Optimization Grand Challenge 2024

수정 비용 기반 점진적 묶음 생성 접근법

Reduced Cost-based Incremental Pattern Generation Approach for the Pickup and Delivery Problem with Time Window

바른열정청년들

2024. 10.24

목차

묶음배송 최적화 문제

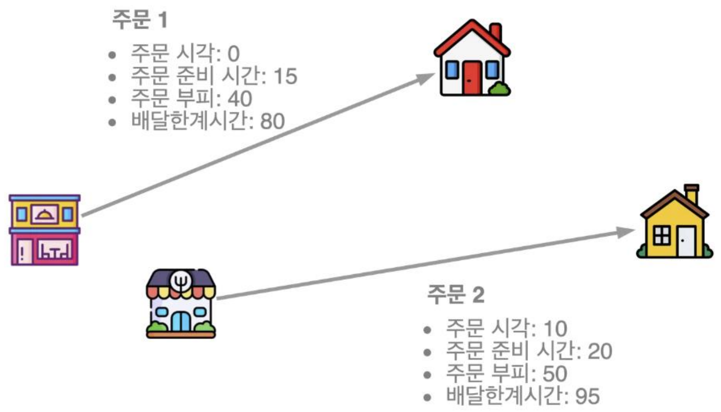
문제 정의

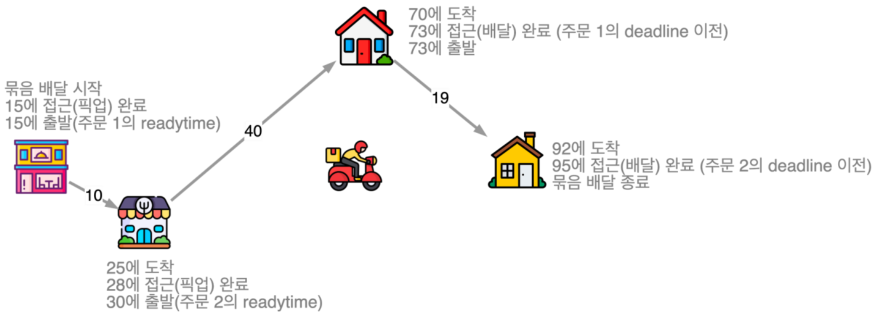
- 묶음배송 최적화 문제
  - 입력 정보: ① 주문 ② 배달원
  - 제약: ① 묶음 배송 제약 ② 주문 만족 제약 ③ 배달원 가용 인원 제약
  - 목표: 평균 배달 비용 최소화

- 주문 $j \in J$
  1) 주문준비시각 (readytime)
  2) 주문 부피
  3) 배달한계시각 (deadline)

- 배달원 $r \in R$
  1) 용량
  2) 속도
  3) 고정/변동비용
  4) 가용인원 n^r

- 묶음 배송 제약
  1) 용량 제약
  2) 시간 제약
  3) 방문순서 제약

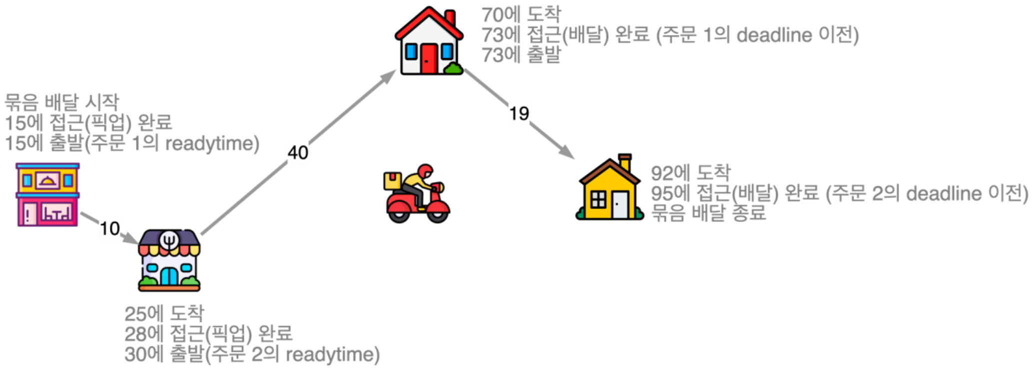




묶음배송 패턴

최적화 모형

- 묶음배송 패턴 $p \in P$
  1. 라이더 r_p
  2. 주문 집합 J_p → **용량 제약 만족**
  3. 픽업/배달 순서 O_p^1 / O_p^2 → **시간 제약 만족**



