_{ij} - p_i x_{ij}, \quad \forall i \in C, j \in D \quad (15)$$

$$\sum_{i \in C} \text{demand}_i x_{ij} \leq N_j y_j, \quad \forall j \in D \quad (16)$$

$$\sum_{j \in D} y_j \geq 2 \quad (17)$$

$$\sum_{\substack{j \in C \\ i \neq j}} w_{ij}^k w'_{ik} = \sum_{\substack{j \in C \\ i \neq j}} w_{ji}^k w'_{ki} = y_k, \quad \forall i \in C, k \in P \quad (18)$$

$$\sum_{i \in C} w'_{ij} = \sum_{i \in C} w'_{ji} = y_j, \quad \forall j \in P \quad (19)$$

$$u_0 = 0 \quad (20)$$

$$u_j - u_k + 1 \leq |D| (1 - z_{jk}), \quad \forall j \in D \cup \{0\}, k \in D, j \neq k \quad (21)$$

$$u_j \leq |D| y_j, \quad \forall j \in D \quad (22)$$

$$0 \leq u_j \leq \sum_{k \in D} y_k, \quad \forall j \in D \quad (23)$$

$$x_{ij} \in \{0, 1\}, y_j \in \{0, 1\}, z_{jk} \in \{0, 1\}, \quad \forall i \in C, j, k \in D, j \neq k \quad (24)$$

$$w_{ij}^k \in \{0, 1\}, w'_{ij} \in \{0, 1\}, \quad \forall i, j \in C, k \in D, i \neq j \quad (25)$$

$$t'_{ij} \geq 0, \quad \forall i \in C, j \in P \quad (26)$$

$$g_i \geq 0, t_j \geq 0, t_{ij} \geq 0, \quad \forall i \in C, j \in D \quad (27)$$

식 (1)은 제안 모델이 UAM의 이동거리와 고객이 원하는 배송 시간과 실제 배송 시간의 차이, 배송지 사용에 따른 비용을 최소화하는 것을 목적으로 함을 나타낸다. 식 (2)는 모든 고객은 하나의 배송지에만 배정될 수 있다는 것을 나타낸다. 식 (3)과 (4)는 각각 배송지를 사용하기로 선택한 경우에만 해당 배송지에 고객이 배정될 수 있음을, 배송지에 배정된 고객이 없으면 해당 배송지는 사용하지 않음을 의미한다. 식 (5)–(7)은 사용되는 배송지들만 UAM이 방문한다는 것과 그 때의 흐름균형제약을 나타낸다. 식 (8)을 통해 UAM이 depot에서 출발해서 depot으로 돌아오도록 한다. 식 (9)와 (10)으로 UAM이 배송지에 도착하는 시간을 계산한다. 식 (11)은 vertistop에 할당된 고객들이 물품을 배송 받는 시간을 계산하며, 식 (12)–(14)을 통해 vertiport에서 드론이 고객들에게 물품을 배송하는 시간을 계산한다. 최종적으로 식 (15)를 통해 고객들의 선호 배송 시간과 실제 배송 시간의 차이를 구한다. 식 (16)은 배송지에 배정된 고객들의 수요총량은 배송지의 최대 수용량을 넘길 수 없음을 나타낸다. 식 (17)은 최소 2개의 배송지가 사용되어야함을 의미한다. 식 (18)과 (19)는 vertiport에서 드론이 방문하는 고객들의 순서에 대한 제약이다. 식 (20)–(23)은 부경로 생성을 방지하는 제약이다. 식 (24)와 (25)는 결정변수들의 이진조건을, 식 (26)과 (27)은 결정변수들의 비음조건을 의미한다.

