Dynamic Pickup and Delivery Problem:

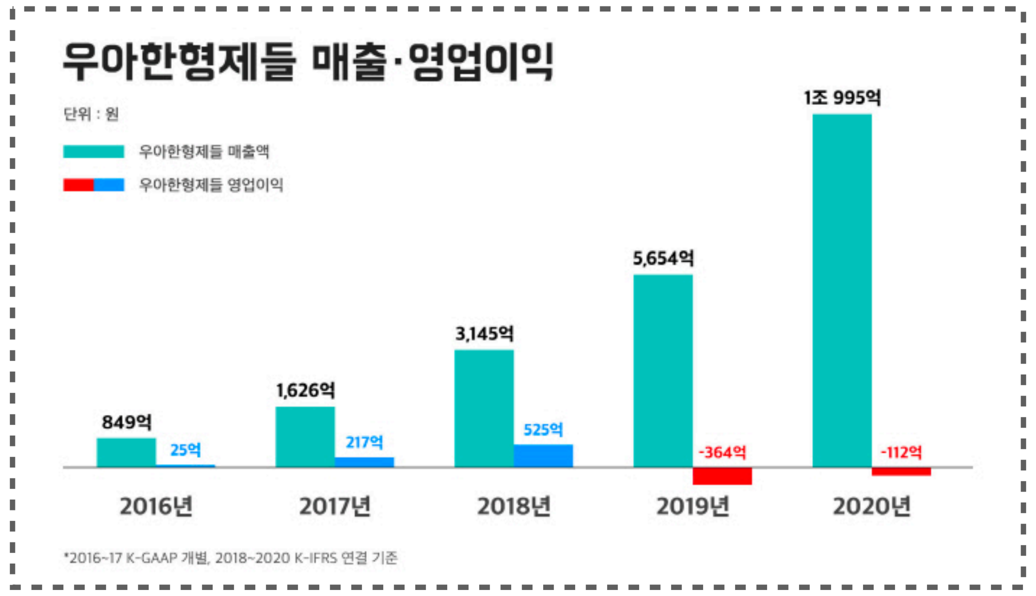
Spatial clustering-based Bundling and Allocation Heuristics, Postponement, and Hierarchical Order Classification

Haemin Park, Kihyun Han

① Problem Introduction

1. Introduction : Routing Problem in Delivery Platform

◆ Competitive Meal Delivery Platform Market



◆ Problems of Current Delivery System

Unable to follow the increase of user population and market size

- ✓ Inefficient delivery system due to market share competition
- ✓ Delivery policy where the cost is imposed by other players (Increasing costs due to market share competition between delivery platforms are shared by other players)
- ✓ Conflicts between players in the delivery market & Failure of public delivery applications

◆ Importance of Delivery Platform Market

- ✓ Growth of delivery market accelerated by COVID-19
- ✓ The platform that has the most influence on people's food life
- ✓ Complex systems that must adequately satisfy the needs of various players (platform, restaurants, couriers, consumers)

1) **Every player are unsatisfied in current delivery market**: 1 order - 1 courier (single order) policy results in unstable operation of the system, decrease in courier's average profit, and delayed delivery to customers

2) Delivery platform companies should consider order bundling but develop advanced delivery policy that resolves problem (such as delivery time exceed) in past bundling policies

◆ Problem Description : Dynamic Pickup and Delivery Problem with Time Window

Vehicle Routing Problem

VRP with Time Window

DPDPTW

OD(Order Dispatching) + VR(Vehicle Routing)

- ✓ Courier: should deal with orders coming in real-time —- move to restaurant, pick-up orders, move to destination and finish delivery
- ✓ Order comes in real-time with uncertainty
- ✓ Each orders should be delivered within specific time (deadline)
- ✓ Platforms should solve an optimization problem at each moment that allocates couriers that can efficiently deliver each order

◆ Why is it hard?

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

2. Purpose

♦ DPDPTW is a Difficult Problem

- DPDPTW is a computationally difficult problem, which is impossible to solve by theoretical methods practically
  ➡ need to find a heuristic-based delivery policy
- Only 2 papers Ulmer et al. (2020) and Reyes et al. (2018) deal with DPDPTW considering all temporal uncertainties in delivery processes
- However, papers above are not applicational in some parts ; large scale

♦ Things to Consider in Actual Operation

- 1 order - 1 courier policy requires many couriers
- When orders outnumbers couriers, there may be some remaining orders (need more couriers for stable delivery system)
- Couriers have lower expected profit with 1 order - 1 courier policy than the policy that adopts bundling
- Unrealistic policies (such as reversing the allocation to courier) should be excluded : Policy should provide benefits to all players
- Efficient and applicable policy is needed

♦ Research Goal

Propose delivery policy that

- **Considers Temporal Uncertainty**
- **Resolves Computational Problem (solve large-scale problem in seconds)**
- **Applicable in actual delivery platform**
- **Considers spatial characteristics of Korean delivery market**

♦ Basic Assumptions of the problem

(1) Delivery time is proportional to distance (Speed: 5 m/s)
(2) Delivery distance is defined as (Euclidean distance) X 1.4 (Boscoe et al. (2012))
(3) Order generation follows Poisson process
(4) Service time (restaurant, destination) is constant
(5) Courier moves through their optimal route after order allocation
(6) Courier stays if no route is assigned

Solve a complex problem involving Online + OD + VR with
1) order bundling with spatial clustering and postponement, 2) dynamic parameter assignment

3. Literature Review

◆ Problem Specifications

Deadline Consider deadline of orders? Each order should arrive until deadline. In practice, customer satisfaction plummets when an order exceeded the golden hour

Scalability Solve quickly for large scale problems? We need to propose an algorithm that deals with numerous computation of Vehicle Routing problem. Actual delivery platform should make a decision in few seconds.

Objective Focus on service level of customers? Delivery platforms should consider not only profit of the platform or delivery enterprise but also all players' satisfaction

Dynamism Use mathematically optimized parameter? As an online problem, situations and environments continuously change, and accordingly, we should take parameters optimized in the situation.

Bundling Consider bundling?
1 order - 1 courier can be inefficient in many cases

Postponement Consider postponement of order allocations?
* **Postponement** : A strategy that allocates courier better by postponing orders for a while to consider more orders, not immediately assigning new order to a courier

◆ 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 characteristics of DPDPTW are considered only in Ulmer et al. (2020) and Reyes et al. (2018)

	Reyes et al. (2018)	Ulmer et al. (2020)
Main Idea	Rolling Horizon Approach	Postponement + Time Buffer
Objective Function	Maximize (#orders per time) - (penalty considering delay)	Minimize sum of time delay
Bundling Method	Bundle orders in same restaurant	Bundle when it reduces time delay
Limitation	Do not consider the location of destinations while bundling Not intuitive objective(Lack of evidence in some penalty term)	Difficult to apply postponement for large scale due to computational issue (in the case of #couriers 15)

4. Problem Definition : Dynamic Pickup and Delivery Problem with Time Window (DPDPTW)

◆ Problem Situation

(1) Orders are generated in arrival rate λ Poisson process for service time [0,T] and service space $\mathcal{S}$, and $\mathcal{N}$ couriers work in the service space

(2) Each order is allocated to an unassigned courier that can efficiently perform (OD + VR)

(3) As soon as orders are assigned, couriers visit the restaurant and the destination in the optimal route

(4) Delivery platform should match courier-order pairs that satisfy all players and keep overall service quality (VR)

◆ Mathematical Modeling

For each decision timing, we should solve the following order allocation problem:

<Variables for each decision timing t>

$O = \{o_i : i \in I\}$: set of orders

$C = \{c_j : j \in J\}$: set of couriers

$\mathcal{Q} = \{(o_1, ..., o_m) \in O^m\}$: set of possible routes(of bundles) : set of order tuples

