

2022 한국SCM학회 춘계학술대회

전동 무인 항공기 전력의 모션 의존성을 고려한 배송 최적화 문제

2022년 4월 29일
대한상공회의소 회의실

*차형주¹, 김동균¹, # 은준엽², # 정태수¹

¹ 고려대학교 산업경영공학과 공급사슬 및 가치망 분석 연구실

² 고려대학교 기술경영전문대학원

* No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—
electronic, mechanical, photocopying, recording, or otherwise – without the permission of the presenter.
This document provides an outline of a presentation and is incomplete without the accompanying oral commentary and discussion

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1F1A105930212).

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)
(No. 2021R1F1A105930212).

* Presenter (hyungjoo_cha@korea.ac.kr),
Corresponding Author (jeun@korea.ac.kr, tcheong@korea.ac.kr)

Introduction

배경

- 배송 업체의 파업, 코로나감염증-19 펜데믹으로 인한 **배송 인력의 부재**
- 배송 시장의 급격한 확대 및 전자 상거래 발달로 인한 **급격한 수요 증가**

“ 수요 공급 불균형으로 인한 배송 시장 혁신 요구 ”

전동 무인 항공기

- 실제 조종사가 직접 탑승하지 않고, 지상에서 무선으로 조종해 **자동 또는 반자동으로** 날아가는 비행체
- 전동 무인항공기 (E-UAV) 시장 규모는 **2025년까지 12% 이상**의 CAGR로 성장할 것으로 예측 ([1])
- 특히, 글로벌 드론 물류 시장은 2030년까지 연평균 54%씩 성장, **390억 달러 규모의 시장으로 성장 예측** ([2])

물류와 드론의 만남

Amazon Prime Air



드론을 활용한 혈액 운송



물류에서의 전동 무인 항공기의 역할

- 하중이 높은 물건 운송은 불가
- 배터리에 의해 운송 범위 결정
- 접근성이 낮은 곳에 빠르게 배송 가능
- 자동 비행이 가능함

Literature review

과거 연구

: 전동 드론과 차량의 협업 배송과 관련된 연구는 다음과 같음

1. TSP with a drone station (Kim and Moon, ([3]))

전동 드론은 특정 station에서 왕복하며 배송을 수행, 차량은 그 외 지역을 방문하여 배달 수행

2. Flying Sidekick Traveling Salesman Problem (Murray and Chu, ([4]))

차량과 드론의 협업 문제 두 가지를 제시, FSTSP 와 PDSTSP라 명명하였음

FSTSP의 경우, 차량으로부터 출발하여 전동 드론은 순서쌍을 거쳐 경로가 생성됨

3. Vehicle Routing Problem with drones (Wang and Sheu, ([5]))

다중 차량과 다중 전동 드론이 협업 배송을 수행, 전동 드론이 출발했던 차량과 다른 차량으로 도착할 수 있음

4. Parallel Drone Scheduling Vehicle Routing Problem (Minh Anh et al., ([6]))

전동 드론의 station으로부터의 비행 가능 범위를 고려하여 다중 차량을 사용한 배송 모형 정의

기존의 연구들은 무한한 배터리 개수 가정, 모션으로부터 전동 드론이 충전하는 경우를 고려하지 않음

Problem Definition

문제 정의

- 본 연구에서는 전동 무인 항공기 전력의 모션 의존성을 고려한 경로 문제를 **Flying Sidekick Traveling Salesman Problem with Drone Recharge (FSTSP-DR)** 로 정의함

모션 의존성

- FSTSP-DR은 전동 무인 항공기가 **모션인 전기 차량의 배터리로 무선 충전 됨을 가정함**

가정

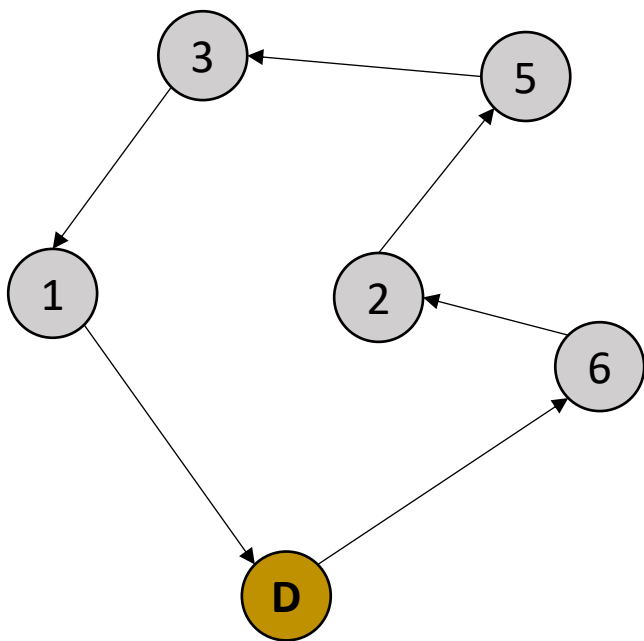
1. 주어진 모든 고객은 전기 차량이나 전동 무인 항공기로 **한 번만 방문 가능하다**
2. 전동 무인 항공기와 모션은 **각각 배터리를 소유한다**
3. 두 운송 수단 모두 운용 될 시 **거리에 비례하여 전력을 소모하며, 소모 비율은 다르다**
4. 전동 무인 항공기는 출발 시 **한 명의 고객을 방문하고, 출발지가 아닌 다른 노드에서 차량과 만난다**
5. 전동 무인 항공기 운용 시 **물건을 적재하는 시간과, 전동 무인 항공기를 수거하는 시간이 추가로 소요된다**

