

2025 Optimization Grand Challenge

# Roll-on Roll off Optimization

---

수학적 최적화를 통한 강제 하역 최소화

서울대학교 산업공학과 공급망관리 연구실  
FORORO 이소현 강세정

# Contents

01

알고리즘 로직 및 구현

02

핵심 아이디어

03

알고리즘 발전 방향

04

경진대회 참여 후기

01. 알고리즘 로직 및 구현

알고리즘 로직

Clustering을 활용한 초기해 제공



재배치 비용 최소화를 위한 MILP 풀이



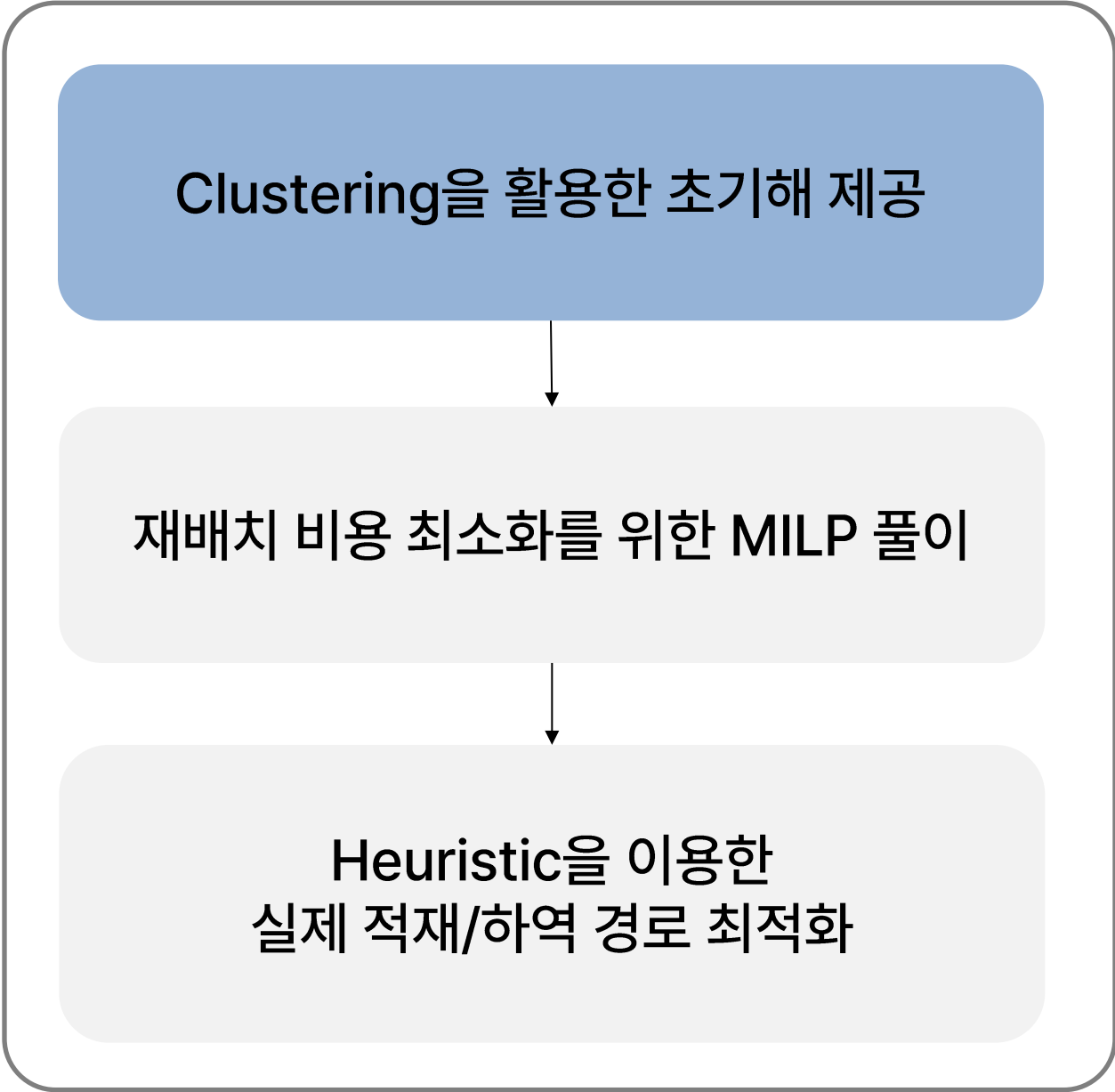
Heuristic을 이용한  
실제 적재/하역 경로 최적화

알고리즘 구현

- ✓ 대회 기본 언어인 Python으로 구현함.
- ✓ 1개의 CPU를 사용하는 default Python
- ✓ 최적화 모델은 Gurobi를 solver로 사용함.