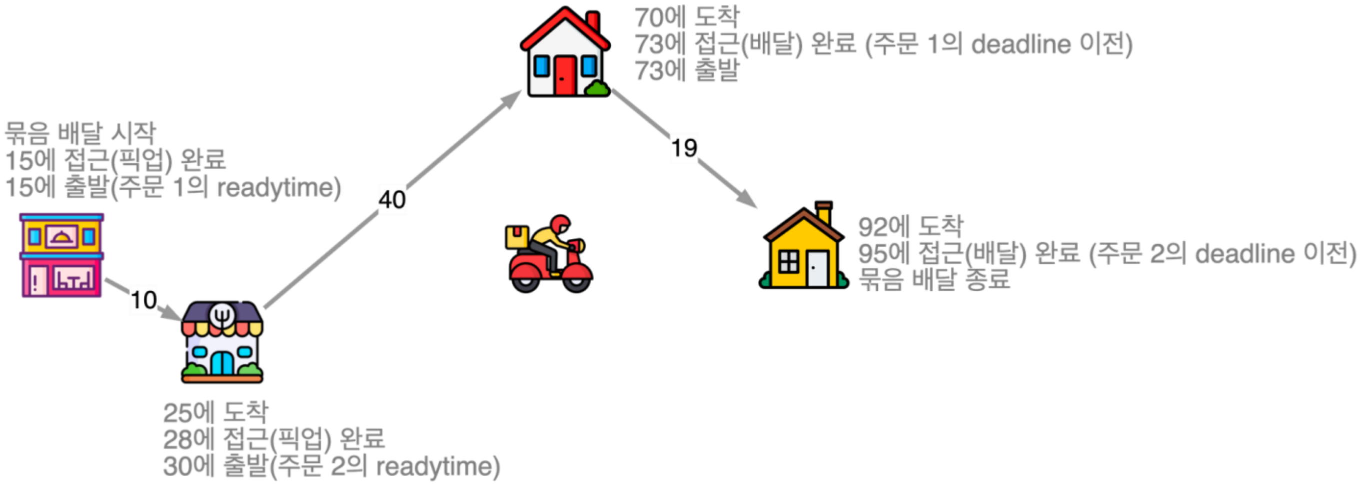
- 묶음배송 패턴 기반 최적화 모형

$$
\begin{aligned}
(\mathcal{P}) \quad \min \quad & \frac{1}{|J|} \sum_{p \in P} c_p x_p && \cdots \ \textbf{평균 배달 비용(=총 배달 비용/ 주문 수)} \\
\text{s.t.} \quad & \sum_{p: j \in J_p} x_p = 1, \qquad \forall j \in J, && \cdots \ \textbf{주문 만족 제약} \\
& \sum_{p: r_p = r} x_p \le n^r, \qquad \forall r \in R, && \cdots \ \textbf{배달원 가용 인원 제약} \\
& x_p \in \{0, 1\}, \qquad \forall p \in P. && \cdots \ \textbf{이진 조건}
\end{aligned}
$$

모형의 계산 부담

최적화 모형

- 가능한지 확인해야 할 주문 조합 자체가 매우 많음
  - 주문 수가 n일 때, 주문 수 k개인 묶음에 대해 $\binom{n}{k}$ 개 조합 고려 필요
- 묶을 주문들의 조합이 결정되어도, 그 안에서 적절한 라우팅 필요
  - 시간 제약
  - 방문순서 제약
- 모든 가능한 묶음을 다 이용하는 것은 현실적으로 불가능 → ★ **<u>"좋은" 패턴을 선별하는 작업 필요</u>**



패턴 생성 전략: 수정비용의 활용

알고리즘 로직

- 수많은 묶음들 중 일부를 선별하는 기준으로써 패턴의 수정비용(reduced cost)을 활용
  - 각 패턴의 선택에 따른 목적함수 변화
  - 묶음배송패턴 기반 모형의 선형계획 완화문제(LP relaxation)를 풀어서 얻을 수 있음
  - 수정비용이 적은 패턴들을 선별하면 비용 측면에서 유리한 패턴 후보들이 나올 수 있음

$$
\begin{aligned}
(\mathcal{P}_{LP}) \quad \min \quad & \frac{1}{|J|}\sum_{p\in P} c_p x_p & & \text{(dual var.)}\\
\text{s.t.} \quad & \sum_{p:j\in J_p} x_p = 1, \quad \forall j \in J, & & (v_j)\\
& \sum_{p:r_p = r} x_p \le n^r, \quad \forall r \in R, & & (w_r)\\
& 0 \le x_p \le 1, \quad \forall p \in P.
\end{aligned}
$$