(Once the bundle and the assigned courier are determined, the optimal route is automatically determined)

$\delta^q_{o,c}$: delivery finish time when an order o of a route $q \in \mathcal{Q}$ is assigned to courier c

Should determine the following decision variables given above variables

<Decision Variables>

$x_{q,c} \in \{0,1\}$: 1 if route $q \in \mathcal{Q}$ is allocated to a courier $c \in C$, otherwise 0

$y_o \in \{0,1\}$: 1 if an order $o \in O$ is allocated to any courier, otherwise 0

◆ Problem Formulation at each decision timing t

*Extended Integer Programming Formulation from Reyes et al. (2018)
(Extended IP from the IP of Reyes et al. (2018) considering the specifications of this study)

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

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

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

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

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

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

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

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

** solve the above problem for every decision timing

Constraints Explanation

(1) : Each courier can be allocated at most one route

(2) : Each route can be allocated at most one courier (c=0 means no allocation)

(3) : Each order is included in a unique allocated route

(4) : Allocation of order requires a courier

(5) : The number of allocation should be maximized (minimum of the number of courier and the route)

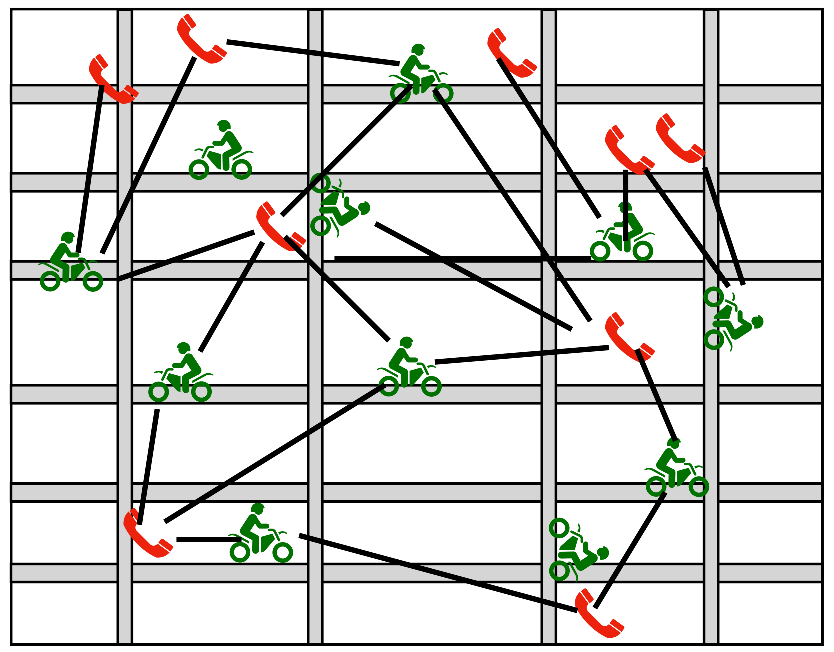
② Policy Proposal

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

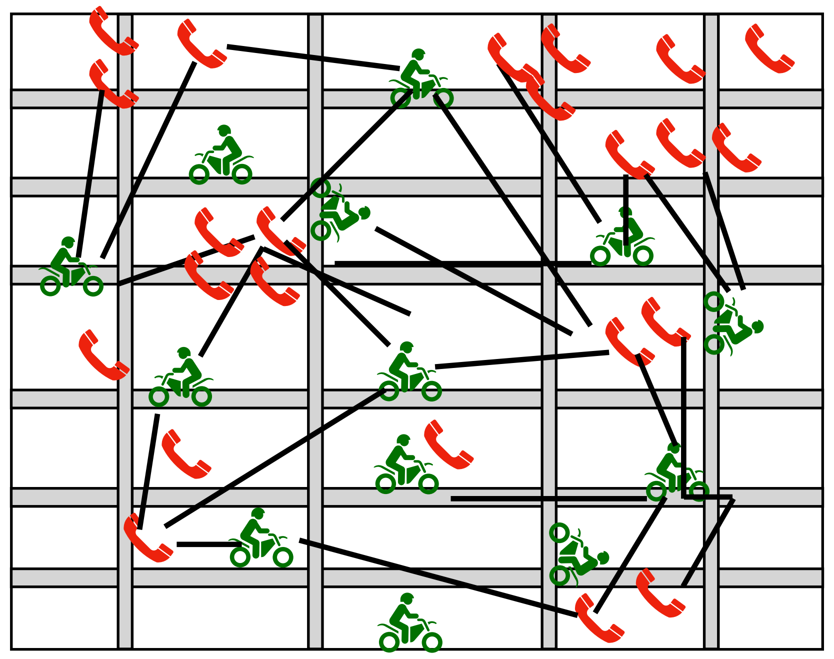
♦ Important Issues

1. Online Problem

- Situation changes with respect to time (ex) the number of orders, available couriers
- Policy should be flexible for each situation, instead of uniform for all service time
- Difficult to make a stable policy by temporal uncertainty
- (Ex) Lunch time, Dinner time, Rain, Snow, Sudden rush of orders



Policy details in two situations should be different



2. Spatio-Temporal Characteristics

- Delivery allocation problem should utilize spatio-temporal characteristics (especially in delivery destinations)
- Performance significant increases with considering the spatial characteristic
- (Ex) Orders of near restaurants, Orders accumulate with respect to time,
  Separation of residential and commercial areas

3. NP-hard Problem : Computation Problem

- Solve a combinatorial optimization problem(allocation) within a few seconds
- (Ex) 20 couriers and 10 orders make 10^{20} computations 10^{20}

➡ **1) Dynamic optimal parameter (considering online feature)**
2) Postponement and Bundling process based on paired-clustering (considering spatial distribution of restaurants)
3) Delivery policy with clustering-based allocation process and hierarchical order classification to solve computation costs

❖ **Remark : Comparison of delivery policies are based on service level derived from simulation, which is a mean of 30 times repetition for each design**

② Policy Proposal

Observation 1 : Imporatant Parameter in Meal Delivery Problem : WORKLOAD

Table 2 **Service Level by Bundle size, #Couriers, Workload**

	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

- Dissatisfication Level < 1%
- Dissatisfication Level < 2%
- Dissatisfication Level < 4%
- Dissatisfication Level < 8%
- Dissatisfication Level< 16%
- Dissatisfication Level< 32%
- Dissatisfication Level< 100%

Simulation Environments

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

Serviec level is highly affected by workload
➡ workload : important parameter
Service level has limitations around workload of 2-2.3
➡ We focus on that interval

* **Service Level** = #Orders arrived within deadline / #Total orders
* **workload** = average #orders that 1 courier should take in 1 hour

Prop 1 The probability of an order exceeding the deadline increases as the number of restaurants decreases

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

Proof)

Let $A = [0,1] \times [0,1]$ be the service space and m be the #total restaurants

Let the locations of i-th restaurant and associated j-th order's destination follow Uniform Distribution in A. i.e.,

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

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

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

and $\sum_{i=1}^{m}\sum_{j=1}^{n_i} E\left[\left|R_i - O_{ij}\right|\right] = 0.5214\, n$

➡ the expectation of delivery distance is independent of the number of restaurants

Now, we take a look at orders exceeding the deadline. $\chi\left(\left|R_i - O_{ij}\right| > 1\right) = 1\ or\ 0$

Is a value whether the order j with restaurant i exceeds the deadline. Then

$S_m = \sum_i \sum_j \chi\left(\left|R_i - O_{ij}\right| > l\right)$ becomes the number of total exceeded orders

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

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

Observation 2 : Spatio-Temporal Characteristics

Table 3 **Service level comparison of 10 restaurants and 10 clusters of restaurants (100 total)**

	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

Is there a difference?