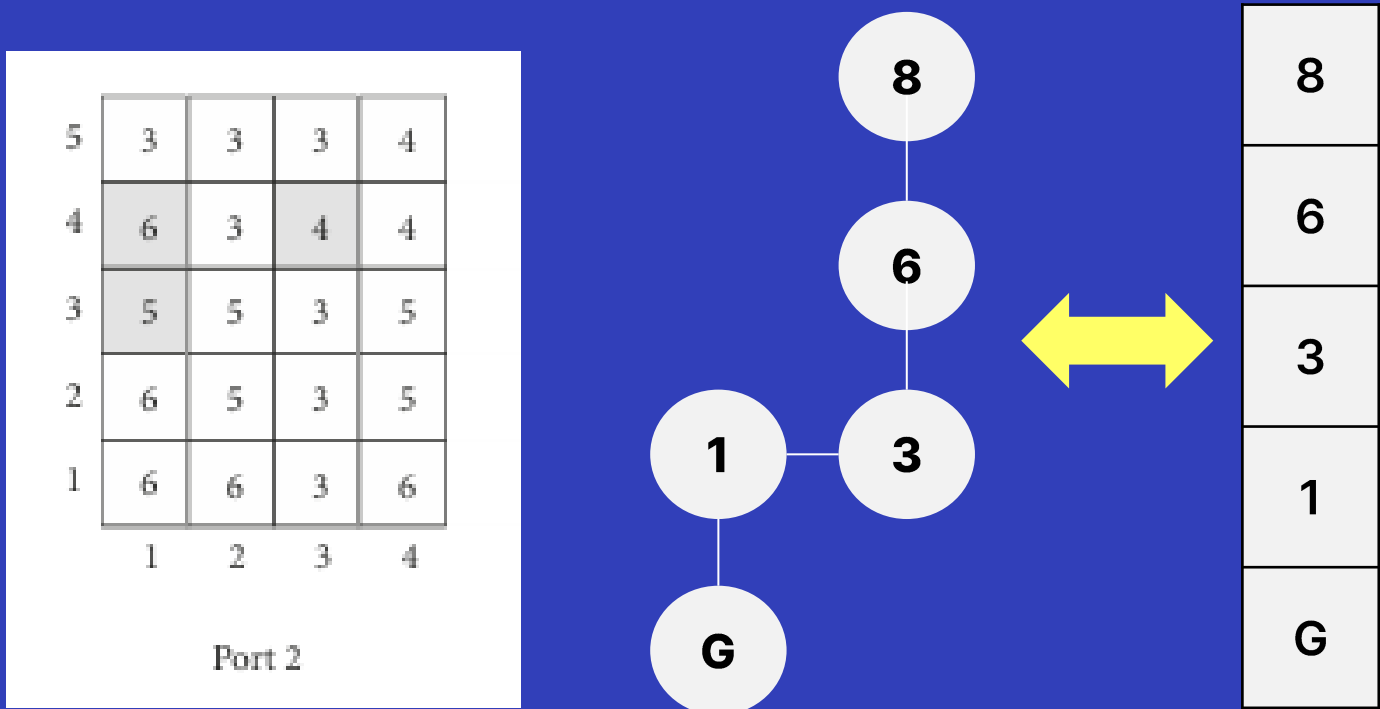
02. 핵심 아이디어

알고리즘 로직



선행 연구 : Multiport Container Ship Stowage Problem

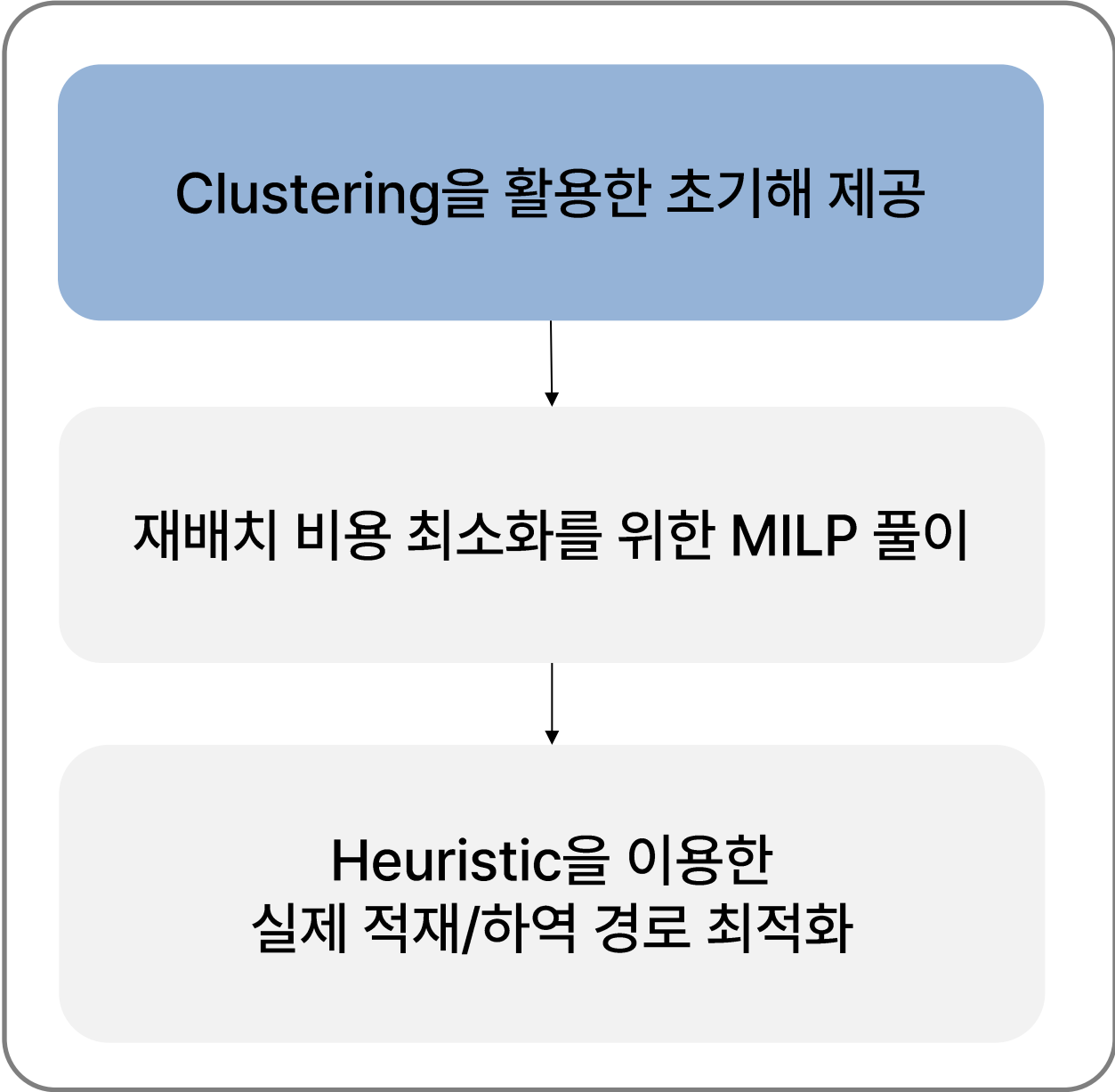
미리 정해진 M rows x N columns 형태의 bay가 존재하지 않아 바로 적용이 어려움



