

2024-1 파란학기

대중교통과 트럭을 활용한 드론 배송 시스템

붕붕

박찬영 강승호 구본우 이원재

Contents

01 Introduction

- 프로젝트 선정 배경 및 목적
- 기대효과
- 선행 연구의 한계

03 Mathematical Modeling

- Notation
- Objective function
- Constraints

05 Result

02 System Outline

- 시스템 개요
- 가정사항

04 Algorithm

- Pseudocode
-

01 Introduction

프로젝트 선정배경

1. **ラスト 마일에서의 드론의 활용 가능성 증가**
 - 이커머스의 성장으로 인한 당일 배송의 중요성
 - 교통 혼잡으로 인한 제약 완화 가능성
 - 드론을 활용한 빠른 배송
2. **배송의 무인화를 통한 비용 절감**
 - 무인화 →ラスト 마일에서 발생하는 인건비 감소
 - 라스트마일 배송은 전체 물류 단계별 비용 구성에서 53%를 차지
 - [MIT 물류센터 연구 결과]
 - 트럭에서 드론 배송으로 변경할 시 배송비용 30% ~ 40% 절감 예측
3. **환경 오염 방지를 통한 ESG 경영 실현**
 - [미국 로렌스 리버모어 국립연구소 연구 결과]
 - 트럭에 비해 최소 20%에서 최대 54% 온실가스 배출량 감소 가능
 - 배터리 기술이 발전될 경우, 더욱 친환경적인 운송 수단으로 부상
4. **드론의 방전을 고려하지 않은 선행 연구의 한계**
 - 드론 배송의 단점 중 하나인 배송과정에서의 '제한된 에너지 소비'
 - 드론의 제한된 에너지 소비를 고려하는 선행연구는 존재
 - FSTSP(Flying Sidekick Traveling Salesman Problem)
 - 스탠포드 대학교 - Round Trip Routing for Energy-Efficient Drone Delivery Based on a Public Transportation Network
 - **그러나**, 드론의 방전 상황을 고려한 선행연구는 존재하지 않음.

➡ 방전된 **드론의 회수**를 고려하는 방법론 제시

목적 드론의 에너지 제약 해결

Problem	Solution
드론의 방전	배송-회수 시스템
긴 배송 시간 & 비효율	트럭과 드론의 동시 사용
드론의 거리 제약	대중교통 활용

- ✓ 배송-회수 시스템: 고객에게 배송을 마친 후 Depot으로 직접 비행하지 않아도 트럭, 대중교통을 활용하여 회수되는 시스템
- ✓ 트럭과 드론, 서로의 이점을 활용한 상호보완적 시스템
 - 트럭: **속도↓** **용량↑**
 - 드론: **속도↑** **용량↓**
- ✓ 기존의 인프라인 대중교통 활용 → 초기 투자비용 감소

