



한국의 거주지 및 식당의 위치적 특성을 고려한 배달 플랫폼의 할당 정책 제안

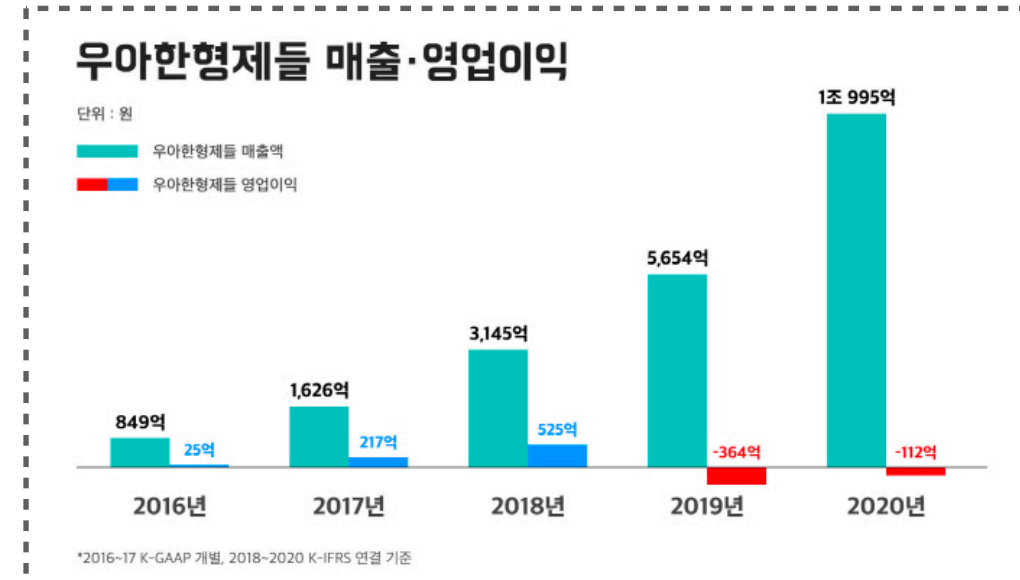
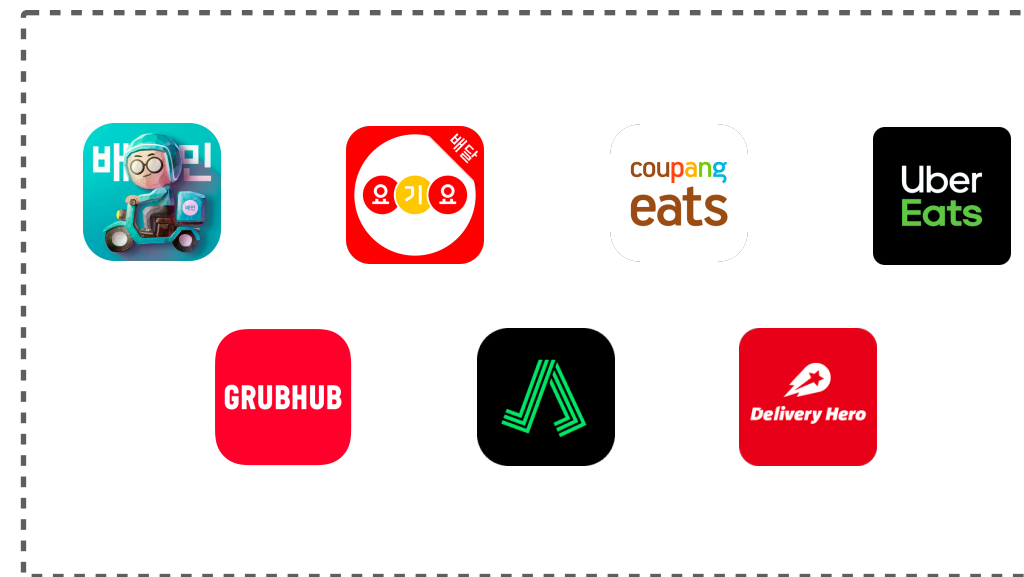
군집화 기반 주문 묶음 처리, 지연 정책, 계층적 주문 처리, 군집화를 활용한 할당 알고리즘

박해민, 한기현

① Problem Introduction

1. 문제 소개 : Routing Problem in Delivery Platform

◆ 치열해지는 음식 및 퀵커머스 배달 플랫폼 시장



◆ 배달 시스템(플랫폼)의 문제점

사용인구와 시장 규모 증가를 따라가지 못하는 배달 시스템 정책의 문제점들

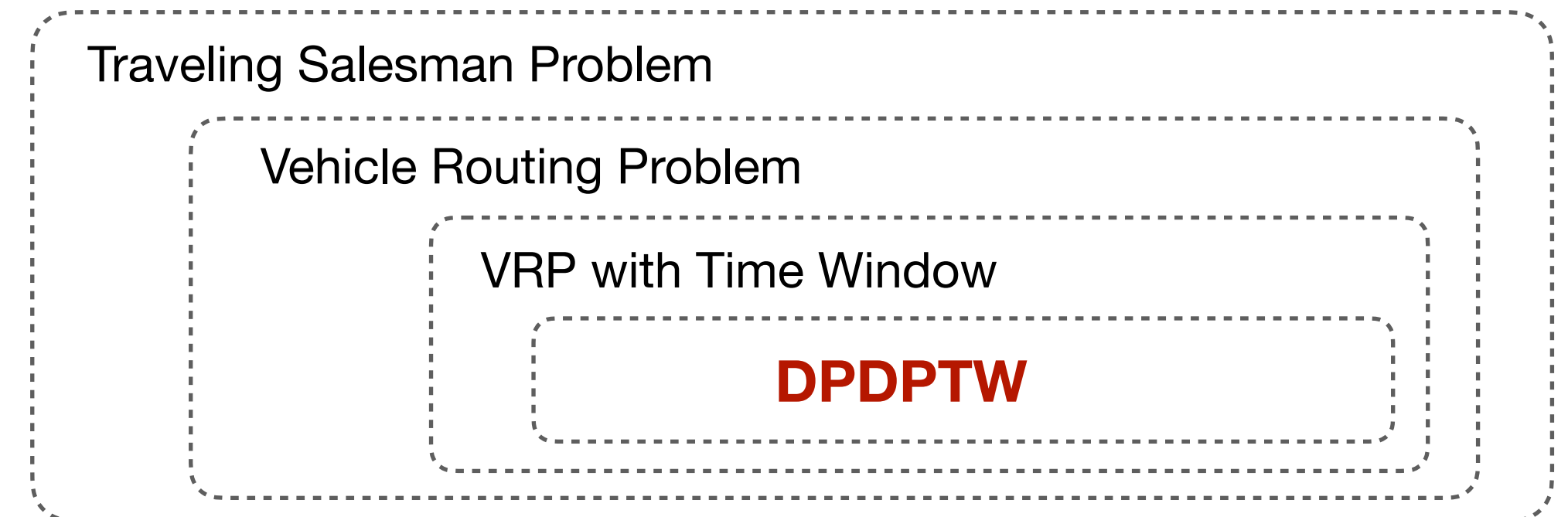
- ✓ 시장 점유율 경쟁으로 인한 비효율적인 배달 시스템 구현
- ✓ 비용이 다른 플레이어들에게 분담되는 배달 정책 (배달 플랫폼들 사이의 점유율 경쟁으로 인한 비용 증가가 플레이어들에게 분담)
- ✓ 여러 갈등의 발생과 이를 해결하기 위한 공공 배달어플의 등장 및 실패

◆ 배달 플랫폼 시장의 중요성

- ✓ COVID-19로 가속화 된 배달 시장의 성장
- ✓ 사람들의 (식)생활에 가장 영향력이 강한 플랫폼
- ✓ 여러 플레이어(플랫폼, 식당, 배달기사, 소비자)의 필요와 요구를 적절히 만족시켜야하는 복잡한 시스템

- 1) 1주문-1배달기사(단건주문) 정책으로 배달 플랫폼 기업도 안정적인 운영에 어려움을 겪고, 배달 기사도 평균 수익이 떨어지며, 소비자는 늦은 배달을 받는 등 **모든 구성원이 만족하지 못하는 현재의 배달 시장**
- 2) 배달 플랫폼 업체들도 묶음 배송을 고려하되, 이전 묶음 배송을 시행하던 시절 발생했던 문제(배송 시간 초과)를 해결한 더욱 발전한 배송 정책을 개발할 필요성이 있음

◆ 문제 구분 : Dynamic Pickup and Delivery Problem with Time Window



OD(Order Dispatching) + VR(Vehicle Routing)의 특성을 가진 문제

- ✓ 실시간으로 들어오는 주문에 대해서 픽업장소로 이동하여 주문 상품을 픽업한 이후 배달지로 이동하여 배송을 완료해야 함
- ✓ 주문이 실시간으로 확률적으로 들어오며
- ✓ 각각의 주문은 특정 시간(데드라인) 안에 배송을 완료해주어야 함
- ✓ 플랫폼은 각각의 주문을 효율적으로 배송할 수 있는 배달기사에게 할당해주어야 하는 할당문제를 매 시간 풀어야함

◆ 왜 어려운 문제인가?

- (1) ONLINE
- (2) ROUTING WITH TIME WINDOW
- (3) CURSES OF DIMENSIONALITY
- (4) UNCERTAINTY IN PROCESSES
- (5) NP-hard

① Problem Introduction

2. 연구 목표

◆ DPDPTW는 굉장히 어려운 문제

- DPDPTW는 굉장히 computational하게 어려운 문제로, 사실상 이론적인 방법으로 해결하는 것이 불가능함 ➡ 따라서 휴리스틱에 기반을 둔 배달 정책을 찾아야 함
- 시간에 따른 여러 배달 과정들의 불확실성들을 모두 고려한 DPDPTW는 Ulmer et al. (2020)과 Reyes et al. (2018) 두 논문에서만 다루어짐
- 하지만 위의 논문들은 몇몇 부분에서 실제 적용가능성이 떨어짐 (대규모 상황에서 적용 불가 등)

◆ 연구의 목표

- 시간에 따른 불확실성을 고려한 배달 정책 제안
- 실제 배달 플랫폼에 적용 가능한 배달 정책 제안
- Computational한 문제를 고려한 배달 정책 제안
- 한국의 배달 시장 특성(공간적 특성)을 고려한 배달 정책 제안

◆ 실제 운영의 고려사항

- 1-1 주문 배달 방식은 많은 배달기사 수를 필요로 함
- 주문의 수가 배달기사의 수에 비해서 훨씬 많을 때, 처리하지 못하는 주문이 발생할 수 있음 (배달 시스템 안정을 위해 필요한 배달기사 수가 많이 필요함)
- 배달기사의 경우 1-1 주문 배달 방식으로 진행할 경우 bundling을 채용하는 배달 방식에 비해 기대 수익이 떨어짐
- 배달기사에게 할당 이후 취소하는 등의 비현실적인(다른 플레이어를 고려하지 않는) 배달 정책은 배제되어야 함
- 효율적이고 실행가능한 policy의 필요성

◆ 문제의 기본 가정

- (1) 배달시간은 거리에 비례함 (속도 : 5m/s)
- (2) 배달거리는 (직선 거리 X 1.4배)로 정함 (Boscoe et al. (2012))
- (3) 주문의 발생은 포아송 과정을 따름
- (4) 서비스 시간(식당, 배달지)은 상수
- (5) 배달기사는 주문의 할당 이후 항상 최적의 경로로 이동함
- (6) 배달기사는 루트가 정해지지 않으면 움직이지 않음