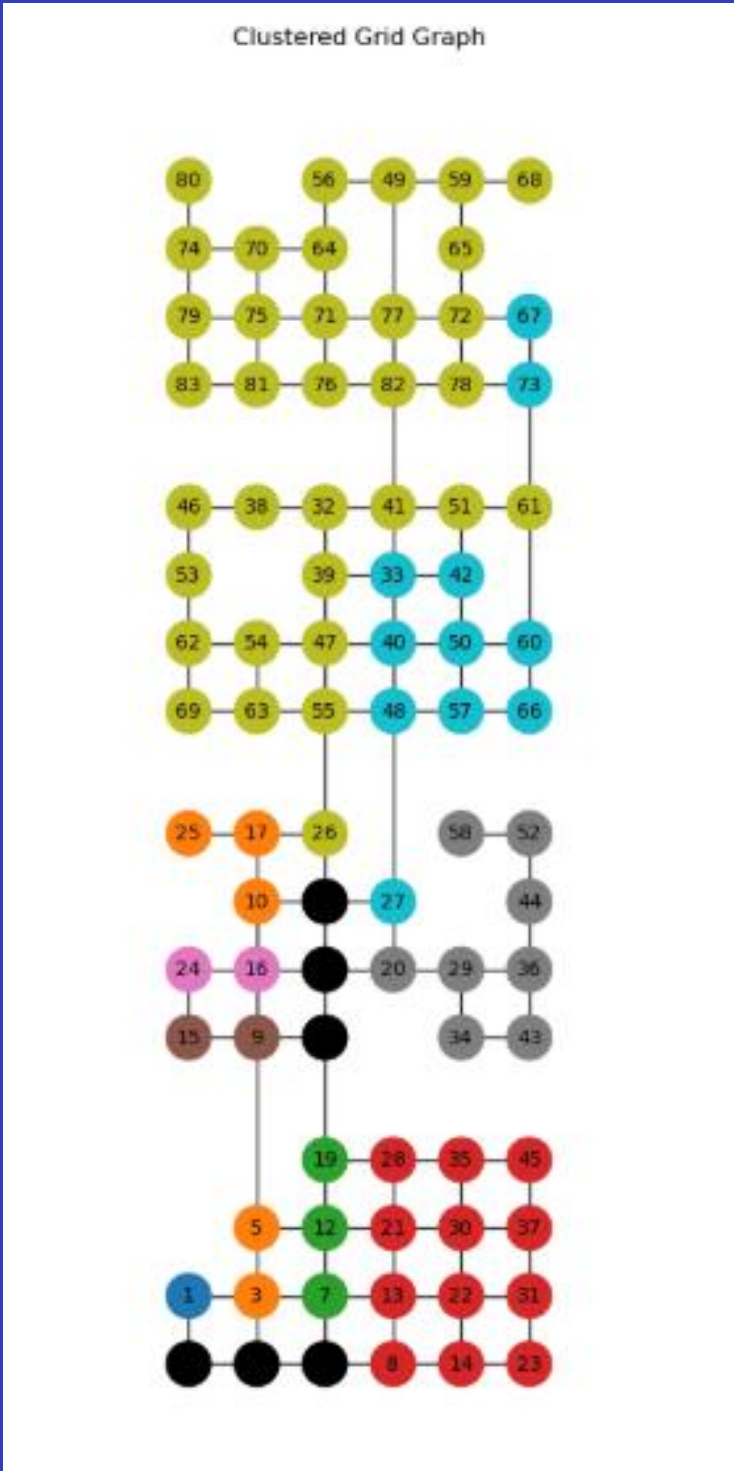
- ✓ Gate까지의 path를 기반으로 어떻게 구역을 만들까?
- ✓ 어떤 차량들을 어떤 구역에 배치할까?
- ✓ 구역 내에서는 어떻게 배치해야 재배치를 최소화할까?