In order to guaranee robustness, we used Nonparametric method (Random permutation test with t-statistic (2000 repetitions))

Bonferroni test : all p-values $< \dfrac{0.05}{m}$

➡ Service levels of two restaurants are same (sig. level 0.05)

As the number of restaurants are small, n restaurants and n clusters of restaurants have same service level. Thus, we should compare how restaurants are spatially distributed.

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 the 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

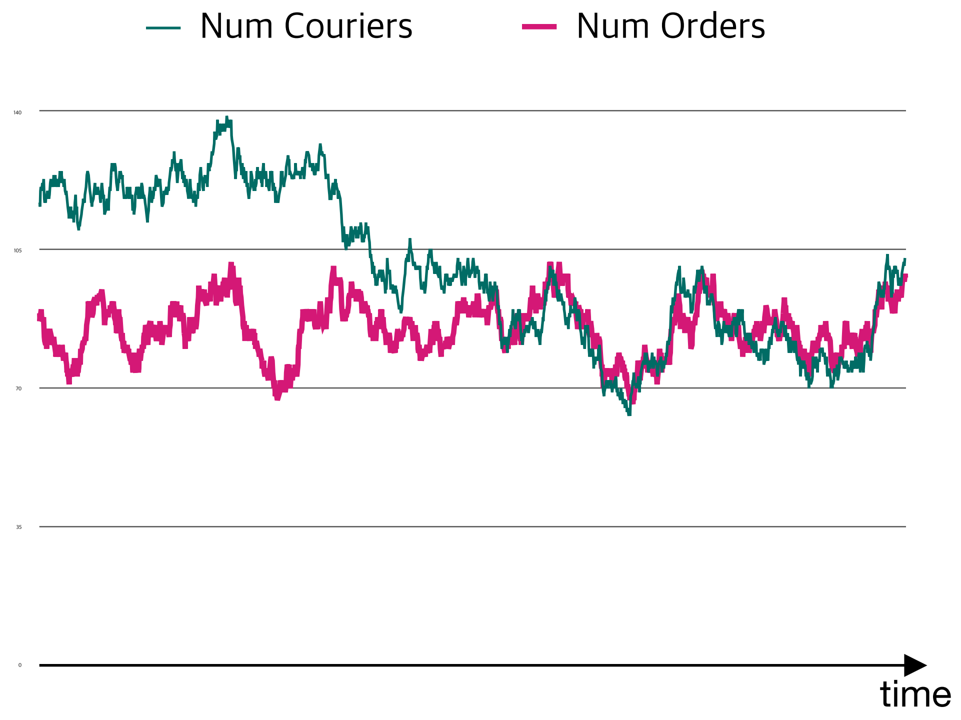
P2 Postponement

- 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

- As service environment changes, set a service parameter appropriate for the new environment (Parameter estimation was made with "Multi-layered Neural Network" model)

* Decision variables: cluster diameter, postponement time



Graph 1 Number of couriers and orders with time

(The system environment changes over time)

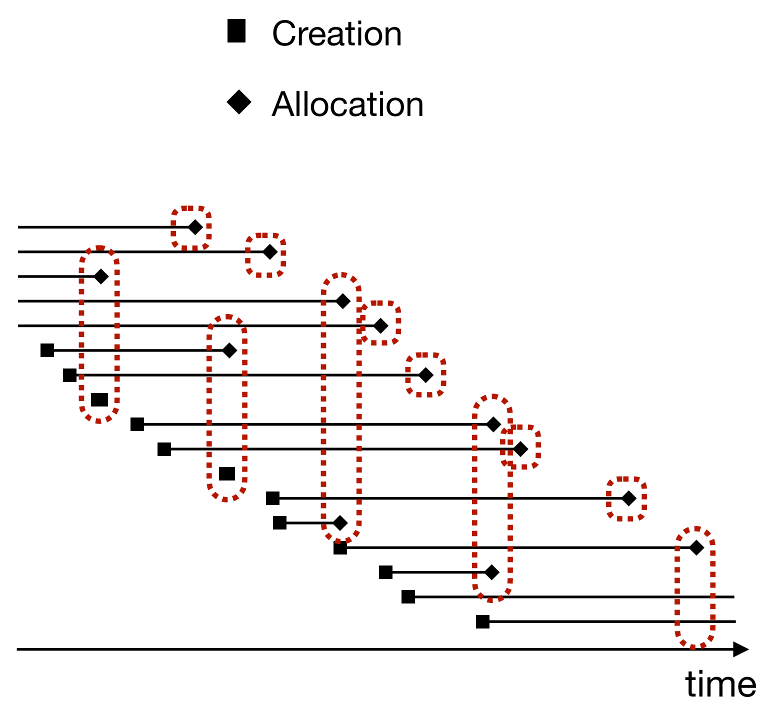


Figure 1 system environment fluctuates

(Collecting Information by Postponement)

② Policy Proposal

Observation 4 : Computation Problem

◆ Huge Amount of Calculations in allocation process

- If the #Order to be processed is m, #Couriers is h, then the total number of cases to consider is h^m (Find the routes of courier 1, 2, …, h for each order and compare them)
- Ulmer et al. (2020) proposed a heuristics that compares the permutations of orders with $m! \cdot m \cdot h$ cases
- For $m = 10,\ h = 36$, the exhaustive method needs to compare 3,656,158,440,062,976 cases, whereas Ulmer et al. (2020) needs to compare 306,368,000 cases
- The delivery platforms have much larger sizes, and the allocation process must proceed within seconds
  ➡ **Need a much faster algorithm**

<Proposal of Delivery Policy Considering the Observations>

P 4 Hierarchical Order Classification

- Orders/Bundles are processed hierarchically by their priority ranked based on the emergency
  New Order → Bundle → Bundle that exceeds the deadline → Urgent Bundle
- Increase service level by processing more urgent orders first
- Reduces #Inputs in each allocation process(reduce computations)

P 5 Spatial-Clustering Based Allocation Heuristics

(IDEA) Orders and couriers are likely to be paired with close ones → Since it is difficult to find order-courier pairs considering the all cases, We **first find the courier-cluster** that is near each order, and **then solve the sub-allocation** problem in each courier-clusters → Repeat this several times and pick the allocation of the best performance

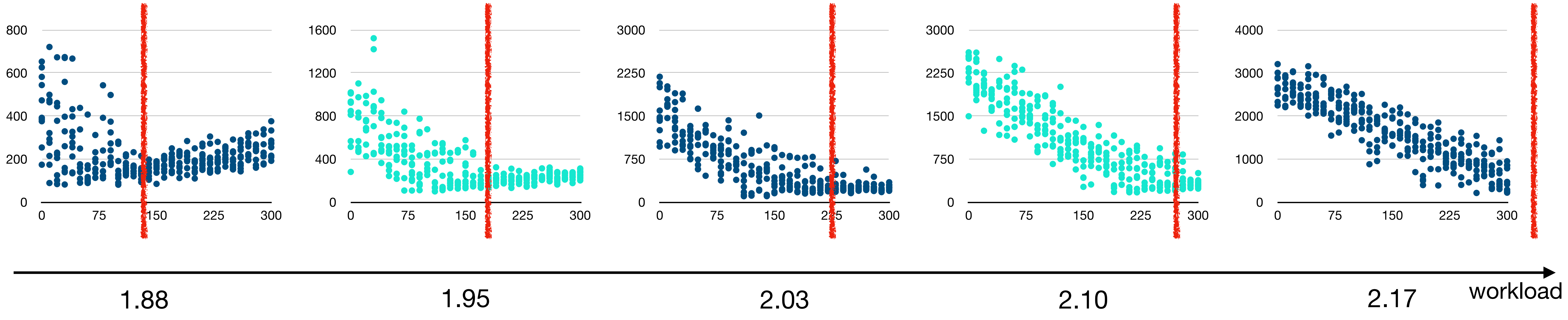
➡ Effect of reducing overlapping routes, which can improve the overall service level