온라인 + 주문 할당 + 배달 라우팅이 결합된 굉장히 복잡한 문제를

1) 위치-군집화와 지연전략을 활용한 주문 묶음처리와 2) 동적인 운영변수 설정을 활용한 정책으로 해결

① Problem Introduction

3. 선행 연구 : Literature Review

◆ Problem Specifications

- Deadline

주문의 데드라인을 고려한 문제인가? 각각의 주문은 특정 시간(데드라인)안에 도착해야 함. 실제 배달에서도 소비자들은 특정 골든 타임을 넘기면 해당 주문에 대해 만족도가 급격하게 감소.
- Scalability

다량의 변수에 대해 빠른 시간에 해결 가능한가? Vehicle Routing Problem의 엄청난 계산량의 문제를 빠른 시간내에 해결가능한 알고리즘을 제안해야 함. 실제 배달 플랫폼에서는 몇 초 내에 의사결정이 이루어져야 함
- Objective

논문의 초점이 배달 주문 고객의 서비스에 초점을 맞추었는가? 배달 플랫폼은 배달 기업이나 플랫폼의 이윤만을 고려해서는 안되고 여러 플레이어의 만족도를 모두 고려한 의사결정 policy를 수립해야 함.
- Dynamism

수리적으로 최적화된 패러미터 사용하였는가? 온라인 문제이기에 문제의 상황과 환경은 매번 변하고, 이에 따라 알고리즘의 패러미터들도 해당 상황에 최적화된 값을 사용해야 함. 즉, 최적화한 것을 사용한 것인지 혹은 그저 Heuristics에 의한 상수를 사용했는지 구분
- Bundling

주문 묶음 처리(Bundling)를 고려하였는가?
1 배달기사 - 1 주문 배달 정책은 여러 상황에서 비효율적일 수 있음
- Postponement

주문 할당의 지연을 고려하였는가?
* **지연** : 새로 들어온 주문을 바로 배달기사에게 할당해주지 않고, 일정 시간까지 시간을 끌며 많은 주문을 고려하여 더 좋은 할당을 해주는 전략

◆ 사전 연구 조사

(IP : Integer Programming, ADP : Approximate Dynamic Programming)

| | Solution Approach | | Model Specification | | | | | |
|-------------------------------|-------------------|-----|---------------------|--------------|----------|----------|-----------|-------------|
| | IP | ADP | Deadline | Postponement | Bundling | Dynamism | Objective | Scalability |
| Reyes et al. (2018) | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Ulmer et al. (2020) | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| Steever et al. (2019) | ✓ | | ✓ | | | ✓ | ✓ | ✓ |
| Yildiz and Savelsbergh (2017) | ✓ | | ✓ | | ✓ | | ✓ | ✓ |
| Our Study | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Online 특성을 고려한 DPDPTW는 Ulmer et al. (2020)과 Reyes et al. (2018)에서만 연구 진행 (ADP 활용)

| | Reyes et al. (2018) | Ulmer et al. (2020) |
|-------------|---|---|
| 주 아이디어 | Rolling Horizon Approach | Postponement + Time Buffer |
| 목적함수 | (시간 당 주문처리 개수) - (음식의 신선도 고려한 penalty 항)을 최대화 | 시간 딜레이의 총합을 최소화 |
| Bundling 기법 | 동일한 식당에 대한 주문 묶음을 고려 | 시간 딜레이를 최소화하는 주문 묶음 고려 |
| 한계 | 주문 묶음 처리시 배달지의 공간적 위치 고려 X. 비직관적인 목적함수 (penalty 항의 상수곱의 객관적 증거 부족) | 계산적인 문제 때문에 큰 규모에는 postponement 고려 어려움. (배달 기사 15) |

① Problem Introduction

4. 연구 문제 정의 : Dynamic Pickup and Delivery Problem with Time Window (DPDPTW)

◆ 문제 상황 설명

- (1) 서비스 시간 $[0, T]$ 동안 주문이 arrival rate λ 의 포아송 과정으로 서비스 공간 \mathcal{S} 에서 랜덤하게 발생하며, 서비스 공간에는 \mathcal{N} 명의 배달기사가 활동한다.
- (2) 각각의 주문에 대해서 이를 효율적으로 배달할 수 있는 현재 주문이 할당되지 않은 배달기사에게 할당시켜야 한다. **(OD + VR 문제)**
- (3) 배달기사는 주문이 할당된 순간 정해진 최적의 루트를 따라 식당과 배달지를 방문한다.
- (4) 배달 플랫폼은 **전체 서비스 질을 유지하면서 모든 플레이어가 만족할 수 있는 배달기사-주문 매칭**을 해주어야 한다. **(VR 문제)**

◆ 수리적 모델링

주문을 배달기사에게 할당해주어야 하는 결정 타이밍(이후 알고리즘에서 정의)마다 다음의 문제를 해결해야 한다.

<각 결정 시간 t마다 주어지는 변수>

$O = \{o_i : i \in I\}$: 주문의 집합

$C = \{c_j : j \in J\}$: 배달기사의 집합

$\mathcal{Q} = \{(o_1, \dots, o_m) \in O^m\}$: 가능 루트(묶음 주문)의 집합 : 주문들의 순서쌍의 집합
(주문들의 묶음과 이에 할당된 배달기사가 결정되면 최적의 루트는 자동으로 정해진다)

$\delta_{o,c}^q$: 루트 $q \in \mathcal{Q}$ 에 있는 주문 o 가 배달기사 c 에 할당되었을 때 배달완료 시간

위와 같은 주문과 배달기사의 세팅이 주어진 상황에서 아래의 결정변수를 결정해야 함

<결정변수>

$x_{q,c} \in \{0,1\}$: 루트 $q \in \mathcal{Q}$ 가 배달기사 $c \in C$ 에 할당되면 1, 아니면 0

$y_o \in \{0,1\}$: 주문 $o \in O$ 가 어떤 배달기사든지 할당되었으면 1, 아니면 0

◆ Problem Formulation at each decision timing t

*Extended Integer Programming Formulation from Reyes et al. (2018)
(Reyes et al. (2018)의 문제를 해당 연구의 specification을 고려하여 변형한 IP)

$$\min_{x,y} \sum_{c \in C} \sum_{q \in \mathcal{Q}} x_{q,c} (\max_{o \in q} \{\delta_{o,c}^q\} - t)$$

$$s.t. \quad \sum_{q \in \mathcal{Q}} x_{q,c} \leq 1 \quad \forall c \in C \quad (1)$$

$$\sum_{c \in C \cup \{0\}} x_{q,c} = 1 \quad \forall q \in \mathcal{Q} \quad (2)$$

$$\sum_{c \in C} \sum_{q \in \mathcal{Q}(o)} x_{q,c} \leq 1 \quad \forall o \in O \quad (3)$$

$$\sum_{c \in C} \sum_{q \in \mathcal{Q}(o)} x_{q,c} = y_o \quad \forall o \in O \quad (4)$$

$$\sum_{q \in \mathcal{Q}} \sum_{c \in C} x_{q,c} \geq \min(|C|, |\mathcal{Q}|) \quad (5)$$

$$x_{q,c} \in \{0,1\} \quad \forall q \in \mathcal{Q}, \forall c \in C \cup \{0\}$$

$$y_o \in \{0,1\} \quad \forall o \in O$$

** 매 결정 시간(할당 문제를 풀어야하는 순간)마다 위의 문제를 해결해야함

제약식 설명

- (1) : 각 배달기사는 최대 하나의 루트에만 할당될 수 있다
- (2) : 각 루트는 최대 하나의 배달기사에만 할당될 수 있다 (c=0은 할당이 안된다는 것 의미)
- (3) : 각 주문은 딱 하나의 할당된 루트에만 속한다
- (4) : 각 주문이 할당되면 배달기사가 할당되어야 한다
- (5) : 현재 가능한 최대로 할당을 해야한다(배달기사 수나 루트의 수 중 작은 값만큼 할당)

② Policy Proposal

1. 요약 : Dynamic Parameter Policy with Clustering-based Bundling and Matching

◆ 이슈

1. Online Problem

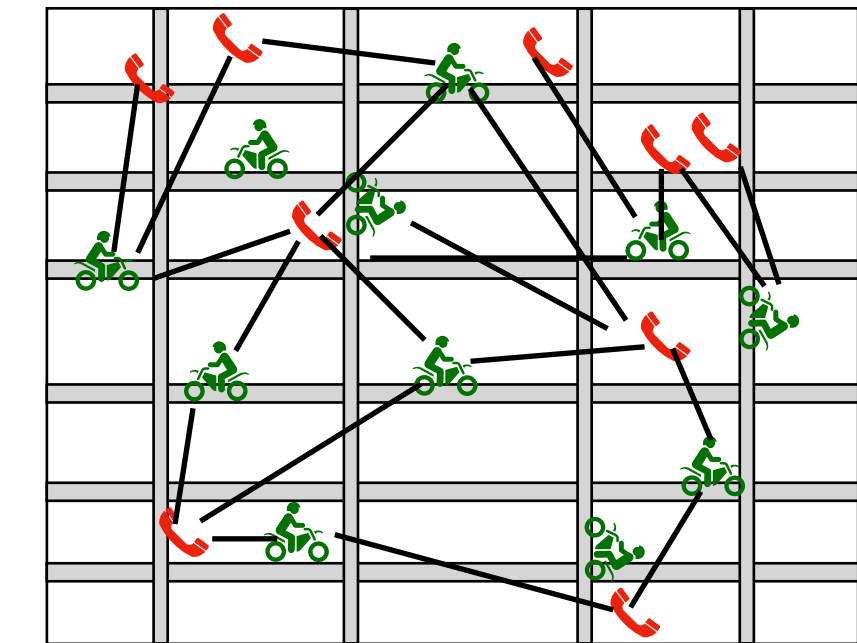
- 시간에 따라 문제의 상황(주문 수, 할당가능한 배달기사 수)이 변함
- 서비스 전체 시간동안 통일된 정책을 수행하기 보다는 각 상황에 맞는 유연한 정책을 수행해야 함
- 시간에 따라 문제의 상황에 변하는 불확실성으로 인해 안정적인 배달 정책을 수립하기 어려움
- (Ex) 점심시간, 저녁시간, 비, 눈, 갑작스런 주문 쏠림

2. Spatio-Temporal Characteristics

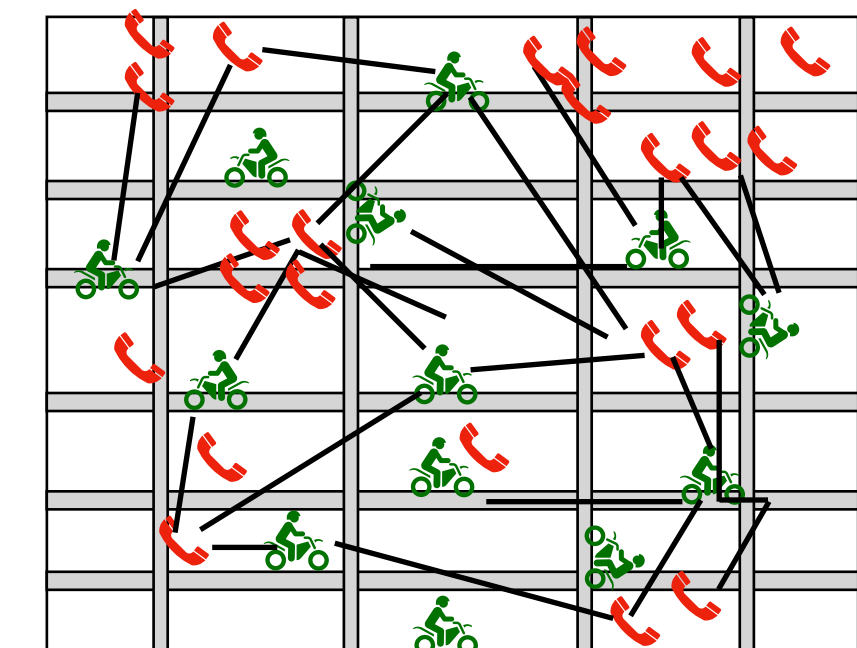
- 배달 할당 문제는 시간과 공간에 따른 정책을 수립하는 문제
- (특히 배달 지역의)시간과 공간적 특성을 적극적으로 활용해야 함
- 배달 할당 문제는 공간적 특성에 따라 성능이 급격하게 변하는 것을 이후 확인
- (Ex) 붙어있는 식당의 주문들, 시간이 지나면서 주문이 쌓임, 주거지역과 상업지역 분리