3.3 휴리스틱 알고리즘

제안 모델의 경우 UAM이 모든 배송지가 아닌 선택된 배송지만을 방문하는 OP를 확장한 형태의 문제이다. Gunawan et al.(2016)에 따르면 OP는 Knapsack Problem과 TSP로 나누어질 수 있기 때문에 TSP의 일반적인 형태로 볼 수 있다. 이는 해당 문제가 TSP 문제의 한 종류임을 의미하며, TSP는 잘 알려진 NP-hard 문제이다. 이러한 문제의 수리최적화모형은 매우 큰 문제에 대해서 최적해를 다항 시간 (polynomial time) 안에 구하기 힘들다. 따라서 본 논문에서는 이를 해결하기 위해 2단계의 휴리스틱 알고

리즘을 제안한다. 1단계에서는 수리최적화 문제 중 하나인 상자 채우기 문제(bin packing problem)를 통해 휴리스틱에서 사용될 초기해를 구한다. 이후 2단계에서는 VNS 기반의 휴리스틱 알고리즘을 활용하여 해를 개선하게 된다.

1) 초기해 생성

초기해 생성에는 상자 채우기 문제의 수리최적화모형을 활용한다. 상자 채우기 문제는 정해진 물품을 모두 담기 위해 필요한 최소 상자의 개수와 어떤 상자에 어떤 물품이 들어갈지를 구하는 문제이다.

아래는 상자 채우기 문제를 풀기 위해 사용된 수리최적화모형이다. 3.1 절에 소개된 인덱스 집합과 상수, 결정변수를 공유한다.

$$\text{Minimize } \gamma \sum_{j \in D} C_j y_j + \delta \sum_{i \in D} \sum_{j \in D} l_{ij} x_{ij} \quad (28)$$

Subject to

$$\sum_{j \in D} x_{ij} = 1, \quad \forall i \in C \quad (29)$$

$$\sum_{i \in C} demand_i x_{ij} \leq N_j y_j, \quad \forall j \in D \quad (30)$$

여기서 δ 는 배송지와 해당 배송지에 배정된 고객 사이의 거리에 대한 비용 가중치를 의미한다. 식 (28)은 상자 채우기 문제의 목적식으로 배송지의 선정과 고객의 배정을 고려할 때, 배송지 선택에 따른 배용과 배송지와 고객 사이의 거리를 최소화하도록 한다. 식 (29)는 모든 고객이 각각 하나의 배송지에 배정되도록 제한한다. 식 (30)은 배송지에 배정된 고객의 수요량의 합이 해당 배송지의 최대 수용량을 넘지 않아야함을 의미한다.

휴리스틱의 초기해를 이러한 상자 채우기 문제를 수리최적화모형을 통해 풀어서 찾게 된다. 이를 통해 모든 고객을 적어도 하나의 배송지에 배정하기 위해 필요한 최소 배송지들과 해당 배송지들에 배정된 고객들을 구한다. 이렇게 구한 배송지 - 고객 쌍에 대해 경로를 생성한 후, 해당 값이 휴리스틱 알고리즘 2단계의 초기해로 사용 된다. 경로 생성 방법은 다음 절에서 설명한다.

2) 경로 생성

본 절에서는 상자 채우기 문제를 통해 구한 해를 휴리스틱 알고리즘의 초기해로 사용하기 전에 배송지 - 고객 쌍에 대해 UAM과 드론의 경로를 생성하는 과정에 대해 설명한다. 본 연구에서는 규칙 기반으로 경로를 생성하도록 하였다. UAM의 경우 상자 채우기 문제를 통해 구한 사용되는 배송지들에 대해서 최근접 이웃(nearest neighborhood) 방식을 적용하여 출발지에서부터 시작해서 가장 가까운 곳으로 이동하도록 하였다. 드론의 경우에도 최근접 이웃 방식을 적용하였지만 이때는 거리가 가장 가까운 고객부터 방문하는 것이 아니라, 배송 선호 시간이 가장 짧은 고객들부터 방문하도록 하였다.

3) 변동 이웃 탐색 알고리즘 (Variable Neighborhood Search Algorithm)

VNS는 메타 휴리스틱 기법으로 해 검색을 진행할 때, 이웃해의 구조를 다양하고 체계적으로 변경하는 방식에 기초한 알고리즘이다. VNS는 크게 다음과 같은 3가지 단계로 나뉘어진다.

