Multi-period Workload Balancing in Last-Mile Urban Delivery

Yang Wang[a], Lei Zhao[a], Martin Savelsbergh[b], Shengnan Wu[c]

[a] *Department of Industrial Engineering, Tsinghua University, Beijing, 100084, China*
[b] *H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 30332, United States*
[c] *JD Logistics, Beijing, 100176, China*

Abstract

In the daily dispatching of urban deliveries, a delivery manager has to consider workload balance among the couriers to maintain workforce morale. We consider two types of workload: incentive workload, which relates to the delivery quantity and affects a courier's income, and effort workload, which relates to the delivery time and affects a courier's health. Incentive workload has to be balanced over a long period of time (e.g., a week or a month) whereas effort workload has to be balanced over a short period of time (e.g., a shift or a day). We formulate a multi-period workload balancing problem under stochastic demand and dynamic daily dispatching as a Markov Decision Process. We propose a balanced penalty policy based on Cost Function Approximation and use a hybrid algorithm combining the modified nested partitions method and the KN++ procedure to search for the optimal policy parameters. A comprehensive numerical study demonstrates that the proposed balanced penalty policy outperforms four benchmark policies and establishes the impact of demand variation and manager preferences on workload balance.

Keywords: Incentive & effort workload, Multi-period workload balancing, Last-mile urban delivery, Markov decision process, Cost function approximation

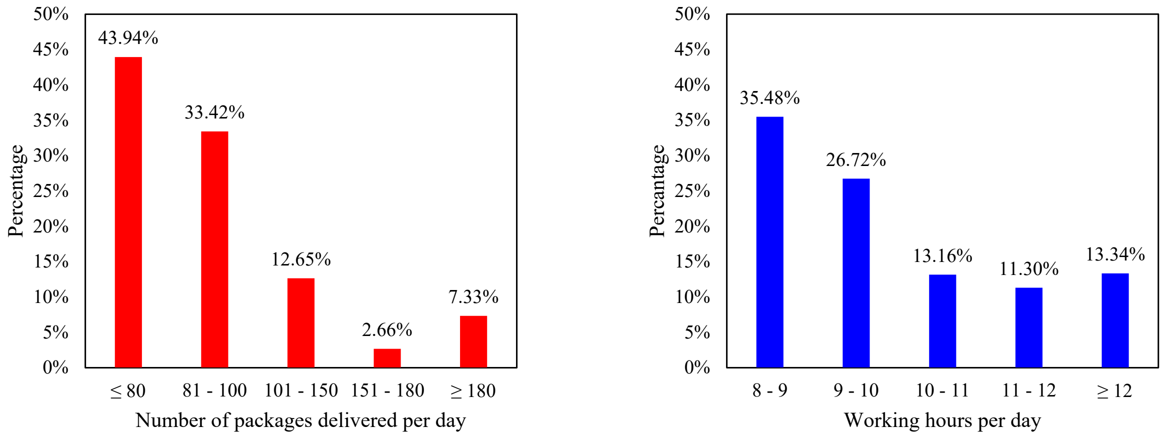
1. Introduction

Urbanization is one of the global forces, along with accelerating technological change, aging, and globalization, that is reshaping the world we live in (Dobbs et al., 2015). According to the United Nation's World Urbanization Prospects, the proportion of the world's population residing in urban areas increased from 30% in 1950 to 55% in 2018, and is expected to reach 68% by 2050. While the most urbanized regions are currently located in Northern America (82%), Latin America and the Caribbean (81%), and Europe (74%), Asia and Africa have faster urbanization rates than the other regions and their urban population proportion is expected to reach 66% and 59%, respectively, by 2050 (United Nations, 2019).

Likewise, the boom in Business-to-Customer (B2C) e-commerce is reshaping business models and supply chains. The worldwide B2C trading volume reached 1.82 trillion US dollars in 2017, to which China contributed 37%, followed by the United States with 23.8% and the United Kingdom with 10.6% (Ecommerce Foundation, 2017). In 2018, the trading volume in China reached 0.70 trillion US dollars with an annual increase of 26.1% (iResearch, 2019).

Urbanization and the advance of B2C e-commerce have both contributed to the rapid growth of urban delivery worldwide. The global delivery package volume reached 87 billion in 2018 (Pitney Bowes, 2019). In China, over 50 billion packages were shipped in 2018, and the quantity exceeded 60 billion in 2019 (State Post Bureau of P. R. China, 2020). About 80% of the express delivery volume in China in 2019 were e-commerce packages (People's Daily, 2019).

To accommodate the growth and to achieve the service level expected of B2C e-commerce, the number of couriers employed by companies providing urban delivery has increased dramatically. The "Survey report for China e-commerce logistics and express employees" (China Federation of Logistics & Purchasing and China Logistics Information Center, 2017), for example, reports that there were 3 million couriers in the express delivery industry in China in 2017. The leading players, SF Express and JD.com, had about 210,000 and 120,000 couriers, respectively. The same report also reveals that couriers work long hours and make many deliveries; Figure 1(a) and Figure 1(b) show the distributions of the number of packages delivered and the working hours per day, respectively. We see that the number of packages delivered per day varies significantly with 43.94% of the couriers delivering fewer than 80 packages per day and 7.33% of the couriers delivering more than 180 packages per day. Furthermore, 35.48% of the couriers work 8 – 9 hours per day, while 13.34% of the couriers work more than 12 hours per day.



(a) Number of packages delivered

(b) Working hours

Figure 1: Distributions of the number of packages delivered and the working hours per day among couriers

The disparities in courier working hours and number of deliveries have become a concern for urban delivery providers as they impact a courier's quality of life (especially the number of deliveries as in many parts of the world a courier's pay is directly linked to the number of deliveries) and, thus, courier retention rates. In this highly competitive, growing market segment, courier retention has become increasingly important. As a consequence, urban delivery providers are starting to explore ways to incorporate workload balance considerations into their operational planning processes. This is the topic of our research. We investigate how an urban delivery provider can achieve low operating cost while also providing workload balance, where workload balance has two components: a courier's delivery quantity, or *incentive workload*, and a courier's delivery time, or *effort workload*. What makes this especially interesting and challenging is that incentive workload has to be balanced over a relatively long period of time (a payroll cycle – a week or a month) and effort workload has to be balanced over a relatively short period of time (a shift or a day).

More specifically, we introduce and study a *multi-period workload balancing problem* in which we seek a daily dispatching policy minimizing expected operating cost and penalties for effort imbalance and incentive imbalance over a planning horizon (e.g., a 4-week period).

The contributions of our research are threefold. First, we study daily dispatching in last-mile urban delivery under stochastic demand while explicitly accounting for two types of workload balance with different periodicity (i.e., effort and incentive workload). Second, we formulate the problem as a Markov Decision Process (MDP), propose a balanced penalty policy based on Cost Function Approximation (CFA), and apply a hybrid algorithm combining the modified nested partitions method and the KN++ procedure to search for the optimal parameters of the CFA policy. Third, we perform a comprehensive numerical study which demonstrates that the proposed CFA-based balanced penalty policy outperforms four natural benchmark policies, and which provides insight into the impact of demand variation and manager importance weighting on operating cost and workload balance.

The remainder of the paper is organized as follows. In Section 2, we review relevant literature. In Section 3, we introduce the multi-period workload balancing problem and formulate it as a Markov Decision Process (MDP). In Section 4, we propose a CFA-based balanced penalty policy and four benchmark policies. In Section 5, we discuss the results of an extensive computational study. Finally, in Section 6, we present a few concluding remarks.

2. Literature review

Workload balance has been widely studied in the Vehicle Routing Problem (VRP) literature. To the best of our knowledge, Sutcliffe and Board (1990) were the first to include equity considerations in the VRP. In the context of transporting mentally handicapped adults to a training center, the authors impose lower and upper bounds on the trip time for each vehicle and the number of seats occupied in the vehicle. Jozefowiez et al. (2002) were the first to formally introduce the VRP with route balancing (VRPRB) and extend the capacitated VRP (CVRP) with a second objective that seeks to minimize the range of the tour lengths.

Matl et al. (2018) provide a comprehensive survey on workload equity in VRP. The authors make a distinction between *equity metrics*, which identify the resources to be balanced (e.g., tour length, delivery quantity) and *equity measures*, which specify the (in)equity calculation for a given resource allocation (e.g., maximum, range, standard deviation). Furthermore, the authors distinguish two types of equity metrics: constant-sum and variable-sum, depending on whether the resource "consumption" is identical for all routing solutions or not. For example, delivery quantity is a constant-sum equity metric because the total delivery quantity is a constant and does not depend on the routing solution, while delivery time is a variable-sum equity metric because the total delivery time is different for different routing solutions. Using the terminology introduced in Matl et al. (2018), the incentive workload considered in this paper (i.e., the delivery quantity) is a constant-sum metric and the effort workload (i.e., the delivery time) is a variable-sum metric. However, as discussed in Section 1, incentive workload is balanced over a long period of time (e.g., a week or a month) whereas effort workload is balanced over a short period of time (e.g., a shift or a day). This suggest that in addition to equity metric and equity measure, it may also be useful to consider *equity period*.

Next, we review the existing literature on single-period workload balancing and on multi-period workload balancing.

2.1. Single-period workload balancing

Table 1 summarizes existing literature on single-period workload balancing based on the equity measure, the model type, and the solution approach, where the model type indicates how workload balance is handled, i.e., as an objective in a pure multi-objective approach (MO), as an objective in weighted sum multi-objective approach (WS), or as a constraint (CN).

Few researchers have focused on incentive balance. Kritikos and Ioannou (2010) study the balanced cargo VRP with time windows (BCVRPTW), which penalizes cargo imbalance, defined as the sum of the absolute deviations from the average load of the vehicles, and propose a heuristic based on the free disposal hull (FDH) method of data envelopment analysis. Sarpong et al. (2013) extend the uncapacitated VRP with a second objective which seeks to minimize the maximum demand served by any of the vehicles, and apply an ϵ-constraint method to find the Pareto frontier (facilitated by column generation with efficient column search and bound computations).

Many studies, however, have focused on effort balance. As mentioned above, VRPRB extends the CVRP to a bi-objective problem with a second objective of minimizing the range of the route lengths of the vehicles. Multi-objective evolutionary algorithms are widely used to solve the VRPRB (Garcia-Najera and Bullinaria, 2011; Jozefowiez et al., 2002, 2006, 2007, 2009; Pasia et al., 2007a,b; Borgulya, 2008; Lacomme et al., 2015). Different from the above studies, Matl et al. (2019a) propose a heuristic box splitting (HBS) algorithm within the framework of ϵ-constraint method for obtaining the Pareto frontier.

Rather than minimizing the range of the route lengths of the vehicles, many researchers seek to minimize the maximum of the route lengths of the vehicles, i.e., the makespan (Corberán et al., 2002; Pacheco and Martí, 2006; Lacomme et al., 2006; Reiter and Gutjahr, 2012; Pacheco et al., 2013; Murata and Itai, 2005, 2007). Most of these studies use (meta)heuristics to obtain a Pareto frontier. Reiter and Gutjahr (2012) apply an adaptive ϵ-constraint method to find a Pareto frontier, in combination with a branch-and-cut algorithm and two genetic algorithms for solving distance-constrained VRP subproblems.

Table 1: Literature on single-period workload balancing

Publications	Equity measure			Model type			Solution approach	
	Max	Range	Other	MO	WS	CN	Exact	Heuristic
Incentive balance								
Kritikos and Ioannou (2010)			✓		✓			✓
Sarpong et al. (2013)	✓			✓			✓	
Effort balance								
Jozefowiez et al. (2002, 2006, 2007, 2009)		✓		✓				✓
Pasia et al. (2007a,b)		✓		✓				✓
Borgulya (2008)		✓		✓				✓
Garcia-Najera and Bullinaria (2011)		✓		✓				✓
Melián-Batista et al. (2014)		✓		✓				✓
Lacomme et al. (2015)		✓		✓				✓
Matl et al. (2019a)		✓		✓				✓
Corberán et al. (2002)	✓			✓				✓
Murata and Itai (2005, 2007)	✓			✓				✓
Pacheco and Martí (2006)	✓			✓				✓
Lacomme et al. (2006)	✓			✓				✓
Reiter and Gutjahr (2012)	✓			✓			✓	
Pacheco et al. (2013)	✓			✓				✓
Lee and Ueng (1999)			✓		✓			✓
Halvorsen-Weare and Savelsbergh (2016)	✓	✓	✓	✓			✓	
Both incentive and effort balance								
Baņos et al. (2013a,b)		✓		✓				✓
Matl et al. (2018, 2019b)	✓	✓	✓	✓			✓	
Sutcliffe and Board (1990)			✓			✓	✓	
Bowerman et al. (1995)			Variance		✓			✓
Apte and Mason (2006)			✓		Length	Load		✓
Liu et al. (2006)		✓				✓		✓

Lee and Ueng (1999) extend the CVRP by seeking to minimize the weighted sum of the total travel time and the sum of the differences between each vehicle's delivery time and the smallest delivery time of any of the vehicles, and propose a saving-based heuristic to solve the problem. Halvorsen-Weare and Savelsbergh (2016) study four bi-objective mixed capacitated general routing problems, each with a different effort balance objective, implement the box method (Hamachera et al., 2007), and demonstrate that minimizing the range of the tour lengths is a robust choice.

A small number of researches study both incentive and effort balancing. Some of these studies extend the CVRP into bi-objective problems with a second objective that seeks to minimize either the incentive imbalance or the effort imbalance (Baņos et al., 2013a,b; Matl et al., 2018, 2019b), while others consider incentive and effort imbalance simultaneously (Sutcliffe and Board, 1990; Bowerman et al., 1995; Apte and Mason, 2006; Liu et al., 2006).