3. NP-hard Problem : Computation Problem

- 매 주문 발생마다 배달기사 할당을 위한 조합최적화 문제를 합리적인 시간(수 초)안에 해결해야 함
- (Ex) 배달기사 20명, 주문 10개의 경우 10^{20} 번의 연산 필요



두 상황에서의
policy 세부사항은 달라야 한다



➡ 1) Online의 특성을 고려한 동적 최적 패러미터 설정, 2) 한국의 위치 분포 특성을 고려한 paired-clustering 기반 지연 및 주문 묶음 프로세스, 3) 계산량 문제 해결을 위한 순차적 주문 처리 및 clustering 기반 할당 프로세스를 도입한 배달 정책 제안

② Policy Proposal

관찰 1 : 배송 문제에서의 중요 패러미터 : WORKLOAD

표 2 Bundle size와 배달기사 수, 작업량에 따른 서비스 레벨

| | Workload | 0.6 | 1 | 1.4 | 1.8 | 1.9 | 2 | 2.1 | 2.2 | 2.3 | 2.4 | 2.6 |
|----------|----------|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|
| Bundle 1 | C50 | 0.3 | 0.5 | 0.6 | 1.8 | 1.0 | 1.8 | 6.6 | 11 | 32.0 | 40 | 49 |
| | C100 | 0.4 | 0.5 | 0.5 | 0.8 | 1.0 | 2.5 | 2.4 | 17.9 | 29.0 | 37.5 | 52.7 |
| | C200 | 0.6 | 0.4 | 0.6 | 0.5 | 0.6 | 1.4 | 5.8 | 17.8 | 37.5 | 39 | 52 |
| Bundle 2 | C50 | 0.4 | 0.4 | 0.9 | 0.9 | 0.9 | 2.3 | 7.6 | 12.4 | 28.8 | 46.3 | 51.2 |
| | C100 | 0.5 | 0.6 | 0.6 | 0.9 | 0.9 | 1.5 | 2.6 | 9.2 | 21.4 | 37.6 | 50.4 |
| | C200 | 0.6 | 0.7 | 0.7 | 1.0 | 0.9 | 1.1 | 1.2 | 2.7 | 6.0 | 11.6 | 44.4 |
| Bundle 3 | C50 | 0.4 | 0.7 | 0.7 | 1.3 | 1.8 | 5.6 | 5.4 | 15.2 | 26.4 | 34.6 | 53.3 |
| | C100 | 0.6 | 0.7 | 0.7 | 0.8 | 1.1 | 1.4 | 1.8 | 7.2 | 20.0 | 36.5 | 49.3 |
| | C200 | 0.7 | 0.7 | 0.9 | 1.1 | 0.9 | 1.0 | 1.4 | 2.0 | 8.3 | 17 | 45.6 |
| Bundle 4 | C50 | 0.8 | 0.5 | 0.8 | 1.3 | 1.3 | 4.0 | 7.9 | 15.7 | 26.8 | 42.8 | 52.4 |
| | C100 | 0.7 | 0.6 | 0.8 | 1.0 | 0.9 | 1.2 | 2.6 | 8.8 | 12.6 | 44.2 | 49.1 |
| | C200 | 0.6 | 0.8 | 0.8 | 1.0 | 1.1 | 1.0 | 1.0 | 1.4 | 5.5 | 15.8 | 43.9 |

- 서비스 불만족 레벨 < 1%
- 서비스 불만족 레벨 < 2%
- 서비스 불만족 레벨 < 4%
- 서비스 불만족 레벨 < 8%
- 서비스 불만족 레벨 < 16%
- 서비스 불만족 레벨 < 32%
- 서비스 불만족 레벨 < 100%

시뮬레이션 환경

Dense Restaurants 50
Couriers 500
Bundle Size 3
Cluster Diameter 350m
Deadline 50min
Simulation Size 5km x 5km

서비스 레벨은 작업량(Workload)에 가장 큰 영향을 받음
➡ 앞으로 작업량을 중요 패러미터로 볼 것

서비스 레벨은 작업량 2~2.3 부근에서 한계를 가짐
➡ 해당 구간에 집중

* 서비스 레벨 = 데드라인 안에 도착한 주문 수 / 총 주문 수
* workload = 한 시간에 한 배달기사가 처리해야하는 평균 주문량

관찰 2 : Spatio-Temporal Characteristics

표 3 식당 10개와 식당 10개 군집(총 100개)의 서비스 레벨 비교

| | Workload | 1.80 | 1.90 | 2.00 | 2.10 | 2.20 | 2.30 | 2.40 |
|----------|----------|-------|-------|-------|-------|-------|-------|-------|
| Bundle 1 | S | 99.29 | 99.55 | 99.50 | 99.08 | 95.46 | 89.12 | 86.08 |
| | C | 99.53 | 99.57 | 99.38 | 98.95 | 95.35 | 87.96 | 65.86 |
| | p-value | 0.904 | 0.624 | 0.373 | 0.963 | 0.946 | 0.707 | 0.092 |
| Bundle 2 | S | 99.13 | 99.40 | 99.38 | 99.08 | 98.22 | 95.84 | 88.57 |
| | C | 99.45 | 99.03 | 99.50 | 99.40 | 99.21 | 95.20 | 83.49 |
| | p-value | 0.931 | 0.992 | 0.754 | 0.340 | 0.700 | 0.153 | 0.626 |
| Bundle 3 | S | 99.24 | 99.21 | 99.57 | 99.16 | 98.64 | 96.75 | 87.67 |
| | C | 99.53 | 99.49 | 99.49 | 99.47 | 98.31 | 93.59 | 79.27 |
| | p-value | 0.137 | 0.612 | 0.447 | 0.009 | 0.223 | 0.053 | 0.039 |
| Bundle 4 | S | 99.28 | 99.42 | 99.32 | 99.51 | 98.71 | 93.72 | 90.96 |
| | C | 99.44 | 99.46 | 99.34 | 99.06 | 97.56 | 96.48 | 84.22 |
| | p-value | 0.853 | 0.961 | 0.893 | 0.040 | 0.455 | 0.229 | 0.037 |

서비스 레벨 차이가 있는가?

비모수적 검정으로 robustness 보장하기 위해, t 검정 통계량
을 이용한 random permutation test (반복수 2000) 진행

➡ Bonferroni test 결과 모든 검정의 p-value < $\frac{0.05}{m}$

➡ 두 식당 분포에 따른 서비스 레벨이 통계적으로 서로 다르
다고 볼 수 없음(유의수준 0.05)

식당 개수가 적어지면 서비스 레벨은 낮아지며, 식당 n개와 식당 n개
군집은 통계적으로 같은 서비스 레벨 분포를 보인다.
따라서 배달 정책에서는 식당이 어떻게 분포해있는가를 고려해야한다.

성질 1 주문 배달이 데드라인을 넘길 확률은 레스토랑의 개수가 적어지면 증가한다

$$Var(S_m) \approx Var(p(R_1)) \frac{n^2}{m} + E\left[p(R_1)(1 - p(R_1))\right] n$$

Proof)
 $A = [0,1] \times [0,1]$ 를 전체 공간으로 정의하고 m 을 전체 식당의 수라고 하자
 i 번째 식당의 위치와 해당 식당에 들어온 j 번째 주문의 위치는 전체 공간 A 에서 Uniform
Distribution을 따른다고 하자. 즉,

$R_i \sim Unif(A)$, $O_{ij} \sim Unif(A)$

$i = 1, 2, \dots, m$ $j = 1, 2, \dots, n_i$ $n_1 + n_2 + \dots + n_m = n$

$E\left[\left|R_i - O_{ij}\right|\right] = 1/15 \left(2 + \sqrt{2} + 5 \log\left(1 + \sqrt{2}\right)\right) \approx 0.5214$ 이므로

$\sum_{i=1}^m \sum_{j=1}^{n_i} E\left[\left|R_i - O_{ij}\right|\right] = 0.5214 n$ 이다. 즉, 배달 거리의 기댓값은 식당 개수와 무관하다.

이제 데드라인을 넘는 주문의 개수를 살펴보자. $\chi\left(\left|R_i - O_{ij}\right| > l\right) = 1 \text{ or } 0$ 는
식당 i 에 들어온 주문 j 가 데드라인을 넘는지 안넘는지에 대한 값이다. 그러면

$S_m = \sum_i \sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right)$ 는 데드라인을 넘는 주문의 총 개수가 된다.

$\sum_{j=1}^{n_i} \left[\chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right] \sim Binom\left(n_i, p(R_i)\right)$ 이므로

$Var\left(\sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right) = n_i p(R_i) (1 - p(R_i))$ 가 된다. 이를 이용하면

$$\begin{aligned} Var(S_m) &= Var\left(\sum_i \sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right)\right) \\ &= Var\left(E\left[\sum_i \sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right]\right) + E\left[Var\left(\sum_i \sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right)\right] \\ &= Var\left(\sum_i \sum_j E\left[\chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right]\right) + E\left[\sum_i Var\left(\sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right) \middle| R_i\right)\right] \\ &= Var\left(p(R_1)\right) \sum_{i=1}^m n_i^2 + E\left[p(R_1)(1 - p(R_1))\right] n \end{aligned}$$

② Policy Proposal

Observation 2 : Spatio-Temporal Characteristics (conti.)

◆ Should Consider the number and the distribution of restaurants

- If the number of restaurants decreases, the service level decreases
- Even if the number of restaurants is large, if restaurants are gathered, it has the same effect as the number of restaurants decreases
- The distribution of restaurants has a significant impact on the performance of delivery policies ➔ **need a delivery policy that considers the distribution of restaurants**
(In particular, restaurants and delivery destinations are gathered in Korea)

◆ Service levels change significantly depending on the Bundling

- Bundling process, which considers spatial characteristics, improves service level
 - (i) low workload : bundling has little impact on service level
 - (ii) medium workload (2.1~2.4) : bundling has significantly positive impact on service level
 - (iii) high workload : despite the bundling, service level is too low
➔ (service failure) : need to increase the number of couriers in the platform

Observation 3 : Online Problem

◆ Online Problem : optimization environment continues to change

- Since ordering is stochastic process, the optimization environment continues to change
 - ➔ **Should focus on optimizing the overall expectation of the service level, not the temporal one**
 - ➔ **Should continuously adjust the service parameters to optimal values that are suitable for service environment**
- As a result of simulation, the number of available couriers and the number of orders are fluctuating over time according to [Graph 1]

◆ Postponement Strategy to deal with uncertainty

- One of the difficulty of online optimization is that you have to make decision with information about the current situation without knowing what will happen in the future : afterwards, orders may suddenly increase or decrease rapidly
- One way to resolve uncertainty is to gather information through postponement : do not process orders immediately after they are placed, but process them with orders that have been placed for postponement time.

