Community Logistics for dynamic vehicle dispatching: the effects of community departure "time" and "space"

Abstract

The rise of e-commerce has completely transformed the dynamicity and problem nature of last-mile delivery, owing to significant B2C customer demand in compact urban areas. Given the unprecedented growths of urban last-mile deliveries, this paper proposes a novel postponement prioritized-route optional approach, namely Community Logistics (CL), as a new logistics tool to manage dynamic arrivals of delivery requests received in e-commerce hubs. Each vehicle is responsible for serving a "community". At each decision epoch, fragmented e-commerce delivery requests arrived at the depot are either allocated to a community or postponed to later epochs for actual last-mile delivery. With an objective of consolidating newly arrived requests, we develop two dynamic policies – temporal and spatial, respectively for temporally delaying vehicle's departure and spatially allocating more pre-partitioned geographical cells into one community. The main contribution of this study lies in a spatiotemporal relativity analysis and a comparative analysis. The former demonstrates the essence of incorporating both dynamic community departure times and dynamic community regions into managing urban e-commerce deliveries, whereas the latter validates the merits of Community Logistics against dynamic vehicle routing solutions. In the end, we call for further developments of community logistics strategies to address the impacts of urban deliveries due to the rise of e-commerce and online shopping.

Keywords Community logistics (CL); e-commerce last-mile delivery; urban logistics; dynamic delivery dispatching; megacities deliveries; serving region generation problem (SRGP)

1. Introduction

The impact of digital business and online shopping is apparent. On the positive side, convenience is one of the major benefits for many consumers, whereas retailers could reach a wider customer base with e-commerce enabling them to cross geographical boundaries. However, an analysis performed by the World Economic Forum on "The future of the last-mile ecosystem" suggested that growing demand for e-commerce last-mile delivery will result in over 30% more delivery vehicles in urban cities by 2030. Consequently, this leads to a rise in both emissions and traffic congestion if no effective intervention is applied (World Economic Forum, 2020). Therefore, the development of innovative strategies and solutions of vehicle dispatching and scheduling continue to play a significant role in order to transform the way vehicles perform last-mile delivery tasks in urban areas.

Given the unprecedented challenges in striking a balance between faster e-commerce delivery and reducing the associated undesired impacts, this paper proposes a novel operating strategy, namely Community Logistics (CL), which deals with a single-depot multi-vehicle dynamic delivery dispatching problem under e-commerce same-day/next-day delivery context. The proposed strategy is general, making it applicable to most common delivery contexts found in today's e-commerce sector. In this section, we uncover the practical relevance of the proposed CL in Section 1.1 by revealing the current vehicle dispatching operational flow in real environment. Then, we review the relevant vehicle dispatching solutions in the literature in Section 1.2 to discuss our research relevance and thus introduce the research questions to be addressed in this paper.

1.1 Practical relevance

Online retailers, like Amazon, dynamically receive new online orders every single minute. Currently, the common practice of e-commerce same-day or next-day delivery performed by online retailers is to aggregate the received customer orders and allocate them to each available vehicle which has a pre-determined service region. Once the vehicle dispatching decision is finalized, orders will be loaded to respective vehicles for actual delivery. It is a common phenomenon that any orders arrived after the vehicle dispatching decision finalized will not be served at the current dispatch cycle.

The current practice can be treated as a static delivery dispatching approach, where the dispatch cycle is fixed and all orders received during the cycle must be dispatched at the end of the cycle. Although giant online retailers like Amazon launched membership feature allowing customers to receive their orders quicker by paying an extra membership fee, late delivery still happens to customers who paid extra costs[12]. As a result of delayed delivery of the orders in the current dispatch cycle, one could imagine the consequences of orders received at future cycles – a continuous delivery backlog problem[3].

To manage the continuously arriving delivery orders flexibly, the CL strategy serves as a new

---

[1] https://www.vox.com/recode/2019/12/5/20997515/amazon-prime-delivery-late-delays-one-day-shipping

[2] https://www.theverge.com/2020/3/15/21180737/amazon-deliveries-delayed-coronavirus

[3] There are numerous root causes of delivery delay in recent years, including the impact of the pandemic on global trade, shortage of labour, etc. Given the uncontrollable external factors affecting delivery efficiency and the large amount of delivery resources (fleets) already in place, we must find better ways of managing the frequent arrivals of online orders.

logistics support tool for "dynamically" generating compact serving regions for a given fleet of vehicles. There are two major differences of the proposed CL strategy compared to any existing approaches: solution format and the logic of formulating a vehicle's serving region. We treat a serving region of each vehicle, defined as "community", rather than the visiting sequence of a vehicle, as the solution format. The logic of community formulation is not as simple as partitioning nodes into clusters based on geographical proximity. As vehicles are allowed to depart dynamically without a fixed departure interval, the fluctuating arrivals of delivery requests with known delivery destinations would influence whether a region containing some pending requests should become a community to be served by a vehicle. Therefore, the tradeoff between fast delivery (temporal perspective) and consolidation (spatial perspective) comes into play. Due to such temporal and spatial considerations, we define "Community Time" and "Community Space" as two inter-related attributes during the CL's decision-making process. The former treats the varying vehicle dispatching times in each community, whereas the latter addresses the varying sizes and geographical locations of a community upon its formulation.

The scope of this study is located within the cross intersections of vehicle districting and dynamic dispatching. The CL strategy applies districting as a means of solving dynamic dispatching problems in same-day and next-day e-commerce delivery context. To explicitly explain the differences between the CL strategy and the existing solution approaches in the dynamic delivery dispatching literature, we systematically review all relevant problem variants from spatial and temporal perspectives. The outcome of the review is a set of research questions set out in this paper that promote the development of dynamic vehicle dispatching in e-commerce last-mile delivery context.

1.2 Research relevance

(i) Managing spatial uncertainties in dynamic delivery dispatching

- **Cluster VRPs**

Cluster VRPs (CluVRPs) are one of the variants with the strongest relation to our CL strategy formulation[4]. It requires all the customers to be visited exactly once, but a vehicle visiting one customer in a cluster must visit all the remaining customers therein before leaving it. In the literature, numerous CluVRPs studies have been performed, with the integration of additional considerations found in real

[4] This VRP variant is a generalization of the capacitated vehicle routing problem (CVRP), which deals with customer demand through clustering customers nodes based on their delivery or pick-up locations, followed by identifying the shortest route in each cluster (Sevaux and Sörensen, 2008).

delivery scenarios. Hintsch & Irnich (2018) regard CluVRP as a three-stage optimization problem, where the customer assignment in each cluster, the vehicle assignment in each cluster and the visiting sequence for each vehicle are optimized sequentially. They also propose a heuristic optimization containing a large multiple neighborhood search method with multiple clusters destroy and repair operators and a variable-neighborhood descent for post-optimization. In general, the CluVRP, which deals with vehicle dispatching problems through logically groups spatially nearby customer demand together for batch delivery, pre-assumes a set of orders being received for solution generation. Hence, this variant does not take the temporal arrival of customer demand into account.

- **Districting problems**

Districting problems aim to divide a large geographical area into several sub-areas or regions to ensure that customers in each sub-area are close to the corresponding service node (Kalcsics & Ríos-Mercado, 2019). This problem domain can be classified into two streams: Static and dynamic districting. Static districting is the partitioning of customer nodes into districts with pre-defined sizes and locations. Dynamic districting allows a more flexible formulation of districts with varying sizes and locations at each departure cycle. Most of the districting models in the literature focused on static districting by determining the fixed size and location of each district given a particular scope, such as districting for routing with stochastic customers (Lei et al., 2012), formulating a two-stage districting problem by establishing districts in the first stage and adapting them to the daily demand realizations in the second stage (Bender et al., 2020), etc. As region districting can significantly reduce the complexity of route optimization and improve the route compactness, districting strategies are widely considered in some delivery contexts, such as parcel pickups and deliveries (González-Ramírez et al., 2011), waste collection (Mourão et al., 2009) and etc. Most research works in this field aim to design appropriate depot locations and serving regions in urban areas (Carlsson, 2012; Carlsson & Delage, 2013)[5]. Zhou et al. (2021) develop a hybrid multi-population genetic algorithm for solving a real-world territory

---

[5] For example, Winkenbach et al. (2016) present a districting model to simultaneously determine the optimal facilities' number and locations, the optimal size and shape of each facility serving region, and the corresponding optimal fleet sizer and composition in each facility. Bender et al. (2020) study a districting problem in a real-word parcel delivery case in Germany, where heterogeneous delivery resources including different drivers and vehicles are allocated to ensure service consistency and adequate delivery resources for fluctuated customer demands. In this situation, the districting results are generally static as it is impossible to frequently relocate the terminal depot and modify its serving region.

design problem encountered by a dairy company. The problem is to partition customers into several districts so that the fixed costs of the districts and the corresponding routing costs are jointly minimized. Banerjee et al. (2022) study a tactical design problem for same-day delivery systems to determine the optimal fleet size and corresponding vehicle service region. Conputational studies demonstrate that such design can improve system efficiency in related vehicle routing settings.

However, when districting is performed for generating vehicle dispatching solutions, the districting results can be more dynamic. For example, Lei et al. (2016) introduce a multi-objective dynamic stochastic districting and routing problem (MDSDRP) to jointly optimize the number of vehicles, district compactness, district dissimilarity and vehicle profit equity. In their study, vehicles serving regions can be slightly revised when new orders arrive. Huang et al. (2018) present a new concept, namely block, to describe a serving region of vehicle, which is comprised of several basic geographic cells. A block design problem is developed to determine the optimal combinations of cells based on different customer density in each cell. Although the modification of vehicle serving region is considered in some studies, vehicle departure time is generally neglected. Obviously, the vehicle departure time can significantly influence vehicle serving regions under e-commerce context.

- **Research gaps associated to managing spatial uncertainties in dynamic delivery dispatching**

Both static and dynamic districting models assume fixed vehicle departure intervals and disallow delayed departure of any specific districts. The major differences between our proposed communities and the existing static or dynamic districting models, as illustrated in Fig. 1, lie in the fact that (1) each community can have their own vehicle dispatch time depending on the real-time fluctuations of order arrivals, referred to as "dynamic community time", and (2) the size and location of a community varies at each departure, leading to the formulation of "dynamic community space" at each vehicle's departure. Through relaxing the departure times, community formulation decision becomes even more dynamic than dynamic vehicle districting. The consideration of delayed departure of a community would influence the future formulation of a new community mix at later decision epochs due to the pending order set carried forward to the next decision epoch. In this regard, this study examines the feasibility and appropriateness of integrating "Community Time" as a decision variable into solving districting problems, which leads to the first research question of this study:

RQ1: Is "Community Time" a necessary parameter to be considered as decision variable in formulating dynamic vehicle dispatching solutions?