02. 핵심 아이디어

알고리즘 로직



1. Gate까지의 path를 기반으로 구역 만들기



1. No node 정의

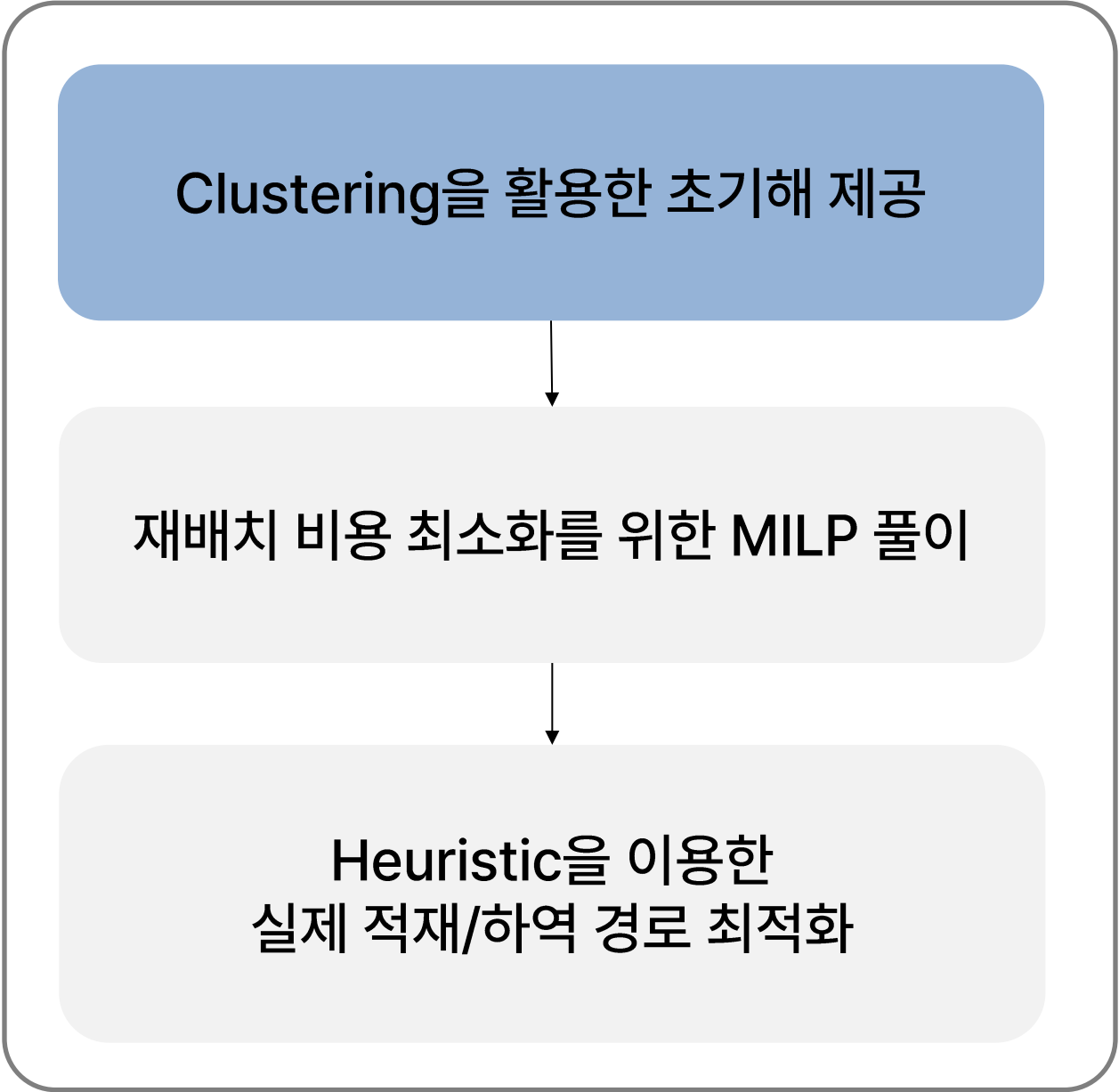
- BFS tree의 leaf node에서 gate로 가는 경로에 포함되는 횟수가 leaf node 개수의 절반 이상인 node
- 최대 7개까지로 개수 제한

2. Clustering

- BFS tree에서의 gate까지의 shortest path 중 No node 이후의 경로 기반으로 clustering 진행
- Cluster 개수 :  $\sqrt{N}$
- Sklearn의 Agglomerative Clustering 사용