기대효과



새로운 드론 배송 패러다임

- 새로운 서비스 모델 창출
- 시장 경쟁력 확보
- 새로운 시장 진입



배송 영역 확장

- 교통 시스템이 미흡한 지역도 배송
 - 인건비 절감
- 사이클 시간 단축으로 인한 영역 확장



경제적 파급 효과

- 효율적 물류 시스템 구축
- 운용 비용 절감



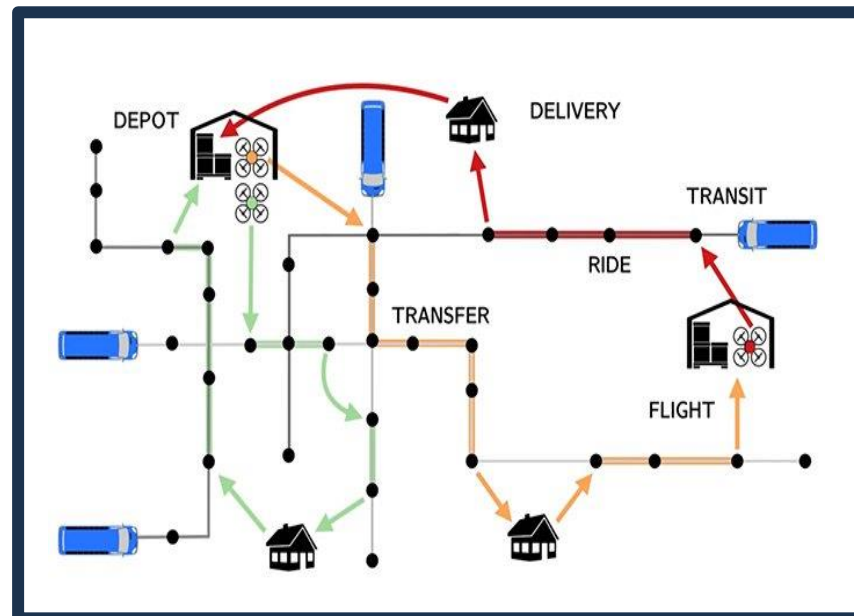
에너지 제한 완화

- 비행 시간 단축
- 탄소 배출량 감소

01 Introduction

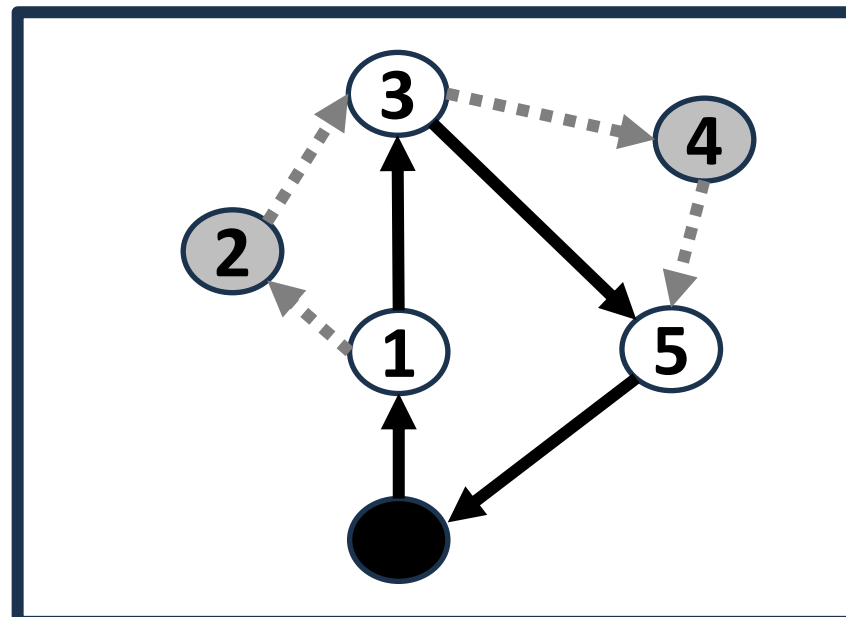
■ 선행 연구의 한계