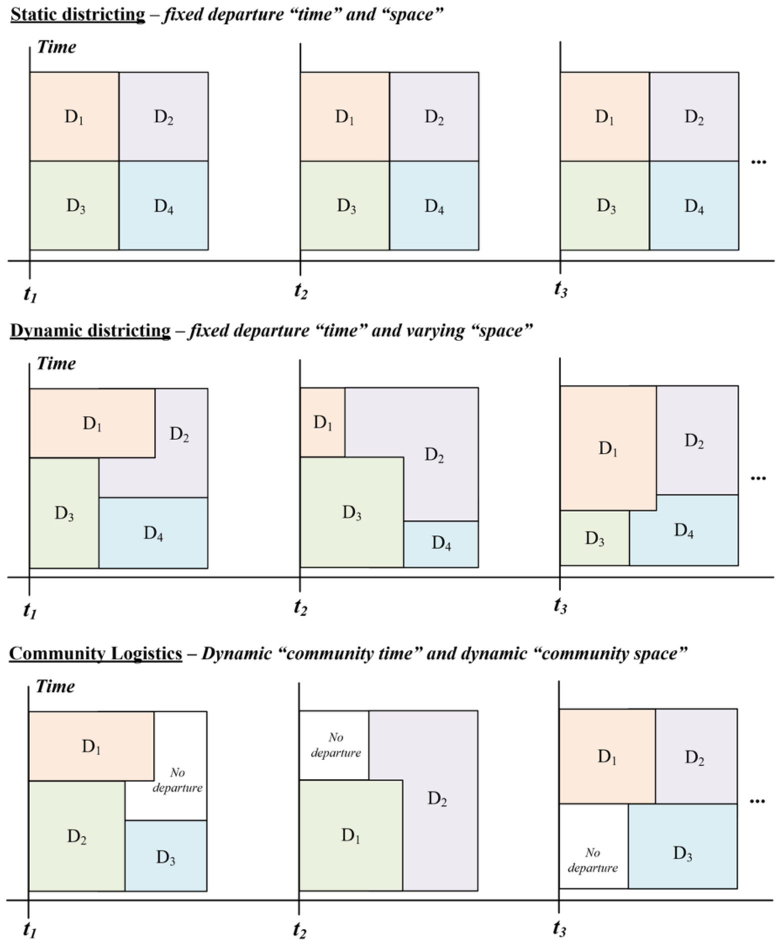


Fig. 1. Differences between static districting, dynamic districting, and community logistics

(ii) Managing temporal uncertainties in dynamic vehicle dispatching

- **Dynamic delivery dispatching problems**

To manage the arrivals of customer demand from the temporal perspective, this type of temporal-oriented problem falls into dynamic delivery dispatching problems (DDPs) and dynamic vehicle routing problems (DVRPs) in the literature. For DDPs, as introduced by Minkoff (1993), the timing and the composition of order batches for dispatching are the two major decisions to make. To minimize operating costs through better consolidation opportunities and more efficient tours, decision-makers have incentive to wait for future orders to arrive. However, dispatching decisions are generally bound by time windows as well as vehicle availability and storage capacity (Mitrović-Minić & Laporte, 2004; van Heeswijk et al., 2019; Kim & Lee, 2011). The literature has a variety of approaches developed to address how stochastic arrivals of customer demand are managed. van Heeswijk et al. (2019) addressed the last-mile dispatching problem with dispatch windows by formulating it as a Markov decision mode.

An approximate dynamic programming (ADP) algorithm is developed to offer flexibility in dispatch times. In contrast to our study, dispatching is not made in real time. Ulmer and Streng (2019) introduced a policy function approximation approach to decide the dispatch time of a vehicle. Similar to our study, they conducted a sensitivity analysis to balance and reflect the tradeoff between fast delivery and consolidation. The difference compared to our study is that they consider order consolidations through restricting customer pickup locations. All orders are delivered to designated pickup stations to facilitate dynamic dispatching of consolidated orders. In contrast, orders considered in our study can be shipped to any location specified by the customers. Order consolidations are realized through grouping orders into communities based on their geographical proximity.

- **Dynamic Vehicle Routing Problems**

The transportation scheduling problems for fulfilling dynamic requests fall into the DVRP variant, which mainly deals with three decisions (Toth & Vigo, 2014; Braekers et al., 2016):

(1) Spatial – the allocation of a set of customers to a vehicle; and/or

(2) Temporal – the dispatching intervals or times of a vehicle or a fleet of vehicles; and/or

(3) Routing – the visiting sequence of a vehicle.

To systematically review the existing DVRP works, we identify the key DVRP literature from a recent DVRP survey study conducted by Rios et al. (2021). We then investigate how the above (1), (2) and (3) decisions are addressed by each of these key literatures. The outcome of this process is the identification of research gaps in DVRP which suggests the motivation of this study. In the DVRP domain, models can be classified in terms of the characteristics of the problem – delivery and pickup. The delivery problem is a classic VRP that considers the deliveries of goods from a depot to a set of customers (Wassan and Nagy, 2014). On the other hand, VRPs with pickup consider the goods to be collected from customers' locations to the depot. In the literature, delivery and pickup problems are referred to as one-to-many and many-to-one problems respectively (Toth and Vigo, 2014).

In the last decade, the development of e-commerce and the advances of information technologies that enable real-time retrieval and processing of information, have promoted a shift of studying dynamic delivery dispatching and vehicle routing problems from solely consolidating delivery requests in distribution centers to simultaneously managing both delivery and pick-up requests. Recently, Savelsbergh and Van Woensel (2016) conducted a survey of the challenges and opportunities in city logistics. They concluded that most of the existing dynamic vehicle routing literature has limited applicability to the delivery context as they focus on the accommodating dynamic arrivals of pick-up requests into the real-time routing formulations, but not delivery requests. Examples are given in Appendix I.

Solution approaches dedicated to managing dynamic delivery requests are relatively fewer (Savelsbergh and Van Woensel, 2016). Kim et al. (2016) considered a DVRP with non-stationary stochastic travel times. Vehicles deliver goods from a depot to customers. Customers are assumed determined before the delivery, and all of them need to be served. Their model concerns the stochastic travel times, not the dynamic dispatch time of vehicle fleets. Thus, the planning horizon is fixed and this paper deals with (3). Jia et al. (2018) developed a dynamic dispatching system. The underlying model of the dispatching system is the dynamic vehicle routing problem which allows new orders being received as the working day progress. Their system focuses on the managing the spatial distribution of dynamic orders by developing a region partitioning scheme. The planning horizon is fixed and therefore this paper deals with (1) and (3). Voccia et al. (2017) has the most similar scope than ours as they introduced a multi-vehicle dynamic pickup and delivery problem with time constraints that incorporates key features associated with same-day delivery logistics. The solution approach integrates information about future requests into routing decisions by identifying when it is beneficial for vehicles to wait at the depot. Therefore, this paper deals with (2).

- **Research gaps associated to managing temporal uncertainties in dynamic delivery dispatching**

A summary of the DDP and DVRP literature, involving their specific scope and instance setup for experiments, is presented in Table 1. From the table we identify two existing gaps:

(i) *Lack of integrated considerations of both spatial and temporal decisions in dynamic districting and dispatching* – As reflected in the Table, no studies consider both (1) and (2) as a joint optimization problem. That said, the spatial and temporal characteristics of both pickup and delivery requests are often independently studied. In our study, we attempt to conduct a spatio-temporal relativity analysis to investigate the effect of different community sizes towards dispatch times. Such evaluations revealing the inter-relationships between the spatial and temporal decisions could enable decision makers to pre-determine a fixed region (community) or fixed dispatching intervals.

(ii) *Small instances for model validation* – Most DVRP models are specifically designed for a specific operating scenario. They are tested through a small instance as depicted in Table 1. Nevertheless, it is sensible considering the computation efforts, i.e. computation time in particular, being expensive in generating a new route whenever a new (set of) request arrives. In real-life e-commerce city logistics delivery context, however, online retailer giants like Amazon are receiving no fewer than thousands of online orders on a daily basis. They are required to be shipped in a guaranteed same-day or next-day delivery.

Table 1. Scopes and experiment settings of the selected DDP and DVRP studies

			Scope		
	Relevant literature	Instance size	(1) Spatial	(2) Temporal	(3) Routing
DDP	van Heeswijk et al. (2019)	3 – 50		✓	
DDP	Ulmer and Streng (2019)	Up to 1000		✓	
DVRP	Gendreau et al. (2006)	24 – 33	✓		✓
DVRP	Sarasola et al. (2015)	50 – 199	✓		✓
DVRP	Yu and Yang (2017)	25 – 100			✓
DVRP	Ulmer er al. (2018)	100		✓	
DVRP	Kim et al. (2016)	10			✓
DVRP	Jia et al. (2018)	72 – 400	✓		✓
DVRP	Voccia et al. (2017)	40 – 200		✓	
* Instance size refers to the number of customer requests used for conducting experiments					

Savelsbergh and Van Woensel (2016) addressed the need to develop a less-tailored approach to keep up with the ever-challenging e-commerce city logistics environment. In addition to this, we consider a fact that the routing decision as the sole solution format in VRP family potentially attributes to having smaller instances for model validation which saves computation efforts. These motivate us to investigate the possibility of recognizing the "region" rather than the "route" as the solution format. By looking into the formulation of dynamic communities, we have a chance to also discover the inter-relationships of both spatial and temporal decisions in dynamic districting and dispatching, serving as the justifications of treating dispatching times as a decision variable. More importantly, our community-based solution approach makes dynamic districting and dispatching tractable at the largest practical scales. We introduce four demand scenarios into our spatio-temporal relativity experiments, feeding our models with approximately a total of 1450, 2600, 5200 and 10200 daily delivery requests under low, medium, high and peak demand scenarios. To summarize, this paper examines the spatio-temporal relativity as a starting point of our community logistics studies. Another two research questions are formulated:

RQ2: What is the spatio-temporal (space versus time) relativity that poses effects towards community formulation under various delivery demand patterns?

RQ3: How does the CL strategy serve as a support tool to formulate communities prior to making routing decisions in each community?

The spatio-temporal relativity analysis taking a variety of demand scenarios into considerations not only inform the inter-relationship between community time and space, but also the degree of emphasis on time and space when modeling vehicle dispatching under different delivery scenarios. To validate the proposed CL strategy, we benchmark it against a static delivery dispatching policy and a DVRP solution method. These rigorous benchmarking studies are conducted to answer our final research question:

RQ4: What are the advantages of CL strategy over conventional route-based delivery strategy for handling the e-commerce last-mile same-day/next-day delivery?

The significances of this study are twofold. In theoretical perspective, the comparative results lead to a better understanding of how the proactive delay of a vehicle's departure could also play a role in generating a vehicle dispatching solution with a higher degree of route compactness. In practical perspective, the proposed CL strategy is a less problem-tailored, more general, and efficient approach to manage the fragmented, fluctuating arrivals of delivery requests in last-mile distribution hubs. It can be integrated into VRP models as a logistics support tool to formulate a community mix prior to optimizing the route of a community, or be extended to a variety of dispatching problems originally solved by VRP models.

The rest of this paper is organized as follows. Section 2 defines the CL strategy that deals with the urban logistics problems. Section 3 provides the problem description, model formulation and two dynamic policies. Section 4 presents the procedures, results, and discussions of the spatiotemporal relativity analysis which benchmark our dynamic policies against a static policy. Section 5 presents a comparative analysis which benchmarks our policies against a DVRP solution method. Section 6 provides the implications of the research to further inform the applicability of the CL strategy. Finally, section 7 gives the conclusions of the study and directions for future research.