02. 핵심 아이디어

알고리즘 로직



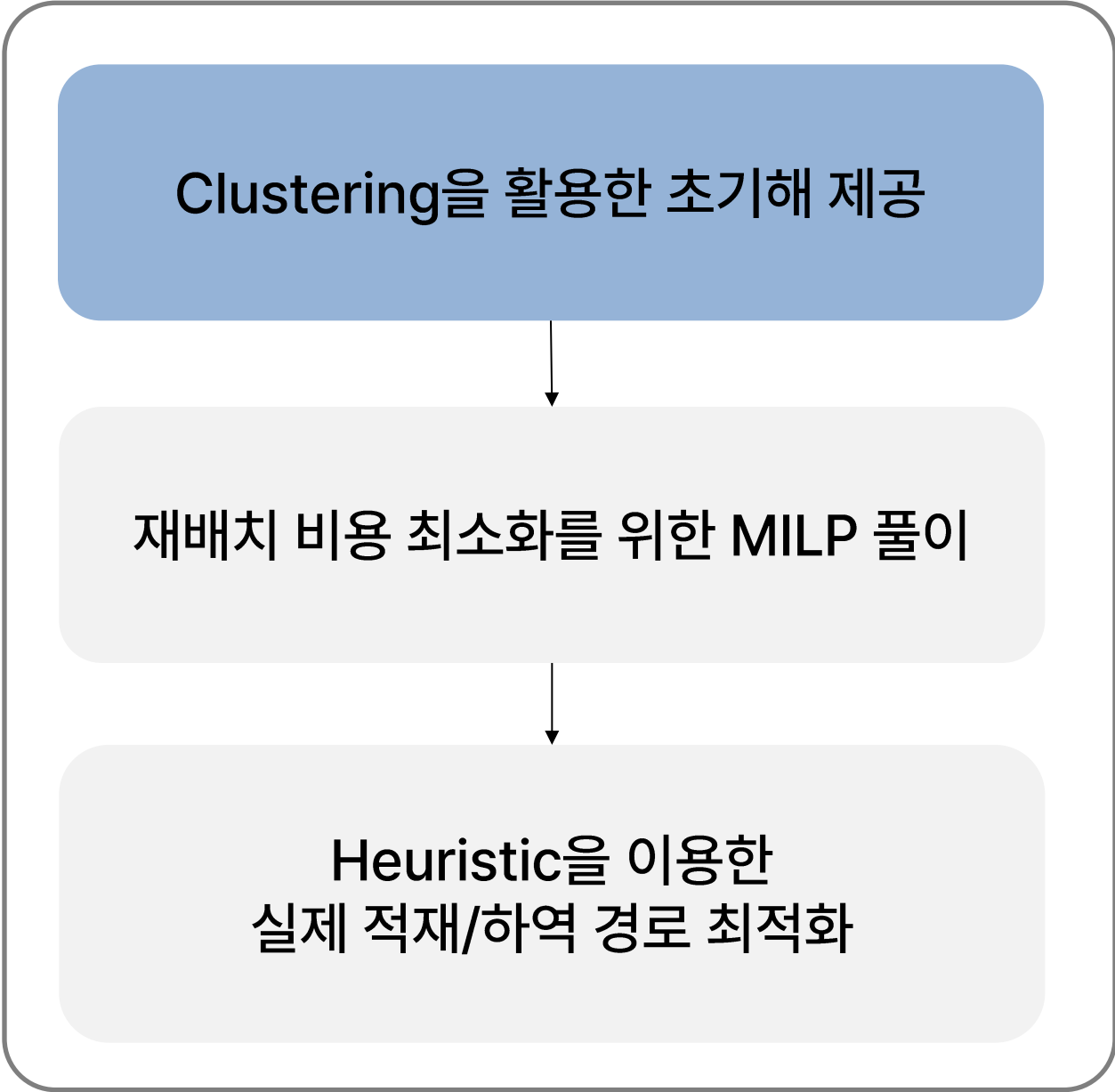
2. 재배치를 최소화 하기 위한 구역 내 배치

Rule : 구역 내에서는 늦게 내리는 차량을 일찍 내리는 차량보다 후면 배치 한다



02. 핵심 아이디어

알고리즘 로직



2. 재배치를 최소화 하기 위한 구역 내 배치

MIP 풀이

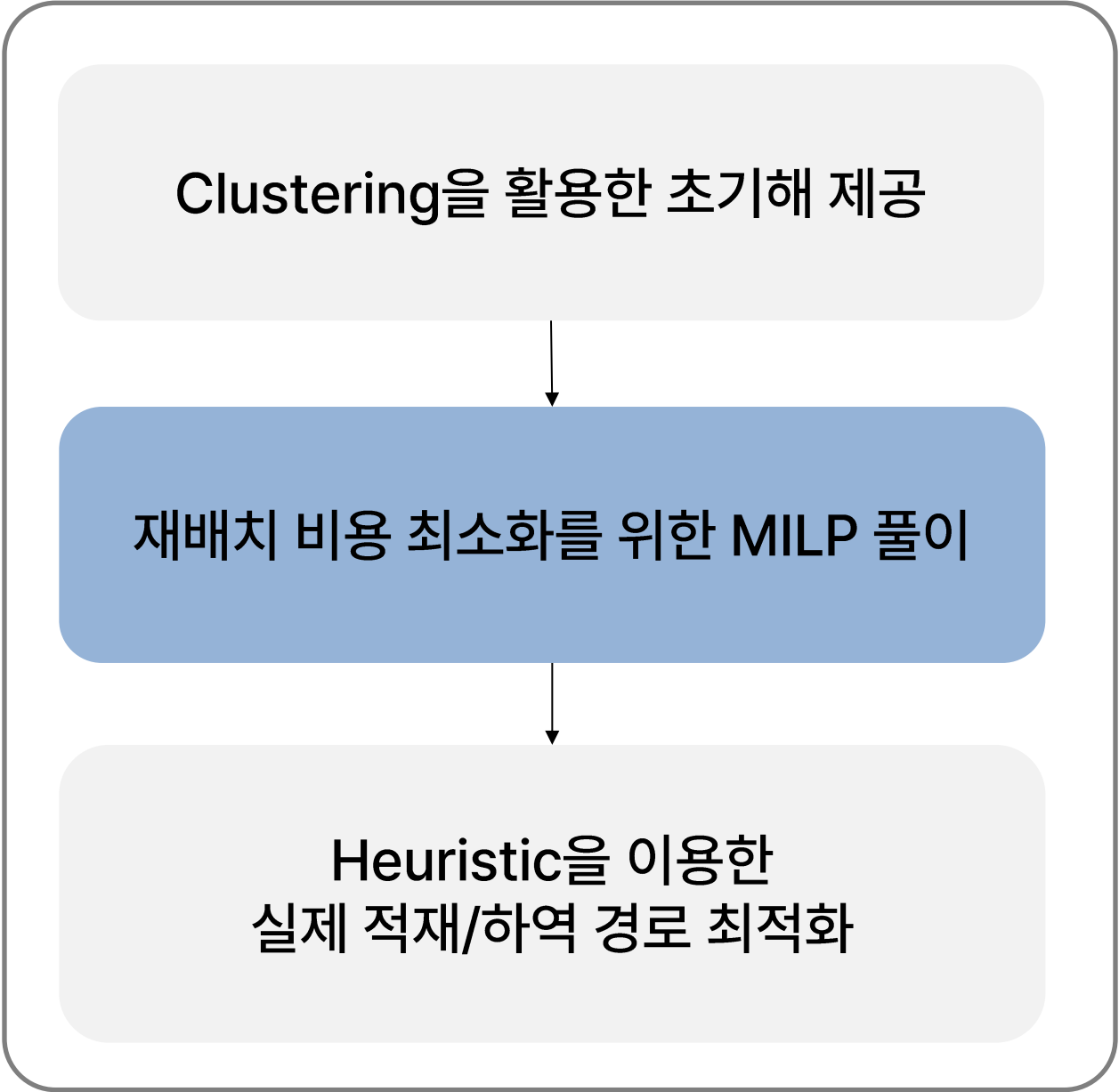
- $Y_{dc}$  : 수요 index d인 차량이 cluster c에 할당된 수량 (INTEGER)
- $Z_{dc}$  : 수요 index d인 차량의 cluster c 할당 여부 (BINARY)
- $RP$  : (일타일내, 늦타늦내) 수요 쌍 집합
- $NP$  : (일타, 늦타) / (늦내, 일타) 수요 쌍 집합