Proposal of Delivery Policy Considering Observations

P1 Spatial-Paired-Clustering Based Bundling

- **paired-clustering** : bundle orders when restaurants and destinations are attached to each other
➔ Improves the service level by the effect of reducing overlapping routes of the couriers
- If both restaurants and destinations are in the cluster diameter respectively, we bundle such orders. Previous studies only considered the location of restaurants, however if the destinations are far away, problems may arise. Especially, **the policy might be effective in the region where each of restaurants and residential areas are concentrated**
- If there are few orders, the orders may not be bundled, and the policy will be ineffective

P2 Postponement

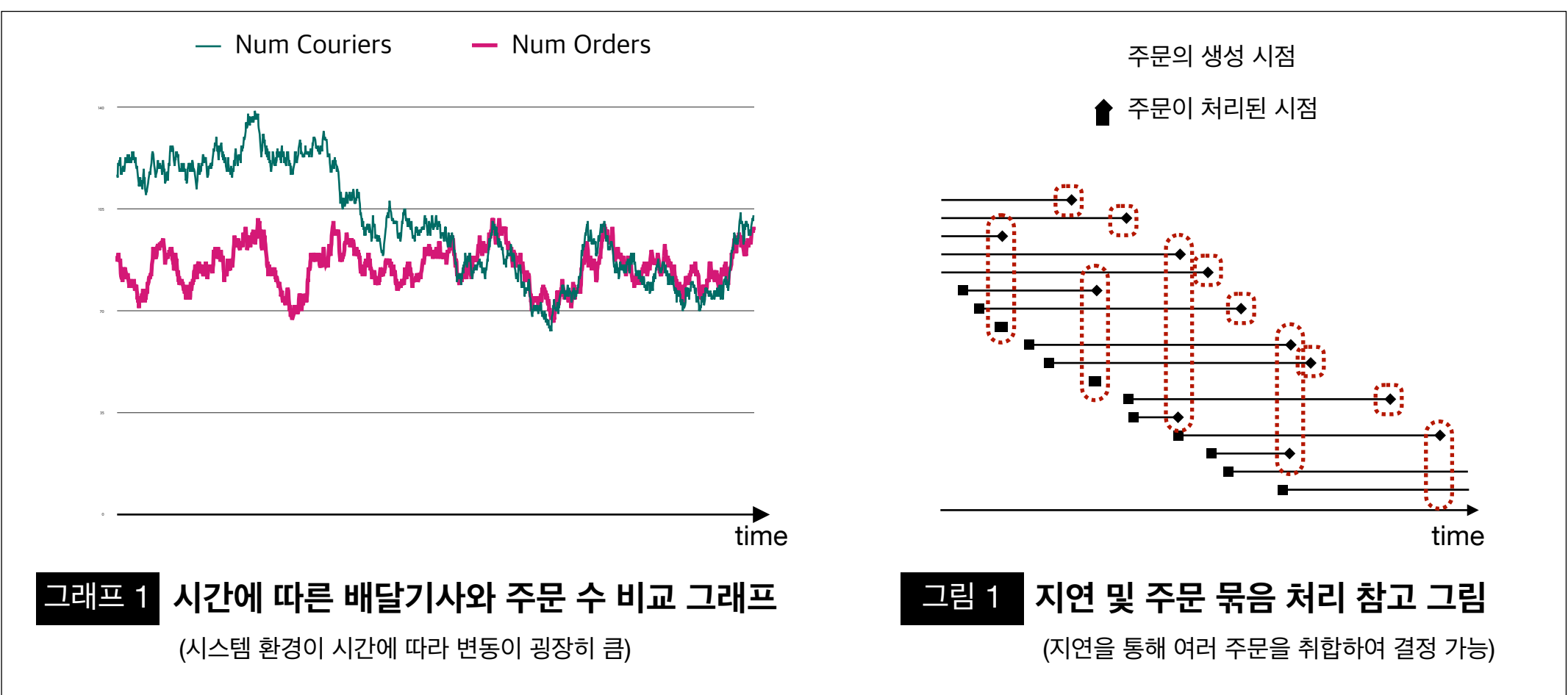
- Through postponement, the amount of information in each decision-making process (especially bundling) increases, resulting in the effect of increasing options and thus better performance
- Synergy occurs when bundling and postponement are used at the same time

P3 Dynamic Optimal Parameter Setting

서비스 환경이 바뀌면, 해당 서비스 환경에 적합한 서비스 패러미터를 설정함

(특정 서비스 환경에 적합한 패러미터는 Multi-layered Neural Network 모델을 통해 추정)

* 결정 패러미터 : 군집 직경(주문 묶음 처리 및 군집화 기반 할당 알고리즘에서 사용), 지연 시간



② Policy Proposal

Observation 4 : Computation Problem of the Vehicle Routing Problem 알고리즘 제안 : 관찰 결과들을 고려한 배달 정책 제안

Computational 주문에 대해서 배달기사를 할당하는 과정에서 엄청난 계산량 발생

만약 현재 할당되어야 하는 주문의 수가 m 개이고, 활용가능한 배달기사의 수가 h 명이 있다면 원래라면 가능한 할당의 방법의 가지수는 h^m (각각의 주문마다 배달기사 1, 2, ..., h 에 대한 배달거리를 구하고 비교)

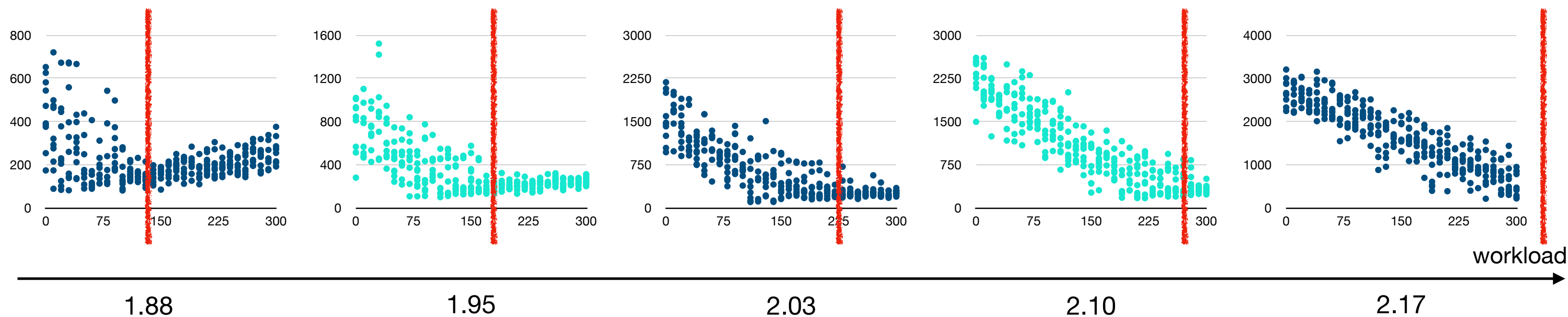
Ulmer et al. (2020)은 각각의 주문의 순서열을 비교하는 방법을 제안하여 $m! \cdot m \cdot h$ 계산량의 휴리스틱을 제안

$m = 10, h = 36$ 만 되어도 원래 방법은 3,656,158,440,062,976가지의 경우의 수가 Ulmer et al. (2020)은 1,306,368,000의 경우의 수가 발생

대부분의 배달 플랫폼은 훨씬 큰 사이즈를 가지고 있고, 의사결정을 위해서는 수 초 이내에 할당 프로세스가 진행되어야 함
→ **훨씬 빠른 알고리즘이 필요함**

그래프 2 workload에 따른 지연시간-서비스 레벨 그래프 (dynamic parameter setting의 필요성)

- 각 그래프에서 x축은 지연시간(postponement time), y축은 데드라인을 넘은 주문의 수로 서비스 레벨로 볼 수 있음
- workload가 증가함에 따라 그래프가 오른쪽으로 stretch되는 모양이 되는 것이 보이며, 이에 따라 최적의 지연시간도 점차 증가함을 볼 수 있음
- 주문 발생량이 많을수록 지연 종료 전에 처리되는 양이 많아지기에, 주문 묶음을 위한 충분한 양의 주문(정보)을 유지시키기 위해 지연시간이 길어져야 함



도입 4 주문 분류 및 계층적 주문 처리(Hierarchical Order Classification)

주문/묶음주문을 긴급성(Emergency)을 기준으로 우선순위를 부여하여 할당을 차등적으로 진행시킨다 :
새로운 주문 > 묶음주문 > 지연시간 초과한 묶음주문 > 긴급 묶음주문

더 긴급한 주문을 먼저 처리하여 서비스 레벨에 기여하며, 각각 할당 과정에서 대상 주문의 수를 줄이는 효과

도입 5 위치-군집화 기반 배달기사 할당 휴리스틱 (Spatial-Clustering Based Allocation Heuristics)

(아이디어) 주문과 배달기사는 가까운 것들끼리 묶일 가능성이 큼 → 전체를 고려하여 주문-배달기사 할당 쌍을 찾는 것이 쉽지 않으니 주문에 적합한 배달기사 군집(거리 기반)을 먼저 찾고 해당 군집 안에서 할당된 주문들끼리 sub-allocation 문제를 해결 → 이를 여러 번 반복하여 가장 좋은 성능의 할당을 업데이트

중복된 경로 이동을 감소시키는 효과가 있어 전체적인 서비스 레벨 개선의 효과를 발생시킬 수 있음

시뮬레이션 환경
Dense Restaurants 50
Couriers 500
Bundle Size 3
Cluster Diameter 350m
Deadline 50min
Simulation Size 5km x 5km