Bowerman et al. (1995) study a school bus routing problem with five goals. The most important goal is to minimize the number of routes, which therefore becomes the primary objective. The remaining four goals are combined, using a weighted sum, into a second objective; these goals include two equity measures (i.e., the variance of the numbers of students served by the buses and the variance of the route lengths of the buses). Apte and Mason (2006) study a library materials delivery problem, which seeks to minimize a weighted sum of the total travel distance and the sum of the differences between the length of a route and a target route length. Lower and upper bounds on the number of stops and on the delivery quantity of a vehicle are also imposed. Liu et al. (2006) extend the VRP with limits on the range of the delivery times and on the range of the delivery quantities of the vehicles.

2.2. Multi-period workload balancing

There are only a few studies on multi-period workload balancing. Some of these naturally arise in the context of the periodic vehicle routing problem (PVRP). Blakeley et al. (2003) study a periodic technician routing and scheduling problem for Schindler Elevator Corporation, which seeks to minimize the total cost over the planning horizon where cost is a weighted sum of total travel time, time-window violations, total overtime, total idle time, and workload imbalance defined as the standard deviation of the route times over the routes. The authors propose a two-phase heuristic involving an assignment procedure and a sequence procedure. Mourgaya and Vanderbeck (2007) study a variation of the PVRP in which only customer to period and customer to vehicle decisions are considered. When assigning customers to vehicles (referred to as clustering), the authors seek to balance vehicle workload and cluster coverage. Balancing vehicle workload is achieved by bounding the total load of a cluster and balancing cluster coverage is achieved by minimizing the sum of the Euclidean distances between customer locations within a cluster. Gulczynski et al. (2011) propose an integer programming-based heuristic for the PVRP with reassignment constraints (PVRP-RC) and the PVRP with balance constraints (PVRP-BC). In the PVRP-BC, the authors penalize the workload imbalance measured as the range of the number of customers served by any of the vehicles in any of the periods of the planning horizon. Liu et al. (2013) study a weekly home health care logistics optimization problem, an extension of the PVRP with time windows, in which they seek to minimize the longest route length over the week, and propose a hybrid tabu search algorithm combined with two intra-route local search strategies.

Ribeiro and Lourenço (2001) consider a multi-period VRP extension which seeks to minimize a weighted sum of three objectives: the total cost, the workload imbalance measured as the standard deviation of the total delivery volumes of the vehicles, and the inconsistency in customer service. The authors provide a small illustrative example, which is solved using LINGO. Groër et al. (2009) study a balanced billing cycle vehicle routing problem encountered by utility companies in which the minimum and maximum numbers of customers served as well as the tour lengths of the meter readers are constrained, and propose a three-phase approach that transforms the initial unbalanced and inefficient billing routes into target routes generated by a modified record-to-record algorithm considering workload balance.

All the problems referred to in our review of the literature are deterministic. We study a multi-period workload balancing problem under stochastic demand and dynamic daily dispatching, in which we explicitly account for two types of workload balance with different equity periods. Incentive workload relates to the delivery quantity and affects a courier's income – balanced across couriers over a long period of time (e.g., a week or a month). Effort workload relates to the delivery time and affects a courier's health – balanced across couriers over a short period of time (e.g., a shift or a day).

3. Problem description and model formulation

A last-mile urban delivery station operates a homogeneous fleet of couriers $\mathcal{K} = \{1, 2, ..., K\}$. Each courier has a daily delivery capacity Q. On a daily basis, the delivery manager dispatches the couriers to deliver packages to customers within its service region. To maintain workforce morale, the delivery manager needs to consider not only the operating cost, but also two types of workload balance: incentive workload balance and effort workload balance. Incentive workload relates to the delivery quantity (i.e., number of packages) and affects a courier's income, while effort workload relates to the delivery time (i.e., travel time plus service time) and affects a courier's health. Couriers care about effort workload balance each day and care about incentive workload balance each payroll cycle. For presentational convenience, we refer to "incentive workload (im)balance" as "incentive (im)balance" and "effort workload (im)balance" as "effort (im)balance. In this study, we consider the workload balancing problem over each payroll cycle (planning horizon). We denote $\mathcal{T} = \{1, 2, \ldots, T\}$ as the set of periods (i.e., days) in the payroll cycle. As shown in Figure 2, period $t+1$ begins with decision epoch t and ends with decision epoch $t+1$. (We will explain the elements in Figure 2 as we describe the problem.)

In practice, for operational simplicity, the last-mile delivery station aggregates the customers in the service region into delivery units (e.g., residential/office buildings) based on geographical and demographical characteristics. Customers within a delivery unit are served by one courier and each courier can serve multiple delivery units. We model the service region as a directed network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. Node set $\mathcal{N} = \{0\} \cup \mathcal{N}^c$, where 0 represents the last-mile delivery station and $\mathcal{N}^c = \{1, 2, \ldots, N\}$ represents the set of delivery units. Each

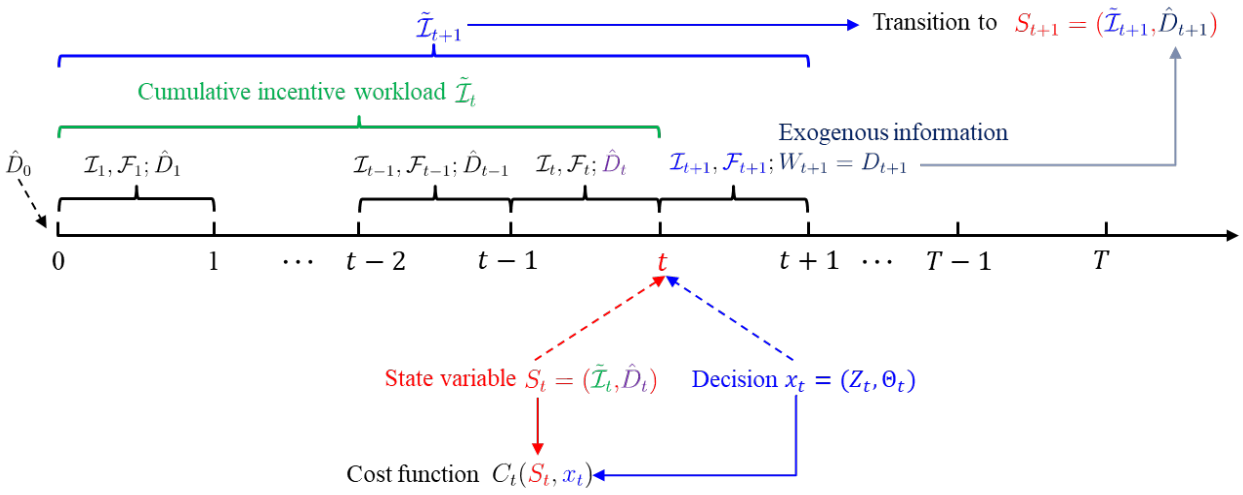

$\tilde{\mathcal{I}}_{t+1}$
Transition to $S_{t+1} = (\tilde{\mathcal{I}}_{t+1}, \hat{D}_{t+1})$
Cumulative incentive workload $\tilde{\mathcal{I}}_t$
Exogenous information
$\hat{D}_0$
$\mathcal{I}_1, \mathcal{F}_1; \hat{D}_1$
$\mathcal{I}_{t-1}, \mathcal{F}_{t-1}; \hat{D}_{t-1}$
$\mathcal{I}_t, \mathcal{F}_t; \hat{D}_t$
$\mathcal{I}_{t+1}, \mathcal{F}_{t+1}; W_{t+1} = D_{t+1}$
0
1
$\cdots$
$t-2$
$t-1$
t
$t+1$
$\cdots$
$T-1$
T
State variable $S_t = (\tilde{\mathcal{I}}_t, \hat{D}_t)$
Decision $x_t = (Z_t, \Theta_t)$
Cost function $C_t(S_t, x_t)$

Figure 2: Problem illustration

customer in delivery unit $i \in \mathcal{N}^c$ has a service time s_i. Arc set $\mathcal{A}$ represents the connections between nodes in $\mathcal{N}$. The travel time on each arc $(i, j) \in \mathcal{A}$ is t_{ij}.

Because each delivery unit aggregates multiple customers, it is reasonable to assume that each delivery unit $i \in \mathcal{N}^c$ generates a positive demand $d_{ti} > 0$ in each period $t \in \mathcal{T}$, to be delivered in period $t + 1$. The demand d_{ti} varies in each period and is represented as a random variable. We denote $D_t = (d_{ti})_{\forall i \in \mathcal{N}^c}$, $\forall t \in \mathcal{T}$. At the beginning of the planning horizon, we start with $\hat{D}_0$, which is the demand generated before decision epoch 0 and is to be delivered in period 1. Note that D_T will be delivered in the next planning horizon and is not included in the current model.

At each decision epoch t, the delivery manager dispatches couriers based on the realized demand in period t and the *cumulative* incentive workload of the couriers up to decision epoch t so as to minimize the operating cost (i.e., total delivery time) and effort imbalance for period $t + 1$ as well as the *expected* incentive imbalance over the (active) payroll cycle. In this paper, we use *cumulative* incentive workload up to decision epoch t to refer to the total incentive workload of a courier from period 1 to period t, and *total* incentive workload to refer to the incentive workload of a courier from period 1 to period T, i.e., over an entire payroll cycle. Next, we formulate the *multi-period workload balancing problem* as a Markov Decision Process (MDP).

3.1. State variable

We define $\hat{D}_{t-1}$ as the generated demand in period $t - 1$, which is the realization of the random variable D_{t-1}, to be delivered in period t. In period t, each courier k delivers the assigned demand, which results in the incentive workload I_{tk} (i.e., number of packages) and the effort workload F_{tk} (i.e., travel time plus service time). We define $\mathcal{I}_t = (I_{tk})_{\forall k \in \mathcal{K}}$ and $\mathcal{F}_t = (F_{tk})_{\forall k \in \mathcal{K}}$. Further, we denote the cumulative incentive workload of courier k up to decision epoch t as $\tilde{I}_{tk} = \sum_{\tau=1}^{t} I_{\tau k}$, and $\tilde{\mathcal{I}}_t = (\tilde{I}_{tk})_{\forall k \in \mathcal{K}}$, $\forall t \in \mathcal{T}$.

At decision epoch t, the state variable $S_t = (\tilde{\mathcal{I}}_t, \hat{D}_t)$ represents the cumulative incentive workload up to decision epoch t and the generated demand in period t (as shown in Figure 2). At decision epoch 0, each courier has no cumulative incentive workload, i.e., $\tilde{I}_{0k} = 0, \forall k \in \mathcal{K}$.

3.2. Decision variable

At decision epoch t, the delivery manager dispatches the couriers to serve the generated demand $\hat{D}_t$ in period $t+1$. We assume a courier serves his assigned demand using the fastest possible delivery route. We define z_{tik} to indicate whether courier k is dispatched to serve delivery unit i in period $t+1$, and $Z_t = (z_{tik})_{\forall i \in \mathcal{N}^c, k \in \mathcal{K}}$. We also define θ_{tk} as the route of courier k to serve the assigned delivery units in period $t+1$, and $\Theta_t = (\theta_{tk})_{\forall k \in \mathcal{K}}$. The decision variable $x_t = (Z_t, \Theta_t)$ (as shown in Figure 2), which needs to satisfy Constraints (1) as described below.

We introduce a dummy depot $N+1$ and extend the network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ to $\overline{\mathcal{G}} = (\overline{\mathcal{N}}, \overline{\mathcal{A}})$. We define the extended node set $\overline{\mathcal{N}} = \mathcal{N} \cup \{N+1\}$ and the extended arc set $\overline{\mathcal{A}} = \mathcal{A} \cup \{(i, N+1) : i \in \mathcal{N}^c\}$ with $t_{i,N+1} = t_{0i}, \forall i \in \mathcal{N}^c$.

We extend the two-commodity formulation in Baldacci et al. (2004) by adding the index of couriers, and introduce the following decision variables.

$$z_{tik} = \begin{cases} 1, & \text{if courier } k \in \mathcal{K} \text{ is dispatched to serve delivery unit } i \in \mathcal{N}^c, \\ 0, & \text{otherwise,} \end{cases}$$

$$\xi_{tijk} = \begin{cases} 1, & \text{if courier } k \in \mathcal{K} \text{ traverses the arc } (i,j) \in \overline{\mathcal{A}}, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_{tijk} = \begin{cases} \text{occupied capacity}, & \text{if courier } k \in \mathcal{K} \text{ traverses the arc } (i,j) \in \overline{\mathcal{A}}, \\ \text{residual capacity}, & \text{if courier } k \in \mathcal{K} \text{ traverses the arc } (j,i) \in \overline{\mathcal{A}}, \end{cases}$$

where ξ_{tijk} can be transformed into the routing decision θ_{tk} by post-processing.

We define the feasible region $\mathcal{X}_t$ of the decision $x_t = (Z_t, \Theta_t)$ in Constraints (1).

$$\sum_{i \in \overline{\mathcal{N}}} (y_{tjik} - y_{tijk}) = 2 d_{ti} z_{tik}, \qquad \forall i \in \mathcal{N}^c, k \in \mathcal{K}, \tag{1a}$$

$$\sum_{j \in \mathcal{N}^c} y_{t0jk} = \sum_{i \in \mathcal{N}^c} \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}, \tag{1b}$$

$$\sum_{j \in \mathcal{N}^c} y_{tj0k} = Q - \sum_{i \in \mathcal{N}^c} \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}, \tag{1c}$$

$$\sum_{j \in \mathcal{N}^c} y_{t,N+1,jk} = Q, \qquad \forall k \in \mathcal{K}, \tag{1d}$$

$$y_{tijk} + y_{tjik} = Q \xi_{tijk}, \qquad \forall (i,j) \in \overline{\mathcal{A}}, k \in \mathcal{K}, \tag{1e}$$

$$\sum_{(i,j) \in \overline{\mathcal{A}}} \xi_{tijk} + \sum_{(j,i) \in \overline{\mathcal{A}}} \xi_{tjik} = 2 z_{tik}, \qquad \forall i \in \mathcal{N}^c, k \in \mathcal{K}, \tag{1f}$$

$$\sum_{k \in \mathcal{K}} z_{tik} = 1, \qquad \forall i \in V^c, \tag{1g}$$

$$\xi_{tijk} \in \{0, 1\}, \qquad \forall (i,j) \in \overline{\mathcal{A}}, k \in \mathcal{K}, \tag{1h}$$

$$y_{tijk} \geq 0, \qquad \forall (i,j) \in \overline{\mathcal{A}}, k \in \mathcal{K}, \tag{1i}$$

$$z_{tik} \in \{0, 1\}, \qquad \forall i \in \mathcal{N}^c, k \in \mathcal{K}. \tag{1j}$$

Constraints (1a) specify that if courier k serves delivery unit i, the net inflow of the two commodities (i.e., the occupied capacity and the residual capacity) of courier k into delivery unit i equals to $2d_{ti}$. Constraints (1b) and (1c) define the outflow and inflow of the two commodities at the depot for courier k, respectively. The commodity outflow of courier k at the depot is the assigned delivery quantity, and the commodity inflow of courier k at the depot corresponds to the residual capacity. Constraints (1d) ensure that the outflow of the residual capacity of courier k at the depot copy $N+1$ equals the capacity. Constraints (1e) enforce that the summation of two commodities of courier k on each traversed arc is the courier capacity. Constraints (1f) state that, if delivery unit i is served by courier k, there must exist two traversed arcs incident to delivery unit i. Constraints (1g) ensure that each delivery unit must be assigned to only one courier. Constraints (1h) to (1j) define the decision variables.