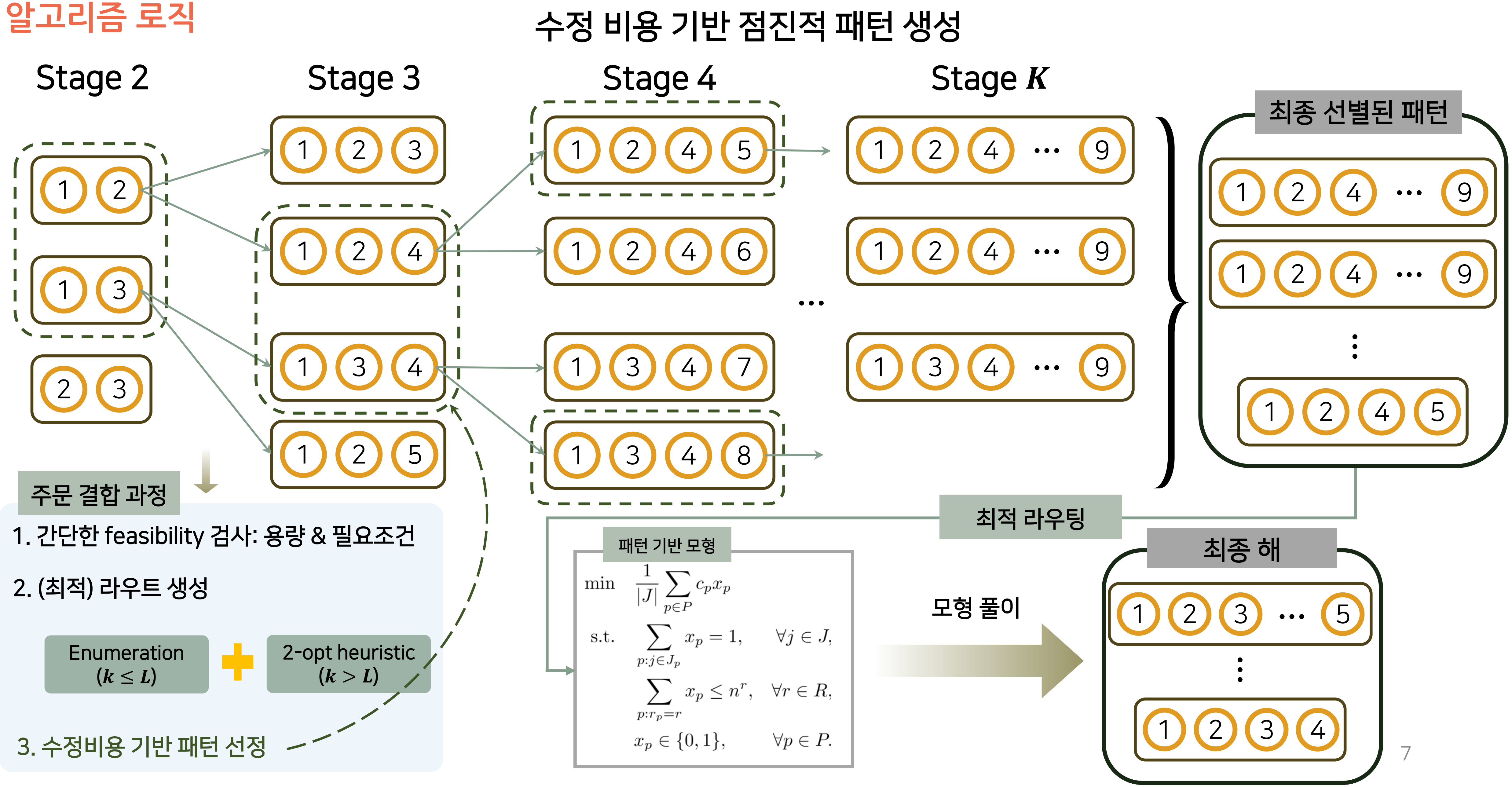
$$\bar{c}_p = c_p - \sum_{j\in J_p} v_j - w_{r_p}$$ ➡ **패턴 p의 수정비용**