2. Community Logistics Framework – What it is and how it works

2.1 Community Logistics: the strategic objective, definition, and its solution format

Community Logistics aims to identify practical, visually compact order dispatching solutions in real time at distribution centers. This goal is achieved by either dynamically introducing a waiting strategy in each delivery batch, and/or flexibly adjusting the size and location of a delivery community. Hence, the CL strategy is defined as *a strategic approach of managing the uncertainty of delivery requests through striking a balance between two dimensions: (i) community time – varying dispatching times and (ii) community space – varying sizes and locations*. As a logistics support tool complementing the existing VRP family, CL solutions determine a vehicle's geographical serving region, that is, the community, that it needs to visit. In other words, only delivery orders with destinations located within the community would be served by the vehicle. Orders outside this community would be served by another vehicle (community). With this solution format, the CL framework is best suitable for e-commerce distribution centres to manage a large number of small, fragmented delivery orders in megacities, which feature high rise buildings and high population density across a compact area. Given a bunch of small orders mostly located within a walkable distance, the CL solution intends not to minimize the traveling distance, but the number of locations to be visited. Ideally, there are enough discrete orders all within walkable distances so that the vehicle only needs to visit one delivery node to fulfil the order set.

2.2 Community Logistics: how it works

The formulation of a delivery community, as discussed above, is based on balancing the tradeoff between time and space dimensions. Theoretically, a large community in terms of geographical serving area would not require waiting time to be introduced because of a sufficiently large number of orders located within the community, and vice versa. However, a large community suffers from a low degree of satisfaction of both customers and delivery person due to an unreasonably long traveling distance and traveling time within the community. Introducing a waiting time for a smaller community is therefore a justifiable option. Thus, the question lies in how to flexibly introduce waiting time and dynamically adjust the size of a community simultaneously.

The execution of the dynamic CL strategy for e-commerce delivery scheduling creates a pooling effect of e-commerce delivery orders such that delivery orders with similar geographical destinations are aggregated for a certain period of time and assigned to a designated vehicle while taking account of the adjustable serving community of each vehicle in each epoch. Once dispatching solution, in a form of "community", is generated for a vehicle, visiting sequences (route) can be generated to facilitate the delivery person in charge to visit the designated delivery destinations specified by individual customers with shortest traveling distances.

To justify the feasibility and appropriateness of jointly considering community time and space in generating real-time vehicle dispatching solutions in e-commerce delivery context, the last-mile e-commerce delivery dispatching problem is formulated and presented in the next section, followed by the introduction of two policies, i.e. temporal and spatial policy, to examine the spatiotemporal tradeoff.

3. Problem definition under the Community Logistics Framework

3.1 Generic operating scenario of e-commerce last-mile delivery considered in this study

In this study, the research object is a terminal depot operated by a 3PL who is responsible for the last-mile delivery of e-commerce orders with specified delivery destinations located within a square region S. These delivery destinations involve not only home addresses, but also the locations of dedicated smart lockers and convenience stores which serve as the customers' parcel pick-up points. Therefore, the e-commerce orders are assumed to be delivered as soon as possible. The daily opening time of the depot is $0-T$. At $t=0$, an initial set of orders $V_0=\{1,2,...,v_0\}$ is accumulated in the depot from the previous day. Hereafter, another set of orders $V=\{v_0+1,v_0+2,...,v_0+v\}$ will continually arrive at the depot with a constant arrival rate λ or time-related arrival rate $\lambda(t)$. Each order $i\in V_0\bigcup V$ is associated with an arrival time t_i, weight w_i and delivery location (x_i,y_i). Since these orders are placed by online end customers, the order weight and delivery location can be recognized as random variables whose probability density functions are $f_{weight}(w)$ and $f_{loc}(x,y)$. In terms of vehicle availability, there are $|J|$ homogeneous vehicle with Q capacity and vel traveling speed during outbound delivery. The dynamic delivery dispatching problem tackled using the CL framework is the real-time identification of possible vehicle's delivery delay and the formulation and resizing of communities for the assignment of pending delivery orders to available vehicles.

3.2 A new problem variant under the Community Logistics Framework

Conventionally, the delivery problem mentioned in Section 4.1 can be tackled by a dynamic vehicle routing problem through dynamically determining a set of shortest vehicle traveling routes. In this section, the proposed CL strategy is deployed to handle the e-commerce last-mile delivery problem without the need to identify the routes in the first place.

With the adoption of the CL strategy, the decision process is firstly simplified by discretizing the temporal and spatial elements. From the spatial perspective, the serving area S is partitioned into cells in advance, denoted as $N = \{1, 2, ..., n\}$, according to the geographical boundaries such as roads, rivers or residential communities. The area covered by cell n is denoted as cr_n. From the temporal perspective, the depot opening hours is decomposed into K decision epochs. At the start of each decision epoch $dt^{(k)}$, i.e., $dt^{(k)} = k \cdot (T / K)$, $k = 1, 2, ..., K$, there is a set of pending orders at the depot, which is denoted as $I^{(k)}$. Decision makers need to identify the number of communities $|M^{(k)}|$ required and then determine serving regions of these $|M^{(k)}|$ delivery communities by appropriately assigning $|N|$ cells into these $|M^{(k)}|$ delivery communities. We denote the cell set contained in community $m \in M^{(k)}$ as $N_m^{(k)}$ ($N_m^{(k)} \subseteq N$), so that the serving area covered by community m can be represented by $\bigcup_{n \in N_m^{(k)}} cr_n$. Subsequently, decision makers need to estimate if the delivery community $m \in M^{(k)}$ should be served at the current decision epoch. If delivery community m is determined to be served at $dt^{(k)}$, a vehicle $j \in J^{(k)}$ will be assigned to serve the order set $I_m^{(k)}$ within this community $(I_m^{(k)} = \{i \in I^{(k)} : (x_i, y_i) \in \bigcup_{n \in N_m^{(k)}} cr_n\})$ by visiting the delivery community m. Otherwise, these orders will be postponed to the next decision epoch $dt^{(k+1)}$. Consequently, the conventional delivery problem is transformed to a new research problem – identifying an optimal policy in partitioning delivery communities to jointly minimize vehicle's serving area, number of vehicles and delivery delay.

3.3 The temporal and spatial policy

In this study, we present two alternatives to generate dynamic policies to tackle this new problem variant in e-commerce last-mile delivery. The practical rationale of formulating two decision policies instead of identifying a single optimal policy to solve the specified problem is that the decision process discussed in Section 3.2 includes a vast number of possibilities. If a Markov decision process model and (approximate) dynamic programming are applied to solve the corresponding Bellman equation, the solution iteration time and computational power would be huge to obtain a merely near-optimal solution, which is not justifiable considering the need to generate a feasible community-based delivery solution in a time-sensitive manner. By developing two alternatives, namely Temporal and Spatial policies, a spatiotemporal relativity study is performed for assessing the interdependencies of the temporal delivery delay and spatial community resizing. The applicability of these policies under varying e-commerce delivery conditions, including order arrival rate, number of communities partitioned, vehicle departure time, etc., is systematically evaluated.

3.3.1 Generating the serving regions of a community

Prior to presenting our Temporal and Spatial delivery policies, we firstly develop a method to generate serving regions of delivery communities based on the cells. Referring to a block design problem proposed by Huang et al. (2018), cells are assigned into several blocks to minimize the number of blocks while satisfying the capacity and cell connectivity constraints. For the $|N|$ cells, a graph $G=\{F,E\}$ is associated where node set F represents all cells in N and arc set E represents pairs of cells $n,n'\in N$. The optimization model is formulated as follows:

$$\min_{x_{p,q},y_q} \sum_{q\in F} y_q \tag{1}$$

Subject to:

$$\sum_{p\in F} d_p \cdot x_{p,q} \le Q\cdot y_q,\ \forall q\in F \tag{2}$$

$$\sum_{q\in F} x_{p,q} = 1,\ \forall p\in F \tag{3}$$

$$x_{q,q} = y_q,\ \forall q\in F \tag{4}$$

$$\text{connectivity constraints} \tag{5}$$

$$y_q \in \{0,1\},\ \forall q\in F \tag{6}$$

$$x_{p,q} \in \{0,1\},\ \forall p,q\in F \tag{7}$$

$x_{p,q}$ and y_q are 0 and 1 variables, in which $y_q=1$ means that cell q represents block q, $y_q=0$, otherwise; $x_{p,q}=1$ indicates that cell p belongs to block q, $x_{p,q}=0$, otherwise. d_p is the accumulated order demand in cell p. Constraint (2) guarantees that the total demand in any block does not exceed the designed capacity. Constraint (3) ensures that each cell must belong to one exact block. Constraint (4) means that the cell selected to represent block q must be belonged to block q. Constraints (5) is the model to ensure the cell connectivity in each block, which can be formulated as follows:

$$\sum_{(g,h)\in E} f_{g,h}^q - \sum_{(h,g)\in E} f_{h,g}^q = \begin{cases} \sum_{l\neq q} x_{l,q}, & g=q \\ -x_{g,q}, & \text{otherwise} \end{cases} \quad \forall g,q\in F \tag{8}$$

$$f_{g,h}^q \le \left(|F|-1\right)\cdot x_{g,q},\ \ \forall (g,h)\in E,\ q\in F \tag{9}$$

$$
\begin{aligned}
& f_{g,h}^q \le \left(|F|-1\right)\cdot x_{h,q}, \ \ \forall (g,h)\in E,\ q\in F \\
& f_{g,h}^q \ge 0, \ \ \forall (g,h)\in E,\ q\in F
\end{aligned}
\tag{10}
$$

where $f_{g,h}^q$ is the flow variable representing the flow in block q. The formulation on the cell connectivity constraints is inspired by multi-commodity flow concepts originated from Huang et al. (2018). Constraint (8) ensures flow conservation and constraints (9) and (10) guarantee that a flow only travels cells in the same block.

Motivated by the block design problem, we propose a serving region generation problem (SRGP). Given a specific number of communities $|M^{(k)}|$ ($|M^{(k)}|\le|N|$) and the demand in each cell d_n ($n\in N$), SRGP seeks to assign these $|N|$ cells into $|M^{(k)}|$ community serving regions to maximize the total cell compactness in all regions while satisfying the demand and cell connectivity constraints in each serving region. The mathematical model of SRGP is described as follows:

$$
\max \sum_{m\in M^{(k)}} F_{comp}\left(N_m^{(k)},\mathbf{\Pi}\right) \tag{11}
$$

Subject to:

$$
\sum_{m\in M^{(k)}} x_{n,m} = 1, \ \forall n\in N \tag{12}
$$

$$
\left|\sum_{n\in N} d_n / |M^{(k)}| - \sum_{n\in N_m^{(k)}} x_{n,m}\cdot d_n\right| \le \delta\cdot\sum_{n\in N} d_n / |M^{(k)}|, \ \forall m\in M^{(k)} \tag{13}
$$

$$
\text{connectivity constraints} \tag{14}
$$

$$
N_m^{(k)} = \left\{n\in N : x_{n,m}=1\right\}, \ \forall m\in M^{(k)} \tag{15}
$$

$$
x_{n,m}\in\{0,1\} \ \forall n\in N, m\in M^{(k)} \tag{16}
$$

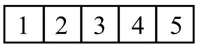
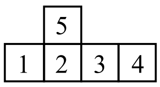
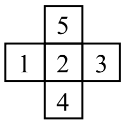
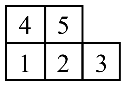
Compared to the block design problem, SRGP only has one type of variable, i.e., $x_{n,m}$, which is binary variable. $x_{n,m}=1$ indicates cell n belongs to the delivery community m, $x_{n,m}=0$, otherwise. Since the number of communities is predetermined, the objective of SRGP is to maximize the total compactness value, i.e., $F_{comp}\left(N_m^{(k)},\mathbf{\Pi}\right)$, in all communities, where $\mathbf{\Pi}$ is the adjacent matrix of all cells. We will discuss this function in detail in the next paragraph. Constraint (12) ensures that each cell will be assigned to one exact community. Constraint (13) guarantees that the accumulated demand of each community are balanced, where N_m is the cell set of community m, d_n ($d_n=\sum_{i\in\{i\in I^{(k)}:(x_i,y_i)\in cr_n\}} w_i$) is the accumulated order demand in cell n and $\delta\in[0,1]$ is a balance coefficient. Constraints (14) is introduced to ensure the cell connectivity in each community. In the work of Huang et al. (2018), the connectivity constraints are formulated according to the concepts of multi-commodity flow. In our SRGP model, since there is no additional variable to designate the representing cell in a community, the above connectivity constraints formulations cannot be directly applied here. Therefore, instead, we used a depth first search (DFS) algorithm to estimate if the cells in a community are connected based on their adjacent matrix $\mathbf{\Pi}$.