Based on the decision x_t, the incentive and effort workload of courier k in period $t+1$ are computed as:

$$I_{t+1,k} = \sum_{i \in \mathcal{N}^c} \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}, \tag{2}$$

$$F_{t+1,k} = \sum_{(i,j) \in \overline{\mathcal{A}}} t_{ij} x_{tijk} + \sum_{i \in \mathcal{N}^c} s_i \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}. \tag{3}$$

A policy π maps a state to a decision. The set of possible policies is denoted by Π. We denote the decision function at decision epoch t for policy π and a given state S_t by $X_t^{\pi}(S_t)$, i.e., $x_t = X_t^{\pi}(S_t)$.

3.3. Exogenous information

At decision epoch t, the demand in period $t+1$ is unknown. We define the exogenous information $W_{t+1} = D_{t+1}$, where $D_{t+1} = (d_{t+1,i})_{\forall i \in \mathcal{N}^c}$ represents the demand of delivery units in period $t+1$ (as shown in Figure 2). We assume D_{t+1} can be described statistically, where the description is obtained from historical order data or from simulation model.

3.4. Transition function

Based on the state $S_t = (\tilde{\mathcal{I}}_t, \hat{D}_t)$, the decision $x_t = (Z_t, \Theta_t)$, and the exogenous information W_{t+1}, the state is transitioned as shown in Figure 2.

$$S_{t+1} = (\tilde{\mathcal{I}}_{t+1}, \hat{D}_{t+1}) = S^M(S_t, x_t, W_{t+1}), \tag{4}$$

where $\tilde{\mathcal{I}}_{t+1} = (\tilde{I}_{t+1,k})_{\forall k \in \mathcal{K}}$, $\hat{D}_{t+1} = (\hat{d}_{t+1,i})_{\forall i \in \mathcal{N}^c}$. Specifically,

$$\tilde{I}_{t+1,k} = \tilde{I}_{tk} + I_{t+1,k}.$$

3.5. Objective function

We measure the operating cost in period $t+1$ by the total travel time of the couriers, i.e., $f(\mathcal{F}_{t+1}) = \sum_{k \in \mathcal{K}} F_{t+1,k}$. Note that $\mathcal{F}_{t+1}$ is a function of S_t and x_t, which are known and decided at decision epoch t. As mentioned earlier, we assume that each courier is self-motivated to minimize his own delivery effort workload and chooses the fastest way to accomplish the assigned delivery task.

We use the maximum daily delivery time among the couriers *in period* $t+1$ to measure the daily effort imbalance, i.e., $\phi(\mathcal{F}_{t+1}) = \max_{k \in \mathcal{K}} F_{t+1,k}$. We use the range of the total incentive workload over the payroll cycle among the couriers to measure the incentive imbalance, i.e., $\varphi(\mathcal{I}_T) = \max_{k \in \mathcal{K}} \tilde{I}_{Tk} - \min_{k \in \mathcal{K}} \tilde{I}_{Tk}$. Note that $\varphi(\mathcal{I}_T)$ does not equal to the summation of daily incentive imbalance.

We choose the above measures of imbalance for theoretical and practical reasons. Theoretically, to measure effort imbalance we use the maximum, a weakly monotonic equity measure, rather than the range, a nonmonotonic equity measure. Using the range of delivery times to measure effort imbalance can result in non-TSP routes and workload inconsistency. Using the maximum of the delivery times to measure effort imbalance avoids such undesirable routes. We refer the readers to Matl et al. (2018) for more detailed discussions. Practically, couriers care most about differences in income (based on the delivery quantity), so using the range to measure incentive imbalance is most appropriate. As long as a courier does not spend too much time making deliveries on any given day, he/she may not care too much about how much time other couriers spend making deliveries (or may not even know how much time they spend), so using the maximum to ensure some effort balance is appropriate.

Given the state $S_t = (\tilde{\mathcal{I}}_t, \hat{D}_t)$ and decision $x_t = (Z_t, \Theta_t)$ at decision epoch t (as shown in Figure 2), the cost function $C_t(S_t, x_t)$ is defined as

$$C_t(S_t, x_t) = \begin{cases} f(\mathcal{F}_{t+1}) + \alpha\phi(\mathcal{F}_{t+1}), & t = 0, \ldots, T-2, \qquad (5a) \\ f(\mathcal{F}_{t+1}) + \alpha\phi(\mathcal{F}_{t+1}) + \beta\varphi(\tilde{\mathcal{I}}_{t+1}), & t = T-1, \qquad (5b) \end{cases}$$

where α and β are the unit penalties for daily effort imbalance and payroll cycle incentive imbalance, respectively, which represents the trade-off between the operating cost and the two types of workload imbalance. In this paper, we assume that α and β are set by the delivery manager.

The objective is to design a daily dispatching policy $\pi \in \Pi$ to minimize the expected total cost over the planning horizon. That is,

$$\min_{\pi \in \Pi} \quad \mathbb{E}\left\{ \sum_{t=0}^{T-1} C_t(S_t, X_t^{\pi}(S_t)) \middle| S_0 \right\}. \tag{6}$$

We summarize the notation in the MDP model in Table 2.

4. Dispatching policies

As described at the beginning of Section 3, the effort workload is balanced daily, but the incentive workload is balanced cumulatively over the planning horizon. However, the dispatching decisions have to be made on a daily basis. Therefore, the dispatching policy needs to incorporate a daily control mechanism to balance the total incentive workload in expectation.

In this section, we introduce the balanced penalty policy, which, at each decision epoch t, penalizes the cumulative incentive imbalance up to t with a proportion of the unit penalty for total incentive imbalance (β). Then, we describe four benchmark policies, namely, the VRP with rebalancing policy, the myopic with rebalancing policy, the fixed territory policy, and the fixed territory with rotation policy. In the VRP with rebalancing policy, we first solve an operating cost minimizing VRP, and then rebalance the cumulative incentive workload up to $t + 1$ when assigning routes to couriers. In the myopic with rebalancing policy, we first solve a myopic problem to minimize the single-period operating cost and effort imbalance (Equation (5a)), and then rebalance the cumulative incentive workload up to $t+1$ when assigning routes to couriers. In the fixed territory policy, each courier only delivers packages in the assigned territory, where the territories partition the service region based on the expected demand. The fixed territory with rotation policy is the same as the fixed territory policy, except that the couriers rotate territories every payroll cycle.

4.1. Balanced penalty policy

We apply the Cost Function Approximation (CFA) approach of Approximate Dynamic Programming (ADP). In CFA, we solve a modified optimization problem at each decision epoch, where the objective function and/or the constraints are modified parametrically (Powell, 2019). CFA can be attractive when the impact of uncertainty is easy to recognize and the decision is multidimensional (Powell, 2014). In the problem under study, the deliveries are made based on the realized demand and thus the effort workload can be balanced deterministically in each period. Therefore, the demand uncertainty only affects the cumulative

Table 2: Notation in the MDP model

Sets	
$\mathcal{T}$	$\{1, 2, \ldots, T\}$, set of periods
$\mathcal{K}$	$\{1, 2, ..., K\}$, homogeneous fleet of couriers
$\mathcal{G}$	$(\mathcal{N}, \mathcal{A})$, delivery network
0	Depot
$\mathcal{N}^c$	$\{1, 2, ..., N\}$, set of delivery units
$\mathcal{A}$	Set of arcs
Data and parameters	
t_{ij}	Travel time on arc $(i, j) \in \mathcal{A}$
Q	Courier capacity
s_i	Service time of each customer in delivery unit i
State variable at decision epoch t	
S_t	State variable, $S_t = (\tilde{\mathcal{I}}_t, \hat{D}_t)$
I_{tk}	Daily incentive workload of courier k, $\forall k \in \mathcal{K}$
F_{tk}	Daily effort workload of courier k, $\forall k \in \mathcal{K}$
$\mathcal{I}_t$	Daily incentive workload, $\mathcal{I}_t = (I_{tk})_{\forall k \in \mathcal{K}}$
$\mathcal{F}_t$	Daily effort workload, $\mathcal{F}_t = (F_{tk})_{\forall k \in \mathcal{K}}$
$\tilde{I}_{tk}$	Cumulative incentive workload of courier k up to decision epoch t, $\forall k \in \mathcal{K}$
$\tilde{\mathcal{I}}_t$	Cumulative incentive workload up to decision epoch t, $\tilde{\mathcal{I}}_t = (\tilde{I}_{tk})_{\forall k \in \mathcal{K}}$
$\hat{d}_{ti}$	Demand of delivery unit i, $\forall i \in \mathcal{N}^c$
$\hat{D}_t$	Demand of delivery units, $D_t = (\hat{d}_{ti})_{\forall i \in \mathcal{N}^c}$
Decisions at decision epoch t	
x_t	Decision variable, $x_t = (Z_t, \Theta_t)$
z_{tik}	Equals to 1 if courier k is dispatched to serve delivery unit i; otherwise, equals to 0.
Z_t	Dispatching decision, $Z_t = (z_{tik})_{\forall i \in \mathcal{N}^c, k \in \mathcal{K}}$
θ_{tk}	Route of courier k to serve the assigned delivery units
Θ_t	Routing decision, $\Theta_t = (\theta_{tk})_{\forall k \in \mathcal{K}}$
ξ_{tijk}	Equals to 1 if courier k traverse the arc (i, j); otherwise, equals to 0.
y_{tijk}	Occupied/Residual capacity, if courier k traverse the arc $(i, j)/(j, i)$.
$\mathcal{X}_t$	Feasible region of x_t
π	Policy
Π	Set of possible policies
$X_t^\pi(S_t)$	Decision function for policy π and a given state S_t
Exogenous information between decision epoch t and decision $t+1$	
W_{t+1}	Exogenous information, $W_{t+1} = D_{t+1}$
$d_{t+1,i}$	Random variable representing demand of delivery unit i, $\forall i \in \mathcal{N}^c$
D_{t+1}	Random variable representing demand of delivery units, $D_{t+1} = (d_{t+1,i})_{\forall i \in \mathcal{N}^c}$
Objective function	
$C_t(S_t, x_t)$	Cost function given the state S_t and decision x_t at decision epoch t
$f(\mathcal{F}_{t+1})$	Operating cost function
$\phi(\mathcal{F}_{t+1})$	Daily effort imbalance function
$\varphi(\tilde{\mathcal{I}}_{t+1})$	Cumulative incentive imbalance function
α	Unit penalty for daily effort imbalance
β	Unit penalty for total incentive imbalance over the horizon

incentive balance of future periods (until the end of the planning horizon). The dispatching and routing decisions are multidimensional.

In the design of the CFA, a natural idea is to penalize the cumulative incentive imbalance at each decision epoch instead of penalizing the total cumulative incentive imbalance only at the end of the planning horizon. We denote the BALANCED PENALTY policy as π^B. The modified cost functions are

$$\begin{aligned} C_t(S_t, X_t^{\pi_p^P}(S_t)) &= f_t(\mathcal{F}_{t+1}) + \alpha\phi_t(\mathcal{F}_{t+1}) + p\beta\varphi_t(\tilde{\mathcal{I}}_{t+1}) \\ &= \sum_{k\in\mathcal{K}} F_{t+1,k} + \alpha \max_{k\in\mathcal{K}} F_{t+1,k} + p\beta(\max_{k\in\mathcal{K}} \tilde{I}_{t+1,k} - \min_{k\in\mathcal{K}} \tilde{I}_{t+1,k}), \forall t = 0, 1, \ldots, T-2, \\ C_t(S_t, X_t^{\pi_p^P}(S_t)) &= f_t(\mathcal{F}_{t+1}) + \alpha\phi_t(\mathcal{F}_{t+1}) + \beta\varphi_t(\tilde{\mathcal{I}}_{t+1}) \\ &= \sum_{k\in\mathcal{K}} F_{t+1,k} + \alpha \max_{k\in\mathcal{K}} F_{t+1,k} + \beta(\max_{k\in\mathcal{K}} \tilde{I}_{t+1,k} - \min_{k\in\mathcal{K}} \tilde{I}_{t+1,k}), t = T-1, \end{aligned}$$

where $p \in \mathcal{P} = (0, 1]$. The cost function includes the single-period operating cost, single-period effort imbalance penalty, and cumulative incentive imbalance penalty based on $p\beta$. We further denote the BALANCED PENALTY policy π^B using parameter p as π_p^P.