Graph 2 **Delay Time - Service Level graph with respect to workload (importance of dynamic parameter setting)**

- x-axis: postponement time, y-axis: #orders that exceed the deadline (≈ Service Level)
- As the workload increases, shape of the graph stretches to the right, and thus the optimal postponement time increases.
- The more orders are placed, the more they are processed before the postponement finishes. Therefore, postponement time should be prolonged to maintain sufficient amount of orders(information)

Simulation Settings

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



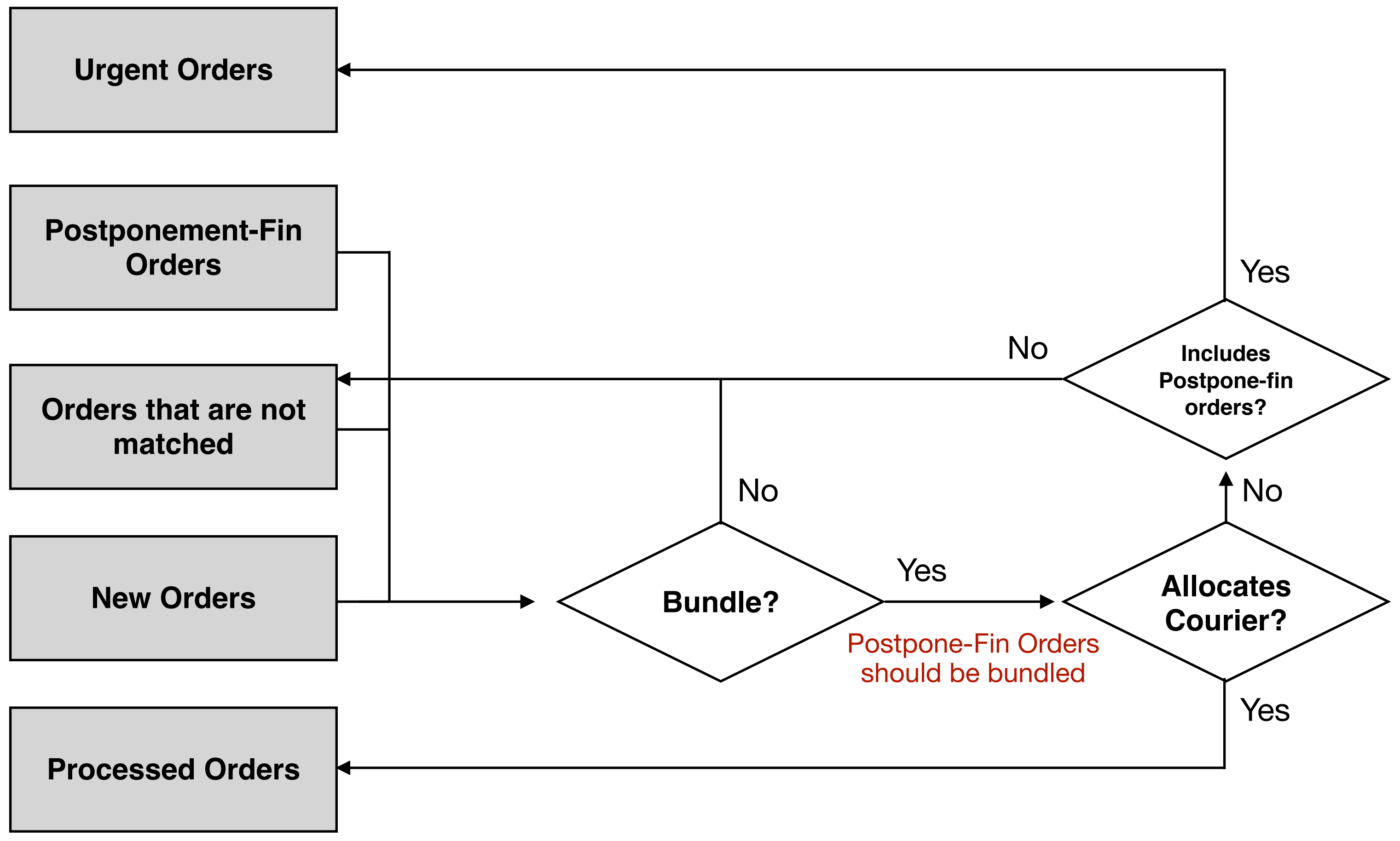


Figure 2 **Order Classification Diagram**

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

In each decision timing t, we need to solve opt. problem and allocate couriers to each bundles

<참고>

Decision timing occurs when 1) new orders are created, 2) postponements of some orders finish

Input of time-t process : orders, couriers, time-t simulation environment parameters(arrival rate, restaurant distribution factor)

Discrete time, system variables changes over time (ex. #orders that are not handled, #available couriers, temporal workload)

The delivery platform should allocate orders to the courier by solving allocation problem at any time required

Algorithm that solves Order-Courier allocation problem is processed as follows:

Process 1 : Emergency Bundle Handling

Process 2 : Pairwise Clustering Bundling with priority orders & general orders

Process 3 : Clustering-Based Heuristics Allocation

Process 4 : Update Courier Routes and Decision Points

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

★1
Policy 1 Dynamic Optimal Parameter Setting

(STEP 1) Find Approximated Service Level Function using Multilayered-Neural-Network Model
(Learning with about 200,000 data that has been simulated with various settings)

bundle size, postponement time, cluster diameter, restaurant distribution factor, arrival rate, initial number of courier ⇨ $S : \mathscr{F} \to \mathbb{R}$ ⇨ Service Level

(STEP 2) Find Optimal Parameter (bundle size, postponement time, cluster diameter) using Approximated Service Level Function

- Stage 1 : Adjust bundle size according to the workload
- Stage 2 : Set optimal (cluster diameter, postponement time) with $\hat{S}$

★2
Policy 2 Hierarchical Order Classification

Orders/Bundles are processed hierarchically by their priority ranked based on the emergency
New Order → Bundle → Bundle that exceeds the deadline → Urgent Bundle

1) Urgent bundles are assigned immediately when a new available courier occurs
2) Orders that exceed the postponement time are grouped first, and they are processed immediately even if they are not bundled with other orders(they are treated as bundle of size 1)
3) Of the orders not assigned by the entire process, the orders that are neither urgent nor postponement-finish orders remain as general orders. The rest of the orders are added to the urgent bundle set.