As mentioned in Section 3, the serving region in a delivery community shall be generated with a high degree of compactness. This goal is achieved by maximizing the cell compactness in each community, an objective function in SRGP. In the literature, there are various mathematical definitions of compactness (Rossit et al., 2019). Most of them, however, are defined to express the compactness of a set of discrete points. In this study, we calculate the cell compactness of a serving region using the following equation:

$$F_{comp}\left(N_m^{(k)},\mathbf{\Pi}\right)=\sum_{n\in N_m^{(k)}} N_{adj}^2\left(n,N_m^{(k)},\mathbf{\Pi}\right) \tag{17}$$

where $N_{adj}(n,N_m,\mathbf{\Pi})$ is the adjacent cell number of cell n, which can be obtained on the basis of cell set $N_m^{(k)}$ and cell adjacent matrix $\mathbf{\Pi}$. Table 2 demonstrates examples of the cell compactness of four types of serving regions with 5 cells. Regions with cells more adjacent to each other give a higher compactness value computed by Eq. (17).

Table 2. Examples of the cell compactness of four types of serving regions with 5 cells

	Region 1	Region 2	Region 3	Region 4
				
$N_{adj}(1,N_m,\mathbf{\Pi})$	1	1	4	2
$N_{adj}(2,N_m,\mathbf{\Pi})$	2	3	1	3
$N_{adj}(3,N_m,\mathbf{\Pi})$	2	2	1	1
$N_{adj}(4,N_m,\mathbf{\Pi})$	2	1	1	2
$N_{adj}(5,N_m,\mathbf{\Pi})$	1	1	1	2
Compactness	14	16	20	22

3.3.2 Temporal policy (the T-policy)

The T-policy provides flexibility in terms of the delivery delay of a vehicle at the depot. The solution generation mechanism of this policy – depart once a vehicle is fully loaded, is fundamentally different from the general rule of thumb of dynamic vehicle dispatching in the sense that the serving region of a vehicle is determined in real time based on the actual geographical distribution of arrived orders.

In the T-policy, the serving area of each community is fixed, so that the numbers of communities at different decision epoch, i.e., $|M^{(k)}|$, are identical, and we let $|M^{(k)}|=|M|$. $|M|$ is a turnable parameter in the T-policy which influences the vehicle dispatching frequency. Once $|M|$ is determined, the serving region of reach community ($SR_m^{(k)},\ m \in M$) can be obtained by solving the SRGP based on the expected daily cell demand $\bar{d}_n$. At the beginning of each decision epoch $dt^{(k)}$, the arrived delivery order set $I^{(k)}$ is partitioned into $|M|$ groups, i.e., $\{I_1^{(k)}, I_2^{(k)}, ..., I_{|M|}^{(k)}\}$, according to their delivery locations. Subsequently, the orders in each delivery community and the available vehicles set $J^{(k)}$ are checked. To determine the served orders $\bar{I}_m^{(k)}$ and delayed orders $R_m^{(k)}$ in delivery community m at $dt^{(k)}$, the following situations are considered:

(i) If the accumulated weight in $I_m^{(k)}$ is less than vehicle capacity threshold $\eta \cdot Q$ ($\eta \in (0,1)$), all orders in $I_m^{(k)}$ will be marked as $R_m^{(k)}$ and intentially delayed to the next decision epoch;

(ii) If the accumulated weight in $I_m^{(k)}$ is within the range of $\eta \cdot Q$ and Q, all orders in $I_m^{(k)}$ will be denoted as $\bar{I}_m^{(k)}$ and served by a vehicle in $J^{(k)}$. $R_m^{(k)}$ is an empty set in this situation.

(iii) If the accumulated weight in $I_m^{(k)}$ exceeds vehicle capacity Q, orders in $I_m^{(k)}$ will be divided into two parts, including $\bar{I}_m^{(k)}$ and $R_m^{(k)}$.

The order set confirmed for dispatching at decision epoch $dt^{(k)}$ is $\bar{I}^{(k)} = \bigcup_{m\in M} \bar{I}_m^{(k)}$; whereas the order set for intended postponement at decision epoch $dt^{(k)}$ is $R^{(k)} = \bigcup_{m \in M} R_m^{(k)}$. $\bar{I}_m^{(k)}$ is determined according to the first-in-first-serve (FIFS) principle until the vehicle capacity is full. Any excess order is denoted as $R_m^{(k)}$. It is noted that if no available vehicle exists in the depot, $\bar{I}_m^{(k)}$ will be identified as a passive postponement order set, which is also denoted as $R_m^{(k)}$. The simulation pseudocode for the T-policy is presented in Fig. 2.

Algorithm 1 (Simulation procedures using the T-policy)
Input: Business time T; Vehicle set J; Vehicle capacity Q; Length of decision cycle Δt; Community number $|M|$; Cell set N; Cell adjacent matrix $\prod$; Expected daily cell demand $\bar{d}_n$ in each cell.
Output: Delivery order $\bar{I}_j^{(k)}$ for each vehicle and $R^{(k)}$ at each decision epoch.

```
Initialization: R^(0) = V_0, J^(1) = {1,2,...,|J|};
Obtain SR_m^(k) by solving SRGP given |M|, N and d̄_n;
K = T / Δt;
for k = 1 to K
    Generate new arrival order set in k th decision cycle I_in^(k);
    I^(k) = I_in^(k) ∪ R^(k-1);
```

for $m=1$ **to** $|M|$
Update $J^{(k)}$ by checking the retunrning time of each vehicle;
Obtain $I_m^{(k)}$ by dividing $I^{(k)}$ based on $SR_m^{(k)}$;
if $J^{(k)} \neq \varnothing$
Dividing $I_m^{(k)}$ into $\overline{I}_m^{(k)}$ and $R_m^{(k)}$ using FIFS principle;
Select a vehicle j in $J^{(k)}$ and asign $\overline{I}_m^{(k)}$ to $\overline{I}_j^{(k)}$;
Compute the retunrning time of vehicle j by solving the traveling salesman problem;
else
$R_m^{(k)} = I_m^{(k)}$;
end if
$R^{(k)} = \bigcup_{m \in M^{(k)}} R_m^{(k)}$;
end for
end for

Fig. 2. Simulation procedures for the Temporal policy

3.3.3 Spatial policy (the S-policy)

In contrast to the T-policy, the S-policy does not allow any delivery delay. In other words, all accumulated orders must be dispatched at each decision epoch $dt^{(k)}$. However, flexible community formulation is possible through merging more nearby cells at $dt^{(k)}$. This solution generation using thiS-policy is different from that of CluVRP in the sense that a community is formulated by merging pre-partitioned cells acorss the entire serving region.

In the S-policy, the departure time of each community is fixed at each decision epoch $dt^{(k)}$. Therefore, in this policy, the length of decision cycle Δt is a ternable parameter which impacts the community merging results. Due to the uncertainty of order arrivals, the number of communities $|M^{(k)}|$ and the corresponding serving regions $SR_m^{(k)}$ are different at each $dt^{(k)}$. The number of delivery communities $|M^{(k)}|$ can be calculated by:

$$|M^{(k)}| = \left\lceil \frac{\sum_{i \in I^{(k)}} w_i}{Q/(1+\delta)} \right\rceil \tag{18}$$

Note that a modified vehicle capacity $Q/(1+\delta)$, instead of the original capacity Q, is used to compute the required number of communities (also the vehicles). By so doing, when the constraint (11) is satisfied, none of the vehicles are overloaded. At each $dt^{(k)}$, $SR_m^{(k)}$ is reorganized by solving the SRGP based on real-time accumulated demand $d_n^{(k)}$ rather than $\overline{d}_n$ in each cell. Subsequently, the served order set $\overline{I}_m^{(k)}$ and delayed order set $R_m^{(k)}$ in community m is determined based on the following rules:

(i) If the accumulate demand in $I_m^{(k)}$ is smaller than Q, all orders in $I_m^{(k)}$ will be denoted as $\overline{I}_m^{(k)}$;

(ii) Otherwise orders in $I_m^{(k)}$ will be divided into two part, i.e., $\bar{I}_m^{(k)}$ and $R_m^{(k)}$.

Similar to the T-policy, $\bar{I}_m^{(k)}$ is determined based on the first-in-first-serve (FIFS) principle until the vehicle capacity is full. Excess orders are classified as $R_m^{(k)}$. Nevertheless, in S-policy, no order will be delayed intentionally unless the available vehicles are insufficient ($|J^{(k)}|<|M^{(k)}|$). For each $\bar{I}_m^{(k)}$, an available vehicle from $J^{(k)}$ will be assigned to fulfill it. $\bar{I}_m^{(k)}$ will be denoted as $R_m^{(k)}$ if no vehicle is available. The order set unfulfilled at the current $dt^{(k)}$, i.e., $R^{(k)} = \bigcup_m R_m^{(k)}$, will be postponed to the next epoch $dt^{(k+1)}$. The simulation pseudocode by using SDP is presented in Fig. 3.

Algorithm 2 (Simulation procedures using the S-policy)
Input: Business time T; Number of vehicles J; Vehicle capacity Q; Length of decision cycle Δt; Cell set N; Cell adjacent matrix $\prod$.
Output: Delivery order $\bar{I}_j^{(k)}$ for each vehicle and $R^{(k)}$ at each decision epoch.