The decision at decision epoch t under π_p^P is the solution of the mixed integer programming (8).

$$\begin{array}{lll} \min & C_t(S_t, X_t^{\pi_p^P}(S_t)) & \quad (8a) \\ \text{s.t.} & \text{Constraints (1a) to (1j)} & \\ & I_{t+1,k} = \sum_{i\in\mathcal{N}^c} \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}, & \quad (8b) \\ & F_{t+1,k} = \sum_{(i,j)\in\overline{\mathcal{A}}} t_{ij} x_{tijk} + \sum_{i\in\mathcal{N}^c} s_i \hat{d}_{ti} z_{tik}, \qquad \forall k \in \mathcal{K}, & \quad (8c) \\ & \tilde{I}_{t+1,k} = \tilde{I}_{tk} + I_{t+1,k}, \qquad \forall k \in \mathcal{K}, & \quad (8d) \\ & I_{t+1,k} \geq 0, \qquad \forall k \in \mathcal{K}, & \quad (8e) \\ & F_{t+1,k} \geq 0, \qquad \forall k \in \mathcal{K}, & \quad (8f) \\ & \tilde{I}_{t+1,k} \geq 0, \qquad \forall k \in \mathcal{K}. & \quad (8g) \end{array}$$

The objective function (8a) is the modified cost function. Constraints (1a) to (1j) define the feasible region of dispatching and routing decisions, as in Section 3.2. Constraints (8b) specify that the incentive workload $I_{t+1,k}$ of courier k in period $t+1$ is the total demand of the delivery units assigned to him/her. Constraints (8c) specify that the effort workload $F_{t+1,k}$ of courier k in period $t+1$ is composed of the travel time and the service time. Constraints (8d) update the cumulative incentive workload for each courier $k \in \mathcal{K}$. Constraints (8e) to (8g) enforce that the decision variables be nonnegative.

4.2. *A hybrid algorithm to search p in π^B*

To search for the optimal parameter p^* in π^B, we propose a hybrid simulation optimization algorithm (Figure 3). Specifically, we use a modified nested partitions (NP) method (Shi and

ÓLafsson, 2000; Chen et al., 2019) as the global search framework to obtain a candidate pool of p, and then employ a ranking and selection (R&S) KN++ procedure (Kim and Nelson, 2006) to determine p^* in the pool.

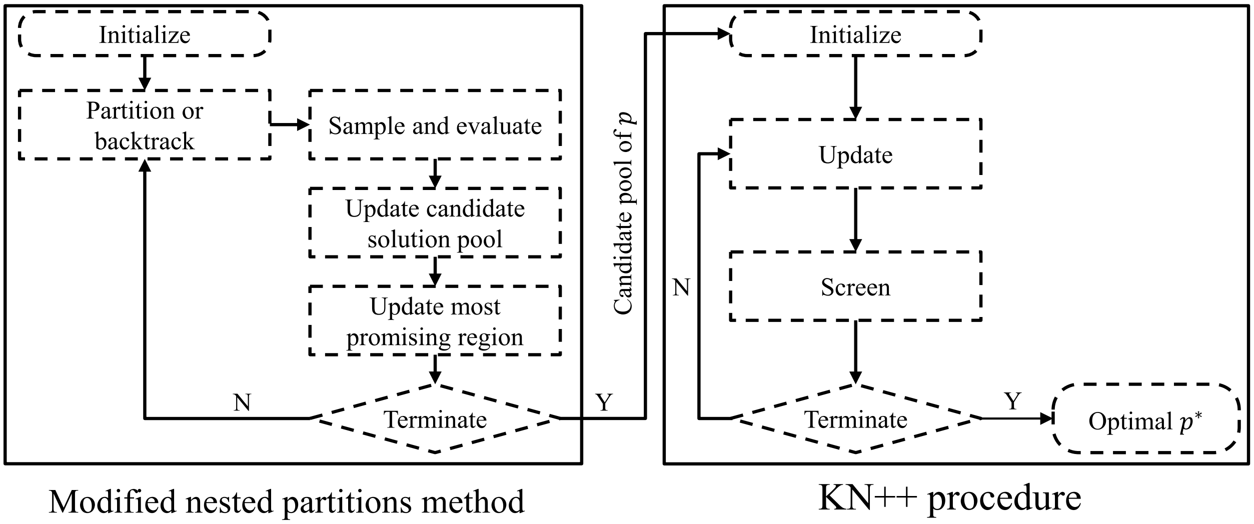


Figure 3: The hybrid algorithm

Normally, the nested partitions method outputs the singleton solution that is the most frequently visited in the last iteration as the best solution (Shi and ÓLafsson, 2000). In each iteration, we need to sample multiple points in the search region of p. To evaluate each sampled point p, we use H demand (stream) observations, which requires to solve $H \times T$ MIPs. If the maximum number of iterations $K^{max} = 50$, the average number of sampled points in each iteration is 10, the number of demand observations for each sampled point $H = 5$, the planning horizon $T = 7$, and the average solution time of a MIP is 10 minutes, we need 175,000 minutes (about 120 days) to run the algorithm once! Even with the speed-up of parallel computing, it still takes about one week. This heavy computational effort limits the number of demand observations H to evaluate each sampled point of p. As a result, the conventional nested partitioned method may be subject to high Type I and Type II errors. Hence, we employ a modified nested partitions method (Algorithm 1 in Appendix Appendix A.1, Chen et al., 2019) to generate a candidate pool of p, instead of only selecting the best singleton solution.

We use a KN++ procedure (Algorithm 2 in Appendix Appendix A.2, Kim and Nelson, 2006) as the statistical "cleanup" procedure to reduce Type I and Type II errors. The KN++ procedure is a fully sequential indifference-zone procedure that (asymptotically) selects the best alternative from a finite set of alternatives with a pre-specified probability of correct selection (PCS). In each iteration, the KN++ procedure screens the evaluation results of p's and eliminates the inferior p's, until one p remains, the optimal p^*.

4.3. Benchmark policies

In this section, we describe four benchmark policies, namely, the VRP WITH REBALANCING policy, the MYOPIC WITH REBALANCING policy, the FIXED TERRITORY policy, and the FIXED TERRITORY WITH ROTATION policy.

4.3.1. VRP with rebalancing policy

At each decision epoch t, we solve an operating cost minimizing VRP (9).

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{K}} F_{t+1,k} && && \text{(9a)} \\ & \text{Constraints (1a) to (1j)} \\ & F_{t+1,k} = \sum_{(i,j) \in \overline{E}} t_{ij} x_{tijk} + \sum_{i \in V^c} s_i \hat{d}_{ti} z_{tik}, && \forall k \in \mathcal{K}, && \text{(9b)} \\ & F_{t+1,k} \geq 0, && \forall k \in \mathcal{K}. && \text{(9c)} \end{aligned}$$

The objective function (9a) is to minimize the daily operating cost. Constraints (1a) to (1j) define the feasible region of dispatching and routing decisions, as in Section 3.2. Constraints (9b) specify that the effort workload $F_{t+1,k}$ of courier k is composed of the travel time and the service time. Constraints (9c) enforce that the decision variables be nonnegative.

However, the solution of (9) only determines the routes without assigning them to couriers. We show that if we assign routes to couriers according to Observation 1 (the proof is given in Appendix Appendix B), we can achieve the minimum range of the cumulative incentive workload $(\tilde{I}_{t+1,k})_{\forall k \in \mathcal{K}}$ from period 1 to period $t+1$. We denote this VRP WITH REBALANCING policy as π^V.

Observation 1. *The minimum range of the cumulative incentive workload $(\tilde{I}_{t+1,k})_{\forall k \in \mathcal{K}}$ among the couriers is obtained when 1) sorting the couriers in ascending order based on the cumulative incentive workload $(\tilde{I}_{tk})_{\forall k \in \mathcal{K}}$, 2) sorting the predetermined routes in descending order based on their numbers of delivery packages, and 3) matching the couriers and predetermined routes accordingly.*

4.3.2. Myopic with rebalancing policy

At decision epoch t, we solve a myopic problem (10) to minimize the single-period operating cost and effort imbalance penalty.

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{K}} F_{t+1,k} + \alpha \max_{k \in \mathcal{K}} F_{t+1,k} && && \text{(10a)} \\ & \text{Constraints (1a) to (1j)} \\ & F_{t+1,k} = \sum_{(i,j) \in \overline{E}} t_{ij} x_{tijk} + \sum_{i \in V^c} s_i \hat{d}_{ti} z_{tik}, && \forall k \in \mathcal{K}, && \text{(10b)} \\ & F_{t+1,k} \geq 0, && \forall k \in \mathcal{K}. && \text{(10c)} \end{aligned}$$

The objective function (10a) consists of the single-period operating cost and effort workload imbalance penalty. Constraints (1a) to (1j) define the feasible region of dispatching and routing decisions, as in Section 3.2. Constraints (10b) specify that the effort workload $F_{t+1,k}$ of courier k is composed of the travel time and the service time. Constraints (10c) enforce that the decision variables be nonnegative.

However, similar to the VRP WITH REBALANCING policy π^V, the solution of the myopic problem (10) only determines the routes without assigning them to couriers. We use the same rebalance method in Observation 1 to assign routes to couriers. We denote this MYOPIC WITH REBALANCING policy as π^M.

4.3.3. Fixed territory policy

In the practice of last-mile delivery, a delivery manager may also partition the service region of the delivery station into multiple territories and assign each courier to serve a fixed delivery territory over the planning horizon.

In order to obtain a reasonably balanced partition, similar to Huang et al. (2018), we solve a single-period workload balancing problem (11), based on the expected demand $\bar{d}_i$ of each delivery unit $i \in \mathcal{N}^c$.

$$\min \quad \sum_{k \in \mathcal{K}} F_k + \alpha \max_{k \in \mathcal{K}} F_k + \beta(\max_{k \in \mathcal{K}} I_k - \min_{k \in \mathcal{K}} I_k) \tag{11a}$$

Constraints (1a) to (1j) without the time index t

$$I_k = \sum_{i \in V^c} \bar{d}_i z_{ik}, \qquad \forall k \in \mathcal{K}, \tag{11b}$$

$$F_k = \sum_{(i,j) \in \overline{E}} t_{ij} x_{ijk} + \sum_{i \in V^c} s_i \bar{d}_i z_{ik}, \qquad \forall k \in \mathcal{K}, \tag{11c}$$

$$I_k \geq 0, \qquad \forall k \in \mathcal{K}, \tag{11d}$$

$$F_k \geq 0, \qquad \forall k \in \mathcal{K}. \tag{11e}$$

The objective function (11a) consists of the single-period operating cost, effort imbalance penalty, and incentive imbalance penalty, using the same α and β as in Equation (5). Constraints (1a) to (1j) define the feasible region of dispatching and routing decisions, as in Section 3.2. Constraints (11b) specify that the incentive workload I_k of courier k is the total demand of the delivery units assigned to him/her. Constraints (11c) specify that the effort workload F_k of courier k is composed of the travel time and the service time. Constraints (11d) to (11e) enforce that the decision variables be nonnegative.

Based on the territory partition from the optimal solution of (11), each courier chooses the fastest way to deliver the realized demand in the assigned delivery territory on a daily basis. We denote the FIXED TERRITORY policy as π^F.

4.3.4. Fixed territory with rotation policy

Under the FIXED TERRITORY policy, a slight difference in the expected demand between the delivery territories may result in the accumulation of incentive imbalance over the planning horizon. Intuitively, the couriers can rotate the assigned territories (e.g., every payroll cycle) to reduce the incentive imbalance. Specifically, we denote the partitioned territories as $\mathcal{R} = \{R_1, R_2, \ldots, R_K\}$, and dispatch each courier $k \in \mathcal{K}$ to serve territory $R_{(k+i-1) \bmod K}$ in the ith payroll cycle. We denote this FIXED TERRITORY WITH ROTATION policy as π^{FR}.

5. Numerical experiments

In this section, we describe the experimental setup in Section 5.1, the performance of the hybrid algorithm in Section 5.2, and the numerical study on the policy comparison on the base case instances in Section 5.3, the impact of demand variation in Section 5.4, and the impact of the unit penalties of workload imbalance (α and β) in Section 5.5.

5.1. Experimental setup

Similar to Huang et al. (2018), in order to experiment on a delivery network that respects the geographical and demographical characteristics in an urban area, while not revealing the real design of the logistics company, we use the publicly available design of regions and districts by the Beijing Bureau of Urban Planning (Figure 4(a)).

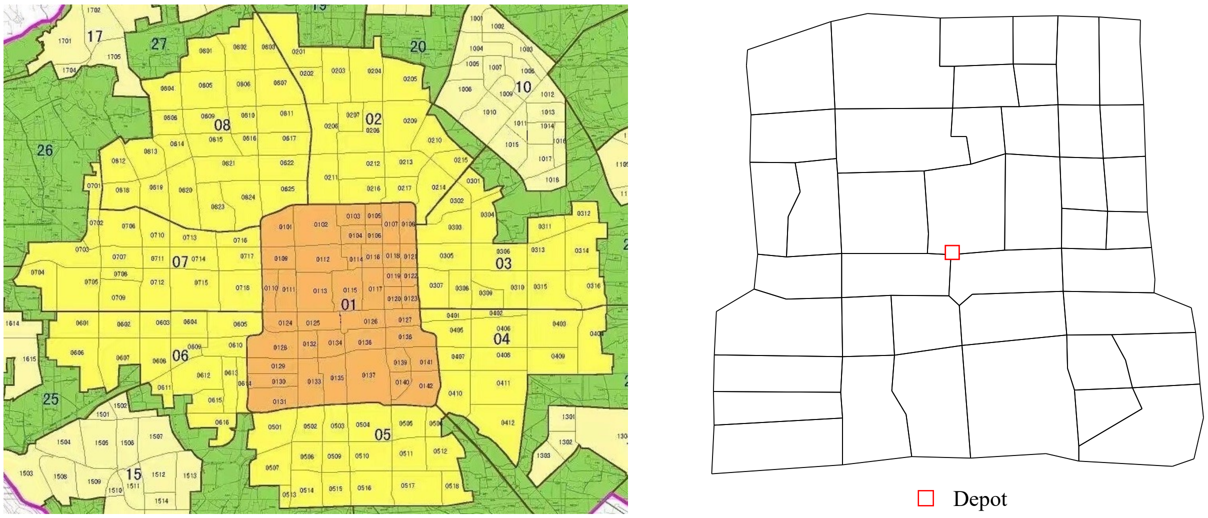
Specifically, we use "shrinked" region 01 to represent the service region of a last-mile station[1]. It has 42 districts, each representing a delivery unit. Each delivery unit can be approximated by a simple polygon, as in Figure 4(b). If two delivery units share a common boundary segment but not just a boundary point, we define that the two delivery units are adjacent.