Drone Delivery Using
Public Transit Networks



Depot 복귀 후 충전

FSTSP



Truck 복귀 후 충전

드론의 방전 고려하지 않음

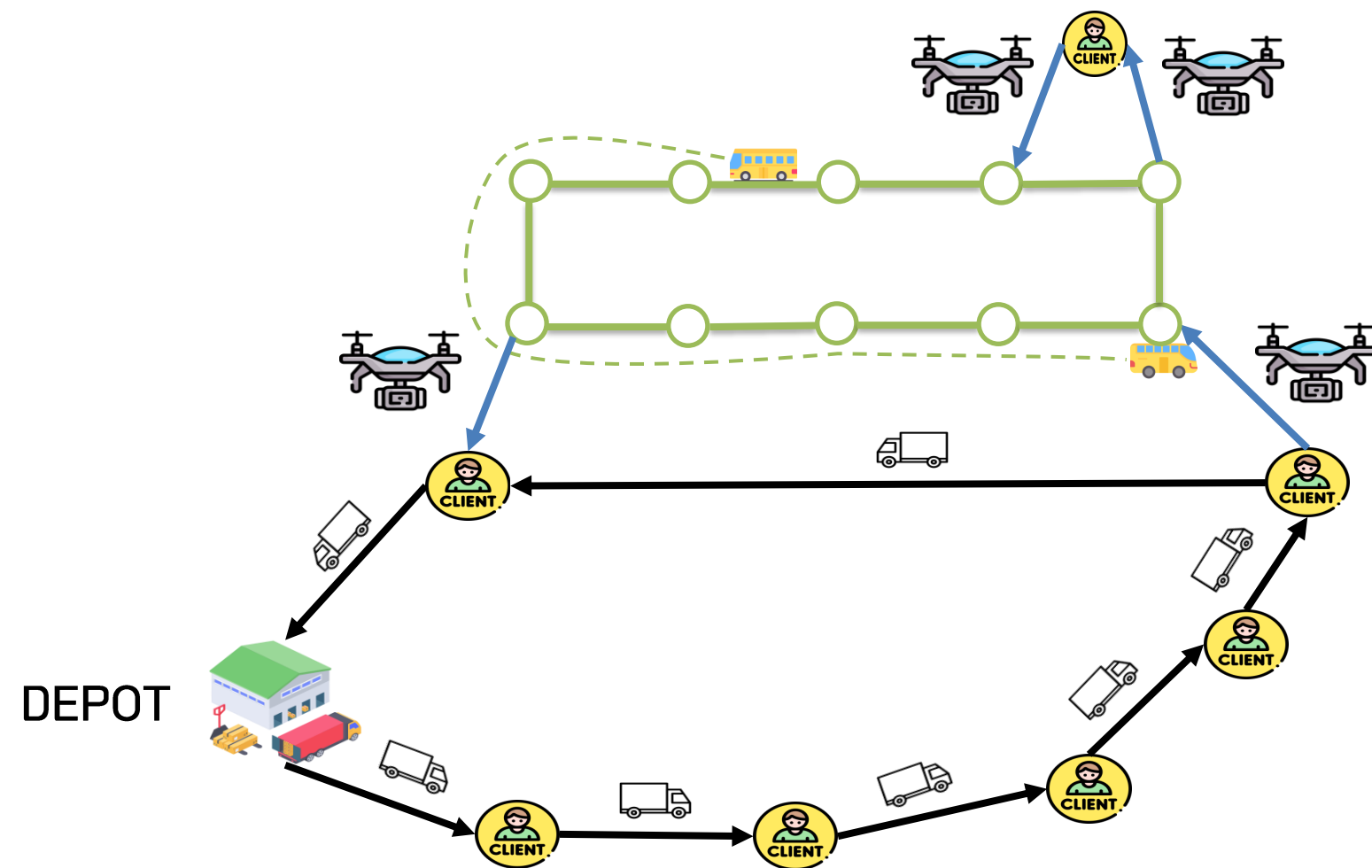


방전된 **드론의 회수**를
고려하는 방법론 제시

System Outline

드론-트럭-대중교통을 활용한 배송 시스템

- To minimize $Cost_드론 + Cost_트럭$



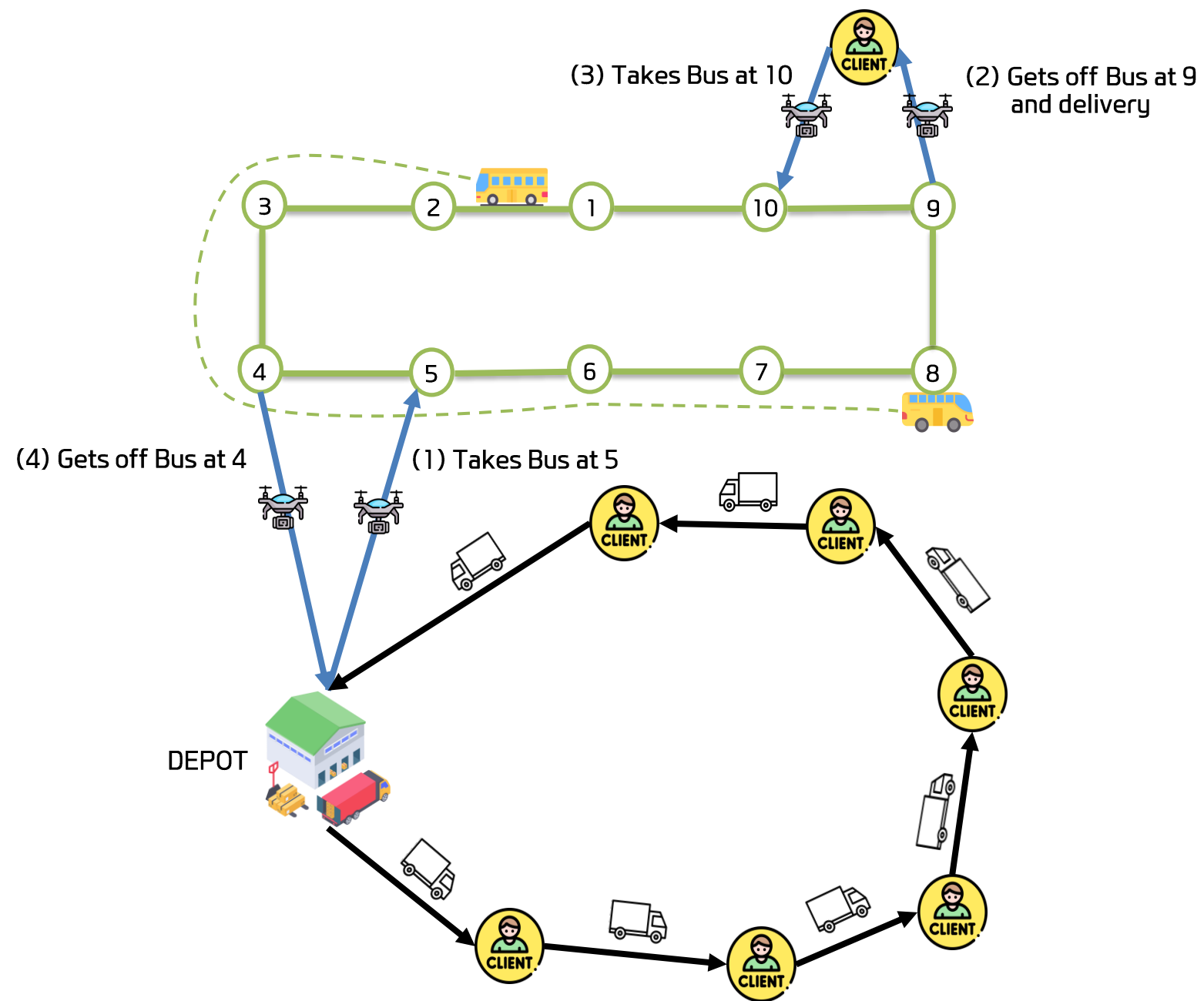
[System Outline]