Shaking : 사용될 이웃해 구조에 따라서 현재 최선해의 이웃해를 생성하는 과정

Local search : 생성된 이웃해들에 대해서 국지적으로 이웃해를 개선하는 과정

Move or Not : 개선된 이웃해가 현재 최선해보다 개선되었다면 현재 최선해를 이웃해로 치환하고, 아니면 다른 이웃해 구조를 시도하는 과정

VNS에서 핵심은 이웃해의 구조를 다양하고 체계적으로 설정하는 것이다. 이렇게 정해진 이웃해 구조를 기반으로 shaking이 진행되기 때문에 VNS의 성능은 이웃해 구조의 설정에 있다고 보아도 과언이 아니다. 따라서 VNS 사용할 때에는 이웃해 구조의 정의가 필수적이며, 아래와 같은 방식으로 해를 개선하게 된다.

[이웃해 구조 결정 및 초기해 도출]

Step 1. Shaking에 사용될 이웃해 구조를 결정한다. $\{N_k | k=1, \dots, k_{\max}\}$ 해당 이웃해 구조를 기반으로 임의의 해가 주어졌을 때 이웃해 집합을 생성하게 된다.

Step 2. 초기해(initial solution), 를 구하고 이를 최선해(best solution), 로 지정한다.

[VNS 기반 탐색]

Step 1. iteration Max_Iteration인 경우 Step 2로 간다. 아니면 Step 7로 간다.

Step 2. 를 1로 초기화 해준다.

Step 3. 미리 결정한 이웃해 구조에 따라서 현재 최선해의 이웃해 집합을 생성한다. (이 때 집합에 속하는 이웃해의 수는 미리 정한 n_{neighbor} 와 동일)

Step 4. 생성된 내의 이웃해에 대하여 국소 탐색(Local search) 방식을 적용하여 해를 개선한다. 이 중 가장 좋은 해를 로 지정한다.

Step 5. 만약 가 보다 좋은 해라면, 를 로 변경한 뒤, iteration을 iteration+1로 변경하고 Step 1로 간다. 아니면, 를 로 변경하고 Step 6으로 간다.

Step 6. 인 경우 Step 3으로 가고, 그렇지 않다면 Step 1로 간다.

Step 7. 를 최종해로 설정한 후 종료한다.

4) 이웃해 탐색 (Shaking)

본 논문에서 가정하는 문제 상황에서 이웃해를 생성하는 방법은 배송지 – 고객 쌍의 변경을 기반으로 하였다. 배송지 – 고객 쌍의 변경이 가능한 상황은 vertiport, vertistop의 추가, vertiport, vertistop의 제거, vertiport간의 교환, vertistop간의 교환, vertiport와 vertistop의 교환, vertistop과 vertiport의 교환으로 총 8가지이다. 배송지의 추가, 삭제, 교환이 이루어지면 이에 따라 필수적으로 고객의 배송지 변경이 필요하다. 특히 새로 추가된 배송지에 몇 명의 고객을 배정할지가 문제가 되는데, 이는 기존 배송지들에 배정된 고객의 수 중에서 가장 작은 값만큼 기존 배송지의 고객을 새로운 배송지로 재배정 해주었다. 앞서 언급한 8가지 경우를 기반으로 아래와 같은 8가지 이웃해 구조를 설정하였다.

$k=1$: 사용되지 않은 배송지 중에서 임의의 vertiport를 추가로 사용하도록 지정한다. 이때, 해당 vertiport에는 기존의 다른 배송지에 배정된 고객들 중 추가되는 vertiport와 가장 가까운 고객을 배정한다.

$k=2$: 사용되지 않은 배송지 중에서 임의의 vertistop을 추가로 사용하도록 지정한다. 이때, 해당 vertistop에는 기존의 다른 배송지에 배정된 고객들 중 추가되는 vertistop과 가장 가까운 고객을 배정한다.

$k = 3$: 사용되는 배송지 중에서 임의의 vertiport를 사용하지 않도록 지정한다. 이때, 해당 vertiport에 배정되었던 고객들은 나머지 사용되는 배송지 중 가장 가까운 곳에 최대 수용량을 초과하지 않는 선에서 추가 배정된다.

$k = 4$: 사용되는 배송지 중에서 임의의 vertistop을 사용하지 않도록 지정한다. 이때, 해당 vertistop에 배정되었던 고객들은 나머지 사용되는 배송지 중 가장 가까운 곳에 최대 수용량을 초과하지 않는 선에서 추가 배정된다.