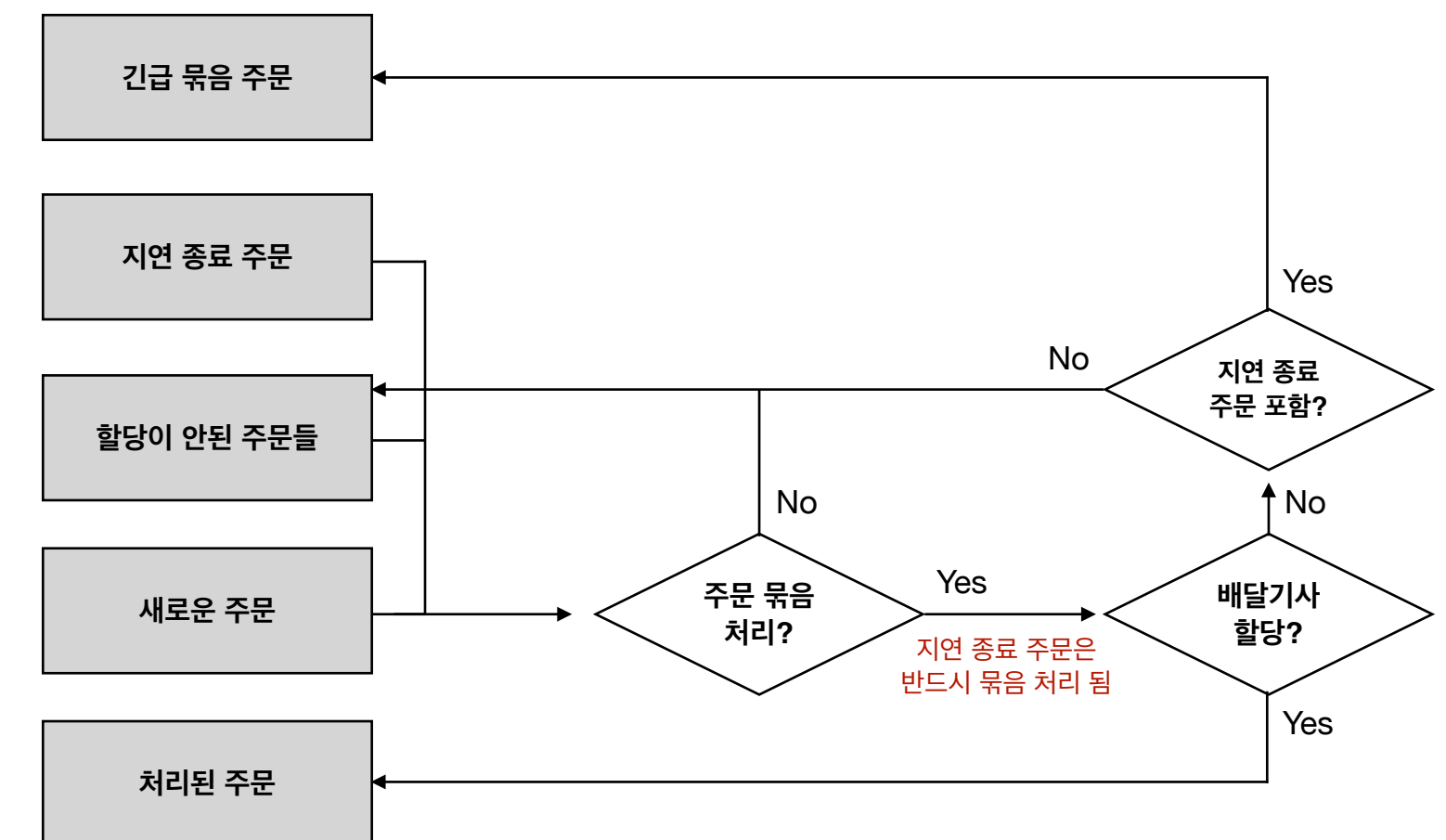
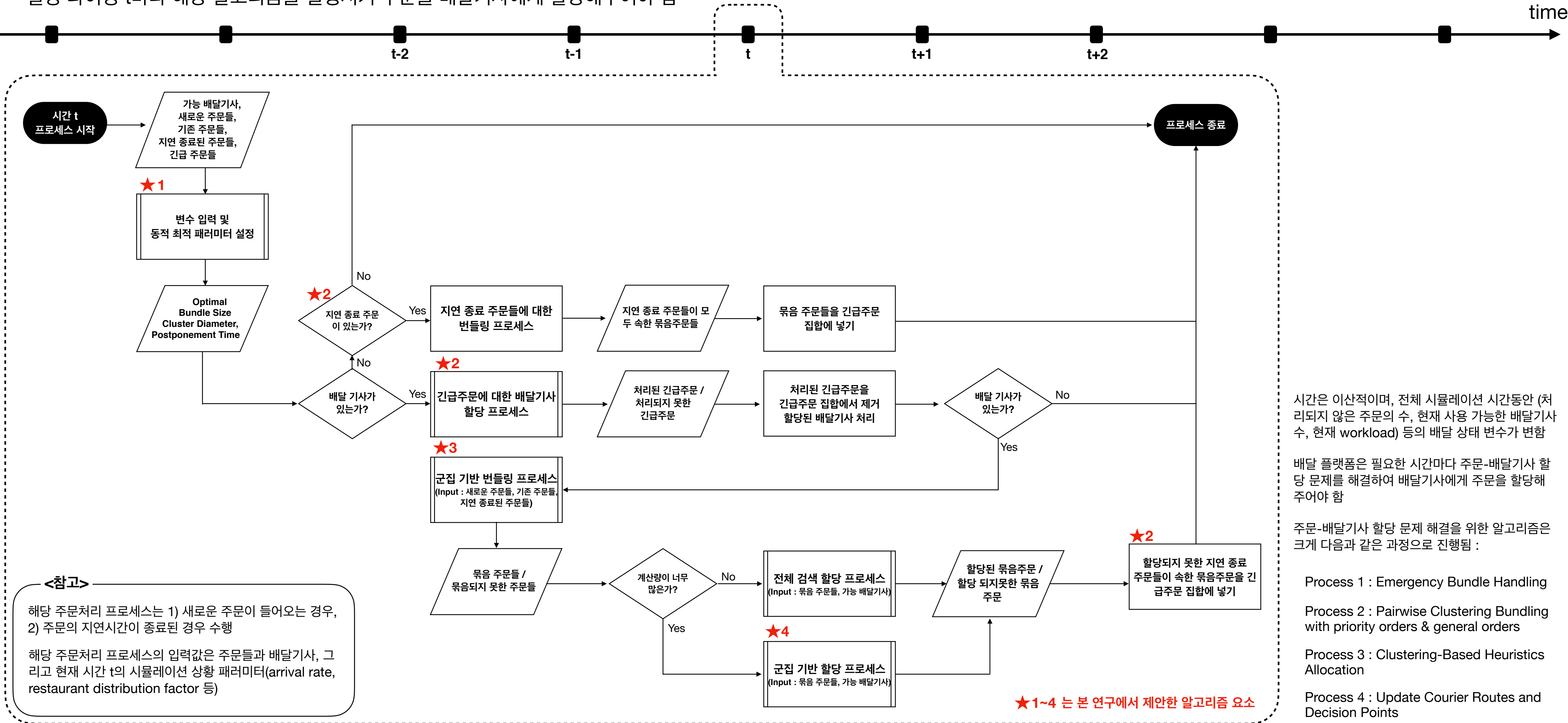


그림 2 주문 분류 다이어그램

3 Algorithm : Spatial Clustering based Bundling and Assignment Heuristics & Policy Parameter Optimization

결정 타이밍 t마다 해당 알고리즘을 실행시켜 주문을 배달기사에게 할당해주어야 함



3

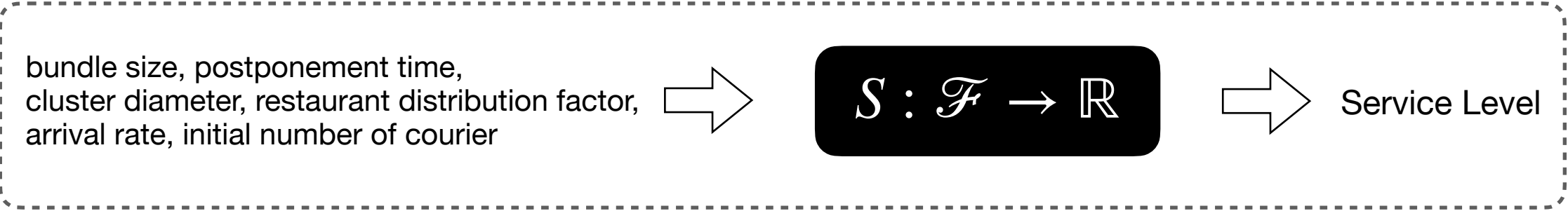
Algorithm : Spatial Clustering based Bundling and Assignment Heuristics & Policy Parameter Optimization

★1

과정 1

동적 최적 패러미터 설정

(STEP 1) Multilayered-Neural-Network Model을 이용하여 Approximated Service Level Function 찾기
(input을 다양하게 조절하여 시뮬레이션을 돌린 약 20만개의 데이터로 학습)



- (STEP 2)** Approximated Service Level Function 을 이용하여 최적의 패러미터 (bundle size, postponement time, cluster diameter) 찾기
- Stage 1 : workload에 따라 bundle size조정
 - Stage 2 : \hat{S} 을 활용하여 cluster diameter 10단위, postponement time 10단위로 탐색하여 최적값 설정

★3

알고리즘

위치-군집화 기반 주문 묶음 처리 알고리즘

| Bundling Process based on Spatial Paired-Clustering |
|--|
| <div>1. Construct clusters consisting of orders considering both locations of the restaurants and the delivery sites</div> <div>2. Set bundle_list = []</div> <div>3. For each cluster C<div>3.1. If there is an order that finishes postponement, then make a bundle of size B_{max} with the order and other arbitrary orders in the cluster C</div><div>3.2. Make bundles of size B_{max} as many as possible with arbitrary remaining orders in the cluster C</div><div>3.3. If the number of remaining orders is more than 1, then make a bundle with the remaining</div><div>3.4. Push all bundles that are made with the cluster C to the bundle_list</div></div> <div>4. End For</div> |

★2

과정 2

주문 분류 및 계층적 주문 처리

주문/묶음주문을 긴급성(Emergency)을 기준으로 우선순위를 부여하여 할당을 차등적으로 진행시킨다 :
새로운 주문 > 묶음주문 > 지연시간 초과한 묶음주문 > 긴급 묶음주문

- 1) 긴급 묶음 주문은 새로운 활용가능한 배달기사가 발생하면 즉시 할당하여 배달 프로세스를 실행

2) 주문 묶음 처리 프로세스에서 지연시간을 초과한 묶음주문은 우선적으로 묶으며, 다른 주문들과 묶이지 않아도 하나의 주문으로 이루어진 묶음 주문으로 처리하여 이후 할당 프로세스 진행

3) 전체 프로세스에서 할당되지 않은 주문들 중에서 긴급 주문도 아니고 지연시간도 초과하지 않은 주문은 그대로 일반 주문으로 남으며, 나머지 주문들은 긴급 묶음주문에 추가

★4

알고리즘

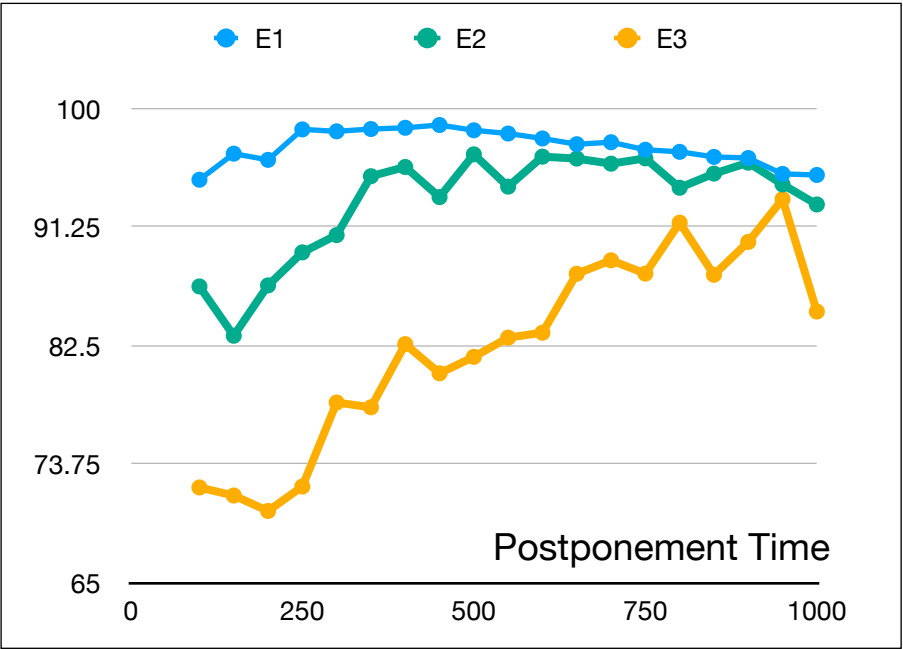
위치-군집화 기반 할당 휴리스틱 알고리즘

| Spatial Clustering-Based Allocation Algorithm for the VRP |
|---|
| <div>1. Construct K-clusters consisting of bundles</div> <div>2. Construct courier-cluster distance <u>matrix</u> (C_{ij})</div> <div>3. While $nbIter < nbIter_{max}$<div>3.1.Put couriers in random order</div><div>3.2.For each courier<div>3.2.1.Put the clusters in increasing distance to the courier</div><div>3.2.2.Find a cluster from the nearest cluster that is not full and put the courier to the cluster</div></div><div>3.3.End For</div><div>3.4.Find the best matching in each cluster</div><div>3.5.Computer overall performance of the allocation and update best allocation of bundle-courier pairs</div></div> <div>4. End While</div> |