시스템 가정사항

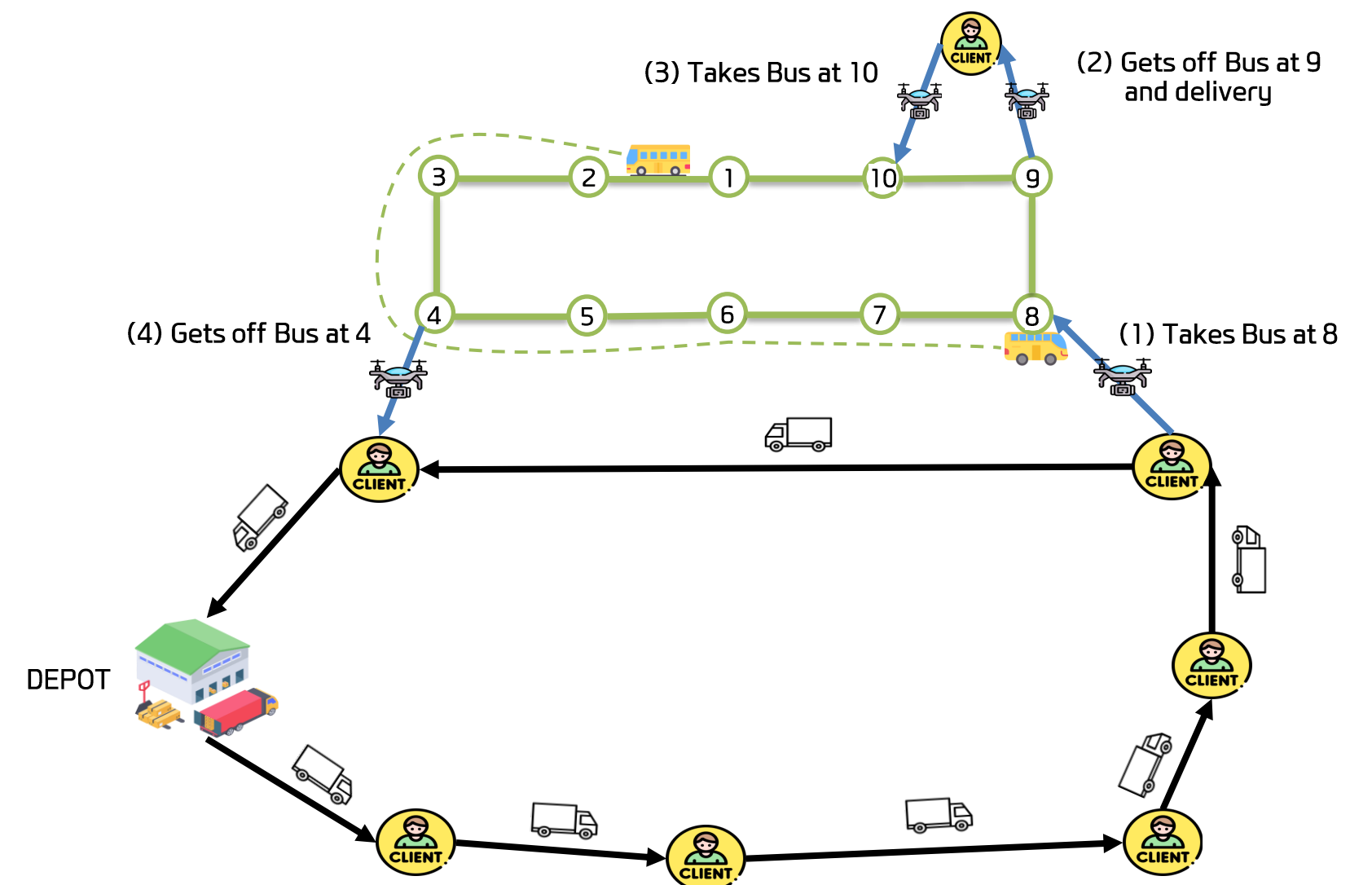
- ✓ 드론과 트럭은 depot에서 동시에 출발
- ✓ N 개의 고객 노드 중 드론이 최대 1개, 트럭은 최소 $(N-1)$ 개 배송
- ✓ 드론은 비행, 트럭 탑승, 버스 탑승 3가지 행동
- ✓ 드론은 트럭 탑승, 버스 탑승, 고객 배송을 위해 반드시 비행으로 이동
- ✓ 드론은 트럭 또는 버스보다 먼저 노드에 도착해야 차량 탑승 가능
- ✓ 드론은 비행시 에너지 소비 발생
- ✓ 드론은 최대 사용가능 에너지인 E 이상을 소비할 수 없음
- ✓ 트럭의 운용 비용은 드론의 운용 비용보다 월등히 큼
- ✓ 드론의 속도가 트럭의 속도보다 빠름
- ✓ 버스의 경로는 고정
- ✓ 각 정류장마다 버스의 도착 시간은 고정

02 System Outline

Case 1: 드론이 버스만을 사용하여 배송



Case 2: 고객 노드에서 드론 회수



Mathematical Modeling

Notation

[Index]

i, j, k : index of nodes

[Set]