➢ 묶음배송비용 – 처리되는 주문들의 가치 – 배달원 사용의 가치

➢ 주문과 배달원의 가치는 해에 따라 달라짐

알고리즘 개요

알고리즘 로직



주문 결합 과정

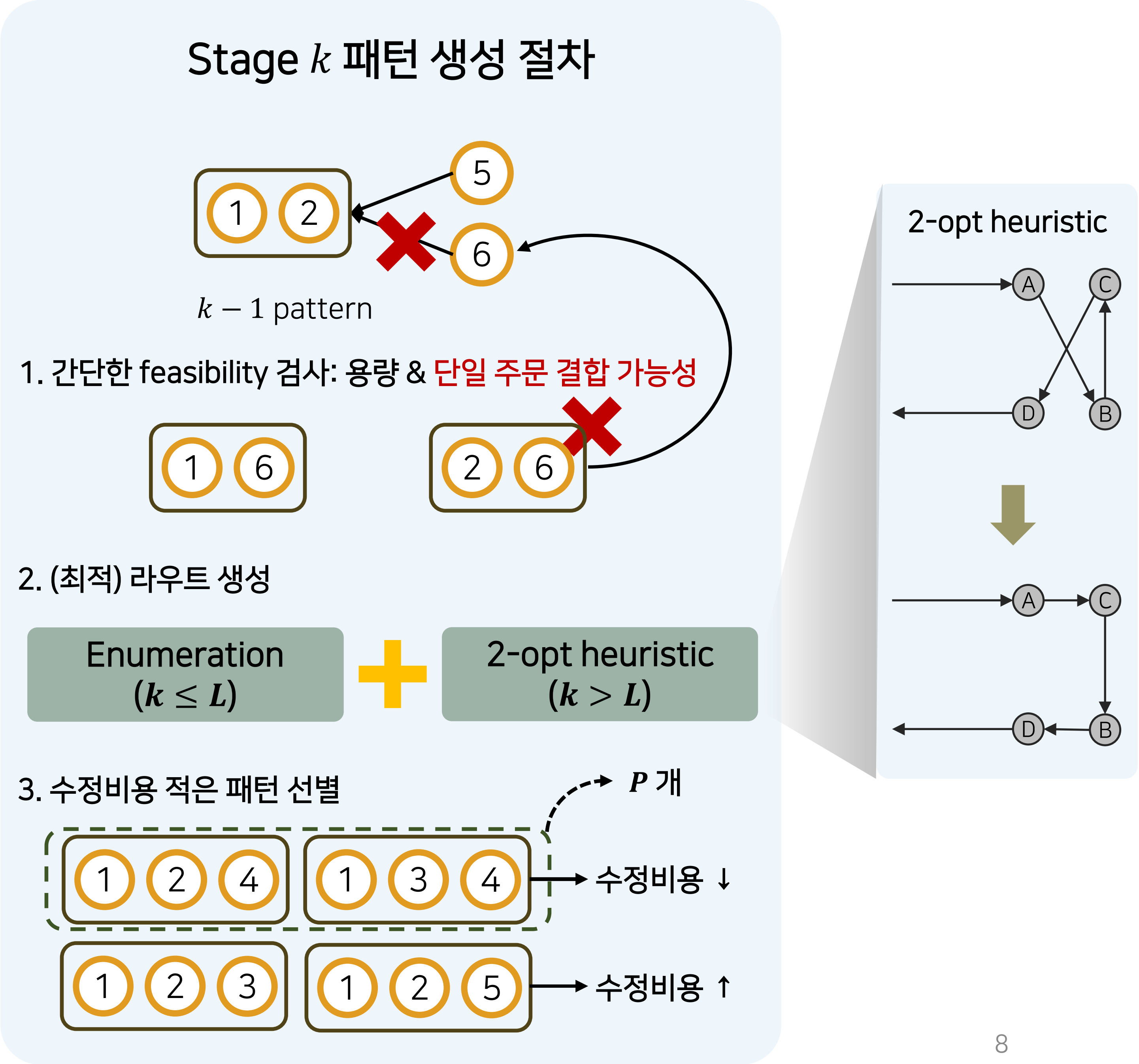
알고리즘 로직

- 용어
  - 패턴 크기: 묶음 패턴에 포함된 주문 개수
  - Stage k : 크기 k인 패턴들을 생성하는 과정

- Stage k 패턴 생성 절차
  1. Stage $k-1$ 에서의 각 패턴에 주문 결합
  2. Feasibility check
  3. 수정비용 기반 패턴 선별

- Feasibility check
  - 용량 & 필요 조건 확인→ 계산 부담 ↓
  - 라우팅: Enumeration + 2-opt heuristic

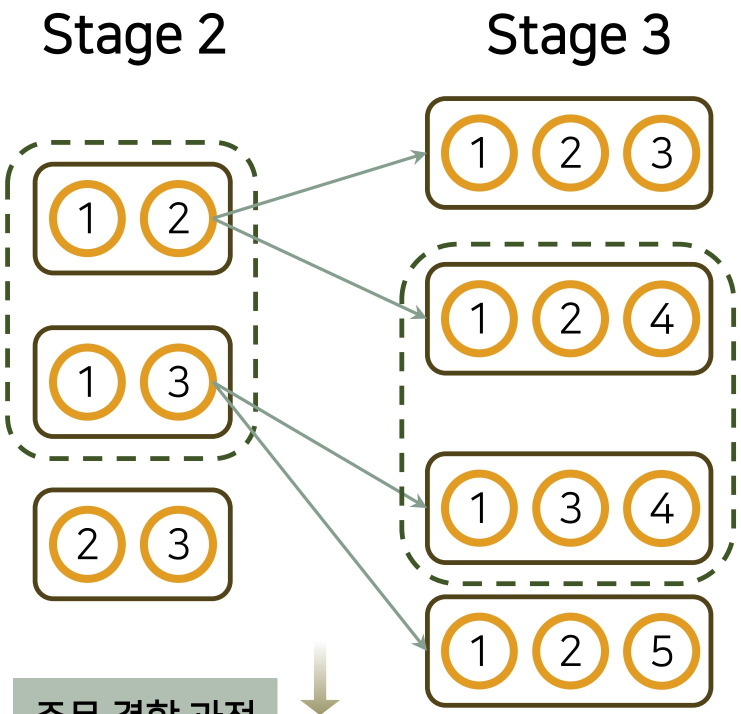
- 알고리즘 파라미터
  1. 생성할 패턴 크기 최댓값 (K)
  2. 매번 선별하는 묶음 개수 (P)
  3. 최적 라우팅하는 패턴 크기 최댓값 (L)
  4. 모형에 최종적으로 포함할 패턴 개수 (M)