④ Results & Analysis

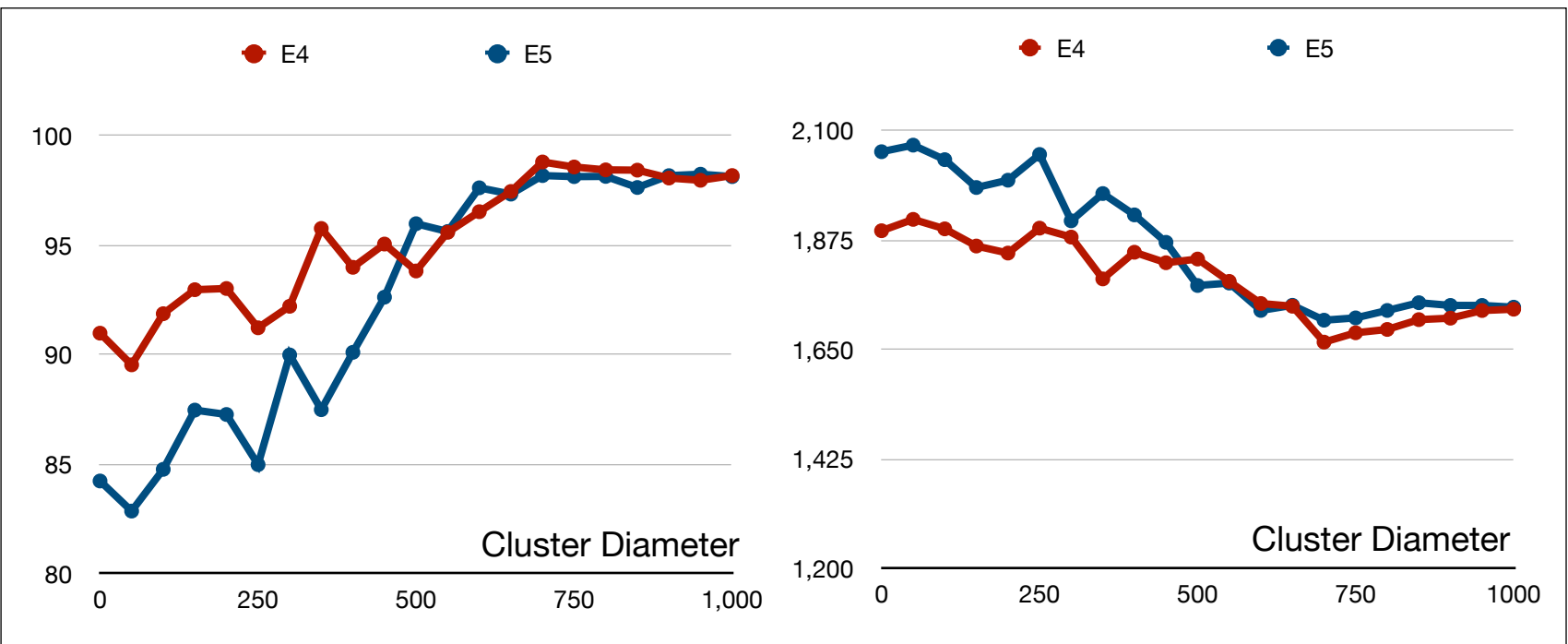
중요 시뮬레이션 결과 : Service Level Function Learning & Effect of the Params

◆ 주요 변수들이 서비스 레벨에 미치는 영향



- 배달량이 많은 상황(workload가 2 이상)에서 지연시간을 증가시키면서 서비스 레벨이 어떻게 바뀌는지 확인
- 초기에는 지연시간이 증가하면 서비스 레벨이 높아지다가 어느 순간부터는 지연시간이 증가하면서 서비스레벨은 감소
- **workload가 높을수록 최적의 지연시간이 증가하는 경향을 보임**
- **대략적으로 그래프가 오목한 모양을 보임 (최적값이 존재할 것으로 예상 가능)**

그래프 5 지연시간(x축)에 따른 서비스 레벨(y축)의 그래프



그래프 6 군집 직경(x축)에 따른 서비스 레벨과 평균 배달시간(y축)의 그래프

- 배달량이 많은 상황(workload가 2 이상)에서 군집 직경을 증가시키면서 서비스 레벨과 평균 배달 시간이 어떻게 바뀌는지 확인
- 초기에는 군집 직경이 증가하면 서비스 레벨이 높아지다가 어느 순간부터는 군집 직경이 증가하면서 서비스레벨은 감소
- **workload가 높을수록 최적의 군집 직경이 증가하는 경향을 보임**

◆ 시뮬레이션 실행 (B, p, c, W, R, N_R, N_C)

각 패러미터 세팅에 대하여 30번의 반복횟수로 시뮬레이션을 실행하여 (총 주문 수, 데드라인을 넘은 주문 수, 묶음 주문 수) 등의 데이터 획득 : 총 약 24만개의 데이터

- 1) 번들 사이즈 B (1, 2, 3)

2) 지연 시간 p (0, 50, 100, ..., 1000 (s))

3) 군집 지름 c (0, 50, 100, ..., 1000 (m))

4) Workload W (0.5, 0.6, ..., 3.0 (개/시간·배달기사))
- 5) 식당 Randomness Factor R (0 ~ 0.24)

6) 식당 수 N_R (10, 50, 100, 200, 400)

7) 배달기사 수 N_C (50, 100, ..., 400)

*Randomness = inter-event distance 기반 spatial randomness
- Deadline : 50분

- 운영시간 8시간 (28,800 초)

- 배달 공간 [0, 5000 m] X [0, 5000 m]

- Distance Factor : 1.4

- 식당 서비스 시간 : 60초

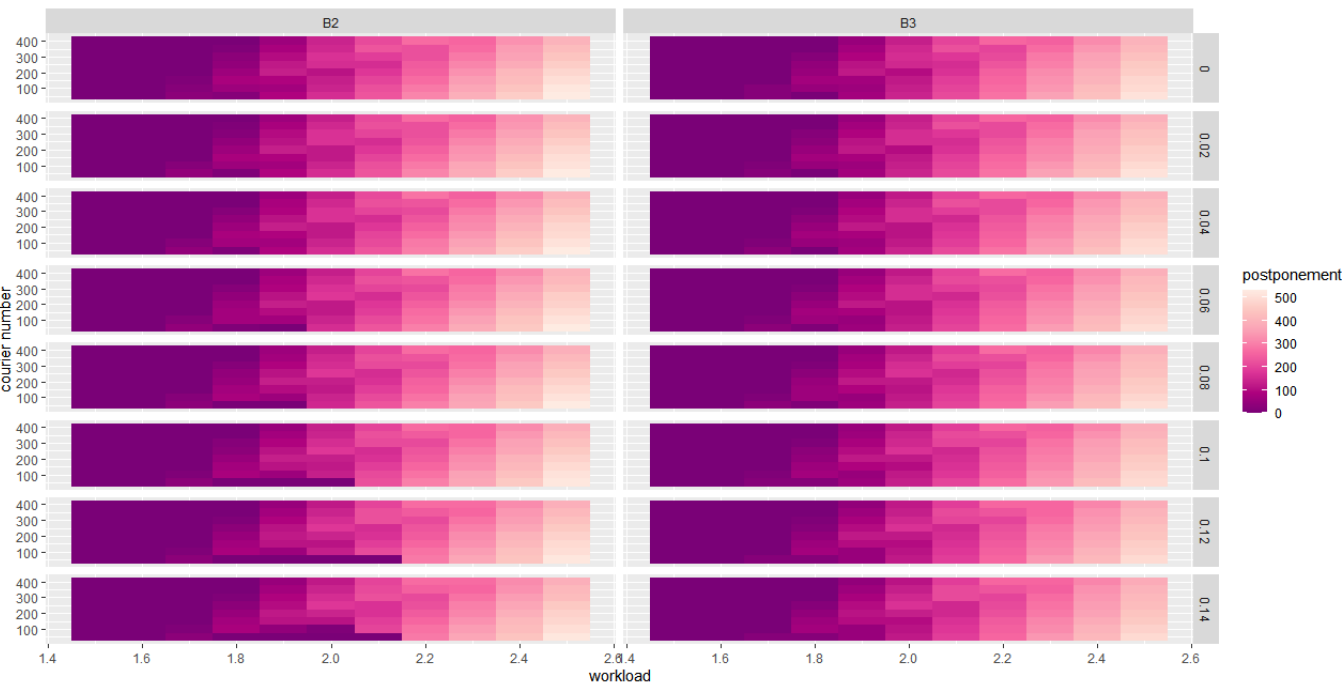
- 배달지 서비스 시간 : 180초
- 식당 음식 준비 시간의 분포 : 삼각분포(MOD, COV)

- MOD는 1:3:3:1.5:1:0.5의 비율로 5, 10, 15, 20, 25, 30분 (배달의 민족 자료 참조)

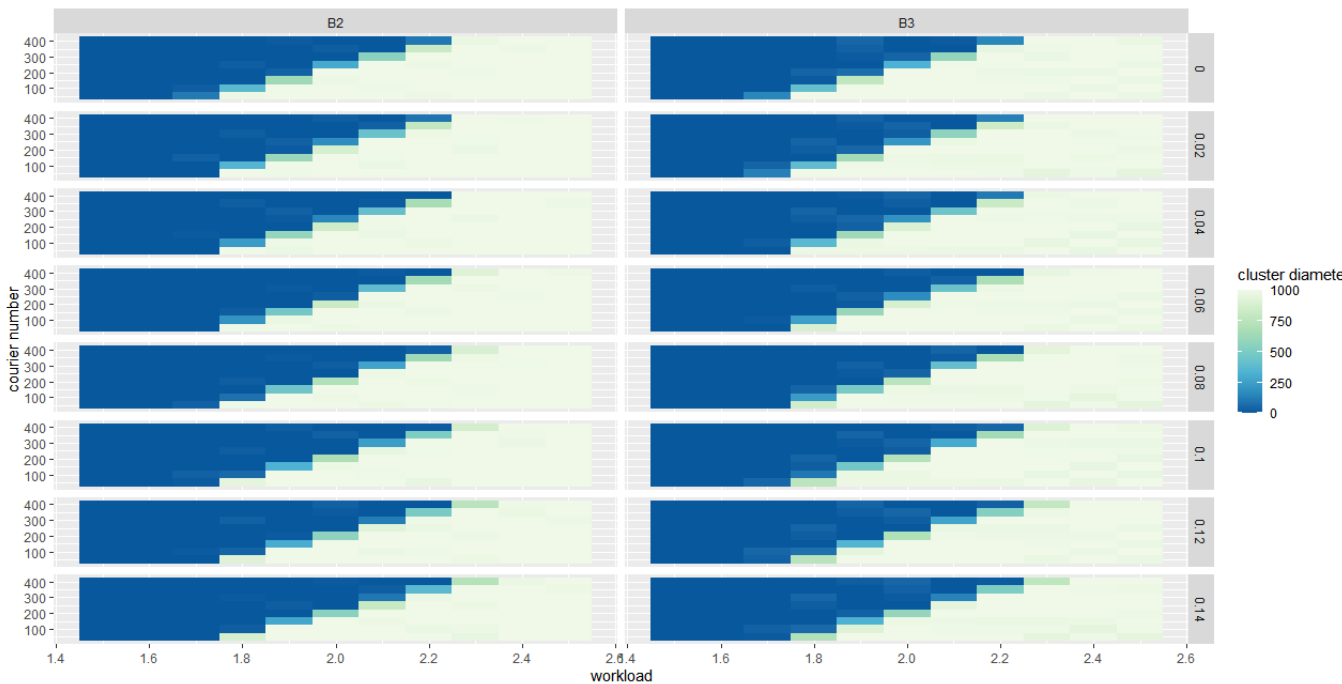
- 준비 시간의 분포의 COV 는 0.1

◆ 서비스 레벨 함수 학습 결과

$$(p^*, c^*) = \operatorname{argmax}_{p, c} \{ \hat{S} | B, W, R, \lambda, N_C, N_R \} = f(B, W, R, \lambda, N_C, N_R) \text{ 학습}$$