We construct a delivery network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. $\mathcal{N} = \{0\} \cup \mathcal{N}^c$, where the depot 0 is located at the centroid of region 01, and $\mathcal{N}^c$ is the set of the centroids of the delivery units. $\mathcal{A}$ consists of the arcs connecting the adjacent delivery units and connecting the depot and the delivery units.

We set the planning horizon $T = 7$ days. For each delivery unit $i \in \mathcal{N}^c$, we randomly generate $\hat{d}_{0i}$ customers at the beginning of the planning horizon and $\hat{d}_{ti}$ customers in period $t, 1 \leq t \leq T-1$. We assume that $\hat{d}_{0i}$ and $\hat{d}_{ti}$ $(1 \leq t \leq T-1)$ independently follow the normal distribution $N(\mu, (\rho\mu)^2)$, where μ is the expected number of customers in a delivery unit. In the base case settings[2], we set $\mu = 12$ and the coefficient of variation $\rho = 0.2$. We will experiment on different values of ρ in Section 5.4. To simplify the experiment, we

[1] Region 01 has a size of 62.52 square kilometer and is too large to serve as the service region of a last-mile delivery station. In 2019, JD logistics (a leading logistics company in China) has 797 last-mile stations in Beijing (`https://www.aikuaidi.cn/outlets/z9o7f88v_40.html`, in Chinese, last accessed on 20 December 2019), and the area of Beijing is 16410 square kilometers. That is, the average size of the service region by a last-mile delivery station is approximately 20.6 square kilometers. Therefore, we balancedly "shrink" the area of region 01 from 62.52 to 21 square kilometers. As a result, the average area of a delivery unit is 0.5 square kilometer.

[2] In 2019, the average number of packages per square kilometer in Beijing is 368.8 (National Bureau of Statistics, `http://data.stats.gov.cn/easyquery.htm?cn=E0103`, in Chinese, last accessed on 20 July 2020), the share of e-commerce packages among overall express packages in China is 80% (People's Daily, 2019), and the market share of JD.com in China is 25.7% (Analysis, 2019). Assuming 3 delivery shifts per day, μ is approximated as $368.8 \times 80\% \times 25.7\%/3 \approx 12$.



(a) Design of regions and districts by the Beijing Bureau of Urban Planning

(b) Region 01 approximated by simple polygons

Figure 4: Network structure

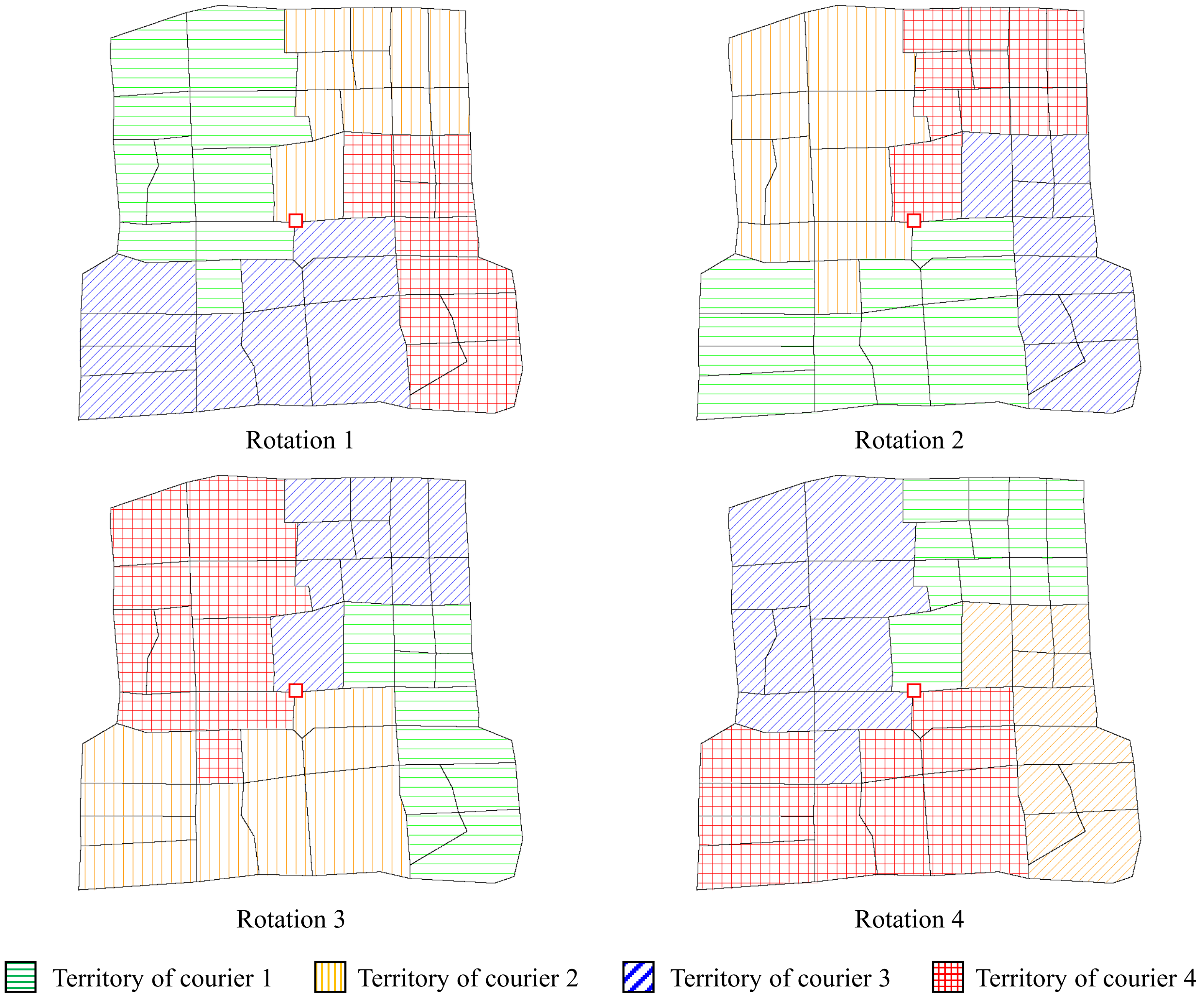
approximate the number of customers by the number of packages. We set the service time of each customer in delivery unit $i \in \mathcal{N}^c$ as $s_i = 3$ minutes, and approximate the travel time within a delivery unit using continuous approximation (Daganzo, 1984).

We set the number of couriers $K = 4$ such that a courier delivers 120 packages per day on average, as in practice. Each courier has a capacity $Q = 200$ and travel speed $v = 15$ km/h. We set the unit penalty for daily effort imbalance as $\alpha = 1$ and the unit penalty for total incentive imbalance as $\beta = 1$. In Section 5.5, we explore different values of α and β and evaluate their impact on performance.

Policies π^B, π^V, π^M, and π^F can be evaluated over a 7-day planning horizon (i.e., a week). However, in policy π^{FR}, couriers rotate every week. In order to evaluate the effect of a complete rotation of 4 couriers, we need at least four weeks. Therefore, we evaluate each policy over four weeks with 25 replications.

The territory partition and rotation in π^{FR} are shown in Figure 5. In π^F, there is no rotation and each courier is assigned to the same territory as in rotation 1 over the entire four weeks. However, the courier assignments under π^B, π^V, and π^M are determined only within each week. To determine the courier assignments of these three policies over four weeks, we sort the cumulative incentive workload of the couriers within each week in ascending order and assign couriers in the same order in each week. That is, the courier with the highest cumulative incentive workload in one week is assigned the highest cumulative incentive workload in every week. Note that this courier assignment favors π^F and π^{FR}.

We program all the policies using C#. The MIP at each decision epoch in each policy is solved by Gurobi 9.0.0 (MIPGap = 0.1%). We run the experiments on three Windows servers with 2.00 GHz, 2.27 GHz, and 2.70 GHz processor, respectively, and identical 64 GB memory.



Rotation 1

Rotation 2

Rotation 3

Rotation 4

Figure 5: Territory partition and rotation settings

5.2. Performance of the hybrid algorithm

We apply the hybrid algorithm in Section 4.2 to search for the optimal p^* in the balanced penalty policy π^B. We describe the implementation details of the hybrid algorithm in Appendix Appendix A.3.

We denote $\bar{V}(p, \mathcal{S})$ as the average total cost over sample $\mathcal{S}$. In Figure 6, we show $\bar{V}(p, \mathcal{S})$ of various values of p under sample size 1, 2, …, 120. We observe that $\bar{V}(p, \mathcal{S})$ stabilizes after the sample size reaches 90. In the following experiments, we use the same sample $\mathcal{S}^{EVAL}$ of size 100 to evaluate $\bar{V}(p, \mathcal{S}^{EVAL})$.

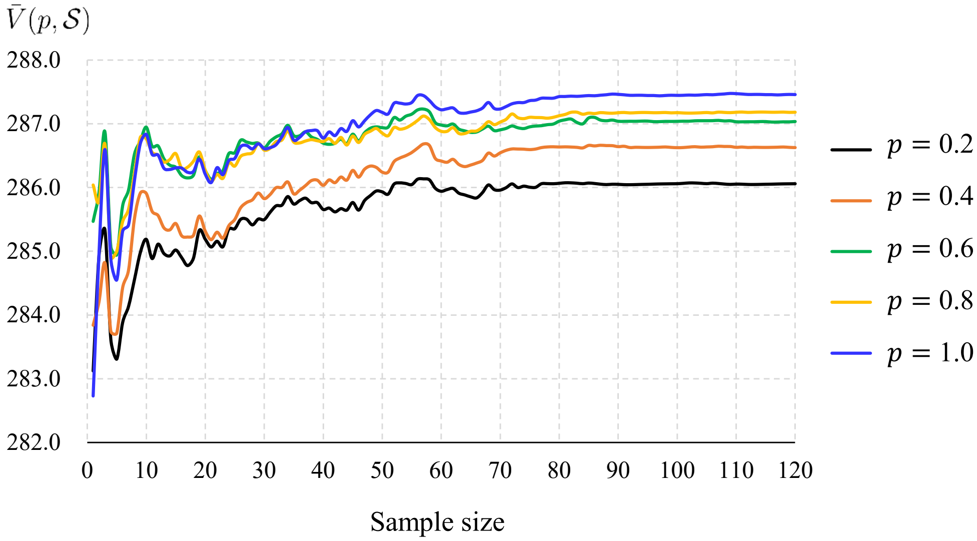


Figure 6: The average total cost under different sample sizes

For the base case instances described in Section 5.1, we replicate the hybrid algorithm 10 times. After obtaining the p_i^* of replication i, we evaluate $\bar{V}(p_i^*, \mathcal{S}^{EVAL})$. In Table 3, columns (1) and (2) show the best value of $\bar{V}(p^*, \mathcal{S}^{EVAL})$ among the 10 replications and the average value of $\bar{V}(p^*, \mathcal{S}^{EVAL})$ over the 10 replications, respectively. The best-average ratio of 99.96% illustrates the stability of the hybrid algorithm. In the real implementation, we run the hybrid algorithm only once to search for the optimal p^*. For the base case settings described in Section 5.1, we select the output of the hybrid algorithm in the first replication, which is $p_1^* = 0.20$.

Table 3: Results of the hybrid algorithm (10 replications)

Best $\bar{V}(p^*, \mathcal{S}^{EVAL})$	Average $\bar{V}(p^*, \mathcal{S}^{EVAL})$	Best-average ratio	Average computational time
(1)	(2)	(3) = (1)/(2)	(4)
286.05	286.18	99.96%	≈ 11 days

The computational time of the hybrid algorithm is about 11 days, averaged over 10 replications. Note that if we enumerate the entire discretized search region $\mathcal{P}$ to search for

$p^* = \arg\min_{p \in \mathcal{P}} \bar{V}(p, \mathcal{S}^{EVAL})$, it takes about 490 days[3]. The hybrid algorithm can significantly reduce the computational time. Once p^* is determined, it takes only about 10 minutes to solve the MIP (8) in the daily dispatching.

5.3. Comparative analysis on the base case instances

Next, for the base case instances described in Section 5.1, we compare the balanced penalty policy π^B with the benchmark policies (see Section 4.3), i.e., the VRP with rebalancing policy π^V, the myopic with rebalancing policy π^M, the fixed territory policy π^F, and the fixed territory with rotation policy π^{FR}.

We compare the policies using the average total cost over 25 replications in Table 4. The row "Difference w.r.t. π^B" represents the difference in the average total cost between a policy and π^B (calculated as $(TC^{\text{Policy}} - TC^{\pi^P})/TC^{\pi^P}$, where TC represents the average total cost). We observe that π^B outperforms the other policies, and that π^F and π^V perform much worse than the other policies. π^{FR} can significantly decrease the total cost compared with π^F, though it still performs worse than π^M and π^B.

Table 4: Total cost and differences with respect to π^B

	π^B	π^M	π^{FR}	π^F	π^V
Average	1144.2	1171.5	1221.1	1547.6	1689.3
Standard deviation	4.7	4.8	9.4	12.0	17.4
Difference w.r.t. π^B	-	2.4%	6.8%	35.3%	47.6%

Figure 7 shows the decomposed cost. Each bar consists of the operating cost, effort imbalance penalty, and incentive imbalance penalty (from bottom to top).