$k = 5$: 사용되는 배송지 중에서 임의의 vertiport와 사용되지 않는 배송지 중에서 임의의 vertiport를 교환한다. 이때, 삭제된 vertiport에 배정되었던 고객들은 k 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에 k 가 1, 2일 때와 같은 방식으로 추가된 vertiport에 고객을 배정한다.

$k = 6$: 사용되는 배송지 중에서 임의의 vertistop과 사용되지 않는 배송지 중에서 임의의 vertistop을 교환한다. 이때, 삭제된 vertistop에 배정되었던 고객들은 k 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에 k 가 1, 2일 때와 같은 방식으로 추가된 vertistop에 고객을 배정한다.

$k = 7$: 사용되는 배송지 중에서 임의의 vertiport와 사용되지 않는 배송지 중에서 임의의 vertistop을 교환한다. 이때, 삭제된 vertiport에 배정되었던 고객들은 k 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에 k 가 1, 2일 때와 같은 방식으로 추가된 vertistop에 고객을 배정한다.

$k = 8$: 사용되는 배송지 중에서 임의의 vertistop과 사용되지 않는 배송지 중에서 임의의 vertiport를 교환한다. 이때, 삭제된 vertistop에 배정되었던 고객들은 k 가 3, 4일 때와 같은 방식으로 먼저 배정되고, 이후에 k 가 1, 2일 때와 같은 방식으로 추가된 vertiport에 고객을 배정한다.

각 이웃해 방식은 선택되었을 때 알고리즘 실행 시 미리 설정해둔 이웃해의 개수만큼 해당 이웃해 방식으로 이웃해를 생성한다. 이렇게 생성된 이웃해 집합은 다음 절에서 설명할 국소 탐색의 입력값으로 사용되어 해의 개선이 이루어진다.

5) 국소 탐색(Local search)

국소 탐색은 UAM 혹은 드론의 배송 및 방문순서의 변경으로 고려하였다. UAM이 방문하는 배송지의 순서를 변경하거나 각 vertiport에서 드론이 방문하는 고객들의 순서를 변경하는 방식으로 국소 탐색을 진행하였다. 자세한 방법은 아래와 같다.

[Pseudocode]

Step 1. iteration \leq Max_Iteration인 경우 Step 2로 간다. 아니면 Step 6으로 간다.

Step 2. UAM과 드론 중 어떤 경로를 변경할지 임의로 선택한다. UAM이 선택되었다면 Step 3으로, 아니라면 Step 4로 간다.

Step 3. UAM 경로에 있는 배송지들 중 임의로 2개를 선택하여 방문순서를 변경하고 소요시간을 계산한다. 만약, 새로운 경로가 기존 경로보다 소요시간이 짧다면 기존 경로를 새로운 경로로 변경한다. Step 5로 이동한다.

Step 4. 고객이 2명 이상 할당된 vertiport 중 하나를 임의로 선택하여, 고객들 중 임의로 2명을 선택하여 방문순서를 변경하고 소요시간을 계산한다. 만약, 새로운 방문순서가 기존 방문순서보다 소요시간이 짧다면 기존 방문순서를 새로운 방문순서로 변경한다. Step 5로 이동한다.

Step 5. iteration을 iteration+1로 변경하고 Step 1로 간다.

Step 6. 현재 경로를 최종 경로로 설정한 후 종료한다.

4. 실험 및 분석

본 절에서는 수리최적화모형과 휴리스틱 알고리즘의 실험 환경과 실험 결과에 대해 설명하고 이를 분석한다. UAM을 통해 배송을 진행할 때 vertiport와 vertistop으로 나누어 구분하며, UAM과 드론의 협업배송을 고려하는 경로 문제는 지금까지 연구된 경우가 없다. 이로 인해, 제안 모델의 성능을 검증하기 위해 비교할 대상이 없다. 따라서 본 논문에서는 제안한 수리최적화모형과 휴리스틱 알고리즘의 성능을 비교하여 검증을 수행하였고, 동일한 문제를 기본적인 TSP로 풀었을 때의 결과를 함께 비교하였다.