Problem Definition

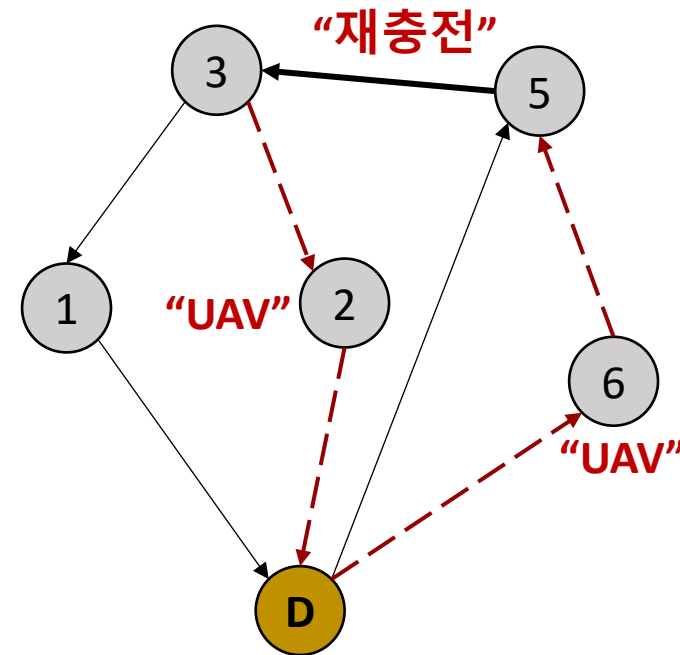
경로 예시

: FSTSP-DR과 일반 외판원 문제의 경로 상의 차이는 아래 그림과 같음

단일 차량 외판원 문제



FSTSP-DR



차량 배터리 : 100 (D) → 80 (6) → 70 (2) → 60 (5) → 50 (3) → 40 (1) → 20 (D)

UAV 배터리 :-

차량 배터리 : 100 (D) → 80 (5) → 65 (3) → 55 (1) → 35 (D)

UAV 배터리 : 50 (D) → 45 (6) → 40 (5) → 45 (3) → 42 (2) → 40 (D)

Total : 80사용

Total : 75 사용

Methodology : Mathematical Model



매개변수와 집합

- C : 모든 고객 집합 $1, 2, \dots, c$
 C_p : 드론 방문 가능 고객 집합, $C_p \subseteq C$
 N : $0, 1, 2, \dots, c, c + 1$
 N_0 : $0, 1, 2, \dots, c$
 N_+ : $1, 2, \dots, c + 1$
 P : 순서쌍 집합 $\langle i, j, k \rangle$,
출발 노드 i , 방문 고객 노드 j , 모선 다시 만나는 노드 k
 τ_{ij}, τ'_{ij} : 고객 i 에서 고객 j 로 가는 모선과 전동 드론의 이동 시간
 S_L, S_R : 소포 탑재 시간과 전동 드론 수거 시간
 α : 전동 드론 충전율
 β : 전동 드론 배터리 소모율
 γ : 모선 배터리 소모율
 Q, I : 모선과 전동 드론의 시작 배터리 양
 B : 드론의 배터리 최대 용량

결정 변수

- x_{ij} : 모선이 고객 i 에서 고객 j 로 이동하면 1 아니면 0
 y_{ijk} : 전동 드론이 고객 i 에서 출발하여 고객 j 에 배송을 하고
고객 k 에서 모선으로 돌아오면 1 아니면 0
 t_j : 고객 j 에 모선이 도착하는 시간
 t'_j : 고객 j 에 전동 드론이 도착하는 시간
 p_{ij} : 전동 드론과 모선의 정확한 랑데부를 위한 보조 변수
 u_i : 부경로 방지 제약을 위한 보조 변수
 o_i : 고객 i 의 위치에 전동 드론이 모선 위에 위치하면 1
아니면 0인 변수
 b_i : 고객 i 에서의 전동 드론 전력 잔량
 v_i : 고객 i 에서의 모선 전력 잔량
 z_i : 고객 i 에서 전동 드론이 최대치로 충전되어 있으면 1
아니면 0인 변수