In π^V, we solve an operating cost minimizing VRP at each decision epoch, and thus π^V achieves the smallest operating cost among the policies. Compared with π^V, the other policies do not significantly increase the operating cost (the difference between the largest and the smallest operating cost is 1.01%). This result indicates that we can achieve workload balance without compromising too much operating cost, which is consistent with the observation in Matl et al. (2018).

We also observe that π^M achieves the best effort balance among the policies. This is intuitive because only π^B and π^M penalize daily effort imbalance, while π^B additionally penalizes cumulative incentive imbalance (see (8a) and (10a)). Note that π^V results in much higher effort imbalance penalty than the other policies.

Further, we observe that π^B achieves the best incentive balance among the policies. This is intuitive in that π^B is the only policy that explicitly considers the cumulative incentive balance in the daily dispatching.

[3] There are 100 points (p) in the entire discretized search region $\mathcal{P}$. For each $p \in \mathcal{P}$, we evaluate $\bar{V}(p, \mathcal{S}^{EVAL})$ using sample $\mathcal{S}^{EVAL}$ of size 100. In each scenario $\omega \in \mathcal{S}^{EVAL}$, we need to solve 7 MIPs (the planning horizon is 7 days). It takes about 10 minutes to solve a MIP. Hence, the total computational time is $100 \times 100 \times 7 \times 10 = 700,000$ minutes ≈ 490 days.

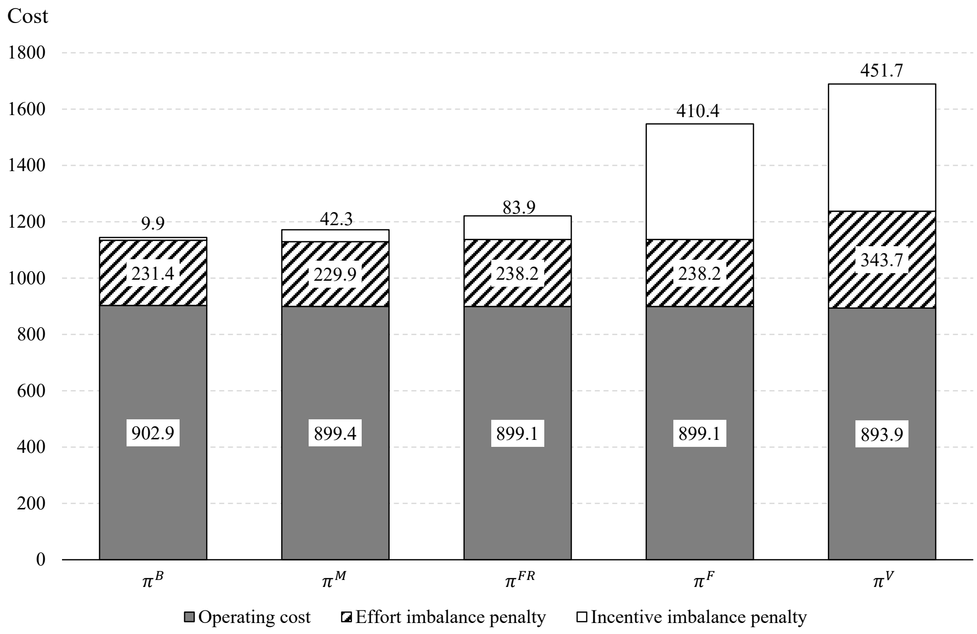


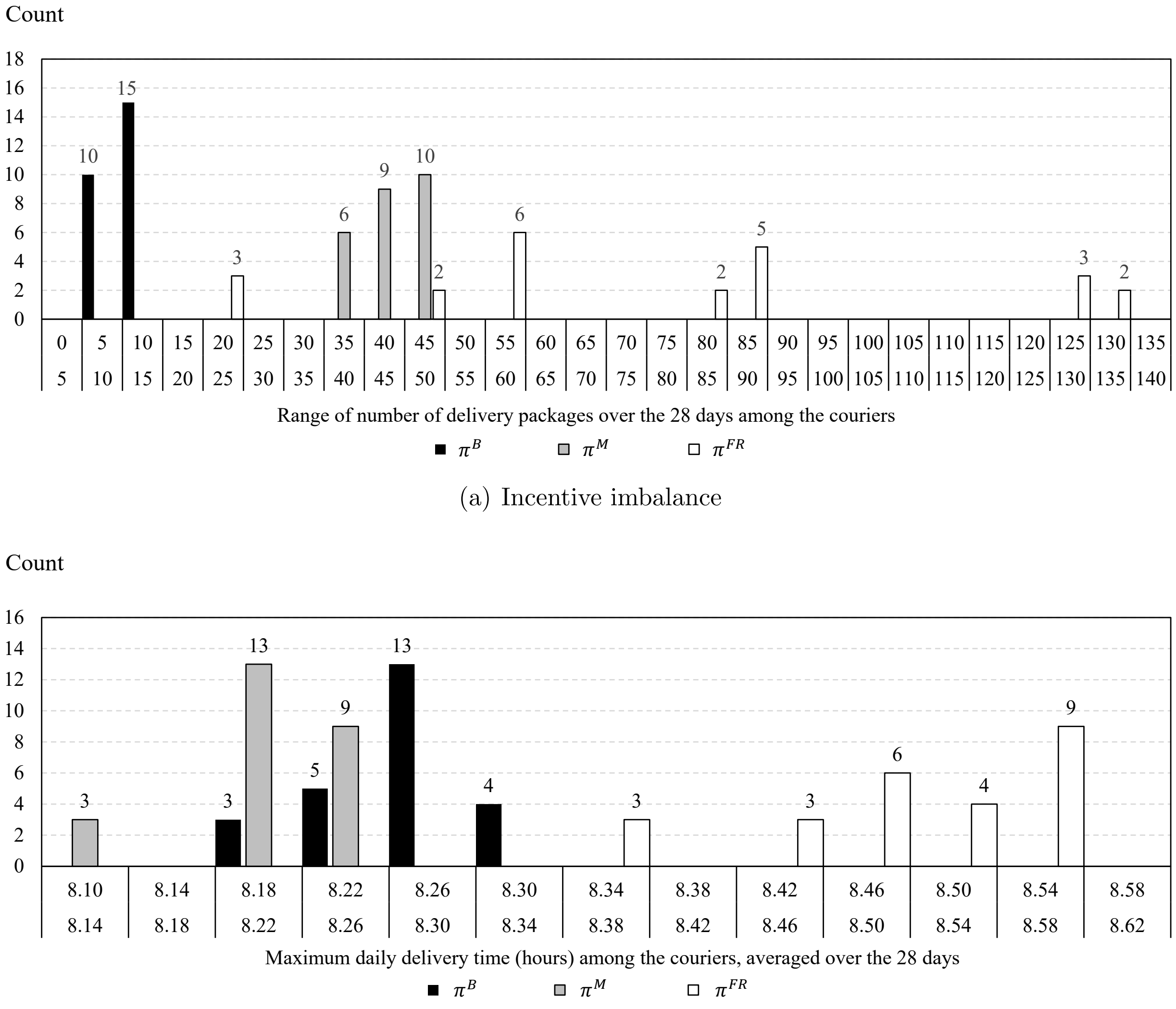
Figure 7: Cost decomposition

Compared with π^F, π^{FR} significantly reduces the total incentive imbalance over the four weeks among the couriers, from 410.4 packages to 83.9 packages (note that $\beta = 1$). Under π^F, courier 1 and courier 3 deliver 120 packages on each day in expectation, while courier 2 and courier 4 deliver 132 packages on each day in expectation. This 10% daily difference accumulates (in expectation) over the four weeks, resulting in more significant incentive imbalance among the couriers. With territory rotation, π^{FR} can effectively mitigate the cumulation of incentive imbalance. Note that, however, π^F and π^{FR} result in exactly the same operating cost and effort imbalance penalty, because the territory rotation only changes the assignment of (homogeneous) couriers to the routes but not the routes themselves and the corresponding (total and maximum) delivery time.

Under π^V, the routes on each day are heavily imbalanced. For example, on day 1 of replication 1, the maximum and minimum numbers of packages are 187 and 14 packages, and the maximum and minimum delivery time are 12.35 and 0.98 hours. Although π^V tries to rebalance the cumulative incentive workload on the daily basis (Section 4.3.1), it still results in high incentive imbalance penalty (similar to π^F). Without effort rebalance, π^V suffers from significantly higher effort imbalance penalty compared with all the other policies.

Except π^V, all the other policies mainly differ in the incentive imbalance penalty. Compared with π^M, π^B results in slightly larger operating cost and effort imbalance penalty, but much smaller incentive imbalance penalty. As a result, π^B outperforms π^M on the total cost. In the following numerical study, we focus on the comparison among π^B, π^M, and π^{FR}, because of their significant outperformance over π^F and π^V.

Next, we look at the distribution of the incentive imbalance and the average effort imbalance over the 28 days over the 25 replications. Figure 8(a) shows the histogram of incentive imbalance, i.e., the difference between maximum and minimum numbers of delivery packages over the 28 days among the couriers. Under π^B, the incentive imbalance is smaller than 15 packages in all the 25 replications. π^{FR} shows much larger variance, with the largest total incentive imbalance reaching 130 packages. Figure 8(b) shows the histogram of average effort imbalance, i.e., the maximum daily delivery hours among the couriers, averaged over the 28 days. π^M performs slightly better than π^B. π^{FR} distributes towards higher values with larger variance. In the current 25 replications, the smallest average daily effort imbalance under π^{FR} is larger than the largest average daily effort imbalance under π^B and π^M.



(a) Incentive imbalance

(b) Effort imbalance

Figure 8: Distributions of the total incentive imbalance and average daily effort imbalance over the 28 days

5.4. Impact of demand variation

As described in Section 5.1, in the base case settings, $\hat{d}_{0i}$ and $\hat{d}_{ti}$ $(1 \le t \le T-1)$ independently follow the normal distribution $N(\mu, (\rho\mu)^2)$, where we set $\mu = 12$ and $\rho = 0.2$. In Table 5, we show different values of $\rho = 0.1, 0.2, 0.3$, and present the corresponding p^*'s in π^B. We observe that p^* decreases in ρ. Intuitively, a larger p value means a higher penalty on the cumulative incentive imbalance and therefore, likely, a more balanced cumulative incentive workload among couriers on each day of the week. However, more balanced incentive workload among the couriers in early days may leave limited room to offset the (unavoidable) difference in package allocation among routes in the remaining days of the week, especially when the demand variation (ρ) is high. As an extreme example, assuming all couriers are perfectly balanced in the first six days of the week, the total incentive imbalance over the week among all the couriers would be the same as the incentive imbalance on the last day.

Table 5: Demand variation settings and the corresponding p^*'s

Number of couriers	Number of customers in a delivery unit on each day		p^*
	Expected value μ	Coefficient of variation ρ	
4	12	0.1	0.31
		0.2	0.20
		0.3	0.11

Figure 9 shows the total costs and the decomposition under different values of ρ. All the values are averaged over 25 replications. Under each policy, the gaps between the average total costs under $\rho = 0.1$ and $\rho = 0.2, 0.3$ are shown on the top of the bars.

We observe that the operating costs under all the policies and the effort imbalance penalties under π^B and π^M are not very sensitive to the change in demand variation. The effort imbalance penalty under π^{FR} increases in ρ. Recall that under π^{FR}, the territories are fixed and the rotation does not affect the maximum daily delivery time among the couriers. A higher demand variation (i.e., a larger value of ρ) may result in larger maximum daily demand and likely larger maximum daily delivery time among the territories. While the incentive imbalance penalties under all the policies increase in ρ, the increase under π^B are the least. As π^B explicitly penalizes the cumulative incentive imbalance in the daily dispatching, it is more adaptive to the change in demand variation than the other policies. As a result, the total cost under π^B increases slightly in ρ, while those under π^M and π^{FR} increase significantly.

5.5. Impact of the unit penalties of workload imbalance

Recall that the unit penalties of the effort imbalance (α) and incentive imbalance (β) are assumed to be set by the delivery manager. Here, we explore the impact of the choice of α and β on the operating cost, effort imbalance, and incentive imbalance.

Figure 10 shows the average daily effort imbalance and the total incentive imbalance over the 28 days under π^B under different values of α (Figure 10(a)) or β (Figure 10(b)), while

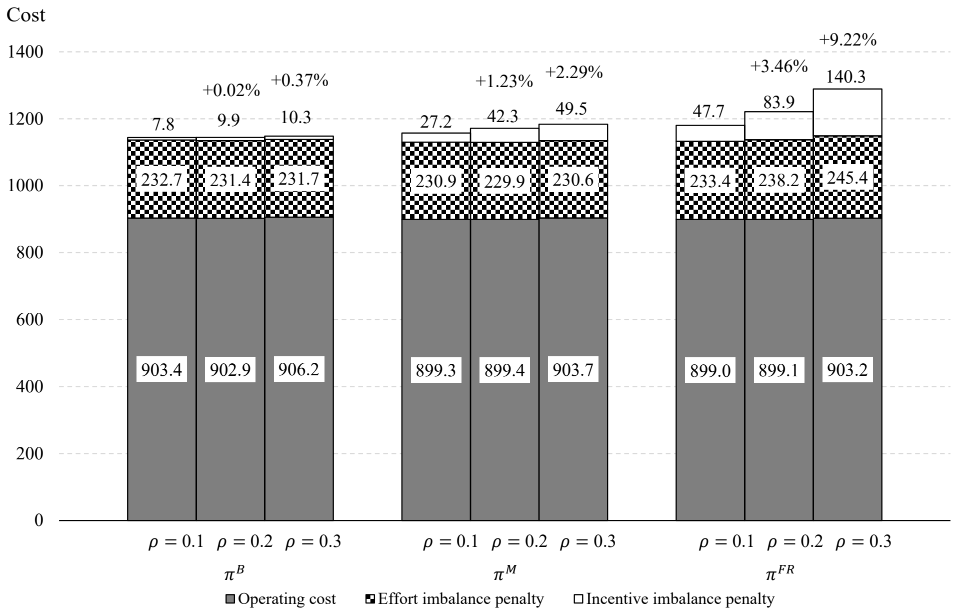


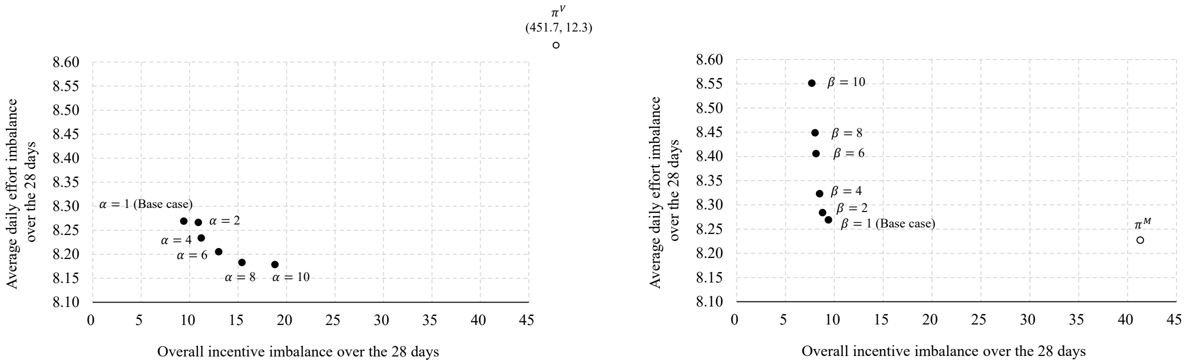
Figure 9: Decomposed cost under different demand variations

fixing the other one at the default value (i.e., $\beta = 1$, or $\alpha = 1$, respectively). We also show the results of π^V (corresponding to $\alpha = \beta = 0$ under π^B) in Figure 10(a) and the results of π^M (corresponding to $\alpha = 1$ and $\beta = 0$ under π^B) in Figure 10(b). All the values in Figure 10 are averaged over 25 replications.