Initialization: $R^{(0)} = V_0$, $J^{(1)} = \{1,2,...,|J|\}$;
$K = T/\Delta t$;
for $k=1$ **to** K
  Calculate $|M^{(k)}|$ $d_n^{(k)}$ based on real-time order accumulation;
  Obtain $SR_m^{(k)}$ by solve SRGP given $|M^{(k)}|$, N and $d_n^{(k)}$;
  Generate new arrival order set in the decision cycle $I_{in}^{(k)}$;
  $I^{(k)} = I_{in}^{(k)} \cup R^{(k-1)}$;
  for $m=1$ **to** $|M^{(k)}|$
    Update $J^{(k)}$ by checking the retunrning time of each vehicle;
    Obtain $I_m^{(k)}$ by dividing $I^{(k)}$ based on $SR_m^{(k)}$;
    if $J^{(k)} \neq \varnothing$
      Dividing $I_m^{(k)}$ into $\bar{I}_m^{(k)}$ and $R_m^{(k)}$ based on FIFS principle;
      Select a vehicle j in $J^{(k)}$ and assign $\bar{I}_m^{(k)}$ to $\bar{I}_j^{(k)}$;
      Compute the retunrning time of vehicle j by solving the traveling salesman problem;
    else
      $R_m^{(k)} = I_m^{(k)}$;
    end if
    $R^{(k)} = \bigcup_{m \in M^{(k)}} R_m^{(k)}$;
  end for
end for

Fig. 3. Simulation procedures for the Spatial policy

3.4 Generate routes after obtaining order set from Temporal and Spatial policies

It should be highlighted that the T and S policies are developed with an aim to assess the suitability of a community-based solution format for solving dynamic vehicle dispatching problems in e-commerce era, which eliminates the need to optimize the visiting sequence of delivery nodes. This assessment is made with respect to four key performance indices to be introduced in Section 4.3, which include traveling distance of a route. Therefore, a route for each vehicle is identified only for the purpose of computing their traveling distance for performance evaluations, which is generated by solving the following traveling salesman problem (Lawler, 1985):

$$\min D_j^{(k)} = \sum_{p\in \bar{I}_i^{(k)}\cup\{0\}} \sum_{q\in \bar{I}_i^{(k)}\cup\{0\}} dist\left(p,q\right)\cdot x_{p,q} \tag{19}$$

subject to:

$$\sum_{p\in \bar{I}_i^{(k)}\cup\{0\}} x_{p,q} = 1,\ \forall q \in \bar{I}_j^{(k)} \cup \{0\} \tag{20}$$

$$\sum_{q\in \bar{I}_i^{(k)}\cup\{0\}} x_{p,q} = 1,\ \forall p \in \bar{I}_j^{(k)} \cup \{0\} \tag{21}$$

$$u_p - u_q + N_j^{(k)} \cdot x_{p,q} \le N_j^{(k)} - 1,\ 1 \le p \ne q \le N_j^{(k)} \tag{22}$$

$$N_j^{(k)} = | \bar{I}_j^{(k)} \cup \{0\} | \tag{23}$$

$$x_{p,q} \in \{0,1\},\ u_{p,q} \in \mathbf{Z} \tag{24}$$

where $x_{p,q}$ is a decision variable; $x_{p,q} = 1$ if the courier needs to travel from location p to location q, 0 otherwise. $\{0\}$ is the indication of the depot. Constraints (20) and (21) ensure that each location in order set $\bar{I}_j^{(k)} \cup \{0\}$ be visited only once. Constraints (22) and (23) guarantee all locations in $\bar{I}_j^{(k)} \cup \{0\}$ are visited by only one vehicle.

4. Spatio-temporal Relativity Study

A series of simulated experiments is performed to evaluate the performance of the proposed Temporal and Spatial policy in managing dynamic arrivals of e-commerce delivery orders in a distribution centre. The simulated delivery environment and identified parameters are based on real business settings of a third-party logistics service provider based in the mainland China. In the proposed two policies, the SRGP is solved by a classic genetic algorithm (GA) (Holland, 1992) and the TSP is solved by a classic Tabu search algorithm (Glover & Laguna, 1998). All numerical experiments are conducted in MATLAB R2020b on a personal server using an Intel(R) Xeon(R) E5-2670 v2 CPU. Following a discussion of the data sets and parameter setting, the performance of the Temporal and Spatial policy is compared in terms of four key performance indicators (KPIs) introduced in this section, namely the average route compactness, postponement time and traveling distance of an order, and the number of unused vehicles at the depot.

4.1 Settings for the test instances

The instance I1 considers an e-commerce distribution hub serving 5 km by 5 km geographical area in Beijing from 8:00am to 18:00pm (600 mins). As mentioned in Section 3, the community serving region is partitioned by combining basic cells in a geographical area. Initially, as shown in Fig. 4 (a), the serving region is partitioned into 33 non-uniform cells according to its geographical characteristics, such as roads and rivers. Subsequently, we can calculate the probability of orders locating in each cell (see Fig. 4 (b)) according to the historical arrival figures of e-commerce orders within the serving region. In actual delivery environment, we assume 30 homogeneous delivery vehicles with 600 kg capacity are available for handling daily delivery operations. Taking traveling speed in urban areas, waiting and serving time at each delivery node into considerations, we determine a vehicle to have an average constant speed of 10 km per hour. The order arrival is regarded as a homogeneous Poisson process with a constant arrival rate λ. Considering that the average number of orders successfully delivered can reach to 330 per square kilometer per day in Beijing (Huang et al., 2018), λ is set to be 1/0.48 (low demand). Based on instance I1, we further introduce three instances I2, I3 and I4 by altering the λ to 1/0.24 (normal demand), 1/0.12 (high demand) and 1/0.06 (peak demand). Table 3 summarizes the parameter settings of the spatio-temporal relativity experiments.

(a) (b)

Fig. 4. Cell districting: (a) Cells partitioning and (b) their probability of order arrival in a cell

Table 3. A summary of the parameters for simulation experiments

Parameters	Configuration
Vehicle availability	
Number of delivery vehicles	30
Maximum capacity of a vehicle	600 kg
Average speed of a vehicle *(including vehicle waiting at delivery nodes)*	10 km per hour
Delivery demand	
Demand scenarios	Low demand (I1) – $\lambda = 1/0.48$ Normal demand (I2) – $\lambda = 1/0.24$ High demand (I3) – $\lambda = 1/0.12$ Peak demand (I4) – $\lambda = 1/0.06$

Number of initial orders in each day	200
Weight of a delivery order	Gamma distribution (20,1)
Dynamic policies	
Fixed no. of communities for Temporal Policy	2, 4, 8, 12, 16, 20 24
Decision epoch for Temporal Policy	Every 2 minutes
Decision epoch for Spatial Policy	Every 15/30/50/60/100/120/150 minutes
Balance coefficient	0.2
Simulation experiments	
Service time length and number of simulations	600 minutes for 20 times

- *Parameter setting for the Temporal Policy*

The 33 non-uniform cells partitioned is used to generate the probabilistic arrivals of orders within the serving region. To evaluate the performance of the Temporal policy, the serving region of delivery community is determined in advance by solving the SRGP, as depicted in Fig. 5. To evaluate the effect of community partitioning towards the solution quality, the number of communities for the Temporal policy across the simulation period, i.e. 20 days, each day 10 hours in operations, is fixed at 2, 4, 8, 12, 16, 20 and 24. Under the Temporal Policy, further community size adjustment is not allowed in the Temporal policy. Only dynamic departure times are permitted. This setting is used to evaluate the performance of dynamic departure policy with different community sizes under changing demand scenarios.

Due to the difference between the proposed policies in terms of their mechanism in solution iterations, decision epochs should have different implications for the Temporal and Spatial policies. In the Temporal policy, the area of communities is predetermined. Decision makers need to check if the vehicle departure condition is satisfied for each community. Nevertheless, frequently checking the orders accumulated in each community is unnecessary because the number of orders cannot vary violently within a short time period. Therefore, the decision epoch for the temporal policy is set to be 2 minutes. This suggests that the algorithm of the temporal policy will determine if the pending delivery orders consolidated in each community fulfill the dispatching criteria in every 2 minutes. Any community that has consolidated enough orders will be dispatched immediately at that decision epoch regardless of the status of other communities.

- *Parameter setting for the Spatial Policy*

In contrast to the Temporal policy, the Spatial policy, while having a fixed departure time, allows flexible size adjustment of community serving regions at each decision epoch by periodically solving the SRGP. Therefore, this setting examines solution quality when the number and size of communities are continuously changing, with the depot serving area being initially partitioned into 33 non-uniform cells as depicted in Fig. 4. For decision epoch configuration, the Spatial policy has a range of decision epochs for performance evaluation – 15, 30, 50, 60, 100, 120 and 150 minutes. Decision epochs in Spatial policy does not only govern the time in which the algorithm of the Spatial policy is run, but also the departure time of each community. Under this policy, the departure times are fixed at the start of each decision epoch. Therefore, if a community did not consolidate enough orders, the policy allows a community to merge with its neighbour. In the contrary, community splitting would take place should a community has received an excessive number of orders that a vehicle could fulfill.

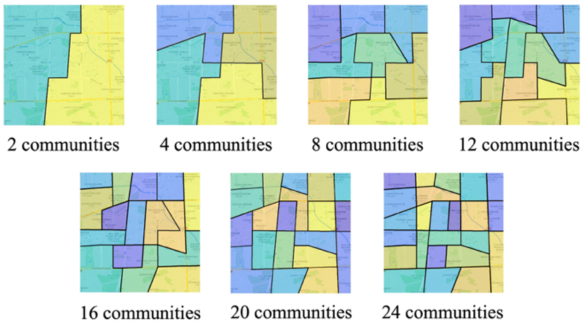


Fig. 5. Pre-determined community partitioning of the serving region for the Temporal policy

4.2 Static policy for benchmarking – static service region and departure time

In this section, a static vehicle dispatching policy (Static policy) based on the conventional delivery strategy is introduced. Subsequently, the static policy is benchmarked with our T and S-policy to validate the performance of our proposed CL delivery strategy. Under the static policy, both vehicle service region and departure cycle are predetermined. Given the static departure cycle Δt^{static}, the static service region number $|M^{static}|$ is computed by:

$$|M^{static}| = \frac{\bar{w} \cdot \lambda \cdot \Delta t^{static}}{Q \cdot (1+\delta)} \tag{25}$$

where $\bar{w}$ is the expected demand of e-commerce orders. The static policy does not vary the number of service region and vehicle departure cycle. Therefore, we identify the number of service regions based on the expected total demand during one vehicle departure cycle. After determining Δt^{static} and $|M^{static}|$, the service region of each vehicle, i.e., SR_m^{static} ($m \in M^{static}$), can be obtained by solving the SRGP. With Δt^{static} and SR_m^{static} ($m \in M^{static}$), vehicles are dispatched according to the rules described in Fig. 6. Different values of Δt^{static} and $|M^{static}|$ are identified to reflect the order arrival volume under different demand levels, as depicted in Table 4. It shall be noted that some values of Δt^{static} cannot be achieved in high and peak demand levels, such as 100 minutes, because no feasible service region exists in these scenarios.

Algorithm 3 (Simulation procedures using the static policy)

Input: Business time T; Number of vehicles J; Vehicle capacity Q; Vehicle departure cycle Δt^{static}; Vehicle service regions SR_m^{static} ($m \in M^{static}$).

Output: Delivery order $\overline{I}_j^{(k)}$ for each vehicle and $R^{(k)}$ at each decision epoch.

Initialization: $R^{(0)} = V_0$, $J^{(1)} = \{1, 2, ..., |J|\}$;
$K = T / \Delta t^{static}$;
for $k = 1$ **to** K
    Generate new arrival order set in k the decision cycle $I_{in}^{(k)}$;
    $I^{(k)} = I_{in}^{(k)} \cup R^{(k-1)}$;
    for $m = 1$ **to** $|M^{static}|$
        Update $J^{(k)}$ by checking the retunrning time of each vehicle;
        Obtain $I_m^{(k)}$ by dividing $I^{(k)}$ based on SR_m^{static};
        if $J^{(k)} \neq \varnothing$
            Dividing $I_m^{(k)}$ into $\overline{I}_m^{(k)}$ and $R_m^{(k)}$ based on FIFS principle;
            Select a vehicle j in $J^{(k)}$ and assign $\overline{I}_m^{(k)}$ to $\overline{I}_j^{(k)}$;
            Compute the retunrning time of vehicle j by solving the traveling salesman problem;
        else
            $R_m^{(k)} = I_m^{(k)}$;
        end if
        $R^{(k)} = \bigcup_{m \in M^{(k)}} R_m^{(k)}$;
    end for
end for

Fig. 6. Simulation procedures for the static policy

Table 4. Parameter settings for the static policy

	Low demand	Mid demand	High demand	Peak demand
Δt^{static}	15, 30, 50, 60, 100, 120	15, 30, 50, 60, 100, 120	15, 30, 50, 60	15, 30
$\lvert M^{static} \rvert$	2, 3, 5, 5, 6, 10	5, 9, 10, 17, 20, 25	5, 10, 17, 20	10, 20

4.3 Policy performance measurement

To evaluate the effectiveness of the proposed policies and identify appropriate parameter combinations, we propose four key performance indicators (KPIs) to evaluate the quality of solutions, namely average route compactness (ARC), postponement time (APT) and traveling distance of an order (ATD), and the number of unused vehicles at the depot (UVN).