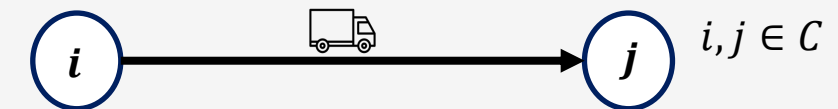
C : Set of customer nodes $(2, \dots, C)$
 S : Set of station nodes $(C + 1, \dots, S = N)$
 N : Set of nodes $(1, \dots, N)$ $(C, S \subset N)$

[Parameter]

B_i : Time of bus arrival at node i $i \in S$
 d_{ij} : Time taken to move from i node to j node $i, j \in N$
 C_d : Drone operating costs
 C_t : Truck operating costs
 E : Maximum usable energy of drone
 M : Sufficiently large number

[Decision Variable]

x_{ij} : 1, when the truck transits the arc i to j
 0, in the other cases



r_{ij} : 1, when the drone reaches station node j from customer node i at delivery process
 0, in the other cases



r'_{ij} : 1, when the drone reaches station node j from customer node i at return process
 0, in the other cases $i \in C, j \in S$

l_{ij} : 1, when the drone leaves station node j from customer node i at delivery process
 0, in the other cases



l'_{ij} : 1, when the drone leaves station node j from customer node i at return process
 0, in the other cases $i \in S, j \in C$

u_i : The order in which trucks visited node i $i \in C$

t_i : Cumulative time when the truck arrives at node i $i \in C$

τ_i : Cumulative time when the drone arrives at node i $i \in N$

Mathematical Modeling

Objective Function

$$\text{minimize } \sum_{i=1}^C \sum_{j=1}^C C_t * d_{ij} * x_{ij} + \sum_{i=1}^N \sum_{j=1}^N (r_{ij} + l_{ij} + r'_{ij} + l'_{ij}) * C_d * d_{ij}$$

Constraints

$$\sum_{i=C+1}^N l_{ij} = \sum_{i=C+1}^N r'_{ji} \quad j = 1, \dots, C \quad (1)$$

$$\sum_{i=C+1}^N l_{ij} + \sum_{i=1}^C x_{ij} = 1 \quad j = 1, \dots, C \quad (2)$$

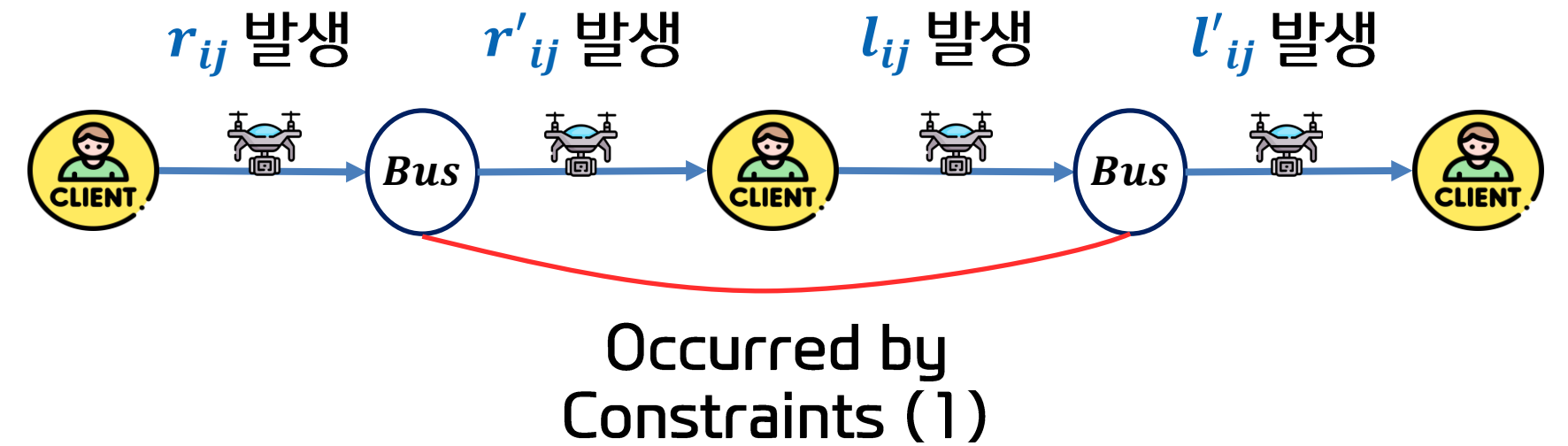
$$\sum_{i=1}^C \sum_{j=C+1}^N r_{ij} \geq \sum_{i=1}^C \sum_{j=C+1}^N r'_{ij} \quad (3)$$

$$\sum_{i=C+1}^N \sum_{j=1}^C l_{ij} \geq \sum_{i=1}^C \sum_{j=C+1}^N r_{ij} \quad (4)$$

$$\sum_{i=C+1}^N \sum_{j=1}^C l'_{ij} \geq \sum_{i=1}^C \sum_{j=C+1}^N r'_{ij} \quad (5)$$

- (1) 드론이 고객에게 배송을 할 시, 두 번의 비행이 발생한다.
 (2) 모든 고객은 트럭 또는 드론으로 반드시 한 번씩 배송된다.
 (3)~(5) $r_{ij}, r'_{ij}, l_{ij}, l'_{ij}$ 간의 선행관계가 존재한다.