As α increases ($\beta = 1$), the daily effort imbalance is penalized more. Figure 10(a) shows that under π^B, when the average maximum daily delivery time among the couriers is reduced by 5 minutes (from 8.27 to 8.18 hours), the difference between the maximum and minimum number of total delivery packages over the 28 days among the couriers almost doubles (from 9.4 to 18.8 packages). Note that the operating cost only changes 0.2% under different values of α. We also observe that compared with π^B under all the values of α, π^V results in much larger average daily effort imbalance and total incentive imbalance over the 28 days.

On the other hand, as β increases ($\alpha = 1$), the total incentive imbalance is penalized more. Figure 10(b) shows that under π^B, when the difference between the maximum and minimum numbers of total delivery packages over the 28 days among the couriers is reduced by 1.7 packages (from 9.4 to 7.7 packages), the average maximum daily delivery time among the couriers increases by 17 minutes (from 8.27 to 8.55 hours). Note that the operating cost only changes 0.8% under different values of β. We also observe that compared with π^B under all the values of β, π^M results in much larger total incentive imbalance but slightly smaller average daily effort imbalance over the 28 days.

We conclude that a reasonable trade-off among the operating cost and the two types of workload imbalance can be achieved as long as both are explicitly considered.



(a) Under different values of α ($\beta = 1$)

(b) Under different values of β ($\alpha = 1$)

Figure 10: Average daily effort imbalance and total incentive imbalance over the 28 days

6. Summary and final remarks

We have studied a setting in which a delivery manager responsible for the daily dispatching of urban deliveries considers workload balance among the couriers to maintain workforce morale. The delivery manager focuses on two types of workload: incentive workload, which relates to the delivery quantity and affects a courier's income, and effort workload, which relates to the delivery time and affects a courier's health. To design effective dispatching policies, we introduce a multi-period workload balancing problem and formulate it as an MDP. We propose and evaluate a balanced penalty policy based on CFA, and use a hybrid algorithm combining the modified nested partitions method and the KN++ procedure to search for the optimal policy parameters. We have conducted an extensive computational study that reveals the efficacy of the balanced penalty policy; it achieves low operating cost while still ensuring workload balance and is robust against demand variation.

Our research introduces several innovations. It considers equity metrics that have to be balanced over different periods; incentive workload has to be balanced over a long period of time (e.g., a week or a month) whereas effort workload has to be balanced over a short period of time (e.g., a shift or a day). It considers workload balance in an environment with uncertain demand; existing research on workload balance has assumed given, known demand.

In practice, many factors impact effort workload. Consistency, i.e., making deliveries at known locations, tends to increase efficiency. Consistency can also have less tangible benefits, e.g., couriers developing familiarity with the delivery recipients, which may increase their level of enjoyment and satisfaction. Geographic and demographic characteristics of an urban area, e.g., residential vs. commercial, low-density vs. high-density, and low-rise vs. high-rise, impact delivery time. Future research may focus on how to include such considerations.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (NSFC, Project No. 71361130017) and the Tsinghua Logistics and Supply Chain Man-

agement Research Center.

References

Analysis, 2019. E-commerce digital process analysis: GMV of B2C market in china reached 1.52676 trillion rmb at the third quarter of 2019. https://www.analysys.cn/article/detail/20019518, in Chinese, last accessed on 11 July 2020.

Apte, U.M., Mason, F.M., 2006. Analysis and improvement of delivery operations at the san francisco public library. Journal of Operations Management 24, 325–346.

Baņos, R., Ortega, J., Gil, C., Ferna̧ndez, A., De Toro, F., 2013a. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. Computers & Industrial Engineering 65, 286–296.

Baņos, R., Ortega, J., Gil, C., Ferna̧ndez, A., De Toro, F., 2013b. A simulated annealing-based paralle multi-objective approach to vehicle routing problems with time windows. Expert Systems with Applications 40, 1696–1707.

Baldacci, R., Hadjiconstantinou, E., Mingozzi, A., 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research 52, 723–738.

Blakeley, F., Argüello, B., Cao, B., Hall, W., Knolmajer, J., 2003. Optimizing periodic maintenance operations for schindler elevator corporation. Interfaces 33, 67–79.

Borgulya, I., 2008. An algorithm for the capacitated vehicle routing problem with route balancing. Central European Journal of Operations Research 16, 331–343.

Bowerman, R., Hall, B., Calamai, P., 1995. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transportation Research Part A: Policy and Practice 29, 107–123.

Chen, Q., Zhao, L., Fransoo, J.C., Li, Z., 2019. Dual-mode inventory management under a chance credit constraint. OR Spectrum 41, 147–178.

China Federation of Logistics & Purchasing, China Logistics Information Center, 2017. Survey report for china e-commerce logistics and express employees. http://www.chinawuliu.com.cn/lhhkx/201704/29/320924.shtml, in Chinese, last accessed on 2 January 2018.

Corberán, A., Fernández, E., Laguna, M., Martí, R., 2002. Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of the operational research society 53, 427–435.

Daganzo, C.F., 1984. The length of tours in zones of different shapes. Transportation Research Part B: Methodological 18, 135–145.

Dobbs, R., Manyika, J., Woetzel, J., 2015. The four global forces breaking all the trends. https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/the-four-global-forces-breaking-all-the-trends, last accessed on 19 March 2020.

Ecommerce Foundation, 2017. Global ecommerce report 2017. https://www.ecommercewiki.org/reports/89/global-b2c-ecommerce-country-report-2017-free, last accessed on 30 September 2019.

Garcia-Najera, A., Bullinaria, J.A., 2011. An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. Computers & Operations Research 38, 287–300.

Groër, C., Golden, B., Wasil, E., 2009. The balanced billing cycle vehicle routing problem. Networks: An International Journal 54, 243–254.

Gulczynski, D., Golden, B., Wasil, E., 2011. The period vehicle routing problem: New heuristics and real-world variants. Transportation Research Part E: Logistics and Transportation Review 47, 648–668.

Halvorsen-Weare, E.E., Savelsbergh, M.W.P., 2016. The bi-objective mixed capacitated general routing problem with different route balance criteria. European Journal of Operational Research 251, 451–465.

Hamachera, H.W., Pedersen, C.R., Ruzika, S., 2007. Finding representative systems for discrete bicriterion optimization problems. Operations Research Letters 35, 336–344.

Huang, Y., Savelsbergh, M.W.P., Zhao, L., 2018. Designing logistics systems for home delivery in densely populated urban areas. Transportation Research Part B-methodological 115, 95–125.

iResearch, 2019. 2019 china's brand e-commerce service industry report. http://report.iresearch.cn/report/201906/3391.shtml, in Chinese, last accessed on 31 October 2019.

Jozefowiez, N., Semet, F., Talbi, E.G., 2002. Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem., in: International Conference on Parallel Problem Solving from Nature, pp. 271–280.

Jozefowiez, N., Semet, F., Talbi, E.G., 2006. Enhancements of nsga ii and its application to the vehicle routing problem with route balancing., in: Artificial Evolution 2005, 7th International Conference (EA'2005), Lecture Notes in Computer Science, pp. 131–142.

Jozefowiez, N., Semet, F., Talbi, E.G., 2007. Target aiming pareto search and its application to the vehicle routing problem with route balancing. Journal of Heuristics 13, 455–469.

Jozefowiez, N., Semet, F., Talbi, E.G., 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. European Journal of Operational Research 195, 761–769.

Kim, S.H., Nelson, B.L., 2006. On the asymptotic validity of fully sequential selection procedures for steady-state simulation. Operations Research 54, 475–488.

Kritikos, M.N., Ioannou, G., 2010. The balanced cargo vehicle routing problem with time windows. International Journal of Production Economics 123, 42–51.

Lacomme, P., Prins, C., Prodhon, C., Rena, L., 2015. A multi-start split based path relinking (msspr) approach for the vehicle routing problem with route balancing. Engineering Applications of Artificial Intelligence 38, 237–351.

Lacomme, P., Prins, C., Sevaux, M., 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. Computers & Operations Research 33, 3473–3493.

Lee, T.R., Ueng, J.H., 1999. A study of vehicle routing problems with load-balancing. International Journal of Physical Distribution & Logistics Management 29, 646–658.

Liu, C.M., Chang, T.C., Huang, L.F., 2006. Multi-objective heuristics for the vehicle routing problem. International Journal of Operations Research 3, 173–181.

Liu, R., Xie, X., Garaix, T., 2013. Weekly home health care logistics, in: Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on, IEEE. pp. 282–287.

Matl, P., Hartl, R.F., Vidal, T., 2018. Workload equity in vehicle routing problems: A survey and analysis. Transportation Science 52, 239–260.

Matl, P., Hartl, R.F., Vidal, T., 2019a. Leveraging single-objective heuristics to solve bi-objective problems: Heuristic box splitting and its application to vehicle routing. Networks 73, 382–400.

Matl, P., Hartl, R.F., Vidal, T., 2019b. Workload equity in vehicle routing: The impact of alternative workload resources. Computers & Operations Research 110, 116–129.

Melián-Batista, B., De Santiago, A., AngelBello, F., Alvarez, A., 2014. A bi-objective vehicle routing problem with time windows: A real case in tenerife. Applied Soft Computing 17, 140–152.

Mourgaya, M., Vanderbeck, F., 2007. Column generation based heuristic for tactical planning in multi-period vehicle routing. European Journal of Operational Research 183, 1028–1041.

Murata, T., Itai, R., 2005. Multi-objective vehicle routing problems using two-fold emo algorithms to enhance solution similarity on non-dominated solutions, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 885–896.

Murata, T., Itai, R., 2007. Local search in two-fold emo algorithm to enhance solution similarity for multi-objective vehicle routing problems, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 201–215.

Pacheco, J., Caballero, R., Laguna, M., Molina, J., 2013. Bi-objective bus routing: an application to school buses in rural areas. Transportation Science 47, 397–411.

Pacheco, J., Martí, R., 2006. Tabu search for a multi-objective routing problem. Journal of the Operational Research Society 57, 29–37.

Pasia, J.M., Doerner, K.F., Hartl, R.F., Reimann, M., 2007a. A population-based local search for solving a bi-objective vehicle routing problem, in: European conference on Evolutionary computation in combinatorial optimization, Springer. pp. 166–175.

Pasia, J.M., Doerner, K.F., Hartl, R.F., Reimann, M., 2007b. Solving a bi-objective vehicle routing problem by pareto-ant colony optimization, in: International Workshop on Engineering Stochastic Local Search Algorithms, Springer. pp. 187–191.

People's Daily, 2019. The trading volume of china's postal service has increased by 7000 times in 70 years. http://www.gov.cn/xinwen/2019-09/18/content_5430750.htm, in Chinese, last accessed on 1 November 2019.

Pitney Bowes, 2019. Parcel shipping index reports continued growth bolstered by china and emerging markets. https://www.pitneybowes.com/us/shipping-index.html, in English, last accessed on 25 March 2020.

Powell, W.B., 2014. Clearing the jungle of stochastic optimization, in: Bridging data and decisions. INFORMS, pp. 109–137.

Powell, W.B., 2019. A unified framework for stochastic optimization. European Journal of Operational Research 275, 795–821.

Reiter, P., Gutjahr, W.J., 2012. Exact hybrid algorithms for solving a bi-objective vehicle routing problem. Central European Journal of Operations Research 20, 19–43.

Ribeiro, R., Lourenço, H.R., 2001. A multi-objective model for a multi-period distribution management problem, in: 4th Metaheuristics International Conference, pp. 97–101.

Sarpong, B.M., Artigues, C., Jozefowiez, N., 2013. Column generation for bi-objective vehicle routing problems with a min-max objective, in: 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13).

Shi, L., ÓLafsson, S., 2000. Nested partitions method for stochastic optimization. Methodology & Computing in Applied Probability 2, 271–291.

State Post Bureau of P. R. China, 2020. Statistics of the postal industry in 2019. http://www.spb.gov.cn/xw/dtxx_15079/202005/t20200519_2154947.html, in Chinese, last accessed on 4 October 2020.

Sutcliffe, C., Board, J., 1990. Optimal solution of a vehicle-routeing problem: transporting mentally handicapped adults to an adult training centre. Journal of the Operational Research Society 41, 61–67.