★3
Alg 1 Spatial-Paired-Clustering Based Bundling

Bundling Process based on Spatial Paired-Clustering

1. Construct clusters consisting of orders considering both locations of the restaurants and the delivery sites
2. Set bundle_list = []
3. For each cluster C
   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
   3.2. Make bundles of size B_{max} as many as possible with arbitrary remaining orders in the cluster C
   3.3. If the number of remaining orders is more than 1, then make a bundle with the remaining
   3.4. Push all bundles that are made with the cluster C to the bundle_list
4. End For

★4
Alg 2 Spatial-Clustering Based Allocation Heuristics

Spatial Clustering-Based Allocation Algorithm for the VRP

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

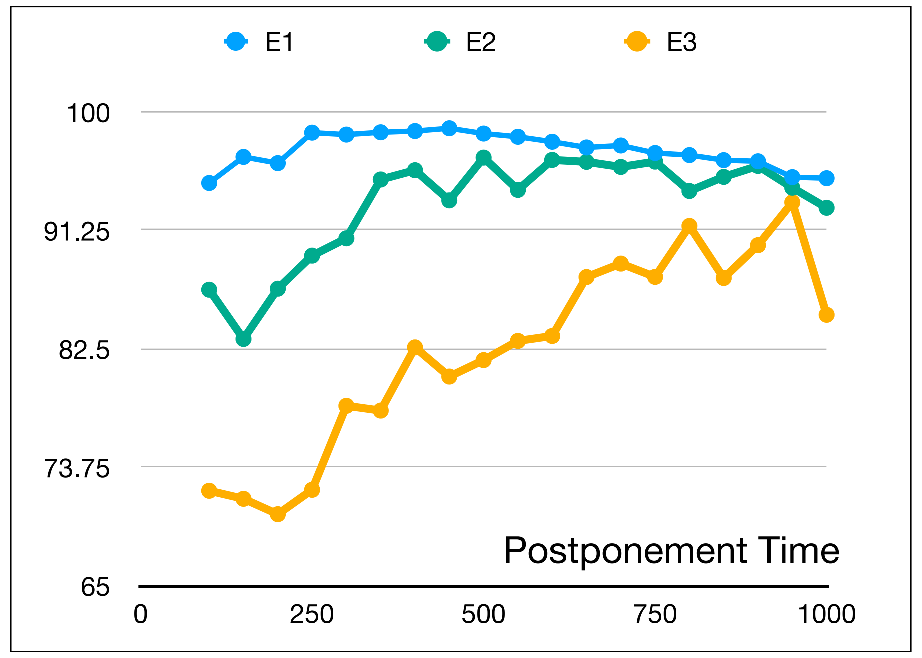
④ Results & Analysis

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)

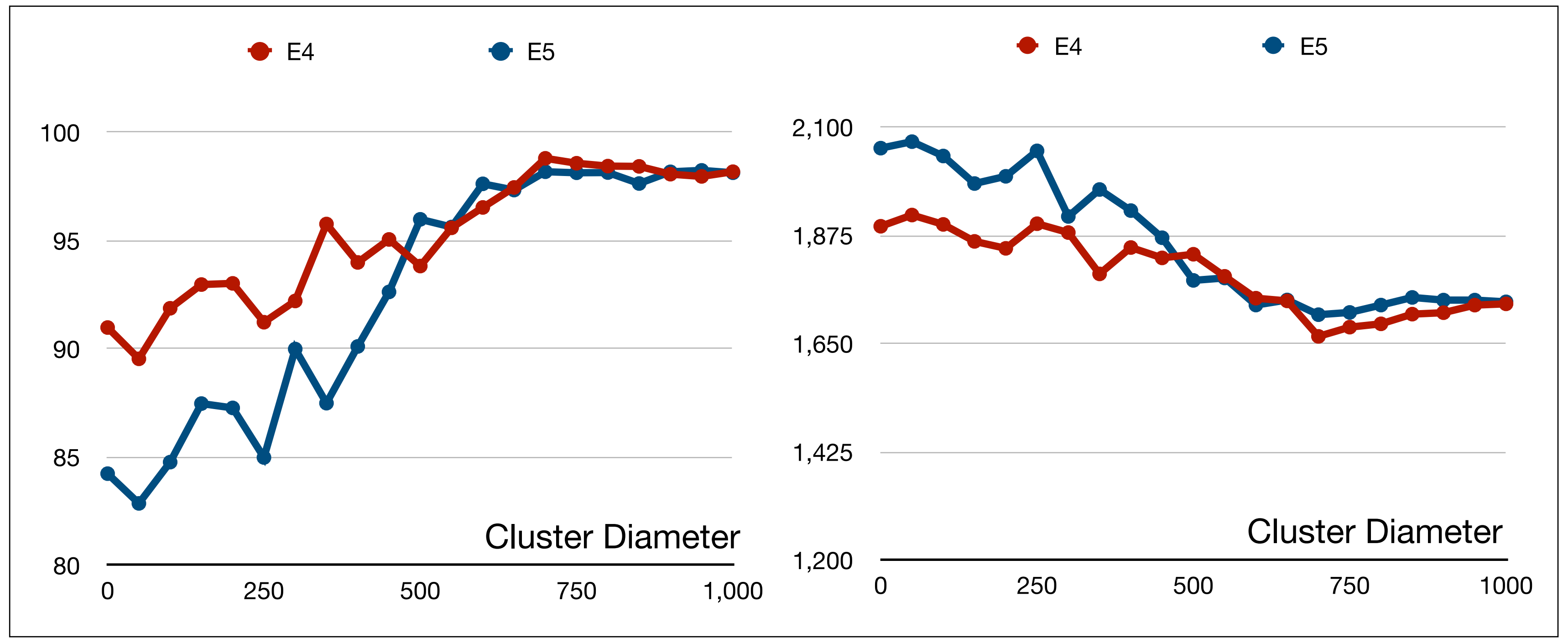
Simulation Result 1 : Service Level Function Learning & Effect of the Params

◆ The effect of the decision variables on service level



- In the workload is high(over 2), we check how the service level changes as the postponement time increases
- In the beginning, the service level increases as p increases, and at some point, the service level starts to decrease as p increases
- **The higher the workload, the longer the optimal p**
- **The graph is approximately concave (Assume it has optimality)**

Graph 5 **Postponement Time - Service Level Graph**



Graph 6 **Cluster Diameter - Service Level(L) & Expected Delivery Time(R) Graph**

- When workload is high(over 2), check how the service level and expected delivery time change as the cluster diameter increases
- In the beginning, the service level increasese as the cluster diameter increases. At some point, as the cluster diameter increases, the service level decreases
- **As the workload increases, the optimal cluster diameter tends to increase**

◆ Simulation $(B, p, c, W, R, N_R, N_C)$

Run Simulation for 30 times repetition with each parameter setting and get (#total orders, #orders exceed deadline, #bundling) datas : total 240,000 data

1) Bundle Size B (1, 2, 3)
2) Postponement time p (0, 50, 100, …, 1000 (s))
3) Cluster Diameter c (0, 50, 100, …, 1000 (m))
4) Workload W (0.5, 0.6, …, 3.0 (EA/hour·courier)
5) Rest. Randomness Factor R (0 ~ 0.24)
6) #Rest. N_R (10, 50, 100, 200, 400)
7) #Courier N_C (50, 100, …, 400)