* Constraints (3)~(5) 추가 설명



$$\sum_{j=C+1}^N (r_{ij} + r'_{ij}) \leq 1 \quad i = 1, \dots, C \quad (6)$$

$$\sum_{j=1}^C (l_{ij} + l'_{ij}) \leq 1 \quad i = C + 1, \dots, N \quad (7)$$

- (6)~(7) 동일한 노드에서 *reaching, leaving*은 두 번 발생할 수 없다

Mathematical Modeling

Constraints

$$\sum_{j=2}^C x_{1j} = 1 \quad (8)$$

$$\sum_{i=1}^C x_{ij} = \sum_{i=1}^C x_{ji} \quad j = 1, \dots, C \quad (9)$$

$$u_1 = 1 \quad (10)$$

$$2 \leq u_i \leq C \quad i = 2, \dots, C \quad (11)$$

$$u_j \geq u_i + 1 - C(1 - x_{ij}) \quad i = 1, \dots, C \quad j = 2, \dots, C \quad (12)$$

$$t_1 = 0 \quad (13)$$

$$t_j \geq t_i + d_{ij} - T(1 - x_{ij}) \quad i = 1, \dots, C \quad j = 2, \dots, C \quad (14)$$

(8) 트럭은 Depot에서 출발한다.

(9) 노드 j에서 트럭이 들어오면 다른 노드로 반드시 나가야 한다.

(10)~(12) 트럭의 경로 중 Sub-tour는 발생할 수 없다.

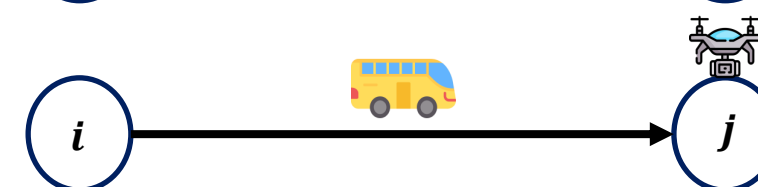
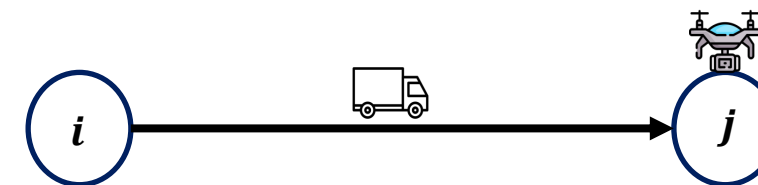
(13)~(14) 트럭은 이동 시 이동 시간만큼 누적한다.

Constraints

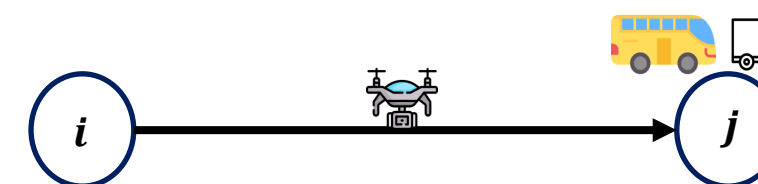
$$\tau_1 = 0 \quad (15)$$

$$\tau_i \leq B_i \quad i = C + 1, \dots, N \quad (16)$$

$$t_j \geq \tau_j - T \left(1 - \sum_{i=1}^C x_{ij} \right) \quad j = 1, \dots, C \quad (17)$$



Drone waits for
Truck or Bus



Truck and Bus
can't wait for Drone



(15)~(17) 드론이 트럭 또는 버스 탑승 시, 반드시 드론이 먼저 도착해야 한다.

Mathematical Modeling

Constraints

$$B_j \geq B_i - T * \left(2 - \sum_{k=1}^C r_{ki} - \sum_{k=1}^C l_{jk} \right) \quad i, j = C + 1, \dots, N \quad (18)$$

$$B_j \geq B_i - T * \left(2 - \sum_{k=1}^C r'_{ki} - \sum_{k=1}^C l'_{jk} \right) \quad i, j = C + 1, \dots, N \quad (19)$$

$$\tau_j \geq \tau_i + d_{ij} - T(1 - r_{ij}) \quad i = 1, \dots, C \quad j = C + 1, \dots, N \quad (20)$$

$$\tau_j \geq \tau_i + d_{ij} - T(1 - r'_{ij}) \quad i = 1, \dots, C \quad j = C + 1, \dots, N \quad (21)$$

$$\tau_j \geq B_i + d_{ij} - T(1 - l_{ij} - l'_{ij}) \quad i = C + 1, \dots, N \quad j = 1, \dots, C \quad (22)$$

$$\sum_{i=1}^N \sum_{j=1}^N (r_{ij} + l_{ij} + r'_{ij} + l'_{ij}) * C_d * d_{ij} \leq E \quad (23)$$

$$x_{ij} \in \{0,1\} \quad i, j = 1, \dots, C \quad (24)$$

$$r_{ij}, l_{ij}, r'_{ij}, l'_{ij} \in \{0,1\} \quad i, j = 1, \dots, N \quad (25)$$