Methodology : Mathematical Model

목적 함수

$\min t_{c+1}$: 배송을 마치고 출발지로 돌아오는 시간을 **최소화**

제약식 : 경로 생성

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} + \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{k \in N_+ \\ <i,j,k> \in P}} y_{ijk} = 1, \quad \forall j \in C$$

(2) : 모든 고객은 하나의 운송수단으로만 방문

$$\sum_{j \in N_+} x_{0j} = 1$$

(3)

$$\sum_{i \in N_0} x_{i,c+1} = 1$$

(4) : 모선의 흐름 제약식

$$u_i - u_j + 1 \leq (c+2)(1 - x_{ij}), \quad \forall i \in C, j \in \{N_+ : j \neq i\}$$

(5)

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} = \sum_{\substack{k \in N_+ \\ k \neq j}} x_{jk}, \quad \forall j \in C$$

(6)

$$\sum_{j \in C} \sum_{\substack{k \in N_+ \\ j \neq i, <i,j,k> \in P}} y_{ijk} \leq 1, \quad \forall i \in N_0$$

(7)

$$\sum_{i \in N_0} \sum_{\substack{j \in C \\ i \neq k, <i,j,k> \in P}} y_{ijk} \leq 1, \quad \forall k \in N_+$$

(8)

$$2y_{ijk} \leq \sum_{\substack{h \in N_0 \\ h \neq i}} x_{hi} + \sum_{\substack{l \in C \\ l \neq k}} x_{lk}, \quad \forall i \in C, j \in \{C : j \neq i\}, k \in \{N_+ : <i,j,k> \in P\}$$

(9)

$$y_{0jk} \leq \sum_{\substack{h \in N_0 \\ h \neq k}} x_{hk}, \quad \forall j \in C, k \in \{N_+ : <0,j,k> \in P\}$$

(10)

$$u_k - u_i \geq 1 - (c+2) \left(1 - \sum_{\substack{j \in C \\ <i,j,k> \in P}} y_{ijk} \right), \quad \forall i \in C, k \in \{N_+ : k \neq i\}$$

(11)

: 부경로 방지 제약

Methodology : Mathematical Model



제약식 : 도착 시간 일치

$$t'_i \geq t_i - M \left(1 - \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall i \in C \quad (12)$$

: 시간 Synchronization

$$t_0 = 0 \quad (23)$$

$$t'_0 = 0 \quad (24)$$

: 시작 시간은 0으로 초기화

$$t'_i \leq t_i + M \left(1 - \sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall i \in C \quad (13)$$

$$t'_k \geq t_k - M \left(1 - \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall k \in N_+ \quad (14)$$

$$t'_k \geq t_k + M \left(1 - \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall k \in N_+ \quad (15)$$

$$t_k \geq t_h + \tau_{hk} + S_L \left(\sum_{\substack{l \in C \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle k, l, m \rangle \in P}} y_{klm} \right) + S_R \left(\sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P}} y_{ijk} \right) - M(1 - x_{hk}), \quad \forall h \in N_0, k \in \{N_+ : k \neq h\} \quad (16)$$

: 도착 시간 전에 다시 출발 할 수 없도록 방지

$$t'_j \geq t'_i + \tau'_{ij} - M \left(1 - \sum_{\substack{k \in N_+ \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall j \in C', i \in \{N_0 : i \neq j\} \quad (17)$$

$$t'_k \geq t'_j + \tau'_{jk} + S_R - M \left(1 - \sum_{\substack{i \in N_0 \\ \langle i, j, k \rangle \in P}} y_{ijk} \right), \quad \forall j \in C', k \in \{N_+ : k \neq j\} \quad (18)$$

$$u_i - u_j \geq 1 - (c + 2)p_{ij}, \quad \forall i \in C, j \in \{C : j \neq i\} \quad (19)$$

$$u_i - u_j \leq -1 + (c + 2)(1 - p_{ij}), \quad \forall i \in C, j \in \{C : j \neq i\} \quad (20)$$

$$p_{ij} + p_{ji} = 1, \quad \forall i \in C, j \in \{C : j \neq i\} \quad (21)$$

$$t'_l \geq t'_k - M \left(3 - \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P \\ j \neq l}} y_{ijk} - \sum_{\substack{m \in C \\ m \neq i \\ m \neq k \\ m \neq l}} \sum_{\substack{n \in N_+ \\ \langle l, m, n \rangle \in P \\ n \neq i \\ n \neq k}} y_{ljk} \right) - p_{il}, \quad \forall i \in N_0, k \in \{N_+ : k \neq i\}, l \in \{C : l \neq i, l \neq k\} \quad (22)$$