*__Randomness__ = inter-event distance based spatial randomness

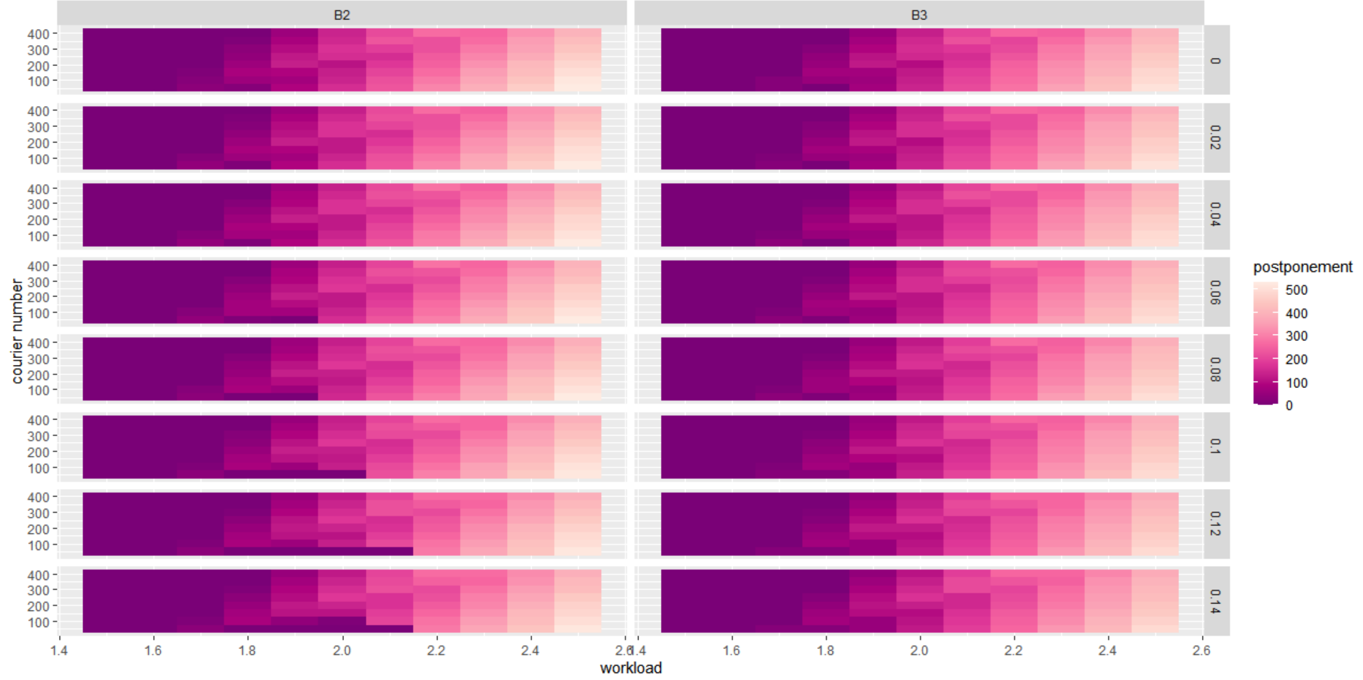
- Deadline : 50분
- Operating hours 8hours (28,800s)
- Sim. Space [0, 5000 m] X [0, 5000 m]
- Distance Factor : 1.4
- Service time at Rest. : 60s
- Service time at Destination : 180s
- Meal ready time : Tri distribution (MOD, COV)
- MOD setting : 5, 10, 20, 25, 30 mins with a ratio of 1:3:3:1.5:1:0.5 (reference from BAEMIN)
- COV of the meal ready time 0.1

◆ Learn Approximated Service Level Function

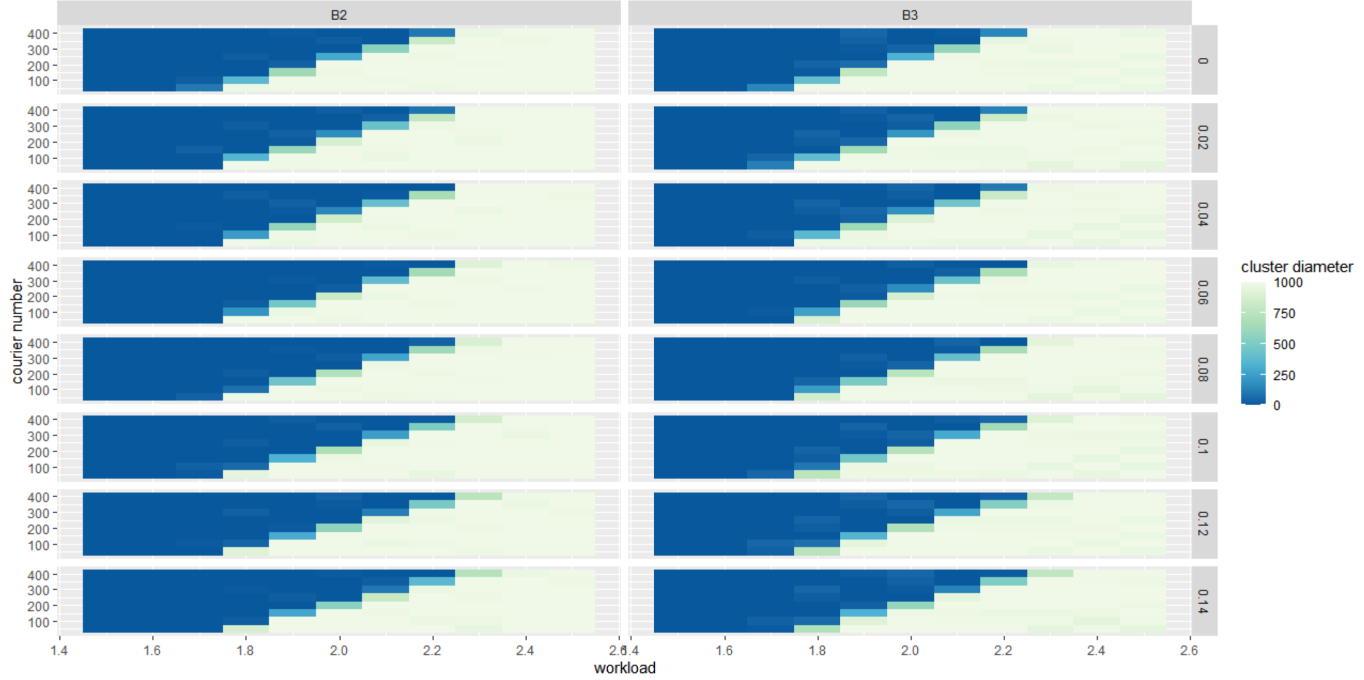
NN Model

Input Layer(7) - Hidden Layer 1(5) - Hidden Layer 2(5) - Output Layer(1)
→ #total params=76, Batch Size=50, Adam

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



Graph 3 **Optimal Postponement Time according to workload and #couriers**



Graph 4 **Optimal Cluster Diameter according to workload and #couriers**

- Same pattern for all parameter settings → Learning $f(B, W, R, \lambda, N_C, N_R)$ was good
- Optimal postponement time and cluster diameter increase as the workload increases

④ Results & Analysis

Simulation Result 2 : real restaurants data

◆ Simulation Setting

Run Simulation for 30 times repetition with each parameter setting and get (#total orders, #orders exceed deadline, #bundling) datas

- Confirm the performance of the policy in this study by running simulation on real restaurant distribution in Seoul
- Cut the space of 3km wide and long of area where there are many restaurants : utilizing the data from <Seoul Restaurant Registration Info>
- #Restaurants = 2834, randomness factor=0.1636

- Deadline : 50분
- Operating hours 8hours (28,800s)
- Sim. Space [0, 3000 m] X [0, 3000 m]
- Distance Factor : 1.4
- Service time at Rest. : 60s
- Service time at Destination : 180s

- Meal ready time : Tri distribution (MOD, COV)
- MOD setting : 5, 10, 20, 25, 30 mins with a ratio of 1:3:3:1.5:1:0.5 (reference from BAEMIN)
- COV of the meal ready time 0.1