4.1 실험 설정

실험에 사용된 데이터는 Solomon 데이터로 이는 Solomon(1987)이 제안한 경로 문제에 주로 사용되는 데이터이다. Solomon 데이터는 크게 C (Clustered), R (Random), RC (Random Clusterd) 형식으로 나누어진다. C 형식 데이터는 고객들이 여러 군집을 이루어 존재하며, R 형식 데이터는 고객들이 군집을 이루지 않고 임의의 위치에 분포되어 존재한다. 마지막 RC 형식 데이터는 C 형식 데이터와 R 형식 데이터를 합친 형태로, 군집을 이루어 존재하는 고객들과 임의의 위치에 분포하는 고객들이 혼재되어 있다. Solomon 데이터의 크기는 25, 50, 100으로 나누어진다.

수리최적화모형의 경우 문제의 크기가 커지면 해를 구하기 매우 어려워지는 특성 때문에, 수리최적화모형은 Solomon 데이터 중 고객의 수가 25, 50인 데이터를 재구성한 작은 문제에 대해 실험을 진행하였다.

Solomon 데이터에는 출발점인 depot과 고객 노드들만이 존재하기 때문에 UAM이 방문하고, 고객들이 배정될 배송지 노드가 존재하지 않는다. 이를 해결하기 위해 배송지 중 vertiport는 임의의 위치에 생성하였고, vertistop은 기존 고객 노드들 중 일부를 선택하여 사용하여 Solomon 데이터의 성질과 형태를 최대한 유지하고자 하였다.

제안 모델에서는 UAM과 드론을 사용하며, vertistop에 배정된 고객의 경우에는 고객이 직접 방문하여 물품을 수령하기 때문에 UAM, 드론, 고객의 속도를 지정해 주어야 한다. Straubinger et al.(2020)에서는 UAM의 속도에 대해 150km/h에서 200km/h, 240km/h 등 다양하게 제시하였다. 본 논문에서는 가장 보수적인 기준을 적용하여 UAM의 속도를 150km/h로 가정하였다. 전기 드론의 속도는 Restas et al.(2021)을 참고하여 50km/h로 가정하였으며, 고객의 이동 속도는 다양한 이동수단의 활용을 고려한 Shin et al.(2022)의 연구를 참고해 10km/h로 가정하여 매개변수를 산정해주었다. 최종적으로 UAM의 속도는 3, 드론의 속도는 1, 사람의 속도는 0.2로 설정하였다.

실험은 AMD Ryzen 9 5900X 12-Core Processor, 64GB RAM에서 수행하였으며, 수리최적화모형을 풀기 위한 solver로 Gurobi 9.5.2 버전을 사용하였다. 휴리스틱 알고리즘은 Python 3.8.11 환경에서 개발하였다.

4.2 실험 결과

본 절에서는 Solomon 데이터를 활용한 수리최적화모형과 휴리스틱 알고리즘의 실험 결과에 대한 설명과 분석을 진행한다.

먼저, 수리최적화모형의 결과에 대해 기술한다. 최적해를 도출할 수 있도록 총 노드 수를 10, 11, 13개로 줄여서 실험을 진행하였다. 이때, 고객의 수를 7, 8, 9명, vertiport의 수를 1, 1, 1개, vertistop의 수를 2, 2, 3개로 설정하였으며, Table 3을 통해 그 결과를 확인할 수 있다. 제안 모델의 성능을 평가하기 위해 경로 문제의 가장 기본인 TSP의 해와 함께 비교하였다. TSP의 경우 Google(2022)의 OR-Tools를 사용하였고, 실험을 진행할 때 도심의 평균 차량의 속도는 Shin et al.(2022)의 연구를 참고하여 30km/h로 가정하였다. 앞서 50km/h로 가정한 드론의 속도 파라미터를 1로 지정하였기 때문에 30km/h로 가정한 차량의 속도 파라미터는 0.6으로 지정하였다. 수리최적화모형은 3600초(1시간)의 연산시간 제한을 두어 최적해를 구하지 못한 경우 최선해의 값을 표기하였다.