Methodology : Mathematical Model

제약식 : 전동 드론 배터리 잔량 계산

$$\begin{aligned} b_0 &= I & (25) \\ b_j &\geq b_i - \beta\tau'_{ij} - M(1 - y_{ijk}) - Mx_{ij}, & \forall \langle i, j, k \rangle \in P & (26) \\ b_k &\geq b_j - \beta\tau'_{jk} - M(1 - y_{ijk}) - Mx_{jk}, & \forall \langle i, j, k \rangle \in P & (27) \\ b_k &\geq b_a + \alpha\tau_{ak}(1 - z_a) - M(1 - x_{ak}) - M \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P}} y_{ijk}, & \forall k \in N_+, a \in N_0 & (28) \\ b_j &\leq b_i - \beta\tau'_{ij} + M(1 - y_{ijk}) + Mx_{ij}, & \forall \langle i, j, k \rangle \in P & (29) \\ b_k &\leq b_j - \beta\tau'_{jk} + M(1 - y_{ijk}) + Mx_{jk}, & \forall \langle i, j, k \rangle \in P & (30) \\ b_k &\leq b_a + \alpha\tau_{ak}(1 - z_a) + M(1 - x_{ak}) + M \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C \\ \langle i, j, k \rangle \in P}} y_{ijk}, & \forall k \in N_+, a \in N_0 & (31) \end{aligned}$$

: 초기 전동 드론 배터리 양

: 전동 드론의 총방전 양 계산

Methodology : Mathematical Model



제약식 : 전동 드론 탑재 여부 및 차량 배터리 잔량 계산

$$z_i \geq 1 - M(B - b_i), \quad \forall i \in N_0$$

$$z_i \leq 1 - \frac{(b_i - B)}{M}, \quad \forall i \in N_0$$

$$o_k \leq o_h + \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C_p \\ j \neq h}} y_{ijk} - \sum_{m \in C_p} \sum_{\substack{n \in N_+ \\ \langle k, m, n \rangle \in P}} y_{kmn} + M(1 - x_{hk}), \quad \forall h \in N_0, k \in N_+, k \neq h$$

$$o_k \geq o_h + \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C_p \\ j \neq h}} y_{ijk} - \sum_{m \in C_p} \sum_{\substack{n \in N_+ \\ \langle k, m, n \rangle \in P}} y_{kmn} - M(1 - x_{hk}), \quad \forall h \in N_0, k \in N_+, k \neq h$$

$$o_i + o_j \leq x_{ij} - \sum_{\substack{k \in N_+ \\ \langle i, j, k \rangle \in P}} y_{ijk} + 1, \quad \forall i \in N_0, j \in C_p$$

$$v_0 = Q$$

$$v_j \leq v_i - \gamma \tau_{ij} + M(1 - x_{ij}) - (b_j - b_i) + M(1 - o_i), \quad \forall i \in N_0, j \in N_p, i \neq j$$

$$v_j \leq v_i - \gamma \tau_{ij} + M(1 - x_{ij}) + Mo_i, \quad \forall i \in N_0, j \in N_p, i \neq j$$

$$v_j \geq v_i - \gamma \tau_{ij} + M(1 - x_{ij}) - Mo_i, \quad \forall i \in N_0, j \in N_p, i \neq j$$

$$v_j \geq v_i - \gamma \tau_{ij} + M(1 - x_{ij}) - (b_j - b_i) - M(1 - o_i), \quad \forall i \in N_0, j \in N_p, i \neq j$$

(32) : 전동 드론 완충 여부 확인

(33)

(34)

(35)

(36)

: 전동 드론 모션 탑재 여부 확인

(37) : 모션 시작 배터리 양 선언

(38)

(39)

(40)

(41)

: 전동 드론 탑재 여부에 따른 모션 배터리 방전 계산 (충전은 없음)

제약식 : 결정 변수 선언

$$p_{0j} = 1, \quad \forall j \in C$$

$$1 \leq u_i \leq c + 2, \quad \forall i \in N_+$$

$$t_i \geq 0, \quad \forall i \in N$$

$$t'_i \geq 0, \quad \forall i \in N$$

$$b_i \geq 0, \quad \forall i \in N$$

$$v_i \geq 0, \quad \forall i \in N$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in N_0, j \in \{N_+ : j \neq i\}$$

$$y_{ijk} \in \{0,1\}, \quad \forall i \in N_0, j \in \{C : j \neq i\}, k \in \{N_+ : \langle i, j, k \rangle \in P\}$$

$$p_{ij} \in \{0,1\}, \quad \forall i \in N_0, j \in \{C : j \neq i\}$$

$$o_i \in \{0,1\}, \quad \forall i \in N$$

$$z_i \in \{0,1\}, \quad \forall i \in N$$

$$u_i \in Z^+, \quad \forall i \in N$$

(43) : Depot에서의 값 초기화

(44) : 부경로 방지 제약 위한 변수

(45)

(46)

(47)

(48)

(49) : 결정 변수에 각각 정수, 양수 조건 부여

(50)

(51)

(52)

(53)