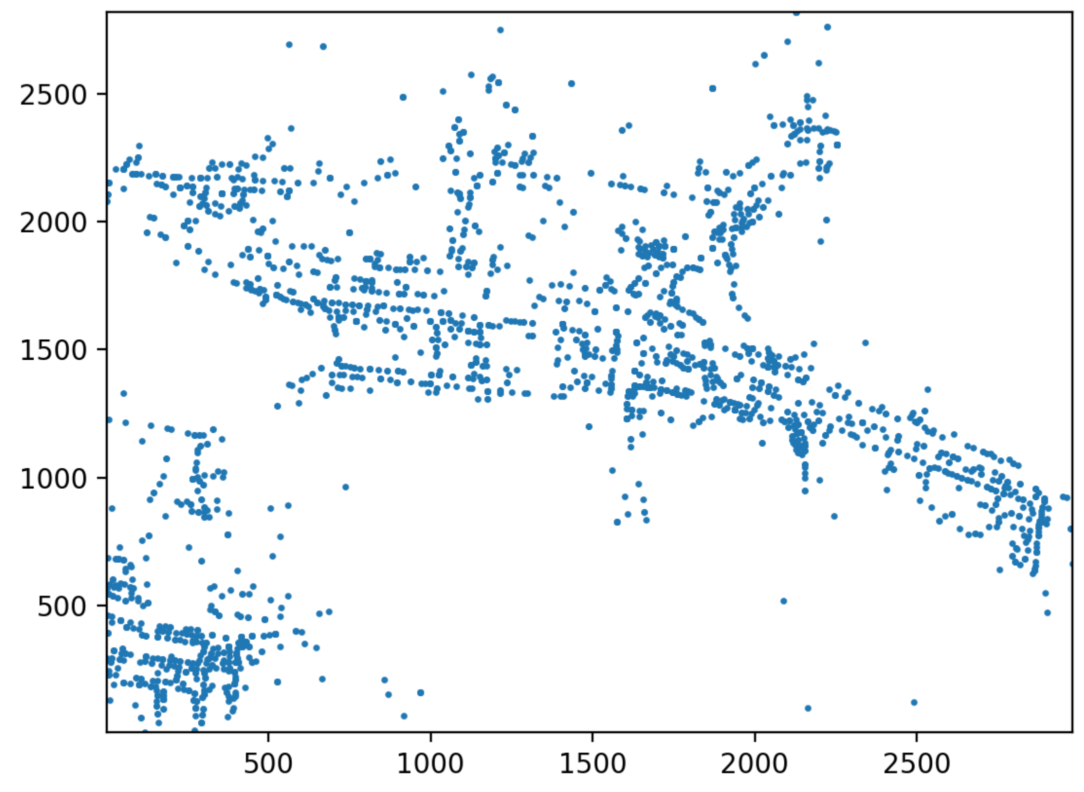
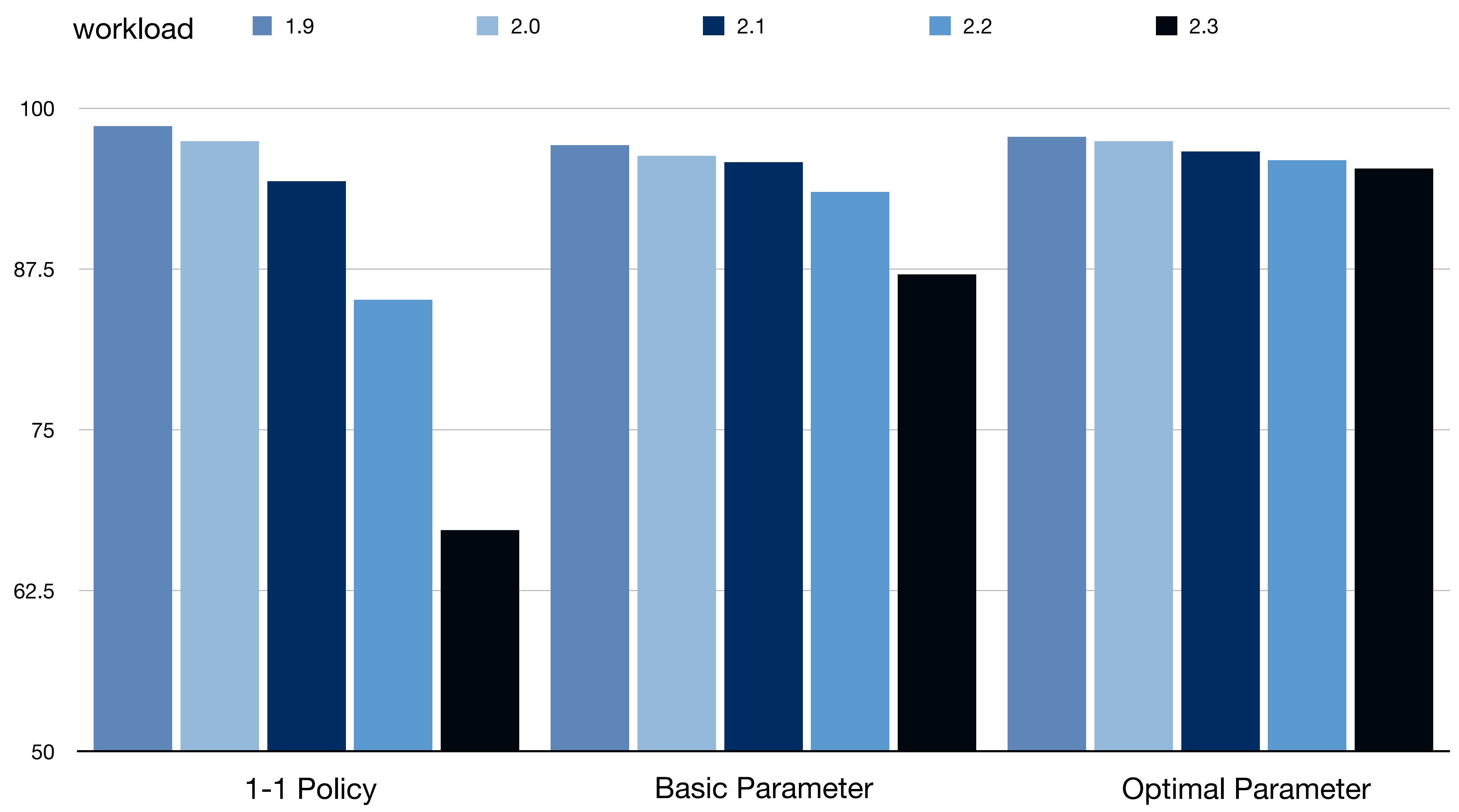


Figure 3 Distribution of restaurants in Seoul

Data from https://data.seoul.go.kr/dataList/OA-16094/A/1/datasetView.do#AXexec (Reference of datas)

◆ Simulation 1 : Compare Service Level for constant workload



Graph 6 **Service Level for each policy (constant workload)**

1) 1-1 Policy: 1 courier - 1 order policy without bundling and postponement
2) Basic Parameter: suggested policy with B=2, p=300s, c=500m
3) Optimal Parameter: suggested policy with optimal parameter from learning results

* Other studies set parameters based on their heuristics, not on theoretical reasons

- **In the case of constant workload, we confirm the performance of the policy in this study overwhelms the performance of the existing policy**
- **In optimal parameter, the policy maintains high service level of 95% with high workload of 2.3**