The first KPI is the order average route compactness (ARC). The route compactness is an intuitive concept and can be defined unequivocally (Rossit et al. 2019). Generally, the higher proximity amongst the destinations of the community, the more compact this route is. In this study, we use the definition proposed in Poot et al. (2002) to measure route compactness, which is formulated as:

$$ARC = \frac{\sum_{k} \sum_{m \in M^{(k)}} \sum_{i \in \bar{I}_m^{(k)}} dist\left(i, c_m^{(k)}\right)}{\sum_{k} \sum_{m \in M^{(k)}} \left|\bar{I}_m^{(k)}\right|} \tag{26}$$

where *dist(a,b)* is a function to calculate the Euclidean distance between point *a* and point *b*. $c_m^{(k)}$ is the geometric centre of the location in the order $\bar{I}_m^{(k)}$. The smaller this value, the more compact the solution is.

The second KPI is the average postponement time of an order (APT), which is a measure of the duration of time an order is pending at the depot for actual delivery. It can be formulated by:

$$APT = \frac{\sum_{k} \sum_{m \in M^{(k)}} \sum_{i \in \bar{I}_m^{(k)}} \left(dt^{(k)} - t_i\right)}{\sum_{k} \sum_{m \in M^{(k)}} \left|\bar{I}_m^{(k)}\right|} \tag{27}$$

The third KPI is the number of unused vehicles (UVN) at the $dt^{(K)}$, which is introduced to describe the usage of available vehicles in the depot. It can be calculated by:

$$UVN = \left| J / \bigcup_k J^{(k)} \right| \tag{28}$$

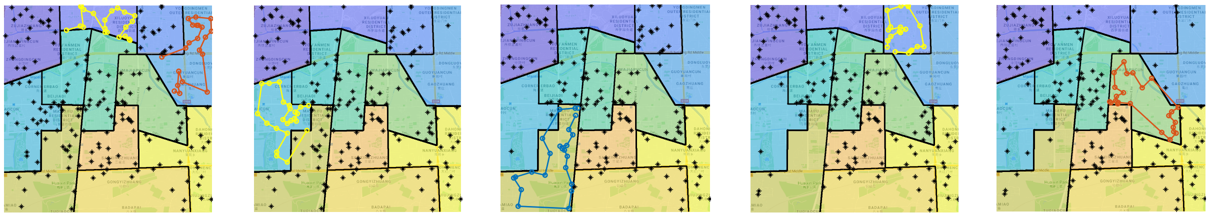
The fourth KPI is the average traveling distance of an order (ATD), which is used to describe vehicles travel efficiency, and is computed by:

$$ATD = \frac{\sum_{k} \sum_{m \in M^{(k)}} D\left(\bar{I}_m^{(k)}\right)}{\sum_{k} \sum_{m \in M^{(k)}} \left|\bar{I}_m^{(k)}\right|} \tag{29}$$

where $D\left(\bar{I}_m^{(k)}\right)$ is the minimum distance for the vehicle *j* to deliver all orders in $\bar{I}_m^{(k)}$. This figure can be obtained by solving traveling salesman problem. Given a specific instance, the S, T are evaluated for 20 times to obtain a mean value of above four KPIs.

4.4 Performance evaluation for the Temporal policy under various demand scenarios

The simulation experiment for the T-policy has decision epochs fixed at a 2-minute interval. An illustrative example of solutions generated by using the Temporal Policy is depicted in Fig. 7. During the decision epochs from 8:02am to 8:20am, only one community has consolidated enough orders for dispatching from the distribution hub. Applying the Temporal policy for checking each community at a two-minute interval, the hub continues the consolidation of orders until 8:12am, 8:14am and 8:16am to dispatch orders from three different communities. This example demonstrates that a community dispatching solution is updated at each decision epoch but is not necessarily be finalized for real dispatching. Real dispatching occurs only when the community has consolidated an adequate number of orders.



At 8:02 am | At 8:12 am | At 8:14 am | At 8:16 am | At 8:20 am

Fig. 7. Examples of community logistics solution generation using Temporal Policy at a 2-minute decision epoch with 12 fixed communities under normal demand period

- **APT performance** – For the T-policy, having a fixed number of communities partitioned and dynamic departure times of each community, simulation experiments of 2, 4, 8, 12, 16, 20 and 24 fixed communities were conducted for assessing the T-policy performance under four demand scenarios. As summarized in Fig. 8, the lowest average postponement (waiting) time of each order is achieved by partitioning the region into the least number of communities, i.e. 2 communities. With a smaller number of communities partitioned across the region, each community shares a larger portion of the geographical area. Therefore, it makes sense that a smaller number of communities would enable each community to consolidate newly arrived delivery requests at a higher rate. This in turn reduces the waiting time of the orders already pending in the order consolidation pool for actual delivery. Generally, the plot in Fig. 8 revealing the relationship between APT and the number of communities demonstrates that partitioning the region into more communities results in a delay of vehicle’s departure. The effect of community partitioning towards delivery postponement is more apparent in low demand scenario. This can be explained by the fact that a low order arrival rate leads to the difficulty in receiving an adequate number of order requests within a spatially compact region. Over-partitioning of communities in lower demand scenarios would further worsen this situation, as the plot indicates that the rate of increase in APT as the number of communities increases is much higher than that during higher demand periods. Interestingly, the APT does not increases with a larger number of communities in peak period. A minimum APT is obtained when there are 16 communities partitioned. The APT in peak demand scenario remains steadily at below 20 minutes. Therefore, the performance of the T-policy in terms of APT suggests that practitioners ought to partition their distribution hub’s serving region into more communities when their order arrival rate is high.

- **ARC and ATD performance** – In terms of the average route compactness and traveling distance of an order, both curves representing their relationships with the number of communities partitioned are similar. It is worth noting that the rate of change of ARC and ATD as more communities are partitioned is almost identical in low, normal and high demand scenarios. This suggests that pre-determining the number of communities within a region would largely limit the route compactness and traveling distance, no matter how the order arrival rate fluctuates. It is until the distribution hub is facing a specific circumstance or events, such as annual shopping festivals, should the ARC and ATD become much lower when a smaller number of communities are partitioned. In short, both ARC and ATD decreases with more communities being partitioned. However, the effect of community partitioning become less significant when communities are partitioned from 16 to 24. Therefore, undertaking experimental studies towards ARC and APT enable practitioners to identify the degree of community partitioning that yield greater benefits.

- **UVN performance** – Under the T-policy, the number of unused vehicles remains high during low and normal demand scenarios. There is no clear sign of how community partitioning is affecting the number of unused vehicles as it remains steady when different number of communities are partitioned in each demand period. This figure suggests that, at the level of resource specified in this experiment, e.g. number of available vehicles and capacity of each vehicles, the T-policy would allow practitioners to save more than 20 and 15 vehicles respectively during low and normal demand scenarios. As for high demand scenarios, partitioning the region into more communities would improve the utilization of available vehicles. In other words, a lower degree of community partitioning requires less vehicles.

In summary, performance evaluations towards the T-policy in terms of the four KPIs provide a sharp indication that the degree of community partitioning shall govern a range of service attributes, from the duration of order delay, route compactness and traveling distance of a trip, to utilization of available resources. After all, practitioners can prioritize the relative importance of these attributes so as to identify an optimal number of communities, if the T-policy is to be deployed.

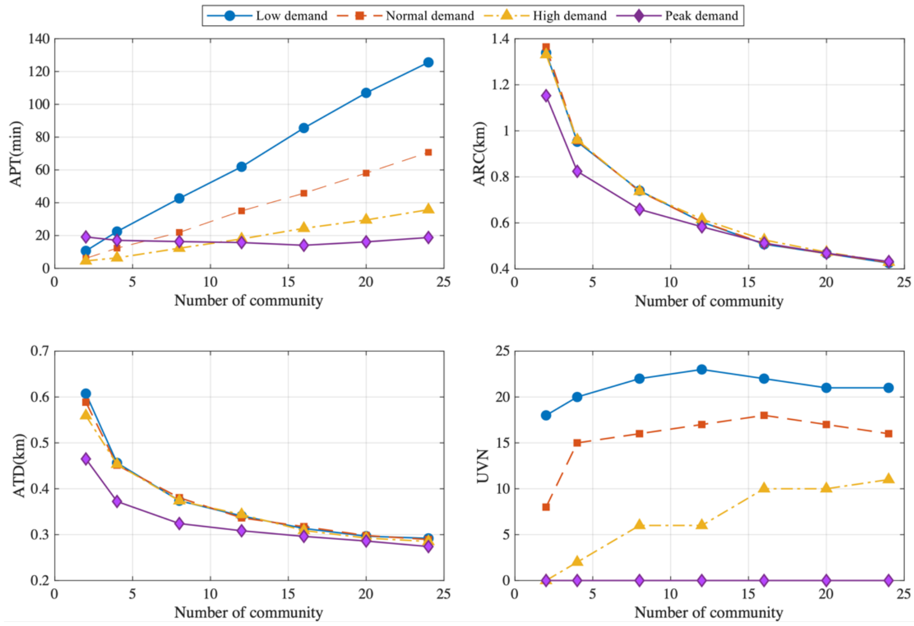


Fig. 8. Solution quality using the T-policy

4.5 Performance evaluation for the Spatial policy under various demand scenarios

The simulation experiment for the S-policy has decision epochs fixed at 15, 30, 50, 60, 100, 120, 150 minutes. At each epoch, the initial 33 non-uniform cells are are allowed to merge in order to utilize available capacity of delivery vehicle at each epoch for departure. An illustrative example is given in Fig. 9 to demonstrate how solution is generated using the S-policy at a 30-minute decision epoch under normal demand. With 200 initial orders in each working day, the first decision epoch, i.e. at 8:30am, requires a total of 11 communities to fulfill the orders. As most of the orders have been dispatching at 8:30 am, only 5 communities are formulated at the next decision epoch. Therefore, the introduction of a range of decision epochs from 15 to 120 minutes shall provide essential indications towards the selection of decision epoch under varying demand scenarios. For ease of further discussion, these fixed decision epochs can be identified as the "maximum allowable duration for communities to consolidate delivery requests". This definition is used to discuss the performance of the S-policy below. Fig. 10 reveals the performance of the S-policy with respect to APT, ARC, ATD and UVN.