그래프 3 Workload와 배달기사 수에 따른 최적의 지연시간



그래프 4 Workload와 배달기사 수에 따른 최적의 군집 직경

- ▶ 모든 세팅에서 동일한 패턴을 모이는 것에서 함수 $f(B, W, R, \lambda, N_C, N_R)$ 이 잘 학습되었다는 것을 확인
- ▶ Workload가 높아질 수록 최적의 군집 직경과 지연 시간이 늘어나는 것을 확인할 수 있음

<참고>
E1 : B 2, c 400m, N_C 100, R 0.999, N_R 100, W 2
E2 : B 3, c 500m, N_C 200, R 0.999, N_R 100, W 2.1
E3 : B 3, c 500m, N_C 200, R 0.999, N_R 100, W 2.2
E4 : B 3, p 400s, N_C 100, R 0.999, N_R 100, W 2.3
E5 : B 2, p 500s, N_C 100, R 0.999, N_R 100, W 2.2
 B (Bundle Size), c (Cluster Diameter), N_C (Number of Couriers), R (Randomness Generating Factor), p (Postponement Time), N_R (Number of Restaurants), W (Workload)

④ Results & Analysis

실제 식당 분포에 대한 시뮬레이션 결과

◆ 시뮬레이션 실행

각 패러미터 세팅에 대하여 30번의 반복횟수로 시뮬레이션을 실행하여 (총 주문 수, 데드라인을 넘은 주문 수, 묶음 주문 수) 등의 데이터 획득

- 실제 서울 특정 지역의 식당 분포를 활용하여 본 연구에서 제안한 배달 정책의 성능을 확인하고자 함
- 식당이 많이 분포한 주택가 지역의 가로, 세로 3km의 공간을 자른 후 <서울특별시 일반음식점 인허가 정보>를 활용하여 지역내의 음식점 위치 정보를 저장
- 해당 지역의 음식점 수는 2834개이며, randomness factor는 0.1636
- Deadline : 50분
- 운영시간 8시간 (28,800 초)
- 배달 공간 [0, 3000 m] X [0, 3000 m]
- Distance Factor : 1.4
- 식당 서비스 시간 : 60초
- 배달지 서비스 시간 : 180초
- 식당 음식 준비 시간의 분포 : 삼각분포(MOD, COV)
- MOD는 1:3:3:1.5:1:0.5의 비율로 5, 10, 15, 20, 25, 30분 (배달의 민족 자료 참조)
- 준비 시간의 분포의 COV 는 0.1

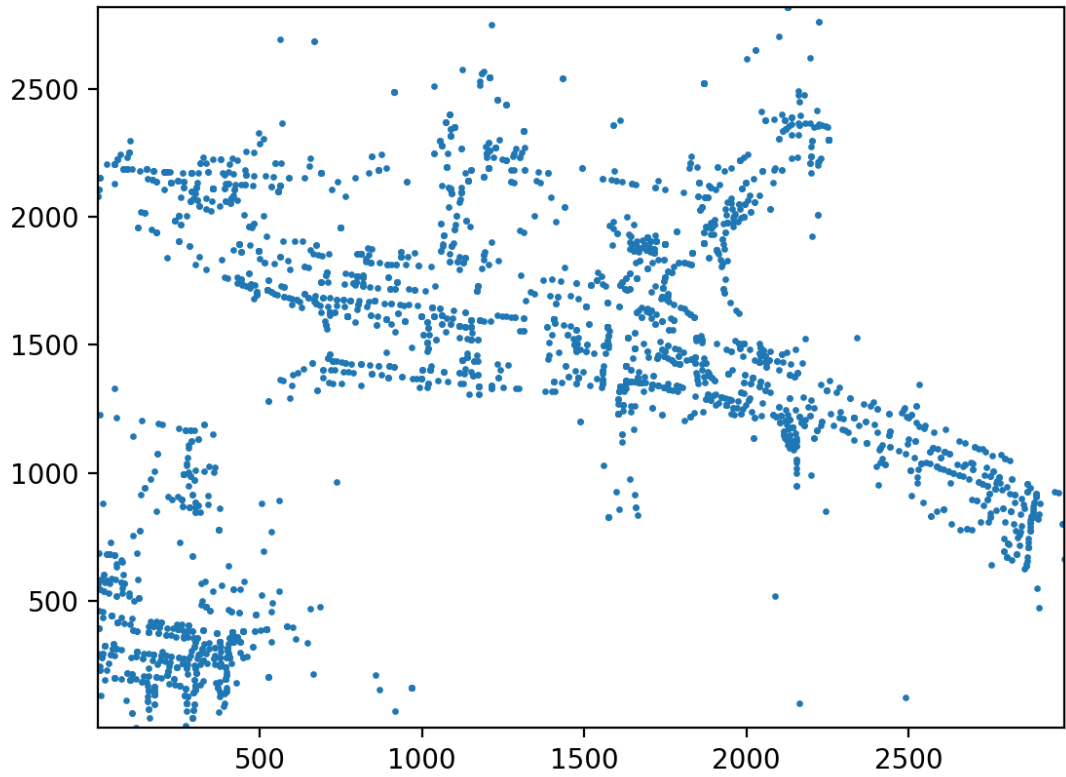
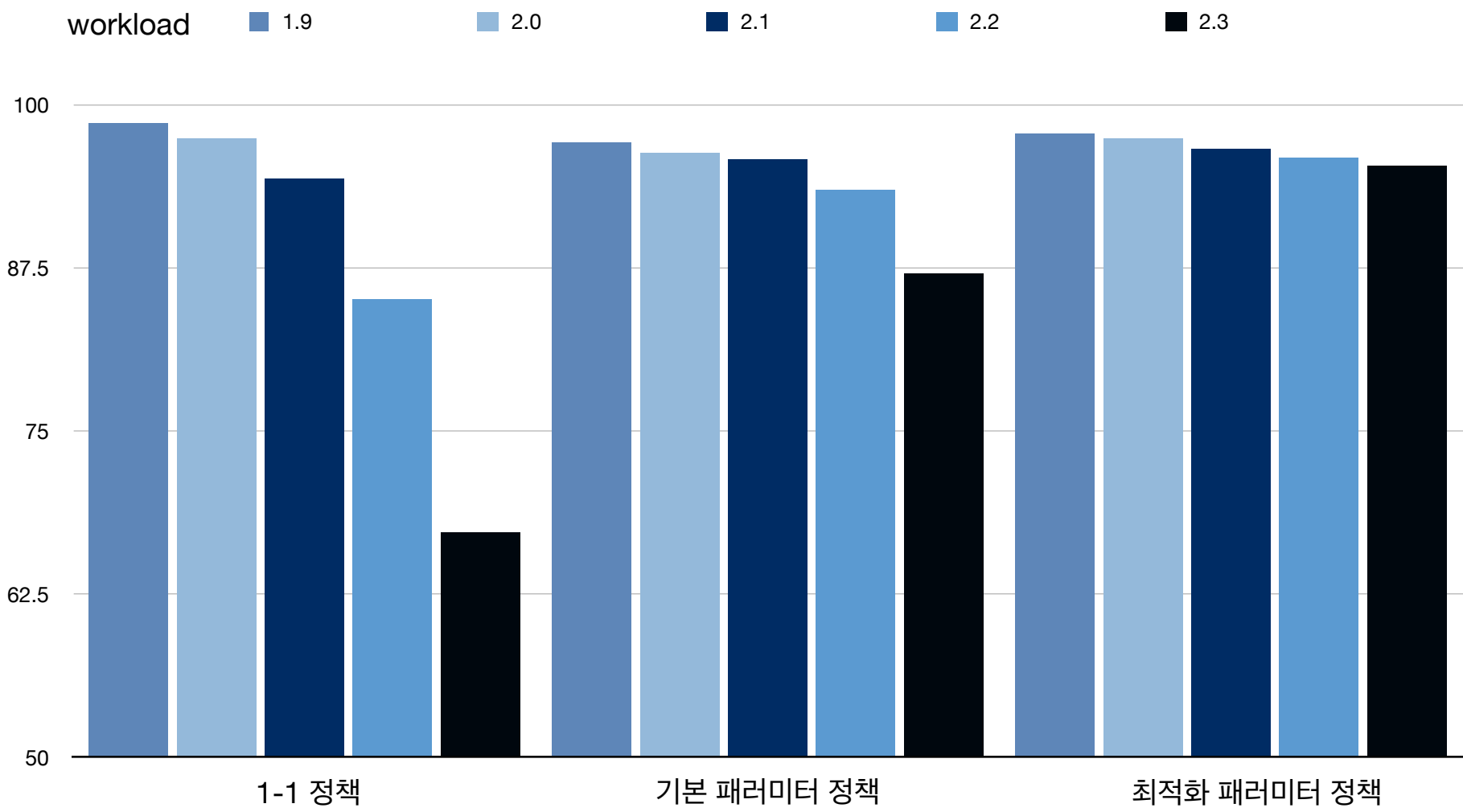


그림 3 서울 시뮬레이션 대상 공간의 레스토랑 분포

Data from <https://data.seoul.go.kr/dataList/OA-16094/A/1/datasetView.do#AXexec> (서울 열린 데이터 광장)

◆ 실험 1 : 고정된 workload에 대한 서비스 레벨 비교



그래프 6 고정된 workload에 대한 각 실험군 별 서비스 레벨비교

- 1) 1-1 정책 : 묶음 배송, 지연 등 아무 것도 하지 않고 하나의 배달기사가 하나의 배송을 하는 정책
- 2) 기본 패러미터 정책 : 묶음 배송 2개, 지연시간 300초, 군집 직경 500m 의 배송 정책
- 3) 최적화 패러미터 정책 : 학습 결과에 따른 최적의 패러미터를 사용한 배송 정책

* 다른 논문에서는 패러미터를 휴리스틱에 근거하여 "고정값으로" 설정함

- 시간에 따른 workload의 변화가 없는 경우, 각 정책별 성능을 비교한 결과 본 연구에서 제안한 배달 정책의 성능이 많은 차이로 우수함을 확인할 수 있음
- 최적의 패러미터를 사용한 정책의 경우 굉장히 높은 workload(2.3)에서도 95% 수준의 서비스 레벨을 유지함

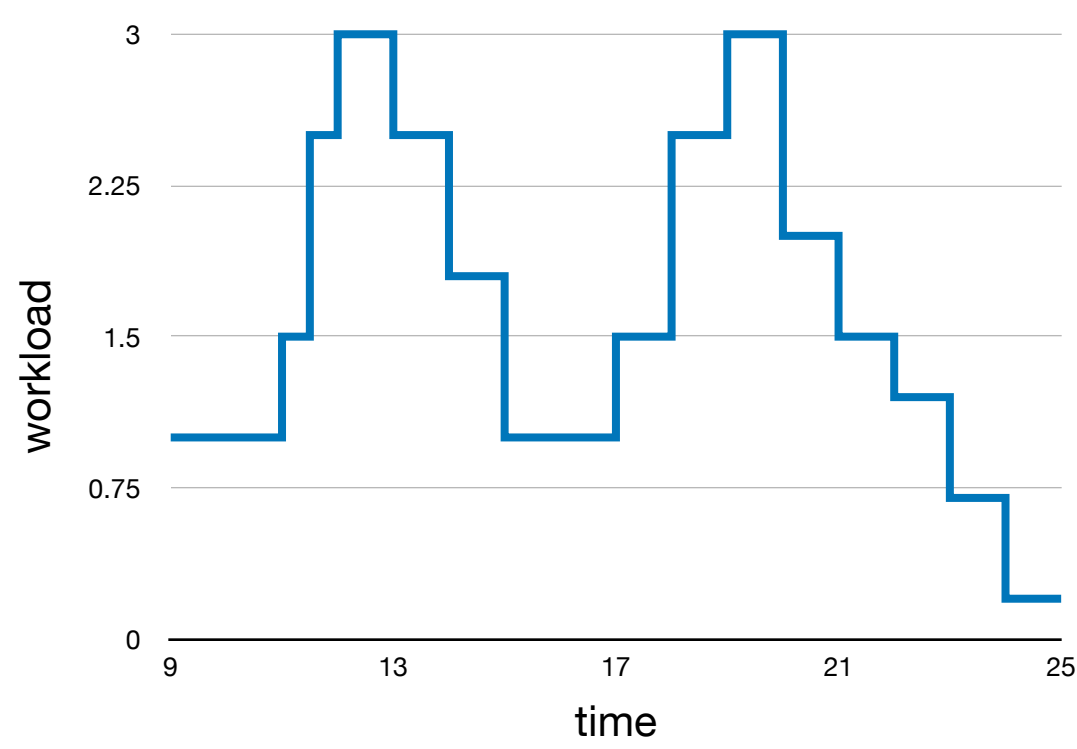
실험 결과

주문 묶음 처리와 지연 정책은 1-1 정책에 비해 서비스 레벨을 크게 개선시킨다. 또한, 이러한 정책을 사용할 때에 본 연구에서 최적화 한 패러미터를 사용하는 것이 더 효과적임을 확인하였다.