구현 세부사항

알고리즘 구현

- **알고리즘 구현**
  - 프로그래밍 언어: C (알고리즘) / Python (데이터 입·출력)
  - 최적화 Solver: Xpress (LP) / Gurobi (MIP)
  - Sorting algorithm: Heap sort
- **계산 속도 개선**
  - **묶음 탐색: 병렬 처리**
  - 수정 비용 도출: Column generation



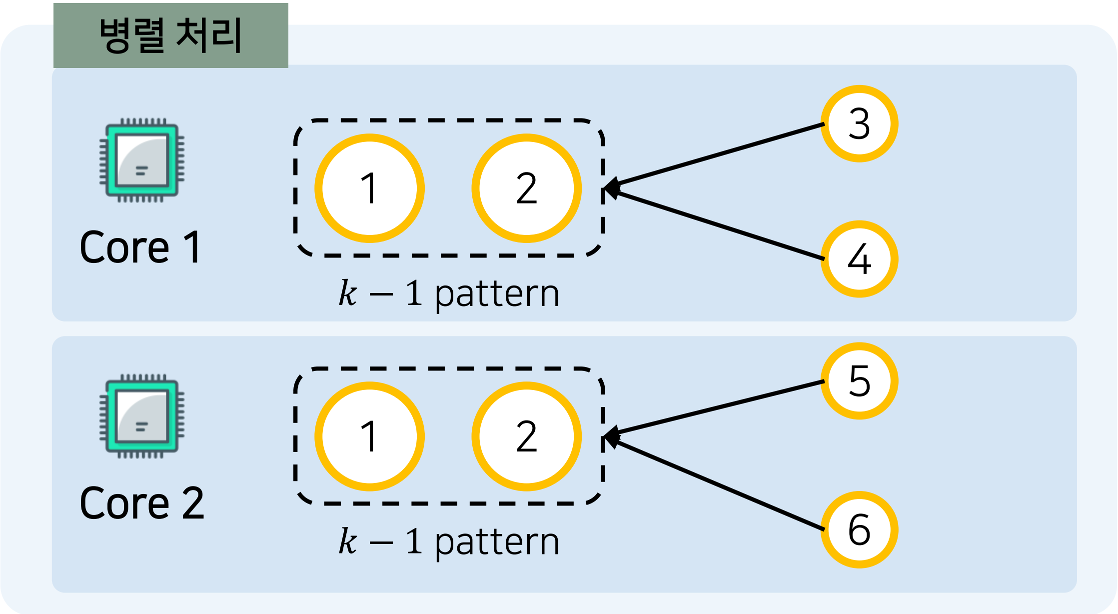
주문 결합 과정

1. 간단한 feasibility 검사: 용량 & 필요조건

2. (최적) 라우트 생성

Enumeration ($k \le L$)	+	2-opt heuristic ($k > L$)