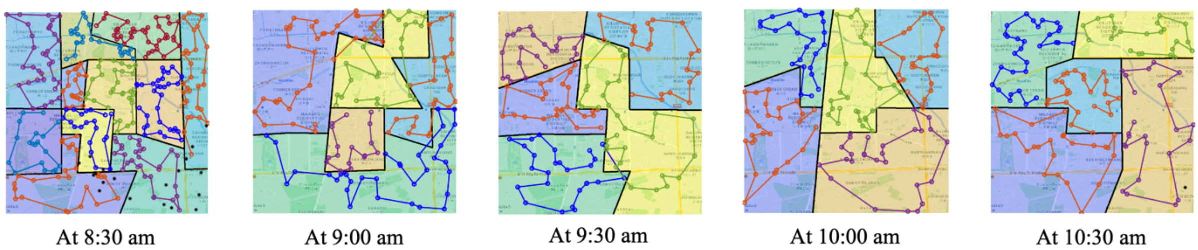


Fig. 9. Examples of community logistics solution generation using Spatial Policy at a 30-minute decision epoch and normal demand period

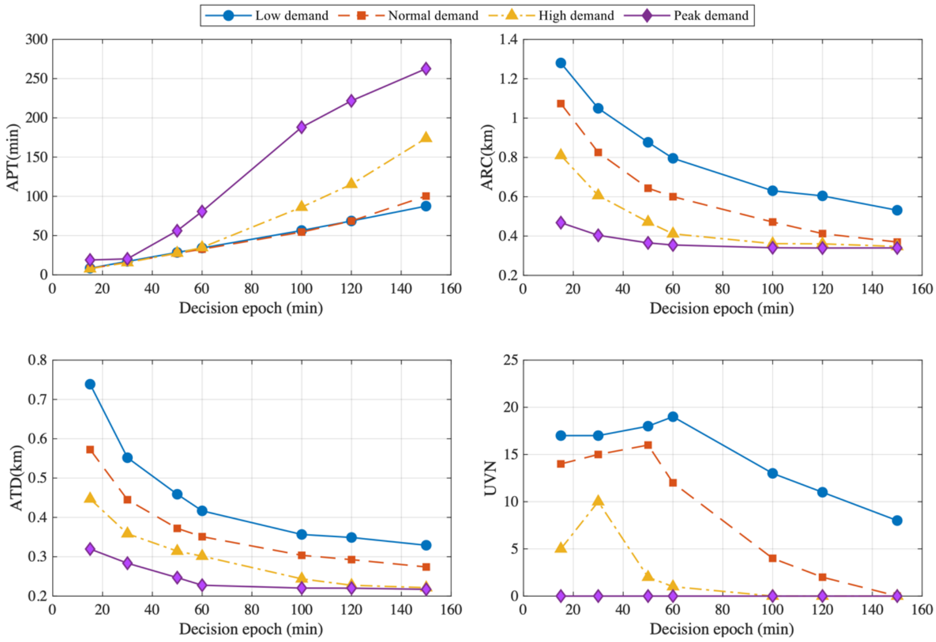


Fig. 10. Solution quality using the S-policy

- **APT performance** – There is an increase in the average duration of delay (APT) by allowing communities to consolidate delivery orders for a longer period. Nevertheless, the relationship tends to be linear only during low and normal demand scenarios. In high demand period, the APT starts to increase at a high rate if the decision epoch is set to be 60 minutes or above. This figure suggests that, when receiving a large number of orders, allowing communities to consolidate more than 60 minutes is unjustifiable. Communities does not require more than 60 minutes to consolidate the incoming orders. Although this figure might only apply to the current resource level and demand scenarios determined in this simulation, results of this experiment indicate the need for practitioners to identify their order arrival pattern and resource level, prior to deciding the timing of a decision epoch, i.e. the fixed departure times of each vehicle serving their respective community, should the S-policy be deployed in real practice.

- **ARC and ATD performance** – The S-policy has a decreasing ARC and ATD as the maximum allowable order consolidation duration increases. This makes sense because giving communities more time for order consolidation in turn suggests that communities need not to merge. However, the effect of relaxing the duration of order consolidation from 15 to 60 minutes is more significant than from 60 minutes onwards. Taking the APT performance into joint consideration, in which the APT starts to increase at a high rate if the decision epoch is set to be 60 minutes or above, simulation results indicate that a decision epoch of 60 minutes or less is more justifiable. Such setting would yield greater benefits in terms of both route compactness, traveling distance and order delivery delay as a short duration of intended postponement of order delivery would have saved a significant traveling distance and route compactness. Moreover, it is noticeable that this policy is more suitable for deployment during higher demand scenarios due to better solution performance in terms of ATD and ARC.

- **UVN performance** – Similar to the T-policy, the S-policy saved a considerable number of vehicles in low, normal and high demand scenarios. From Fig. 10, it is observed that the highest number of unused vehicles is obtained when the decision epoch is fixed at 60, 50 and 30 minutes respectively during low, normal and high demand period. There is noticeably a shift of the optimal decision epoch to achieve the maximum number of unused vehicles, i.e. the minimum number of vehicles used for delivery. This trend indicates that extending the duration of order delay to consolidate more orders at lower order arrival rates is a justifiable option to yield economies of scale. In particular, if a practitioner aims to reduce the number of vehicles to fulfil delivery requests, order consolidation should be made at every 60 minutes during low demand scenario. However, the order consolidation cycle becomes 30 minutes under high demand scenario.

Overall, it depends on the primary objective of a practitioner when deciding the optimal decision epoch. There is no definite answer to the optimal decision epoch in different order arrival rates. However, the simulation study reveals the value of Community Logistics Strategy and how one could examine their community size and departure cycle under current resource levels and order arrival patterns in order to deploy the strategy in real operating environment.

4.6 Comparisons between the Temporal, Spatial and static policies

In section 4.4 and 4.5, it is observed that there is a positive correlation between ARC and ATD, which makes sense due to the fact that improving the degree of route compactness of a trip allows the vehicle to travel for a shorter distance. In this section, therefore, we pick ARC for making further comparisons and evaluations between the T, S and static policy. Such comparisons could inform us of the merits of the T and S-policy. Then, this would justify the appropriateness of the inclusion of "Community Time" as a decision variable into solving districting problems.

- **Identifying the spatial-temporal relativity based on the evaluation of ARC performance at a given APT**

To evaluate the performance of the T and S-policy towards ARC at a given APT, we generated fitting curves of ARC against APT for both policies under each demand scenarios, as shown in Fig. 11. It is worth noting that no obvious correlation between APT and ARC is observed when the demand level is peak. Hence, no fitting curve can be plotted for the peak demand scenario. The fitting curves, which reveal the relationship between ARC and APT, represent the spatial-temporal relativity of how community logistics solution is being more compact as the intended duration of postponement of an order increases.

In general, there is an inverse correlation between ARC and APT. It is mathematically determined as displayed in Fig. 11 and Table 5. In other words, an improvement in (reduction of) ARC is resulted from an increased APT, which is sensible because the intention of allowing orders to be pending in the distribution hub at a longer period of time is to expect the arrivals of more new delivery orders with a high degree of proximity. However, the effect of increasing APT in hopes of improving the ARC varies depending on the demand scenarios and the policy. Comparing the T and S-policy, results shown in Fig. 11 reveal that the S-policy outperforms the T-policy in high demand scenario. At any given APT, the T-policy (the fitting curve in blue) generates a more compact community logistics solution than the S-policy (the fitting curve in orange) during high demand scenario. The S-policy still performs better during normal demand. However, solution quality using the T and S-policy in terms of route compactness becomes very close. Interestingly, there is an intersection point of route compactness of the T and S-policy during low demand scenario. At an average order postponement time of 30 minutes, the T-policy starts to generate a more compact solution, as illustrated in Fig. 11. This suggests that, when practitioners impose a waiting strategy by postponing the delivery of orders at the depot for more than 30 minutes, the T-policy, i.e. flexible merging of communities during a pre-defined APT in order to consolidate orders for delivery in bulk, appears to be a slightly better tactic. However, other than such a specific circumstance in low demand scenario, the S-policy is proved to be a better one.

Comparing the T and S-policy against the static policy, from Fig. 11 it is observed that the fitting curves of both T and S policies are always positioned below the static policy. Theoretically, this implies that at any APT, the ARC from the T ad S policies is always better than that of the static policy. In other words, when orders are intentionally delayed for dispatching, the T and S-policy can always generate a more compact community logistics solution. It is worth reminding that the T-policy has a fixed "community space" but a varying "community departure time", vice versa for the S-policy. This ARC comparison demonstrates that introducing the "community departure time" as one of the decision variables in addition to "community space" is able to yield better dispatching solutions in terms of route compactness.

To determine improvement rate of choosing the S-policy over T-policy, Fig. 12 graphically presents the improvement rate under different scenarios. Results uncover the fact that, except for low demand scenario, the S-policy outperforms the T-policy in terms of route compactness. The performance improvement is more significant: (i) under high demand scenario, and (ii) when orders are having a longer waiting time at distribution centres (higher APT). Furthermore, apart from route compactness, looking into the ATD (average traveling distance) performance of the T and S-policy respectively shown in Fig. 8 and 10, it is observed that the S-policy is able to generate solutions with small traveling distances among each trip. Same as route compactness, a more significant improvement in terms of traveling distance can be obtained during higher demand scenarios.

Informed by these figures, the S-policy, having a stricter control over delivery dispatching time but allowing flexible community sizing adjustment, would improve the quality of vehicle dispatching solutions in terms of both route compactness and traveling distance. This scientifically explains why most of the existing dynamic vehicle districting literatures attempt to generate dispatching solutions by adjusting the serving areas at each decision epoch, but not relaxing the "time" parameter as a decision variable. Nevertheless, benchmarking results between the T and static policy prove that varying departure times for improved order consolidation prior to bulk dispatching could also yield a more compact dispatching solution. This justifies the feasibility and appropriateness of introducing dynamic departure times as another decision variable. Hence, future studies should incorporate both "community departure time" and "space" as the decision variables to generate optimal policy based on real-time demand arrivals.

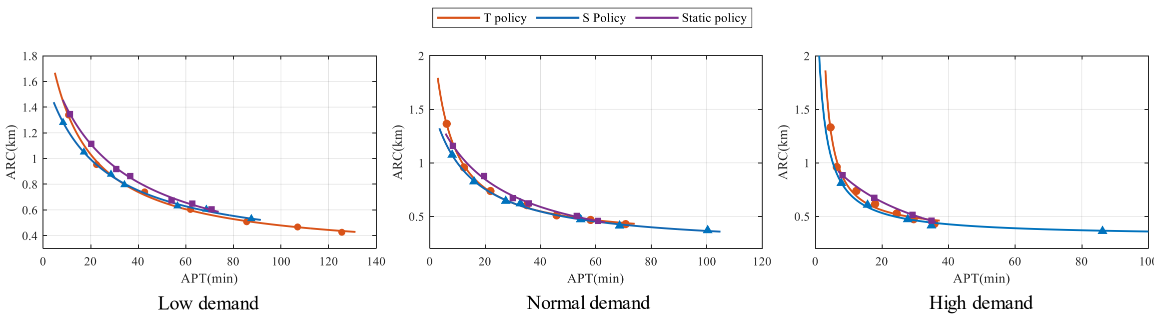


Fig. 11. Fitting curves of ARC against APT using T, S and Benchmark policies under different demand scenarios