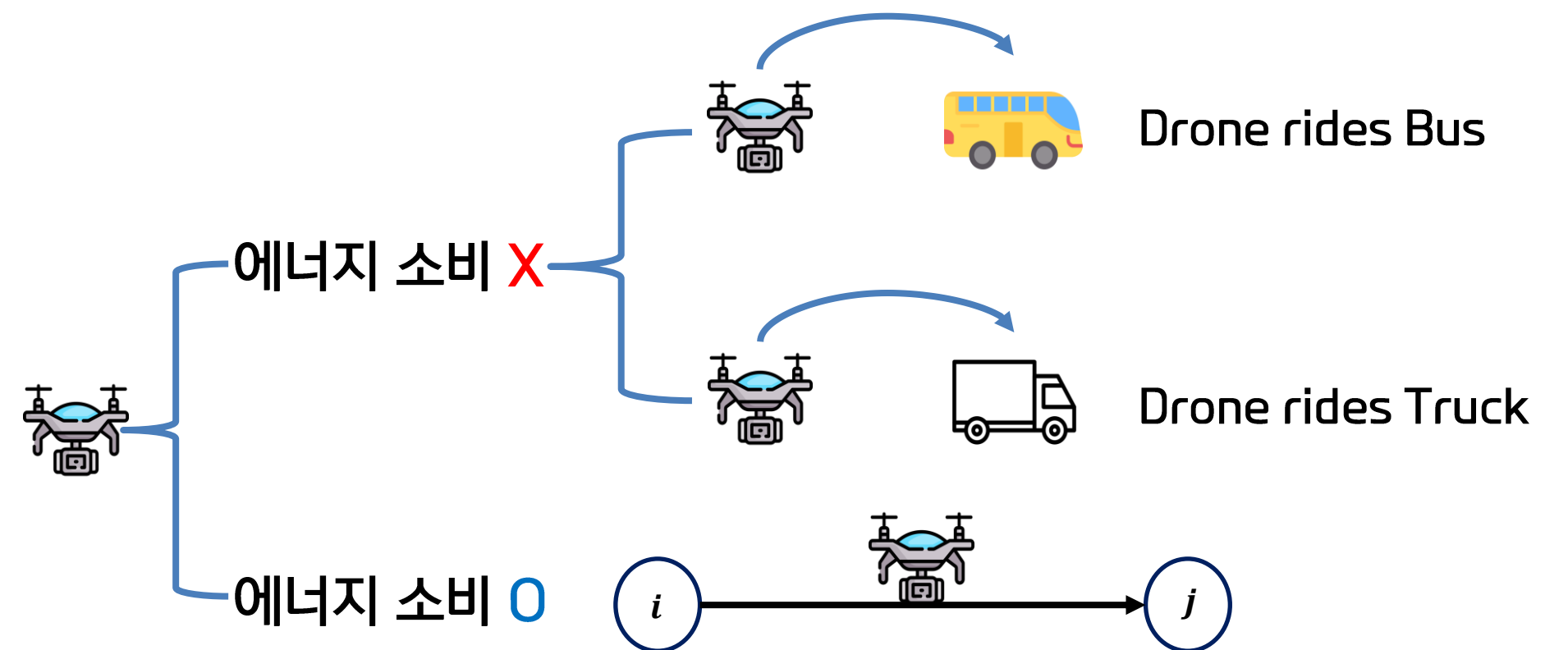
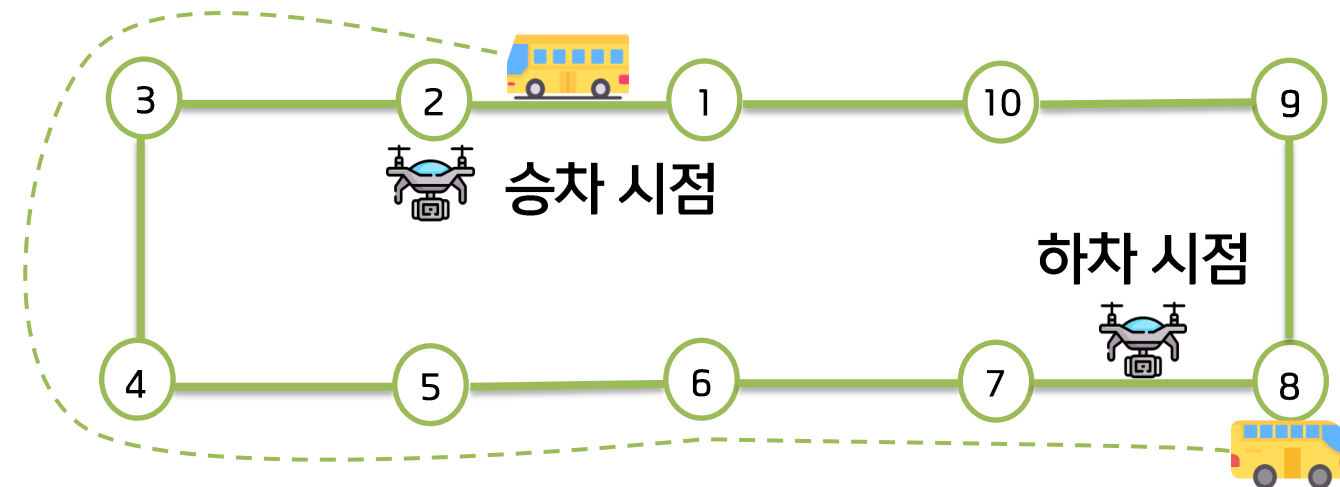
(18)~(19) 버스 하차 시점은 승차 시점 이후이다.