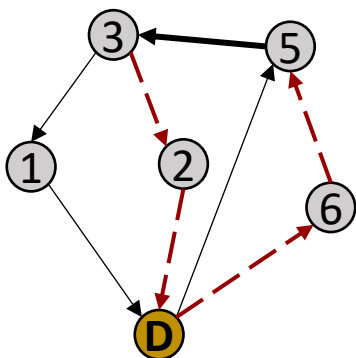
Methodology : Heuristic Algorithm

휴리스틱 알고리즘

- 소개한 수리모형 (P) 의 경우 정수 조건과 부경로 방지 제약, Bound가 Tight 하지 못한 몇 가지 제약이 존재
- 특히, \mathcal{NP} -hard로 알려진 TSP 문제가 FSTSP-DR의 특수 경우(Special Case) 임으로, **FSTSP-DR도 \mathcal{NP} -hard 임**
- 차원의 저주가 존재, 더 빠르게 좋은 가능해를 찾기 위해 **유전 알고리즘 기반의 휴리스틱 알고리즘** 구축

유전 알고리즘

- 생물의 진화 메커니즘으로부터 아이디어를 착안하여 만든 알고리즘
- 유전자, 염색체, 개체를 정의하여 세대를 거듭하며 해를 찾음
- **FSTSP-DR에서 유전자와 염색체는 다음과 같이 정의 됨**



: 염색체 1

* 염색체는 하나의 가능해를 나타냄

Methodology : Heuristic Algorithm

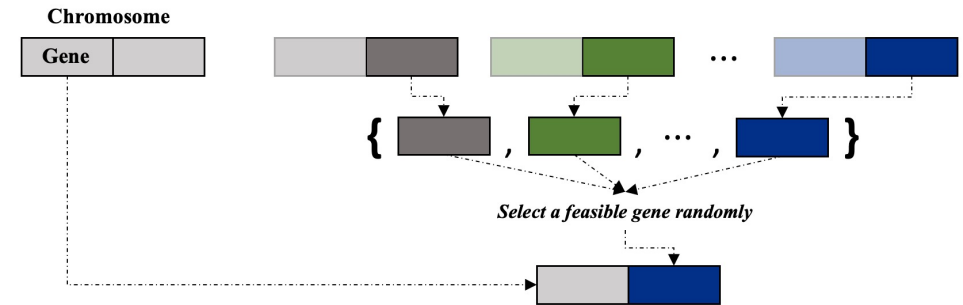
연산자 : 교차 및 변이

- 지역 최적점을 회피하기 위해 두 개체를 교차시켜 새로운 해를 생성
- 교차의 경우, 단순한 방법은 한 **고객을 여러번 방문하는 불가능 해를 생성**
- 따라서, 속해있는 세대 안에 ***연속해서 올 수 있는 유전자를 찾아 집합을 구성, 해당 집합에서 무작위 추출 진행**

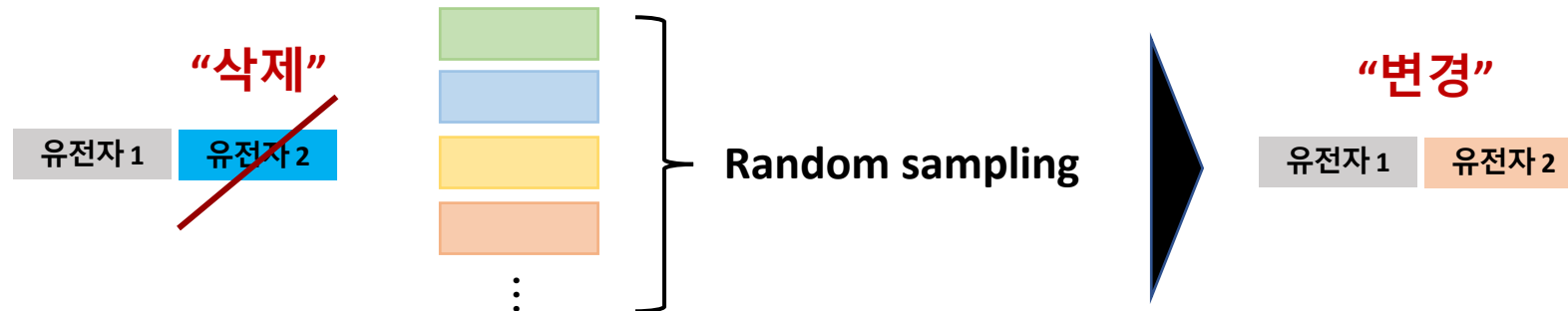
*연속해서 올 수 있는 유전자?