Table 3. Experiment result of the MILP and OR-Tools TSP

Problem	Number of Nodes				MILP		TSP	
	Customer	Verti port	Vert istop	Total	Solution (gap)	Run Time	Solution	Run Time
C101	8	2	3	13	20.00 (0%)	0.11	173	0.003
C101	9	2	4	15	15.00 (0%)	0.02	182	0.004
C101	10	3	5	18	25.00 (0%)	0.08	179	0.006
R101	8	2	3	13	35.00 (0%)	21.06	295	0.003
R101	9	2	4	15	58.62 (42.5%)	3600	276	0.003
R101	10	3	5	18	49.52 (9.1%)	3600	338	0.005
RC101	8	2	3	13	30.00 (0%)	5.27	299	0.006
RC101	9	2	4	15	31.15 (0%)	10.32	300	0.004
RC101	10	3	5	18	45.00 (14.3%)	3600	329	0.004

Table 3을 살펴보면 최적해와 Google OR-Tools의 TSP 모델과 비교하기 위해 작은 크기로 데이터를 조절하였지만, R101 데이터의 경우 총 노드 수가 15개, 18개일 때, RC101 데이터 경우 총 노드의 수가 15개일 때 3600초 동안의 연산에도 최적해를 구하지 못한 것을 확인할 수 있다. 또한, R101 데이터의 경우 15개 노드 수의 문제에 대해 42.5%, 18개 노드 수의 문제에 대해 9.1%의 gap을 보였고, RC101 데이터의 경우 18개 노드 수의 문제에 대해 14.3%의 gap을 보였다. 결과를 보면 대체로 군집화 되어있는 경우에 더 좋은 해를 도출하는 것을 볼 수 있다. 이는 제안 모델이 가지는 성질 때문으로 보이며, 이는 Table 4의 결과와 함께 다시 설명하도록 하겠다. 노드의 수를 더 줄여서 실험을 진행할 수도 있지만, 고려하는 vertiport와 vertistop의 수가 더 줄어들면 고정적인 해를 도출할 위험이 있어 위와 같은 설정으로 실험을 진행하였다.

제시한 수리최적화모형이 최적해를 도출한 경우 TSP와 비교하였을 때, 80% 이상 개선된 해를 도출한 것을 확인할 수 있다. 이는 제안 모델의 목적식의 비용 고려를 위한 파라미터 설정에 의한 차이와 더불어, UAM과 드론의 속도가 차량의 속도보다 훨씬 빠르다는 점이 크게 작용한 것으로 보인다.

다음으로 휴리스틱 알고리즘이다. 앞서 살펴본 것처럼 수리최적화모형은 문제가 조금만 커지더라도 최적해를 찾는 데 매우 시간이 오래 걸리게 된다. 이에 따라 큰 문제에 대해서도 적절한 해를 제공할 수 있는 2단계 휴리스틱 알고리즘을 개발하였다. 휴리스틱 알고리즘의 경우 Solomon 데이터 중 노드 수가 25개, 50개, 100개인 모든 경우에 대해서 실험을 진행하였고, 생성할 이웃해는 50개, 국소 탐색 100회, 총 반복 100회로 설정하여 실험을 진행하였다. Table 4를 통해 그 결과를 확인할 수 있다. 마찬가지로 수리최적화모형의 경우 3600초의 연산시간 제한을 두고 실험을 진행하였다.

Table 4. Experiment result of the MILP and the heuristic algorithm for proposed model

Problem	Number of Nodes				MILP		Heuristic	
	Customer	Verti port	Verti stop	Total	Solution (gap)	Run Time	Solution	Run Time
C101	18	2	4	25	30.00 (0%)	0.23	35.81	87.34
C101	35	4	11	50	60.00 (0%)	6.22	108.85	105.80
C101	71	9	20	100	251.90 (48.4%)	3600	200.90	127.29
R101	18	2	4	25	202.23 (85.2%)	3600	218.63	93.21
R101	35	4	11	50	677.20 (91.8%)	3600	175.83	90.00
R101	71	9	20	100	2956.76 (96.4%)	3600	214.82	117.51
RC101	18	2	4	25	376.49 (89.4%)	3600	218.63	105.57
RC101	35	4	11	50	793.90 (90.6%)	3600	223.17	92.38
RC101	71	9	20	100	6026.57 (97.9%)	3600	331.07	138.27

휴리스틱의 경우 수리모형이 최적해를 구한 경우에는 근소한 차이로 유효해를 찾은 경우도 있지만 최적해와 차이를 보이는 경우도 있다. 수리최적화모형이 3600초 동안 최적해를 구하지 못한 경우에 대해서는 한 경우를 제외하고 모두 수리최적화모형보다 더 좋은 해를 도출하였다.