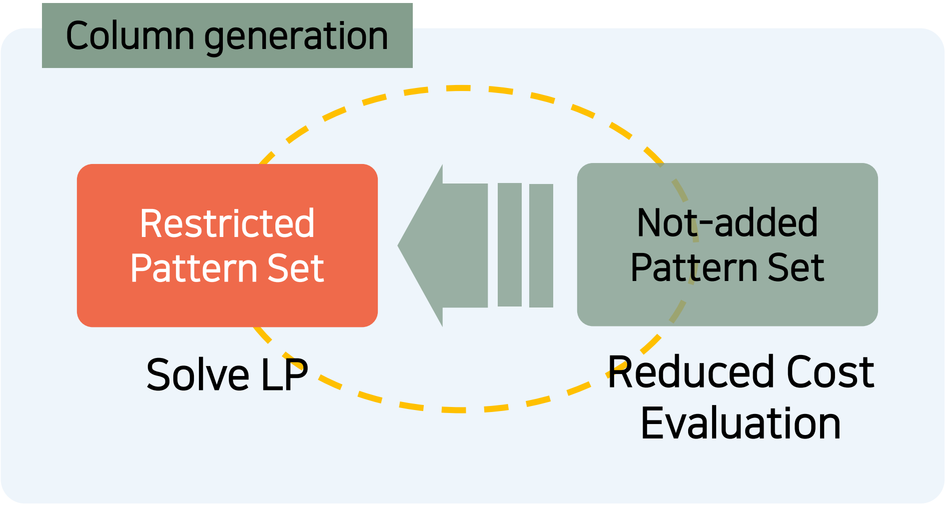
3. 수정비용 기반 패턴 선정

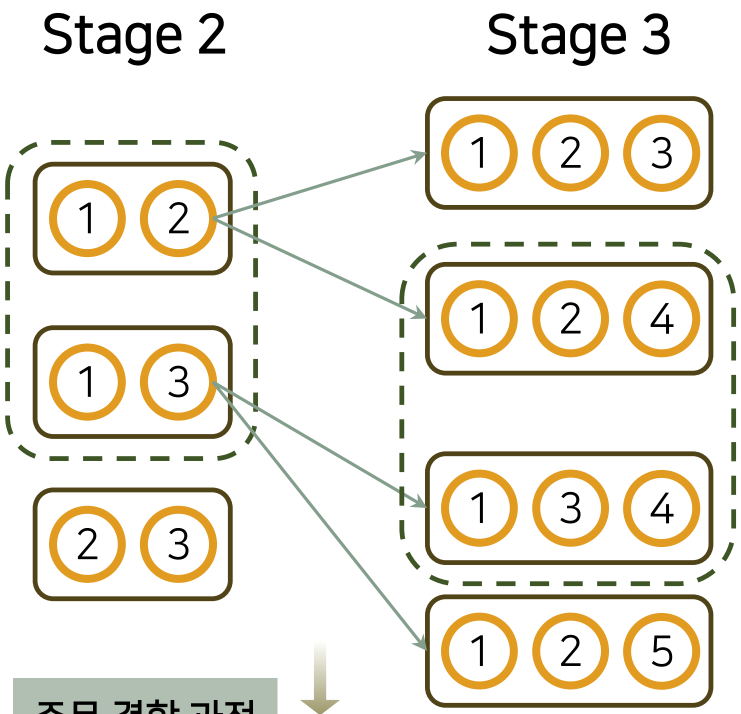


구현 세부사항

알고리즘 구현

- **알고리즘 구현**
  - 프로그래밍 언어: C (알고리즘) / Python (데이터 입·출력)
  - 최적화 Solver: Xpress (LP) / Gurobi (MIP)
  - Sorting algorithm: Heap sort
- **계산 속도 개선**
  - 묶음 탐색: 병렬 처리
  - 수정 비용 도출: Column generation





주문 결합 과정

1. 간단한 feasibility 검사: 용량 & 필요조건
2. (최적) 라우트 생성

Enumeration ($k \leq L$) + 2-opt heuristic ($k > L$)