(20)~(22) 드론이 비행 시 비행 시간만큼 누적한다.

(23) 드론의 소비 에너지는 최대 에너지(E)보다 적어야 한다.

(24)~(25) $x_{ij}, r_{ij}, l_{ij}, r'_{ij}, l'_{ij}$ 는 이진 변수이다.

* Constraints (18)~(19), (23) 추가 설명



04 Algorithm

Algorithm Pseudocode

Algorithm 1 Graph Initialization

- 1: Initialize the graph with cost matrix, number of customers, and number of buses
 - 2: Set the pheromone levels for each edge
 - 3: Set the unchosen probability for each node
-

Algorithm 2 ACO Initialization

- 1: Set the number of ants, generations, and other parameters
 - 2: Initialize pheromone levels and related variables
-

Algorithm 3 Ant Colony Optimization with Drone Delivery

- 1: **for** generation = 1 to max_generations **do**
 - 2: Initialize the ants
 - 3: **for** each ant in the colony **do**
 - 4: Select an unvisited node based on probabilities
 - 5: **for** each node in the graph **do**
 - 6: Select the next node based on pheromone and heuristic information
 - 7: Update the ant's tour and cost
 - 8: Update the time and pheromone delta for the ant
 - 9: **end for**
 - 10: Calculate the total cost for the ant's tour
 - 11: If the drone is used, update the drone route and cost
 - 12: Update the pheromone levels for the ant
 - 13: **end for**
 - 14: Update the global pheromone levels based on all ants
 - 15: Print the current best solution
 - 16: **end for**
 - 17: **return** The best solution found
-

04 Algorithm

Algorithm Pseudocode

Algorithm 4 Ant Initialization

- 1: Initialize the ant's memory, position, and other related variables
 - 2: Set the initial node for the ant
-

Algorithm 5 Select Next Node

- 1: Calculate the probabilities for moving to the next node
 - 2: Select the next node based on the calculated probabilities
 - 3: Update the ant's memory and position
 - 4: Update the total cost and pheromone delta
-

Algorithm 6 Pheromone Update

- 1: **for** each edge in the graph **do**
 - 2: Update the pheromone level based on the ant's tours
 - 3: **end for**
 - 4: Update the unchosen probabilities for each node
-

Algorithm 7 Drone Delivery Calculation

- 1: **for** each node in the ant's tour **do**
 - 2: Reset the drone's local cost and time
 - 3: **for** each bus node **do**
 - 4: Calculate the route and cost if the drone is used
 - 5: Update the best cost and route for the drone
 - 6: **end for**
 - 7: **end for**
 - 8: **return** The best drone route and cost
-

05

Result

Optimization Solver

Energy	15 nodes	20 nodes	25 nodes
100	2653	2933	3356
50	2653	2953	3450
30	2709	3040	3547

AntColony Algorithm

Energy	15 nodes	20 nodes	25 nodes
100	2788	3047	3503
50	2725	3041	3563
30	2729	3090	3566

05 Result

Optimization Solver



✓ 참고 문헌

- <http://m.irobotnews.com/news/articleView.html?idxno=13133>, "드론 배송 서비스, 온실가스 배출 우려 높아"
- <https://www.klnews.co.kr/news/articleView.html?idxno=304713>, Part 1. 코로나 팬데믹에 다시 주목받는 '드론 배송'
- Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C*, 54, 86–109 .
- Huang, H., Savkin, A. V., & Huang, C. (2020). Round Trip Routing for Energy-Efficient Drone Delivery Based on a Public Transportation Network. *IEEE Transactions on Transportation Electrification*, 6(3), 1368–1376 .
- Choudhury, S., Solovey, K., Kochenderfer, M. J., & Pavone, M. (2021). Efficient Large-Scale Multi-Drone Delivery Using Transit Networks. *Journal of Artificial Intelligence Research*, 70, 757–788 .

감사합니다.

아주대학교 박찬영 강승호 구분우 이원재