Table 4의 실험 결과 역시 Table 3에서와 비슷한 성향을 보인다. 고객들이 군집을 이루는 형태인 C101 데이터의 경우 문제를 푸는 속도가 R101, RC101 데이터에 비해 빠르고, 더 낮은 해를 도출하는 것을 볼 수 있다. 총 노드 50개를 고려하는 문제의 경우 C101 데이터는 6.22초 만에 최적해를 구한 반면, R101과 RC101 데이터는 3600초 동안 최적해를 찾지 못한 모습이다.

이러한 이유가 발생하는 제안 모델의 특성은 다음과 같다고 판단된다. 제안 모델의 경우 UAM이 방문하는 배송지의 위치와 고객의 위치에 따라 해의 값이 크게 영향을 받는다. 배송지들과 고객들이 가깝게 위치한다면, 고객에게 물품 배송까지의 시간이 크게 줄어들게 되고 이 경우 UAM의 이동 거리가 주요 비용으로 작용하게 된다. 반면에 배송지들과 고객들 사이의 거리가 멀다면, UAM이 아무리 빠르게 배송을 완료하여도 고객이 직접 물품을 수령 하는 시간은 늦어지게 된다. 이런 경우에는 UAM의 이동 거리와 더불어 고객이 배송받기를 원하는 시간과 실제 배송을 받은 시간의 차이가 주요 비용으로 작용한다. 이러한 특성으로 인해 제안된 모델은 문제를 풀기 위한 네트워크의 상태에 따라 해가 매우 크게 변동할 수 있음을 알 수 있다.

5. 결론

본 논문에서는 UAM과 드론의 협업 배송모형을 제시한다. 본 연구는 UAM을 배송에 활용하며, UAM의 방문 노드를 UAM 운영에 필수적인 기반 시설인 vertiport와 vertistop로 나누어 구분한다. 또한 vertiport와 vertistop은 각각 다른 방식으로 고객에게 최종 물품을 전달한다. 이러한 문제 상황을 반영한 수리최적화모형을 제시하였다. 문제 크기가 커짐에 따라 수리모형이 유효해를 구하기 어려워지는 문제로 인해 수리최적화와 메타-휴리스틱을 함께 사용하는 2단계의 휴리스틱 알고리즘을 제시하였다. 휴리스틱 알고리즘은 1단계에서는 수리최적화모형인 상자 채우기 문제를 풀어 메타-휴리스틱에서 사용할 초기해를 찾는다. 2단계에서는 VNS 기반의 휴리스틱 알고리즘을 사용하여 1단계에서 찾은 초기해를 바탕으로 보다 개선된 해를 도출한다.

제시한 수리최적화모형과 휴리스틱 알고리즘의 성능평가를 위해 Solomon 데이터를 활용한 실험을 진행하였다. 실험 결과를 통해 제시한 수리최적화모형이 UAM이나 드론을 사용하지 않는 방식보다 우수함을 확인할 수 있었으며, 작은 문제 크기에 대해 휴리스틱 알고리즘이 수리최적화모형과 근접한 해를 도출하는 것을 확인하였다. 이를 통해 본 논문에서 제시한 수리최적화모형과 휴리스틱 알고리즘이 기존 방식을 이동거리 및 시간 면에서 크게 발전시킬 수 있다는 것을 확인하였다.