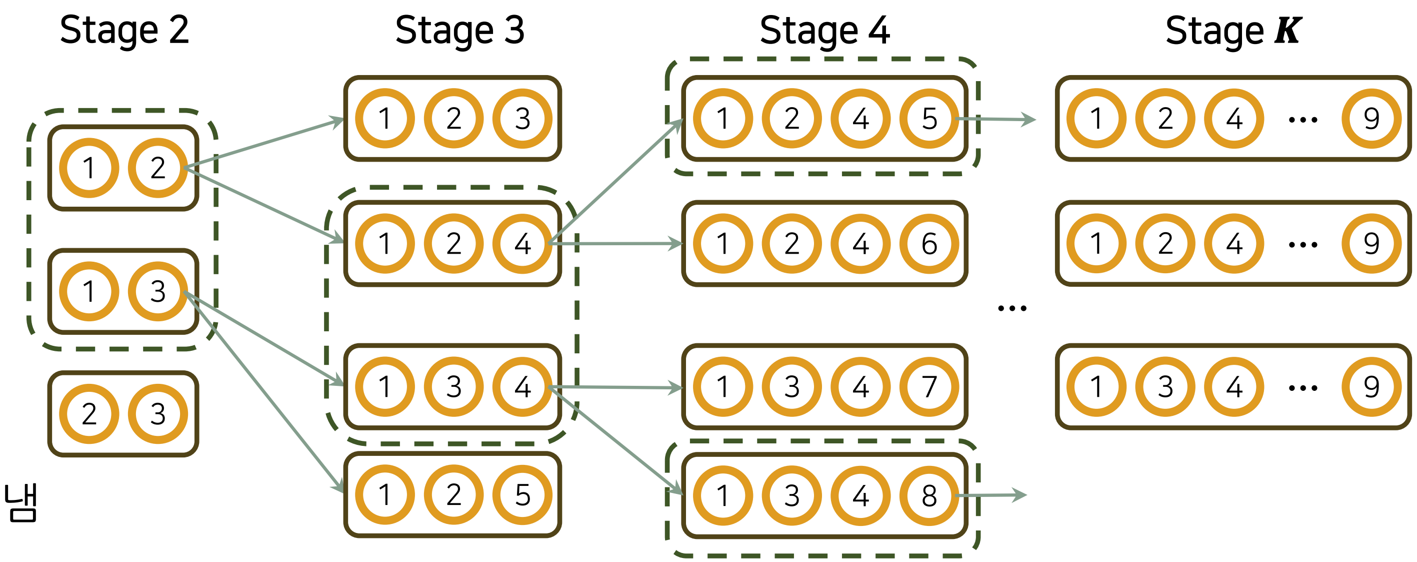
3. 수정비용 기반 패턴 선정

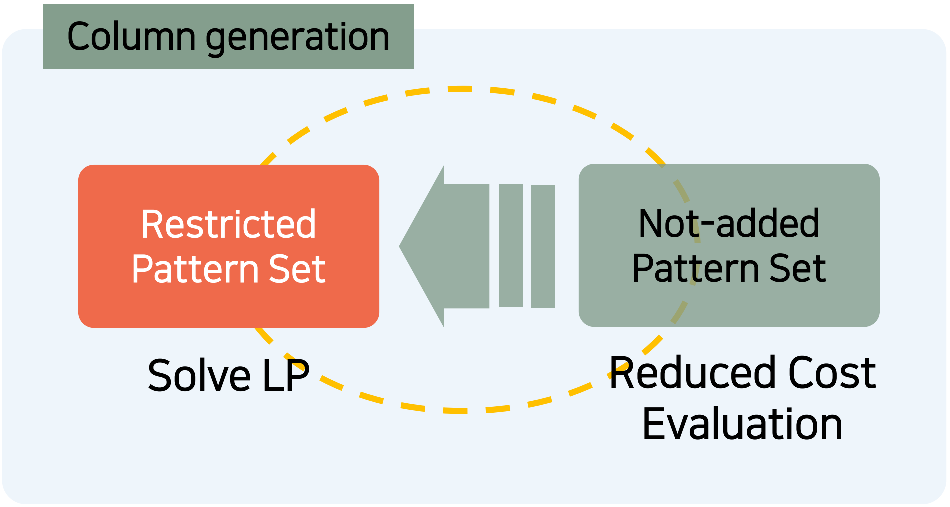
알고리즘 장점

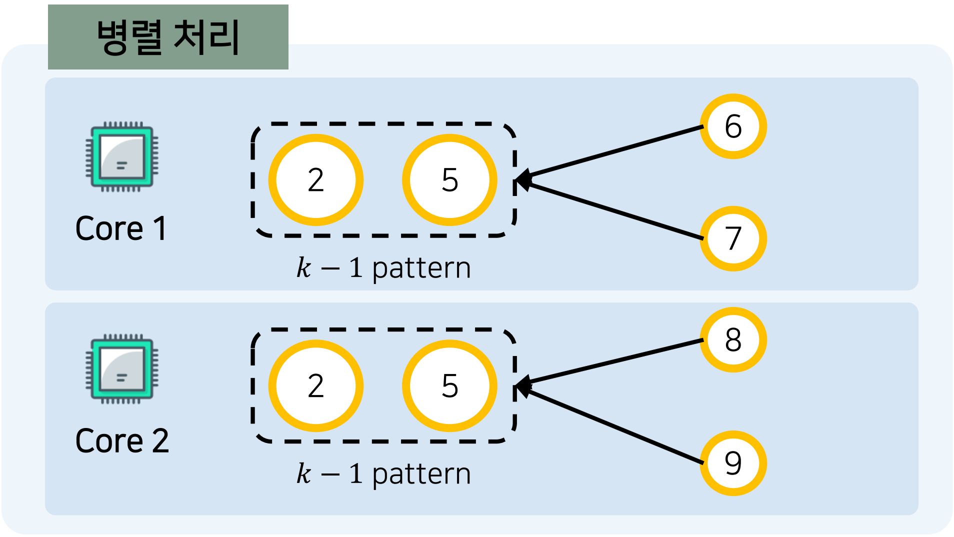
알고리즘 특장점

- **알고리즘 장점**

1. 최적화 이론의 활용
    - 수정 비용, Column generation, 2-opt
2. 실용적인 구현 방식
    - 병렬 처리, 필요조건을 활용한 계산 부담 단축
3. 파라미터들을 충분히 늘리면 최적해 구할 수 있음
    - 주어진 제한시간내에 Gap 2% 이내의 답을 찾아냄