첫 번째 유전자안의 원소는 두 번째 유전자의 원소안에
첫 번째 유전자의 도착 노드를 제외하고 올 수 없다

Ex. [1,3,5] 뒤에는 [2,3,5] or [5,3,1] 등은 올 수 없다



- 변이의 경우 교차와 비슷하나, 세대 내에서 찾는 것이 아닌 연속해서 올 수 있는 집합 전체에서 무작위 추출



Methodology : Heuristic Algorithm

모선 경로 생성

- 모선 경로의 경우 아래 두 가지 규칙으로 구성
 - 1) 현재 전동 드론 출발 노드에 있다면 **가장 가까운 전동 드론 도착 노드**로 직행
 - 2) 그렇지 않다면, **가장 가까운 고객**으로 출발

실행 가능 여부 파악

- 가능해인지 여부 확인 절차 필요
 - * 가능해란?
 - 1) 경로 상에 모선과 전동 드론의 출/도착, 분리 및 합체가 실제로 가능한 상태로 나온 해
 - 2) 두 운송 수단 모두 **배터리 방전이 되지 않는 해**
- 불가능해로 판별되면 삭제하는 것이 아닌 좋지 않은 적합도를 줌
 - 유전 알고리즘의 교차, 변이 등으로 **추후 좋은 해로 발전 가능**
 - 특히, 최적해는 **가능 영역과 불가능 영역의 경계면에 존재함**

Experiments

실험 환경

- MacMini Apple M1 16 Core, 16GB Random Access Memory
- Python 3.8, IBM Cplex Solver 12.10.0 사용

실험 인스턴스

- VRP 인스턴스인 Solomon Instance 을 사용
- 주어진 고객 수 보다 줄여서 실험을 진행하여 수리모형과 비교

파라미터 환경

- 모선과 전동 드론 배터리 초기 값은 각 각 40000, 1000으로 셋팅
- 전동 드론은 최대 1000까지 충전 가능함
- 전동 드론의 충전율은 1, 사용율은 0.5로 설정
- 모선의 배터리 사용율은 2로 지정



배터리 용량 : 55kWh
소비 전력 : 16.8kWh



배터리 용량 : 1.5kWh
소비 전력 : 9kWh

Experiments



실험 결과 1

: 수리모형으로 TSP와 FSTSP-DR을 비교해보았음

Instance	$ N $	TSP		FSTSP-DR			
		Obj.val	Run time (s)	Obj.val	Run time (s)	Imp. over TSP(%)	EUAV usage (%)
C101	8	11.64	27.82	9.69	38.34	20.12	37.5
C101	9	12.16	581.61	10.18	447.99	19.44	33.3
C101	10	12.66	6311.12	10.69	22338.65	18.42	30
Avg:	-	<i>12.15</i>	<i>2306.86</i>	<i>10.18</i>	<i>7608.32</i>	<i>19.32</i>	<i>33.6</i>
R101	8	35.15	23.39	24.37	48.07	44.23	37.5
R101	9	36.53	323.06	26.16	596.05	39.64	33.3
R101	10	41.09	20582.18	29.29	8220.12	40.28	30
Avg:	-	<i>37.59</i>	<i>6976.21</i>	<i>26.60</i>	<i>2954.74</i>	<i>41.38</i>	<i>33.6</i>
RC101	8	23.23	22.38	20.23	35.68	14.82	37.5
RC101	9	23.97	304.28	20.96	452.32	14.36	33.3
RC101	10	34.41	18600.09	23.19	4627.61	48.38	20
Avg:	-	<i>27.20</i>	<i>6308.91</i>	<i>21.46</i>	<i>1705.20</i>	<i>25.85</i>	<i>30.2</i>

결과 1 : FSTSP-DR이 모든 경우에 대해 TSP보다 더 빠른 배송을 가능하도록 하였음

결과 2 : FSTSP-DR의 해 값은 최소 14.36%, 최대 48.38% TSP를 능가하였음

결과 3 : 가장 고객이 산재되어 있는 R-instance에서 가장 많은 전동 드론 사용률을 보였음

Experiments

실험 결과 2 : 휴리스틱 알고리즘의 성능을 비교해보았음

Instance	$ N $	Genetic Algorithm			
		Obj.val	Run time (s)	Gap to MILP(%)	EUAV usage (%)
C101	8	9.69 [†]	0.84	0.00	37.5
C101	9	10.53	0.90	3.43	33.3
C101	10	10.79	1.16	0.93	30.0
C101	25	40.63	21.22	*	24.0
C101	50	135.33	480.43	*	24.0
C101	100	499.22	13551.47	*	25.0
Avg:	-	<i>117.69</i>	<i>2342.673</i>	-	<i>28.96</i>
R101	8	24.75	0.65	1.55	37.5
R101	9	26.68	0.81	1.98	33.3
R101	10	31.84	2.25	8.70	30.0
R101	25	110.77	35.08	*	32.0
R101	50	254.95	383.07	*	24.0
R101	100	549.11	10880.01	*	25.0
Avg:	-	<i>166.35</i>	<i>1883.64</i>	-	<i>30.30</i>
RC101	8	21.05	0.74	4.05	37.5
RC101	9	22.05	1.04	5.20	33.3
RC101	10	23.88	1.30	2.97	30.0
RC101	25	76.90	18.34	*	24.0
RC101	50	266.30	370.89	*	24.0
RC101	100	628.45	13971.59	*	25.0
Avg:	-	<i>173.10</i>	<i>2393.98</i>	-	<i>28.96</i>