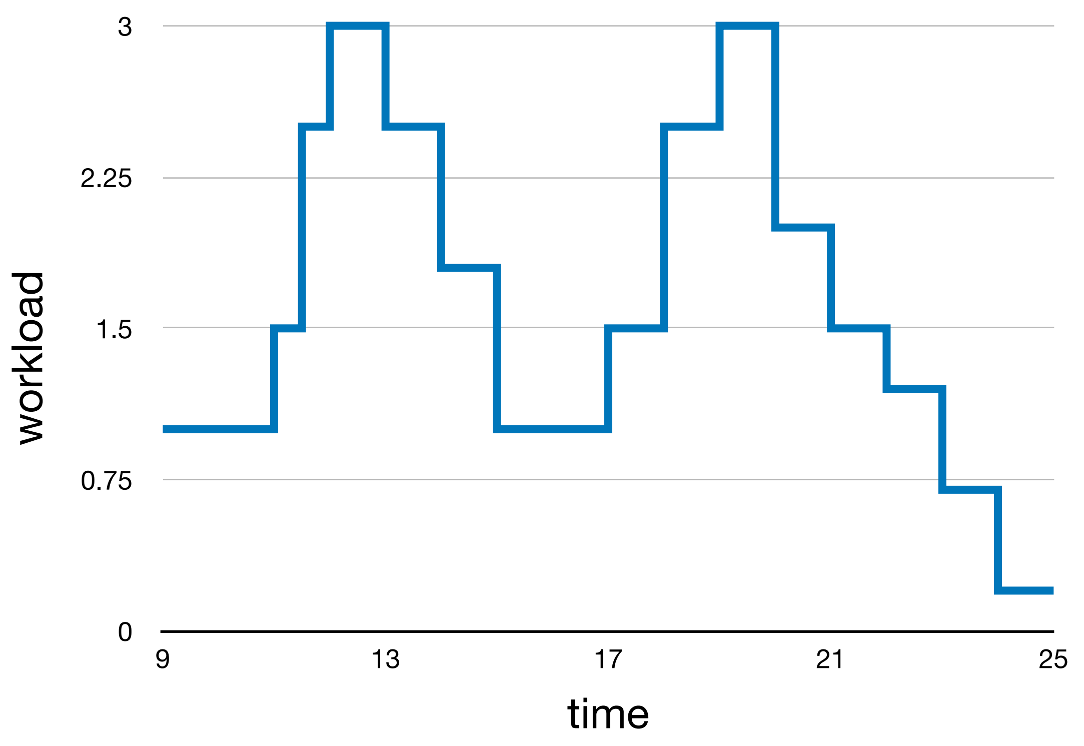
Simulation Results

Bundling and postponement significantly improve service levels compared to the control policy. In addition, it was confirmed that it is more effective to use the optimal parameters.

④ Results & Analysis

Simulation results 2 : real restaurants data & Analysis by each player

◆ Simulation 2 : Compare service level in the case when arrival rates change over time



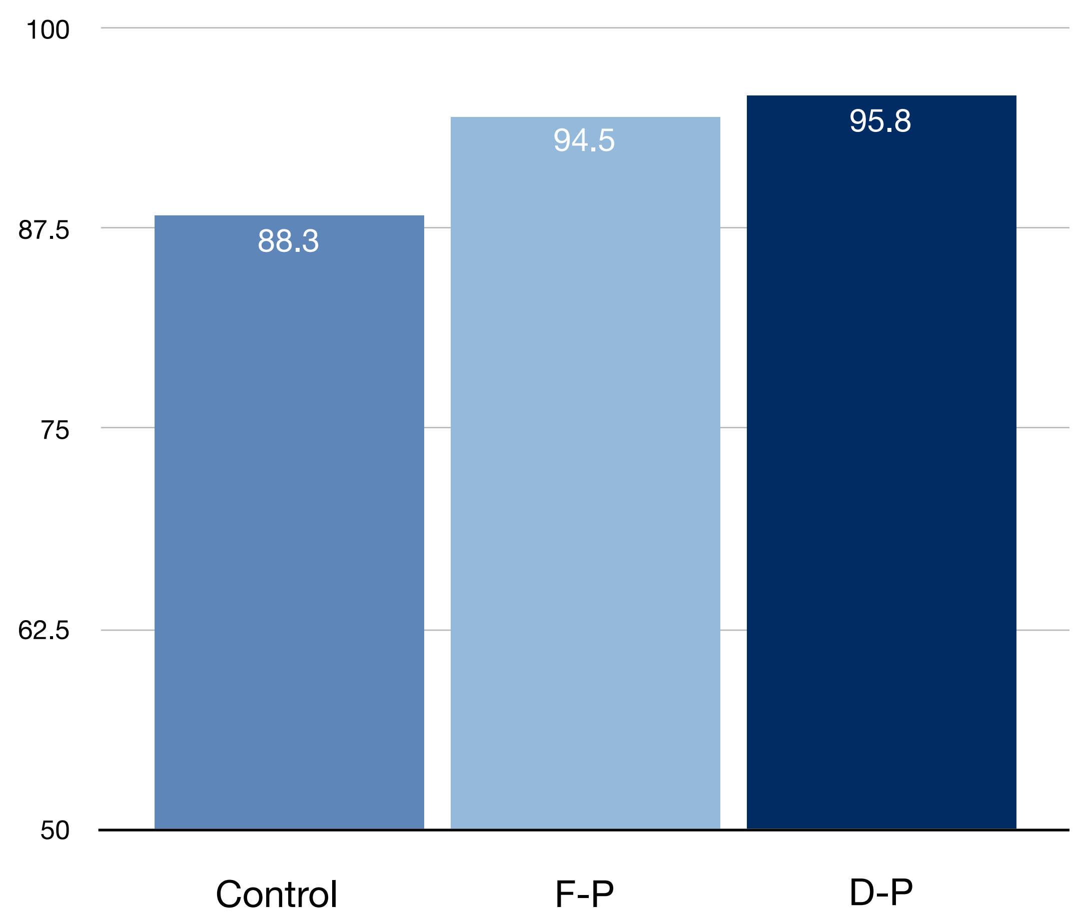
Graph 7 **Workload schedule graph by time-line**

Simulation Setting

- Runs for 16 hours from 9 AM to 1 AM the next day
- Reflecting the characteristics of orders that are concentrated during lunch and dinner time
- Assume that #couriers(300) is the same over the entire time

Delivery Policy

1) Control : 1 order - 1 courier policy without bundling, postponement
2) F-P : policy that uses the suggested heuristics with constant optimal parameters (for average environment settings)
3) D-P : policy that uses the suggested heuristics with dynamic optimal parameters (for each environment)



Graph 8 **Service level of each delivery policy**

- **Workload changes over time : The control policy shows a low service level of 88.3%, but when the policy proposed in this study is adopted, it maintains a high level of 95.8%.**
- **The service level is good even when the parameter was used fixed, but the service level is better when the parameters are dynamically adjusted according to the situation.**

<Remark>

As a result of t-test for the datas that has been experimented 30 times each, it was confirmed that the two distributions(F-P and D-P) were significantly different from a very small value with a p-value of less than 0.001.

◆ Analysis by each player in the delivery platform with the suggested policy

1. Meal Delivery Platform

- It can operate the system more stably even when the workload increases
- Good service level Even if the workload is high, and the number of minimum couriers to maintain certain service level decreases. In other words, stable operation is possible even with a small number of couriers
- Ex) To maintain service level of 95% in the space of 5km X 5km with arrival rate of 2000 EA/hour, control policy requires about 600 couriers whereas the suggested policy requires only 420 couriers.

2. Customer

- Even if the situation changes rapidly (surge in workload, a drop in the number of couriers), customers can receive orders stably within their deadlines
- Because of the bundling, customers can save money by dividing the cost together

3. Courier

- Earn more money since they can deliver more than one order at once (#orders processed per unit time increases, and they receive a delivery fee per delivery)
- If #couriers on the platform decreases, the average revenue will increase

4. Restaurant

- High service levels are always maintained and it guarantee the food quality
- Improved consumer satisfaction with restaurants

Simulation Results

1. **The delivery policy proposed in this study shows stable performance and large-scalability as we take into account the uncertainty caused by the online characteristics and the spatial distribution of restaurants.**
2. **Paired-Clustering Based Algorithms show powerful performance and are intuitive. We suggest that applying this algorithm in a type of SCM problem where location distribution is important would be a good research direction.**

5 Conclusions

1. Algorithm Proposal for Online Delivery Platform

Considering
1) Spatial distribution and temporal variation of orders
2) Characteristics of Online Problem
3) NP-Hardness of Vehicle Routing Problem
4) High population and separation of areas in Korea

Resulting in
5) Significant improvement in service level
6) Satisfaction of all players in the delivery platform market

2. Interpretations and Implications to Operational Circumstances

1) In problems related to delivery, Spatio-temporal Characteristics play an important role in performance, so we need to take these into account

2) In online problem, postponement leads to better decision by collecting more information, and specification of the policy should be updated with regard to fluctuating situations

❖ References

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/2021/03/30/2021033000096.html