$$\underset{\substack{c \in C \\ (k1, k2) \in RP}}{\sum \sum Y_{k1c} Z_{k2c}} + \underset{\substack{(k1, k2) \in NP}}{\sum 0.1 Y_{k1N} Q_{k2}}$$

같은 cluster 내 conflict 최소화      No node에서 발생하는 재배치 최소화

02. 핵심 아이디어

알고리즘 로직



재배치 비용 최소화를 위한 MILP

결정변수

- $X(k,n)$  : 수요 index  $k$ 인 차량의 node  $n$  할당 여부 (BINARY)
- $Y(n,p)$  : Port  $p$ 에서 node  $n$ 에 위치한 차량의 강제 하역 여부 (CONTINUOUS)
- $A(n,p)$  : Port  $p$ 에서 node  $n$ 의 차량 존재 여부 (CONTINUOUS)
- $Z(n,p)$  : Port  $p$ 에서 node  $n$ 의 적재/하역 경로 포함 여부 (CONTINUOUS)

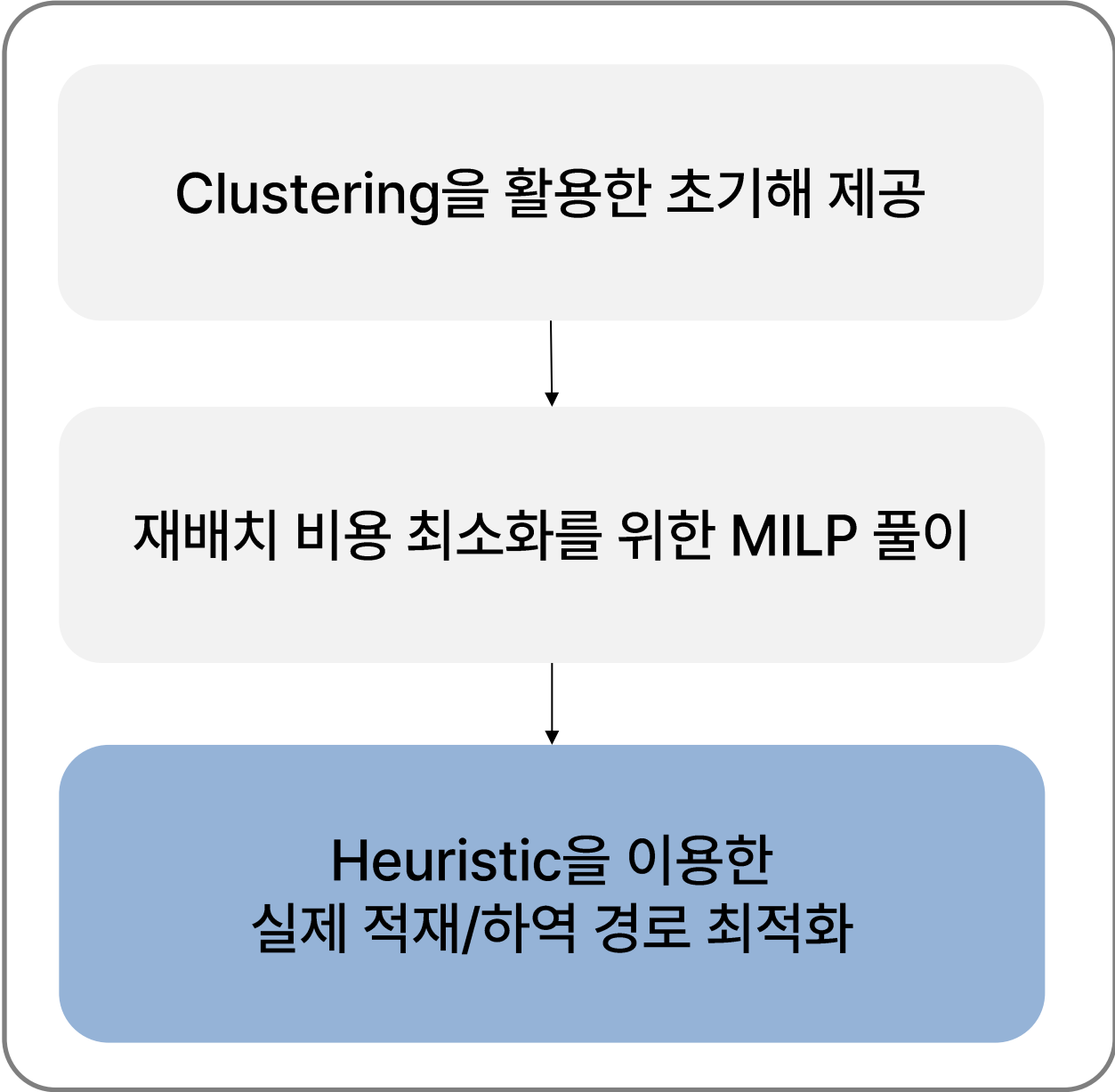
목적함수

$$\underset{\text{재배치 비용}}{\text{minimize } \sum_{n \in N} \sum_{p \in P} F \times Y(n,p)} + \underset{\text{적재/하역 시 경로 길이에 따른 비용}}{\sum_{n \in N} \sum_{k \in K} dist(n) \times X(k,n)}$$

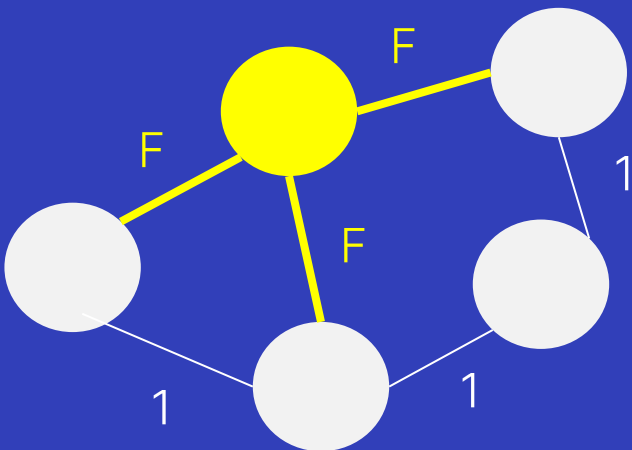


02. 핵심 아이디어

알고리즘 로직



(1) 강제 하역 횟수를 최소화하는 적재/하역 경로 선택



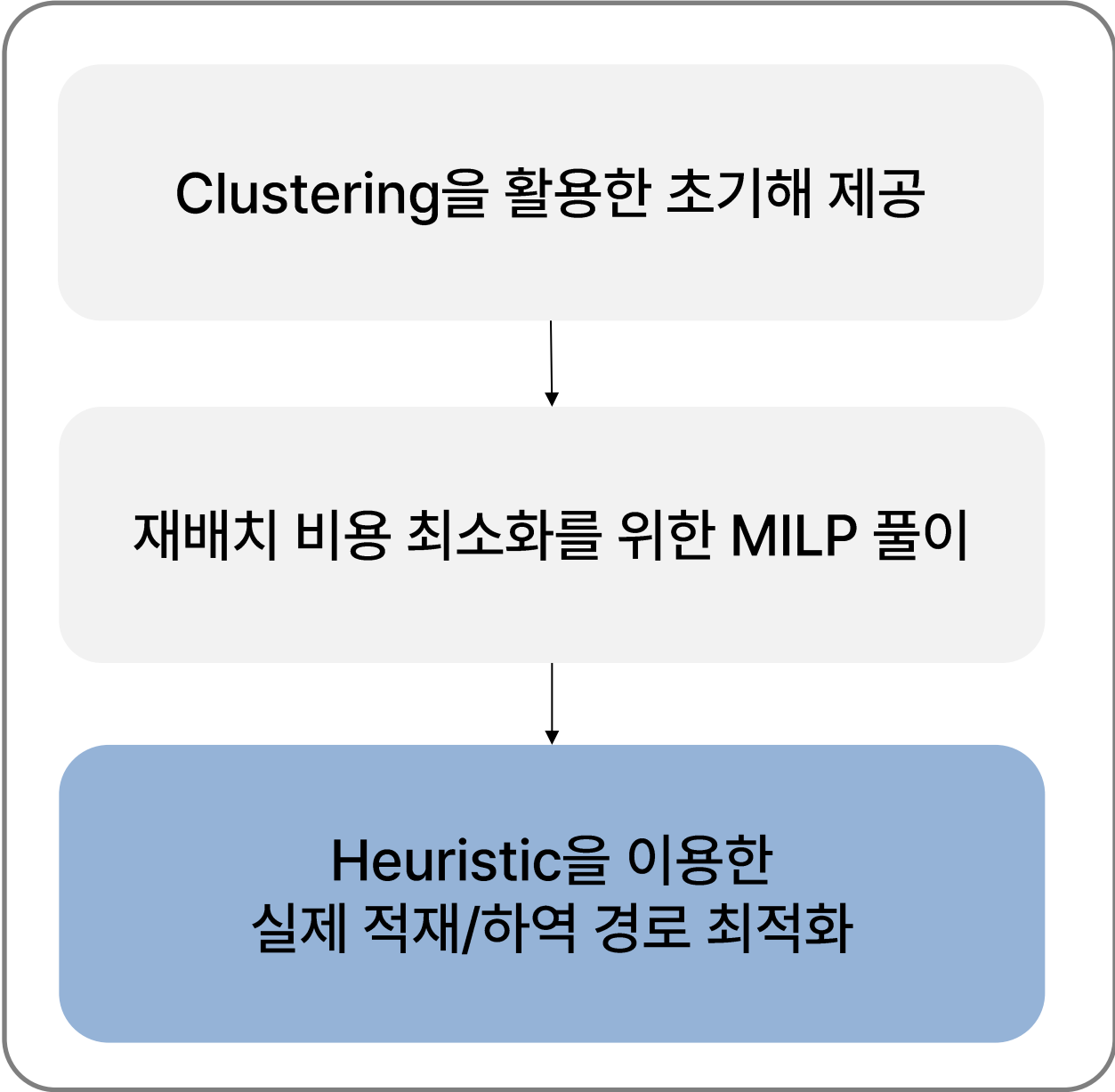
차량이 점유하고 있는 node에  
연결된 edge들의 **가중치 수정**