결과 1 : 작은 노드 수에 대해서 최적해를 찾음

결과 2 : 모든 경우 수리모형 보다 더 빠른 시간 안에 해를 찾음

결과 3 : 100명의 고객에 대해서는 약 3시간의 연산 시간이 필요했으나, 50명까지는 최대 9분 이내 해 도출

결과 4 : 고객 수가 증가 하면 전동 드론 사용율이 감소

* R101 25의 경우는 많아짐

충전에 필요한 시간이 확보되어야 해서 드론 사용율 감소

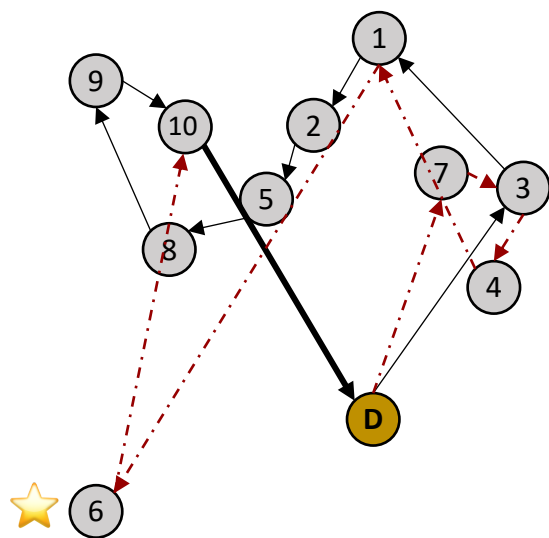
결과 5 : R-instance에서 평균적으로 해가 가장 빨리 도출

Experiments

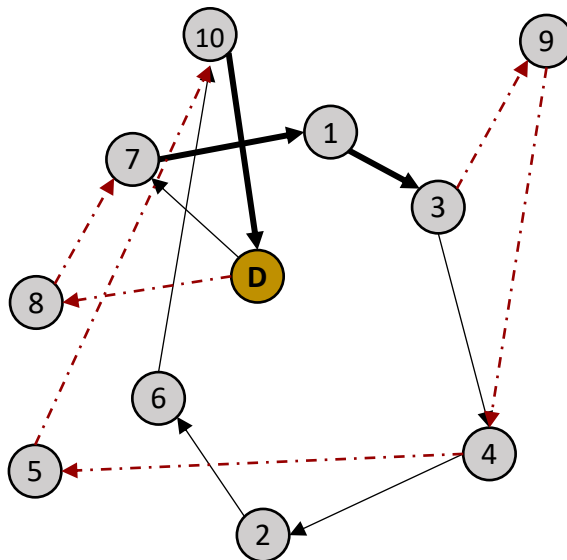
추가 실험 1

: Solomon 문제에 대해 추가적인 실험을 진행해보았음

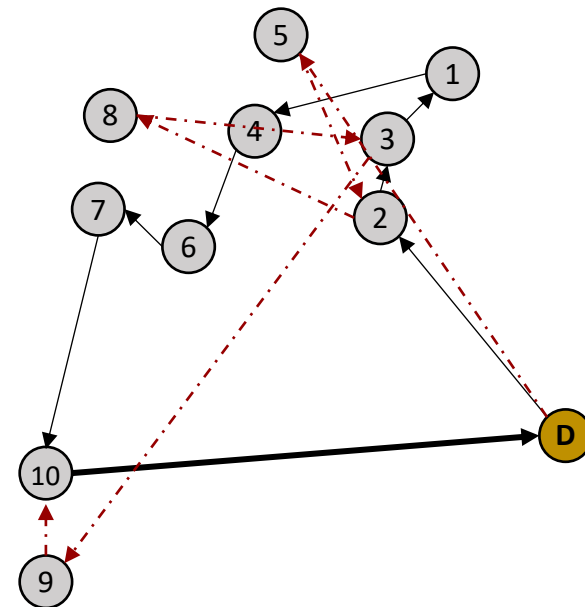
C Class : 군집된 고객 네트워크



R Class : 퍼져있는 고객 네트워크



RC Class : 일부만 퍼져있는 고객 네트워크



결과 1 : C class 처럼 군집된 곳에서는 군집을 전동 드론이 다 처리 하지 않으면 모선이 회수를 위해 자주 이동해야 함

결과 2 : C class 6번 고객처럼 매우 먼 고객은 전동 드론이 효과적으로 처리

결과 3 : 군집 처리가 필요없는 R-class에서는 매우 균형 있는 경로가 형성되었음

“네트워크에 대한 심도 있는 이해가 FSTSP-DR의 효율성을 극대화 할 수 있는 요소로서 작용할 수 있음”