본 연구에서는 UAM과 드론의 협업 배송 시스템이라는 새로운 문제를 제안하였기 때문에 이를 발전시킬 추후 연구 방향은 다양할 것으로 판단된다. Vertiport와 Vertistop에서의 배송 방식을 확장하여 vertiport에서는 차량을 통해 하나의 경로를 형성하여 배송을 하고, vertistop에서는 하나의 드론을 통해 고객에게 배송을 진행하는 방식의 연구가 다음 연구 주제가 될 수 있다. 또한, 본 연구에서는 vertiport에서 하나의 드론만을 사용하지만, 다수의 드론을 활용하여 배송을 진행하는 방식의 연구도 추후 연구 방향으로서 제시한다.

참고문헌

- Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6), 831-842.
- Bennaceur, M., Delmas, R., & Hamadi, Y. (2022). Passenger-centric Urban Air Mobility: Fairness trade-offs and operational efficiency. *Transportation Research Part C: Emerging Technologies*, 136, 103519.
- Cheon Seunghoon., Kim, Seongmin., & Lee, Chaeyoung. (2020). Paradigm Change in Traffic Congestion Cost Estimation and Results of Traffic Congestion Cost Estimation in 2017. *Monthly KOTI Magazine on Transport*, (), 11-16.
- Choi, J-S., Lee, S-H., Baek, J-S., Hwang, H-W. (2021). A Study on Vertiport Installation Standard of Drone Taxis(UAM). *Journal of the Korean Society for Aviation and Aeronautics*, 29(1), 74-81.
- Fadhil, D. N. (2018). A GIS-based analysis for selecting ground infrastructure locations for urban air mobility. inlangen]. Master's Thesis, Technical University of Munich.
- Geem, Z. W., Tseng, C. L., & Park, Y. (2005, August). Harmony search for generalized orienteering problem: best touring in China. In *International conference on natural computation* (pp. 741-750). Springer, Berlin, Heidelberg.
- Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, 34(3), 307-318.
- Google OR-Tools (2022). https://developers.google.com/optimization/routing/tsp#cpp_8. (Accessed on 10/07/2022)
- Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2), 315-332.
- Hassanzadeh, A., Mohseninezhad, L., Tirdad, A., Dadgostari, F., & Zolfagharinia, H. (2009). Location-routing problem. In *Facility Location* (pp. 395-417). Physica, Heidelberg.
- Hong, J., Lee, M., Cheong, T., & Lee, H. C. (2019). Routing for an on-demand logistics service. *Transportation Research Part C: Emerging Technologies*, 103, 328-351.
- Hyundai Motor Group. (2022), <https://tech.hyundaimotorgroup.com/kr/mobility-device/uam/>. (Accessed on 10/07/2022).
- Ilhan, T., Iravani, S. M., & Daskin, M. S. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, 40(4), 406-421.
- Jeong, J., So, M., & Hwang, H. Y. (2021). Selection of Vertiports Using K-Means Algorithm and Noise Analyses for Urban Air Mobility (UAM) in the Seoul Metropolitan Area. *Applied Sciences*, 11(12), 5729.
- Kim, S., & Moon, I. (2018). Traveling salesman problem with a drone station. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 42-52.
- Labadie, N., Mansini, R., Melechovský, J., & Calvo, R. W. (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research*, 220(1), 15-27.
- Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European journal of operational research*, 177(2), 649-672.

- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109.
- Restás, Á., Szalkai, I., & Óvári, G. (2021). Drone Application for Spraying Disinfection Liquid Fighting against the COVID-19 Pandemic—Examining Drone-Related Parameters Influencing Effectiveness. *Drones*, 5(3), 58.
- Shin, H., Lee, T., & Lee, H. R. (2022). Skyport location problem for urban air mobility system. *Computers & Operations Research*, 138, 105611.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35 , 254-265.
- Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K. D., Kaiser, J., & Plötner, K. O. (2020). An overview of current research and developments in urban air mobility—Setting the scene for UAM introduction. *Journal of Air Transport Management*, 87, 101852.
- Verbeeck, C., Sörensen, K., Aghezzaf, E. H., & Vansteenwegen, P. (2014). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, 236(2), 419-432.
- Wu, Z., & Zhang, Y. (2021). Integrated Network Design and Demand Forecast for On-Demand Urban Air Mobility. *Engineering*, 7(4), 473-487.