요약 및 추후 개선 방향

결론

▪ **요약**

- 수정 비용에 기반한 효율적인 패턴 생성
- 병렬 처리 등 활용한 계산 속도 단축

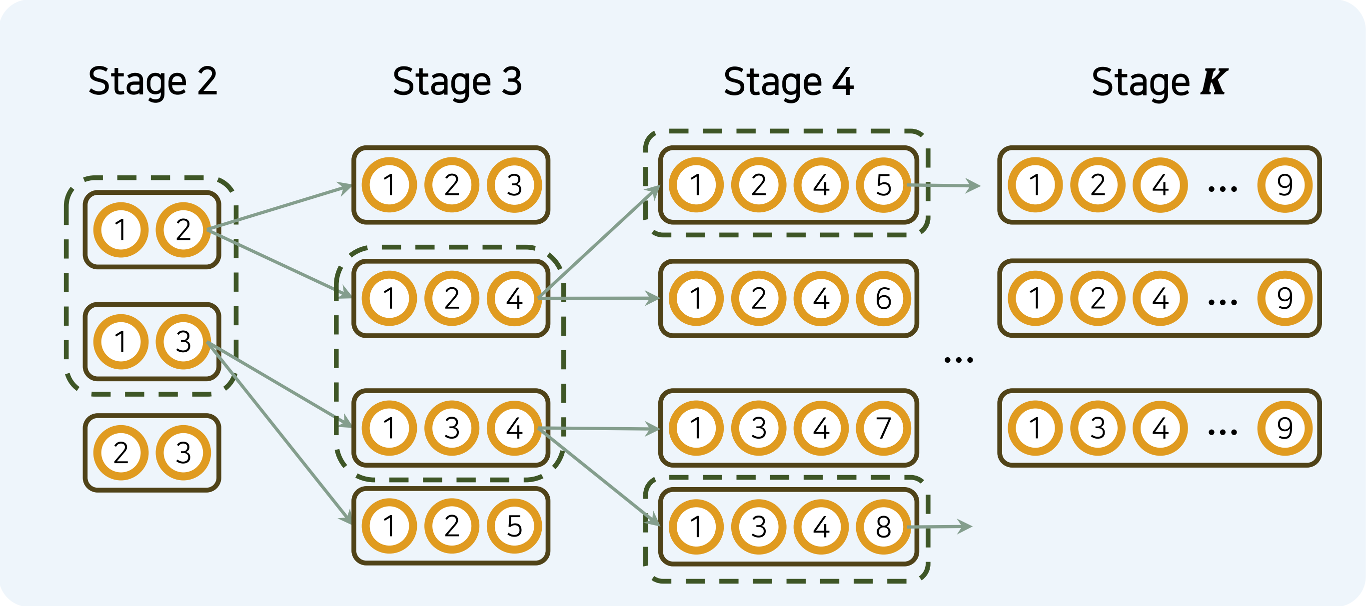
▪ **추후 개선 방향**

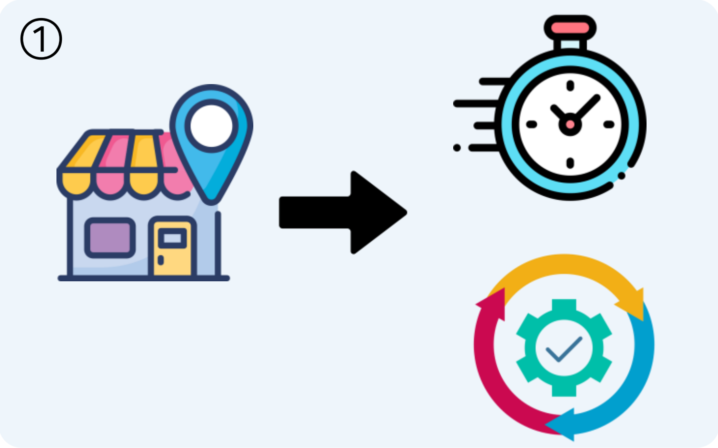
① 주문들의 위치 정보 활용
  ➢ 묶음 생성 효율성 제고
  ➢ 파라미터 튜닝 자동화

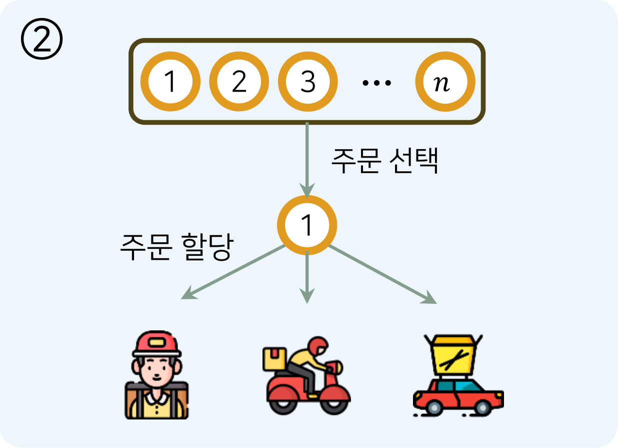
② 근사 동적 계획 알고리즘
  - 주문에 대한 순차적 의사결정
  - 패턴 기반 모형을 통한 가치 함수 근사

▪ **소감**

- 컴퓨터와 최적화 Solver의 연산 속도↑ → **범용적 문제해결도구로써 최적화의 활용도↑**







Q & A

감사합니다