Experiments

추가 실험 2

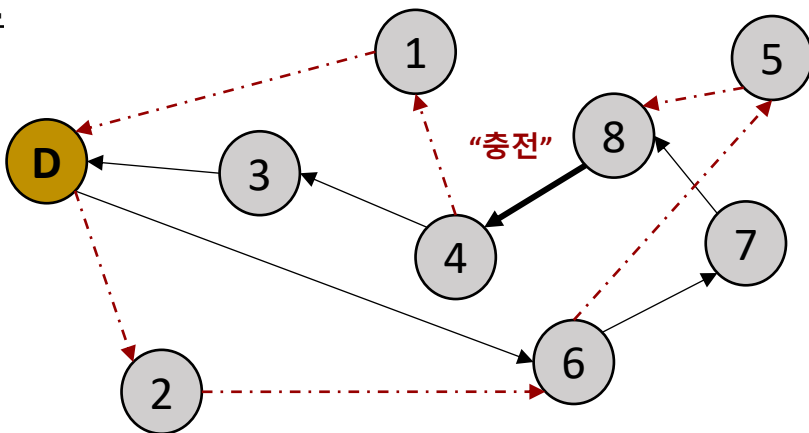
: 실제 제주도 지역에 배송을 수행하는 Case Study 진행

지도

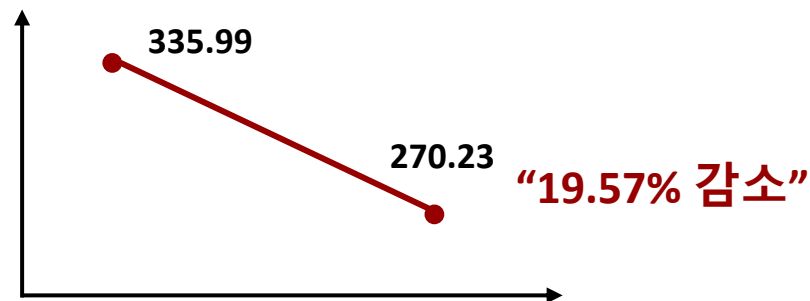


“각 지점이 섬의 sector를 나타내게끔 지정”

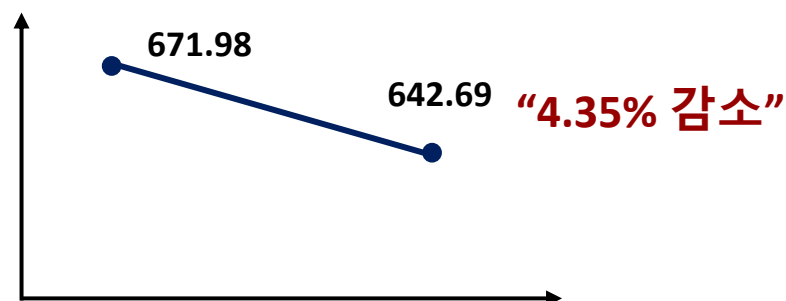
최적 경로



시간 비교 (with TSP)



총 전력 소모량 비교 (with TSP)



“섬의 외곽은 전동 드론이 방문하여 효율성 도모”

“단일 차량 운송보다 더 빠르게 전력을 덜 쓰고 배송을 수행”

Conclusion



- 이번 연구에서는 기존의 드론-차량 협업 배송에 대해 더 현실 적용 가능성을 높이기 위한 전동 드론 전력의 모션 의존성을 고려한 FSTSP-DR 문제를 정의하였음
- 문제 해결을 위해 최적해를 구할 수 있는 정수계획법 기반 수리모형을 구축하였음
- 최적해 도출에 매우 오래 시간이 걸림으로, FSTSP-DR에 맞는 유전 알고리즘 기반의 휴리스틱 알고리즘을 제작하였음
- 실험 결과, 수리모형은 단일 운송수단 방법인 TSP보다 모두 목적함수가 우월한 값을 주었음
- 유전 알고리즘은 제시한 수리모형 (p) 보다 더 빠른 시간안에 좋은 해를 주었음
- 네트워크 위상에 따른 추가 분석과, 제주도 Case study를 통해 insight를 도출하였음
- 추후 연구 가능 방향으로 새로운 충전 방법을 고려하거나, 강화학습과 같은 새로운 방법으로 문제를 해결하는 방향이 존재할 것으로 사료됨



References



- [1] <https://www.giikorea.co.kr/report/moi964495-electric-unmanned-aerial-vehicle-market-growth.html>
- [2] 한국교통연구원 물류기술연구센터
- [3] Kim, Sungwoo, and Ilkyeong Moon. "Traveling salesman problem with a drone station." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.1 (2018): 42-52.
- [4] Murray, Chase C., and Amanda G. Chu. "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery." *Transportation Research Part C: Emerging Technologies* 54 (2015): 86-109.
- [5] Wang, Zheng, and Jiu-Bing Sheu. "Vehicle routing problem with drones." *Transportation research part B: methodological* 122 (2019): 350-364.
- [6] Nguyen, Minh Anh, et al. "The min-cost parallel drone scheduling vehicle routing problem." *European Journal of Operational Research* 299.3 (2022): 910-930.