(2) 하역 시 강제 하역 필수 여부 확인

- (1)에서 도출된 강제 하역 후보군에서 gate에 가까운 순으로  
**강제 하역 vs 기타 노드로 대피 (이후 수요와 충돌 X, 이동 가능) 중 선택**
- 강제 하역 후보군을 1개씩 처리할 때마다 내릴 수 있는 차량 하역

02. 핵심 아이디어

알고리즘 로직



(3) 강제 하역된 차량 재배치

MILP의 기본 가정 : 한 번 배정된 자리는 내릴 때까지 유지한다

강제 하역된 차량을 재 적재할 때 제자리가 아닌 다른 자리에 둘 수 있을까?

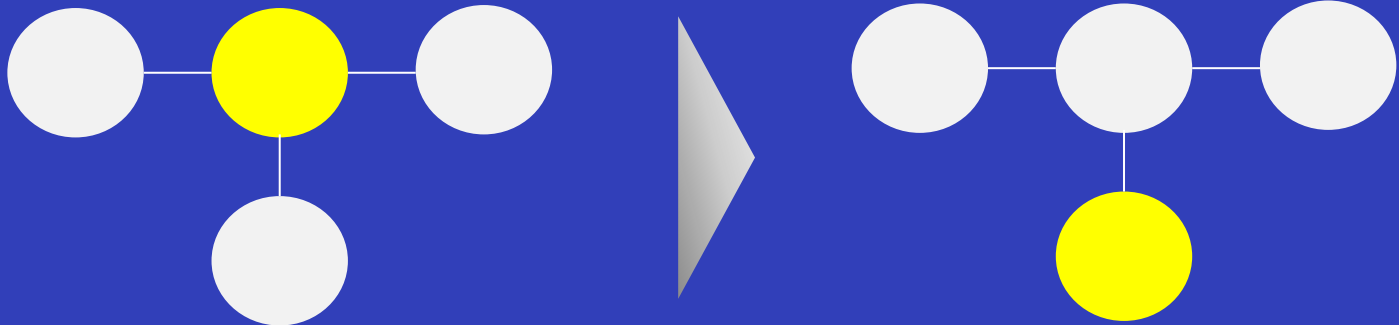
재배치 후보지

- 추후 차량과의 점유 시간 충돌 X
- 추후 차량의 적재/하역 경로 blocking X
- 강제 하역된 port에서 도달 가능



강제 하역된 차량 중 늦게 내리는 차량부터 재배치 여부 확인

강제 하역 재발생을 예방하는 효과를 가짐



강제 하역 재발생 가능성 높음

강제 하역 재발생 가능성 낮음

03. 알고리즘 발전 방향

하드코딩 요소의 일반화  
및 확장성 확보

초기해 목적함수 및 MILP 모형 제약 조건 내 상수를  
데이터 기반 파라미터로 전환

MILP 모형의 충돌 정의 고도화

현재는 최단경로 상에 먼저 타 있는 차량을 충돌로 정의  
대체 경로를 고려하여 충돌 정의 고도화

초기 배치 휴리스틱 다각화

Clustering 기반 초기 배치 외 여러 휴리스틱 적용  
특히 대규모 문제에서 초기해의 품질 중요

## 04. 경진대회 참여 후기

---

### 최적화의 힘을 경험할 수 있었던 기회

- Roll on Roll off 라는 실제 현대 글로비스가 겪고 있는 문제를 접할 수 있어 소중한 경험이었습니다.
- OGC 2024와 다르게 문제의 상황을 수학적 모형으로 표현할 수 있었고 컴퓨팅 속도 높이기/ 병렬화에 많은 노력을 들이지 않아도 가능했다는 점에서 수학적 모형화에 더 많은 노력을 기울일 수 있었습니다.
- 같은 문제여도 어떤 변수, 제약식으로 문제를 표현하는가에 따라 모델의 수렴 속도가 현저히 달라짐을 체감할 수 있었습니다.
- 모형화 이외에도 초기해 제공, 추후 휴리스틱을 통한 모형의 한계점 극복의 과정이 얼마나 중요한지 깨달을 수 있었습니다.

### 건의사항

- 산업공학의 가치를 체감하며 실제 기업의 문제를 풀이해볼 수 있는 OGC 공모전의 지속적 개최 및 네트워킹을 건의드립니다.

# THANK YOU

✉ 이소현 : [yegirna530@snu.ac.kr](mailto:yegirna530@snu.ac.kr)

✉ 강세정 : [sjkang6870@snu.ac.kr](mailto:sjkang6870@snu.ac.kr)