United Nations, 2019. World urbanization prospects: the 2018 revision. https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf, last accessed on 31 October 2019.

Appendix A. Hybrid algorithm

In this section, we describe the modified nested partitions method and the KN++ procedure in the hybrid algorithm (Figure 3). For presentational convenience, we denote $\hat{V}(p,\omega)$ as the total cost associated with p in the balanced penalty policy π^B evaluated under scenario ω, where $\omega = (\hat{D}_t)_{t=0,1,...,T-1}$ represents a stream of realized demand over the planning horizon. We further denote $\bar{V}(p,\mathcal{S}) = \sum_{s=1}^{|\mathcal{S}|} \hat{V}(p,\omega_s)/|\mathcal{S}|$ as the average total cost over sample $\mathcal{S}$.

Appendix A.1. Modified nested partitions method

In Algorithm 1, given the search region $\mathcal{P}$ of all possible values of p, the modified nested partitions method outputs a candidate pool $\mathcal{P}^C = \{(p, v(p))\}$ in K^{max} iterations, where $v(p)$ is the latest estimated total cost associated with p.

In Step 0, we initialize the most promising region σ^k, the candidate pool $\mathcal{P}^C$, and the iteration number k.

In Step 1, we partition the most promising region σ^k into M_{σ^k} disjoint subregions, and aggregate the surrounding regions as subregion $\sigma^k_{M_{\sigma^k}+1}$.

In Step 2, we randomly select $N_{\sigma^k_i}$ points of p in each subregion σ^k_i, evaluate each p^j_i by $\bar{V}^k(p^j_i, \mathcal{S}^k)$, and update $\mathcal{P}^C$ with newly-generated or improved p^j_i, $\forall i = 1,2,...,M_{\sigma^k}+1$, $j = 1,2,...,N_{\sigma^k_i}$.

In Step 3, we update the most promising region σ^{k+1} based on $\bar{V}^k(p^j_i, \mathcal{S}^k)$ in Step 2.

In Step 4, we terminate the algorithm and output the candidate pool $\mathcal{P}^C$ in iteration K^{max}, or return to Step 1 to start iteration $k+1$.

Appendix A.2. KN++ procedure

We denote $\mathcal{P}^K$ as the candidate pool of p generated from the candidate pool $\mathcal{P}^C = (p, v(p))$, which is outputted by the modified nested partitions method (Algorithm 1). In Algorithm 2, given the candidate pool $\mathcal{P}^K$ and the indifference-zone parameter δ, we use the KN++ procedure to select the optimal p^* under a confidence level $1-\alpha$.

In Step 0, we evaluate $\hat{V}(p,\omega)$ for each $p \in \mathcal{P}^K$ under each scenario ω in the first stage sample $\mathcal{S}$, and initialize the observation counter r.

In Step 1, we update the variance estimator $\mathbb{S}^r(p,p'), \forall p,p' \in \mathcal{P}^c$. Based on the $\mathbb{S}^r(p,p')$, we compute the maximum observation number, and determine whether to terminate the algorithm.

In Step 2, we screen the $\bar{V}^r(p,\mathcal{S})$ of the remaining $p \in \mathcal{P}^K$ and eliminates the inferior p's with certain statistical confidence.

In Step 3, we terminate the algorithm if only one p remains in $\mathcal{P}^K$ as the optimal p^*. Otherwise, we evaluate $\hat{V}(p,\omega^{r+1})$ for each remaining $p \in \mathcal{P}^K$ under a newly-generated scenario ω^{r+1}, and return to Step 1.

Algorithm 1: The modified nested partitions method

```
Input: The entire search region $\mathcal{P}$. The size of candidate pool $C$. The maximum
        number of iterations $K^{max}$.
Output: The candidate pool $\mathcal{P}^C$.
Step 0 Initialization.
    Set the most promising region $\sigma^k$ as the entire search region $\mathcal{P}$.
    Set $\mathcal{P}^C = \emptyset$.
    Set the iteration counter $k = 0$.
Step 1 Partition.
    If the most promising region $\sigma^k$ is a singleton, keep it unchanged (no partitions).
    Else, partition $\sigma^k$ into $M_{\sigma^k}$ disjoint subregions $\sigma^k_1, \sigma^k_2, ..., \sigma^k_{M_{\sigma^k}}$.
    Aggregate the surrounding regions as subregion $\sigma^k_{M_{\sigma^k}+1}$.
        Note that if $\sigma^k = \mathcal{P}$, $\sigma^k_{M_{\sigma^k}+1} = \emptyset$.
Step 2 Sampling and evaluating $p$, and updating $\mathcal{P}^C$.
    Generate a sample $\mathcal{S}^k$ of size $H$.
    In each subregion $\sigma^k_i$, $i = 1, 2, ..., M_{\sigma^k} + 1$, randomly select $N_{\sigma^k_i}$ points of $p$,
        denoted as $p^j_i, j = 1, 2, ..., N_{\sigma^k_i}$.
    For each $p^j_i$, $i = 1, 2, ..., M_{\sigma^k} + 1$, $j = 1, 2, ..., N_{\sigma^k_i}$:
        Evaluate $\hat{V}^k(p^j_i, \omega)$ under each scenario $\omega \in \mathcal{S}^k$, and calculate $\bar{V}^k(p^j_i, \mathcal{S}^k)$.
        If $p^j_i$ does not exist in $\mathcal{P}^C$, insert $(p^j_i, \bar{V}^k(p^j_i, \mathcal{S}^k))$ into $\mathcal{P}^C$.
        Else if $\bar{V}^k(p^j_i, \mathcal{S}^k) < v(p^j_i)$, update $v(p^j_i) = \bar{V}^k(p^j_i, \mathcal{S}^k)$.
        If $|\mathcal{P}^C| > C$, remove the $(p, v(p))$ pair with the largest $v(p)$ from $\mathcal{P}^C$.
Step 3 Updating the most promising region.
    Let $i^* = \arg\min_{i \in \{1,2,...,M_{\sigma^k}+1\}} \min_{j \in \{1,2,...,N_{\sigma^k_i}\}} \bar{V}^k(p^j_i, \mathcal{S}^k)$.
    If $i^* \leq M_{\sigma^k}$, update the most promising region as $\sigma^{k+1} = \sigma^k_{i^*}$.
    Else, backtrack the most promising region to the entire region as $\sigma^{k+1} = \mathcal{P}$.
Step 4 Stopping rule.
    If $k = K^{max}$, terminate and output $\mathcal{P}^C$.
    Else, set the iteration counter $k = k + 1$. Go to Step 1.
```

```
Algorithm 2: The KN++ procedure
Input: The candidate pool $\mathcal{P}^K$.
       The confidence level $1-\alpha$.
       The indifference-zone parameter $\delta > 0$.
       The first stage sample size $n_0 \geq 2$.
Output: The best $p^*$.
Step 0 Initialization.
  Generate a sample $\mathcal{S}$ of size $n_0$.
  For each $p \in \mathcal{P}^K$, evaluate $\hat{V}(p,\omega)$ under each scenario $\omega \in \mathcal{S}$.
  Set the observation counter $r = n_0$.
Step 1 Update.
  Calculate $\eta = \frac{1}{2}[(\frac{2\alpha}{k-1})^{-\frac{2}{r-1}} - 1]$, and $h^2 = 2\eta(r-1)$.
  Compute the estimator $\mathbb{S}^r(p,p')$, the sample variance of the difference between the total
    costs associated with $p$ and $p'$.

    $$\mathbb{S}^r(p,p') = \frac{1}{r-1}\sum_{\omega\in\mathcal{S}}(\hat{V}(p,\omega) - \hat{V}(p',\omega) - [\bar{V}^r(p,\mathcal{S}) - \bar{V}^r(p',\mathcal{S})])^2, \quad \forall p \neq p' \in \mathcal{P}^K,$$

  Compute $N^r(p,p') = \lfloor \frac{h^2\mathbb{S}^r(p,p')}{\delta} \rfloor, \forall p \neq p' \in \mathcal{P}^c$, and $N^r(p) = \max_{p' \neq p} N^r(p,p')$.
  If $r \geq \max_{p\in\mathcal{P}^c} N^r(p) + 1$, terminate and output $p^*$ with the minimum $\bar{V}^r(p^*,\mathcal{S}), p^* \in \mathcal{P}^K$.
  Else, go to Step 2.
Step 2 Screening.
  Set $\mathcal{P}^{K-old} = \mathcal{P}^K$
  Let $\mathcal{P}^K = \{p : p \in \mathcal{P}^{K-old}$ and $\bar{V}^r(p,\mathcal{S}) \leq \bar{V}^r(p',\mathcal{S}) + W^r(p,p'), \forall p' \neq p \in \mathcal{P}^{K-old}\}$,
    where $W^r(p,p') = \max\{0, \frac{\delta}{2r}(\frac{h^2\mathbb{S}^r(p,p')}{\delta^2} - r)\}$.
Step 3 Stopping rule.
  If $|\mathcal{P}^K| = 1$, terminate and output the only element $p^*$.
  Else:
    Generate a new scenario $\omega^{r+1}$ and add it into the sample $\mathcal{S}$.
    For each $p \in \mathcal{P}^K$, evaluate $\hat{V}(p,\omega^{r+1})$.
    Set the observation counter $r = r + 1$.
    Go to Step 1.
```

Appendix A.3. Implementation details

In this section, we describe the implementation details of the hybrid algorithm. Note that $\mathcal{P} = (0, 1]$ is the entire search region of p, as described in Section 4.1. $\mathcal{P}$ is discretized into 100 points with a precision of 0.01, i.e., $\{0.01, 0.02, \ldots, 1\}$.

Table A.6: Parameter settings of the hybrid algorithm

Parameter	Value
Modified nested partitions method	
Entire search region, $\mathcal{P}$	100 points: (0:0.01:1]
Size of candidate solution pool, C	10
Maximum number of iterations, K^{max}	50
Number of partitioned subregions of σ^k, M_{σ^k}	$\min\{5,\ L_{\sigma^k}\}$
Number of sampled points in each subregion σ_i^k, $N_{\sigma_i^k}$	$20\%L_{\sigma_i^k}$
Number of demand observations for each sample point, H	5
KN++ procedure	
Confidence level, $1-\alpha$	0.99
Indifference-zone parameter, δ	0.5
First stage sample size, n_0	10

The algorithm parameter settings are shown in Table A.6. In the modified nested partitions method (Algorithm 1), we denote the number of points in a subregion σ_i^k in iteration k as $L_{\sigma_i^k}$, $\forall i = 1, 2, ..., M_{\sigma^k} + 1$. In Step 1, if $L_{\sigma^k} \geq 5$, we partition the most promising subregion σ^k into $M_{\sigma^k} = 5$ subregions; otherwise, we partition σ^k into $M_{\sigma^k} = L_{\sigma^k}$ singleton subregions. In Step 2, we uniformly sample $N_{\sigma_i^k} = 20\%L_{\sigma_i^k}$ (round up to the nearest integer) points in each subregion σ_i^k, and use $H = 5$ demand observations to evaluate each p.

Further, to speed up the hybrid algorithm, we use parallel computing method when evaluating multiple p in Step 2 of the modified nested partitions method (Algorithm 1), and in the initialization and sequential observations of the KN++ procedure (Algorithm 2).

Appendix B. Proof of Observation 1

Proof. Proof. Without loss of generality, we assume that the cumulative incentive workload of the couriers $\tilde{I}_{t1} \leq \tilde{I}_{t2} \leq \ldots \leq \tilde{I}_{tK}$. Given an arbitrary order of numbers of delivery packages $\{I_{t+1,1}, I_{t+1,2}, ..., I_{t+1,K}\}$ of the routes, if there exist any routes $k_1 < k_2$ with $I_{t+1,k_1} \leq I_{t+1,k_2}$, we can swap them and find that

$$\tilde{I}_{tk_1} + I_{t+1,k_1} \leq \tilde{I}_{tk_1} + I_{t+1,k_2} \leq \tilde{I}_{tk_2} + I_{t+1,k_2},$$
$$\tilde{I}_{tk_1} + I_{t+1,k_1} \leq \tilde{I}_{tk_2} + I_{t+1,k_1} \leq \tilde{I}_{tk_2} + I_{t+1,k_2}.$$

The swap can reduce (at least remain) the range of cumulative incentive workload $\tilde{\mathcal{I}}_{t+1}$ among couriers in three situations as follows:

1. If $\tilde{I}_{tk_1}+I_{t+1,k_1}$ is the minimum cumulative incentive in the original assignment, the swap can increase the minimum cumulative incentive workload ($\tilde{I}_{tk_1}+I_{t+1,k_1} \leq \tilde{I}_{tk_1}+I_{t+1,k_2}$).
2. If $\tilde{I}_{tk_2}+I_{t+1,k_2}$ is the maximum cumulative incentive in the original assignment, the swap can decrease the maximum cumulative incentive workload ($\tilde{I}_{tk_2}+I_{t+1,k_1} \leq \tilde{I}_{tk_2}+I_{t+1,k_2}$).
3. If $\tilde{I}_{tk_1}+I_{t+1,k_1}$ and $\tilde{I}_{tk_2}+I_{t+1,k_2}$ is not the minimum and maximum cumulative incentive workload in the original assignment, the swap will not change the minimum and maximum cumulative incentive workload.

In summary, after the swap, the minimum cumulative incentive workload will increase or remain the same, and the maximum cumulative incentive workload will decrease or remain the same, such that the range of cumulative incentive workload among couriers will decrease or remain the same.

Thus, if there exists any two couriers $k_1 < k_2$ with $I_{t+1,k_1} \leq I_{t+1,k_2}$, we can swap them an obtain smaller or at least the same range of the cumulative incentive workload. After a limited number of swaps, we finally obtain the minimum range of the cumulative incentive workload among couriers from period 1 to period $t+1$ with a descending sort of numbers of delivery packages $\{I_{t+1,1}, I_{t+1,2}, ..., I_{t+1,K}\}$. □ □