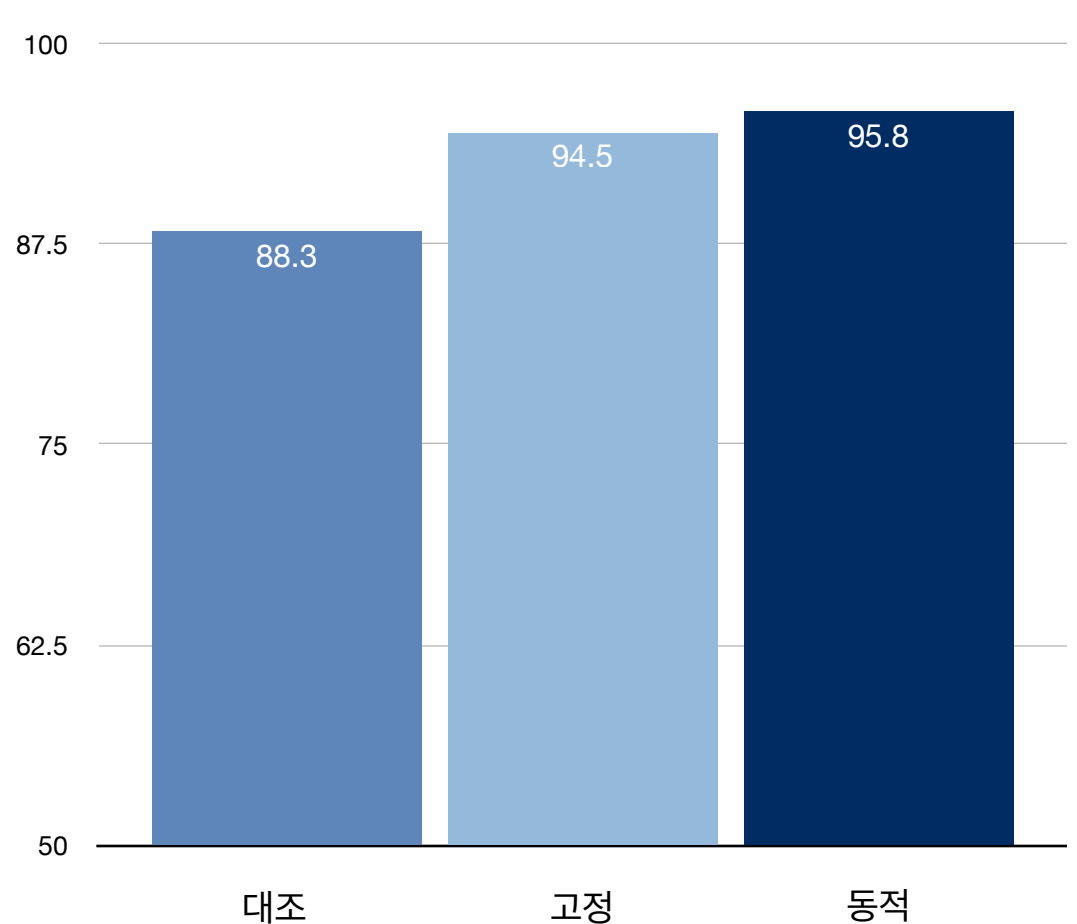
④ Results & Analysis

실제 식당 분포에 대한 시뮬레이션 결과 & 배달 플랫폼 플레이어 별 분석

◆ 실험 2 : 시간에 따른 주문량의 변화가 있는 경우 서비스 레벨 비교



그래프 7 시간에 따른 workload 스케줄 그래프



그래프 8 시간에 따라 workload가 변화할 때 각 정책별 서비스 레벨

실험 세팅

- 오전 9시부터 다음날 새벽 1시까지 16시간 동안 운영되는 스케줄
- 점심과 저녁시간에 집중되는 주문을 반영하여 스케줄 작성
- 전체 시간동안 배달기사의 수(300명)는 동일하다고 가정

배달 정책 세팅

- 1) 대조 : 묶음 배송, 지연 등 아무 것도 하지 않고 하나의 배달기사가 하나의 배송을 하는 정책
- 2) 고정 : 학습 결과에 따른 최적의 고정된 패러미터를 사용한 배송 정책
- 3) 동적 : 학습 결과에 따른 최적의 패러미터를 사용한 배송 정책

- 시간에 따른 workload의 변화가 있는 경우, 1인 1주문 정책은 88.3%의 낮은 서비스 레벨을 보이나, 본 연구에서 제시한 배달 정책을 채용하는 경우 높은 수준의 서비스 레벨 95.8%를 유지함
- 최적의 패러미터를 고정으로 사용하는 경우에도 좋은 서비스 레벨을 보였으나, 동적으로 상황에 맞춰 패러미터를 조절하는 경우 더 좋은 서비스 레벨을 보임

<참고>

30회씩 실험을 진행한 데이터(서비스 레벨)의 분포에 대한 t-test결과 고정된 패러미터를 사용한 경우의 분포와 동적인 패러미터를 사용한 경우의 분포에 대한 p-value가 0.001보다 작은 매우 작은 값이 나와 두 분포가 유의미하게 다르다는 것을 확인

◆ 배달 정책 반영 시 배달 플랫폼 시장 플레이어 별 분석

1. 배달 플랫폼

- 배달량이 많아지는 경우에도 보다 안정적으로 운영이 가능함
- workload가 높아져도 준수한 서비스 레벨을 보이며, 특정 서비스 레벨을 유지하기 위한 최소한의 배달 기사 수가 줄어듦. 즉, 서비스 레벨이 유지하면서, 적은 배달 기사 수로도 안정적인 운영이 가능해짐
- Ex) 5km X 5km 지역에 시간당 평균적으로 2000개의 주문이 들어온다고 할 때, 95%의 서비스 레벨을 유지하기 위해서는 1인-1배달 정책에서는 약 600명의 배달기사가 필요한 반면, 해당 연구의 배달 정책에서는 약 420명의 배달기사로도 가능함

2. 소비자

- 배달 상황이 급격하게 변하더라도(배달량 폭증, 배달기사 수 급감 등) 안정적으로 주문을 원하는 시간 안에 받을 수 있음
- 배달기사가 1회 할당에 받을 수 있는 주문의 수가 많아지면, 배달비용을 주문자끼리 분담하여 절약 가능

3. 배달기사

- 한 번에 여러 개의 주문을 처리할 수 있기 때문에 1인당 벌 수 있는 돈이 많아짐 (단위시간에 처리하는 주문의 개수가 많아지며, 배달기사는 배달 1회당 배달 비용을 받음)
- 전체 플랫폼 상의 배달기사 수가 적어지면 수익 평균 증가 (배달기사는 처리하는 주문의 개수에 비례하여 수익)

4. 식당

- 높은 서비스 레벨이 항상 유지가 되어 음식의 품질이 보존된 상태로 배달됨.
- 식당에 대한 소비자 만족도 향상.

실험 결과

1. 본 연구에서 제시한 배달 정책은 문제 자체의 실시간적 성질로 인한 불확실성과 서울의 공간 분포 특성을 고려한 대로 실제 데이터에 대해 안정적인 성능과 빠른 문제해결 속도를 보여줌. 이는 배달 플랫폼 시장의 모든 플레이어에게도 모두 이익이 되는 정책으로, 배달 플랫폼 기업 혹은 공공 배달앱에 적용을 제안.
2. 본 연구에서 보인 VRP 문제에서의 위치-군집화 기반 알고리즘들은 강력한 기능을 하면서도 직관적임. 위치 분포가 중요한 비슷한 유형의 SCM 문제에서 이 알고리즘을 적용해보는 것도 좋은 연구 방향이 될 것이라 제안.

5 Conclusions

1. 배달 플랫폼의 알고리즘 제시 : Dynamic Pickup and Delivery Problem with Time Window

- 1) 2차원 공간의 분포와 시간에 따른 변화를 고려한 알고리즘 제시
- 2) 온라인 문제(Online Problem) 특성을 고려한 알고리즘 제시
- 3) 배달기사 할당 문제(VRP)의 NP-Hardness를 고려한 알고리즘 제시
- 4) 우리나라 (특히, 서울)의 특성(높은 인구 밀도, 구역의 분리)을 고려한 알고리즘 제시
- 5) 기존 정책에 비해 높은 수준의 서비스 레벨 개선을 보임
- 6) 배달 플랫폼 시장의 모든 플레이어가 만족할 수 있는 배달 정책



2. 결과 해석에 따른 경영 상황에서의 시사점

- 1) 배달 관련 문제를 다루는 경우에는 시간-공간적 특성이 성능에 많은 영향을 준다는 것을 알고, 적극적으로 반영해야 함
- 2) Online 문제에서는 지연 전략을 통해 정보를 축적하여 더욱 좋은 결정을 내릴 수 있으며, 시간에 따라 바뀌는 상황에 맞도록 정책의 세부 사항을 지속적으로 업데이트 해야함

❖ 참고문헌

1. Boscoe, F. P., Henry, K. A., & Zdeb, M. S. (2012). A Nationwide Comparison of Driving Distance Versus Straight-Line Distance to Hospitals. The Professional geographer : the journal of the Association of American Geographers, 64(2)
2. Baris Yildiz, Martin Savelsbergh (2019) Provably High-Quality Solutions for the Meal Delivery Routing Problem. Transportation Science 53(5):1372-1388.
3. Reyes, D., Erera, A.L., Savelsbergh, M.W., Sahasrabudhe, S., O'Neil, R, (2018) The meal delivery routing problem. Accessed December 20
4. Steever, Z., Karwan, M., Murray, C.(2019) Dynamic courier routing for a food delivery service. Computers & Operations Research 107, 173 - 188
5. Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N.(2020) The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. Transportation Science
6. Z. K. Fkaier and B. F. Chaar (2013) Online K-means based heuristic for the dynamic pickup and delivery problem solving, 2013 World Congress on Computer and Information Technology (WCCIT), pp. 1-6
7. 박소정 (2021), 우아한형제들 매출 1조 돌파…年 거래액 15조, 뉴데일리 경제, <http://biz.newdaily.co.kr/site/data/html/2021/03/30/20210330000096.html>