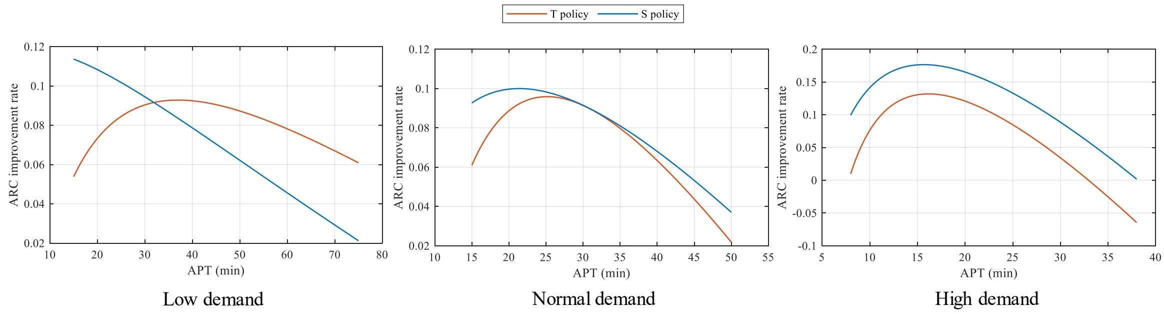


Fig. 12. Route compactness improvement by adopting T and S-policy

Table 5. Relationship between ARC and APT using the T, S and static policy

Policy	Demand scenario		
	Low	Normal	High
T-policy	$ARC = \dfrac{0.25*APT+29.82}{APT+13.69}$	$ARC = \dfrac{0.28*APT+13.71}{APT+5.22}$	$ARC = \dfrac{0.35*APT+4.00}{APT-0.29}$
S-policy	$ARC = \dfrac{0.25*APT+36.41}{APT+21.64}$	$ARC = \dfrac{0.21*APT+20.36}{APT+12.47}$	$ARC = \dfrac{0.31*APT+5.30}{APT+1.70}$
Static	$ARC = \dfrac{0.15*APT+46.93}{APT+24.77}$	$ARC = \dfrac{0.03*APT+36.61}{APT+24.29}$	$ARC = \dfrac{-0.09*APT+31}{APT+26}$

- **Resource management through the evaluation of UVN**

The number of unused vehicles in each demand scenario using the T and S-policy is summarized in Fig. 13. Apart from the peak demand scenario where both policies utilize all available vehicles to fulfill the orders, community logistics solutions generated under the T-policy require less vehicles to perform last-mile delivery operations in low, normal and high demand period. For example, the T-policy would save a minimum of 20 vehicles during low demand scenario, whereas the S-policy could only save no more than 20 vehicles. The results of UVN are more significant in normal and high demand scenarios. This suggests that the S-policy requires a larger delivery fleet to improve route compactness and reduce traveling distance of each vehicle trip. In fact, this KPI informs two tactical implications – resource re-allocation and identification of the optimal number of communities under T-policy or the optimal decision epoch (order postponement duration) under the S-policy. If the S-policy is picked, this comparison suggests that a maximum of 19, 16 and 10 vehicles respectively can be re-allocated during low, normal and high demand scenarios for other delivery tasks. Should a practitioner intend to save the largest number of vehicles for last-mile delivery, this simulation result is able to justify the optimal number of communities under each demand scenario. For example, the largest UVN is obtained when the decision epoch, or the vehicle departure cycle is fixed at every 60, 50 and 30 minutes respectively during low, normal and high demand scenarios.

All in all, evaluations and benchmarking of the proposed and static policies demonstrate the potentials of the S and T-policy under the community logistics framework. Continuous serving region adjustment based on real-time arrivals of delivery demand, to our best knowledge, is rarely applied in real business setting, due to the difficulty of solution generation. However, the comparative and spatio-temporal relativity analyses presented in this section inform us of the essence of flexible size adjustment of communities as a means of consolidating new orders to be arrived at distribution hubs.

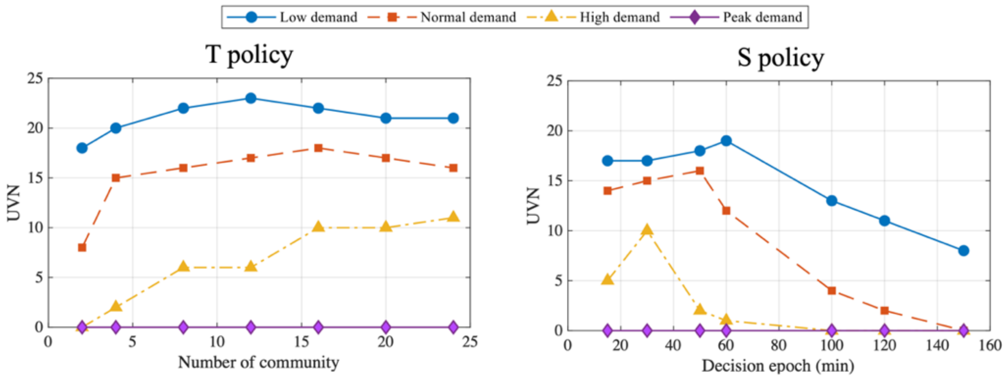


Fig. 13. Comparison of vehicle usage under T and S-policy

5. Strategy comparison study

In this section, we tackle the problem introduced in Section 3.1 with a conventional route-based delivery strategy, which solves the delivery problem by dynamically determining vehicle traveling routes. The delivery solutions obtained using this strategy and the CL delivery strategy are evaluated to exhibit the superiority of our proposed one against route-based strategies.

5.1 Dynamic vehicle routing problem

Using route-based delivery strategies, the original e-commerce last-mile delivery problem is transformed into a dynamic vehicle routing problem. The DVRP optimizes over a finite decision moment set $\{dt^{(1)}, dt^{(2)}, ..., dt^{(k)}, ..., dt^{(K)}\}$. Given a pending order set $I^{(k)}$ and available vehicle set $J^{(k)}$ at the k th decision moment $dt^{(k)}$, decision makers need to determined $|J^{(k)}|$ traveling routes $\{\Lambda_j^{(k)}\}_{j \in J^{(k)}}$ ($\Lambda_j^{(k)} \in I^{(k)}$) to serve the pending orders while satisfying vehicle capacity constraints. A route $\Lambda_j^{(k)}$ is expressed by a sequence of order index, for example, $\Lambda_j^{(k)} = (2,1,3)$ means the arrival sequence of vehicle j is order 2, order 1 and order 3. If $\Lambda_j^{(k)} = \varnothing$, vehicle j has no delivery task at $dt^{(k)}$ and it will wait at the depot until $dt^{(k+1)}$. If an order is not delivered by any vehicle at $dt^{(k)}$, it will be delayed until $dt^{(k+1)}$ and we denote the delay order set as $R^{(k)}$. The objective of DVRP is described from two aspects: decision makers hope to wait more orders to save the traveling distance; on the other hand, they want to deliver orders as soon as possible to achieve a higher customer satisfaction and save inventory costs. Consequently, the objective of the DVRP is to seek an optimal routing policy to minimize the overall cost comprised of the vehicle traveling cost and order delay cost.

In this study, we formulate the DVRP with a Markov decision process model (MDP) which is solved by a scenario-based planning approach. The sampling horizon and number of sampling scenarios are set to be 60 minutes and 20, respectively. The decision interval time T / K is set to be 30 min. Interested readers can refer to the work of Voccia, Campbell, & Thomas (2019) for the modeling and solution details. Here, we emphasize the cost and objective function which are formulated as:

$$C^{(k)} = \begin{cases} \chi \cdot \sum_{j \in J^{(k)}} F^{TD}\left(\Lambda_j^{(k)}\right) + |R^{(k)}| \cdot \frac{T}{K}, & \text{if } k < K \\ \chi \cdot \sum_{j \in J^{(k)}} F^{TD}\left(\Lambda_j^{(k)}\right) + |R^{(k)}| \cdot \left(1440 - T\right), & \text{if } k = K \end{cases} \tag{30}$$

$$C = \sum_{k=1}^{K} C^{(k)} \tag{31}$$

where χ is a parameter controlling the preference between vehicle traveling distance and order delay time. It should be noted that orders not delivered at $dt^{(K)}$ will be delayed to tomorrow, so that the order delay time at $dt^{(K)}$ is 1440-T.

5.2 Strategy comparison results

To compare the proposed CL strategy with the conventional routing-based delivery strategy, we also evaluate the T-policy and S-policy with Eq. (31) and report the minimum C of these two policies. The preference parameter χ is set to be four values, including 100, 200, 300, 400. By considering the four values of χ in instance I1 and I2, we obtain eight new instances denoted as I5-I13 for strategy comparisons. It should be noted that we do not evaluate the routing-based delivery strategy in high and peak demand instances because it is extremely time-consuming to obtain optimal or even near-optimal route solutions in these instances. In this case, the solution time at a decision epoch will exceed the decision interval time, which is impractical for real applications. An illustrative example of the route solutions obtained by using route-based delivery strategy in instance I12 is presented in Fig. 14. As observed in this figure, a distinguished difference between the solutions generated by route-based delivery strategy and CL strategy can be discovered. Delivery solutions using the route-based delivery strategy are a set of vehicle traveling routes across several subareas. On the contrary, delivery solutions using the CL strategy are a set of delivery communities in which the served customers locate compactly. The numerical results of using different delivery strategy in instances I5-I13 are shown in Tables 6 and 7.

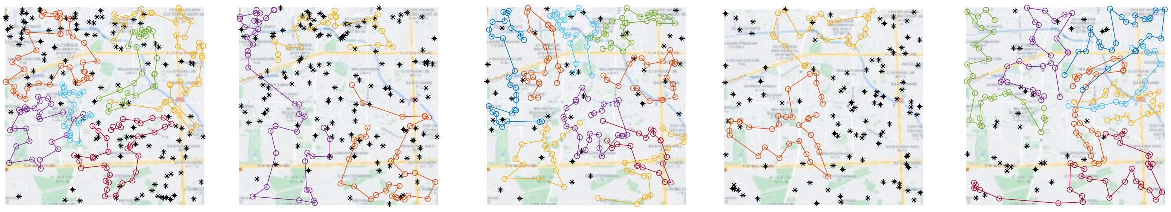


Fig. 14. Solution examples of DVRP using scenario-based sampling approach in instance I12

From Tables 6 and 7, the S-policy always performs better than T-policy in terms of C and ARC. The reason is that a community is served with the T-policy only if the accumulated demand reaches the vehicle capacity threshold. Such a vehicle dispatching rule inevitably causes some orders being postponed overnight at the final decision epoch $dt^{(K)}$, resulting in an additional order delay cost. When comparing the S-policy to the SPA of the route-based delivery strategy, we find that SPA yields lower cost (C) than the S-policy in the instances I5-I7 and I9-I11. Nevertheless, the cost gap gradually narrows with the increase of χ and finally reverses when χ raises to 400 (see instances I8 and I12). As increasing χ means practitioners prefer delaying order fulfillment to save more delivery cost, this indicates that the CL strategy gains more benefits from postponing order delivery than the route-based delivery strategy. It is reasonable that the route-based delivery strategy performs better than the CL strategy when we evaluate them with a DVRP objective, because the former one minimizes the vehicle traveling distance directly and the latter one focuses on optimizing cell compactness.

Nevertheless, the ARC values of the S-policy are much lower than those of SPA especially in Instance I8, where S-policy achieves a 43.8% lower ARC than SPA's. Low ARC can bring several practical benefits, such as improving the route robustness and saving delivery time. Another merit of the CL strategy is the short solution time, the simulation time using the T-policy is around 100 seconds under the low demand instances (I5 – I8) while SPA requires over 9,000 seconds for one simulation for the same instances, as reflected in Table 6. The difference between our policies and SPA in terms of computation time under normal demand (I9 – I12) is more significant as SPA requires over 20,000 seconds to obtain a solution, while our policies take no more than 320 seconds as shown in Table 7. The solution iteration process of e-commerce last-mile delivery problems with route-based delivery strategy involves tackling a set of static vehicle routing problems, which is time-consuming especially in large-scale instances. Therefore, the simulation time of SPA dramatically increases if the demand level doubles from low demand to normal demand.

Table 6. Strategy comparison results in the instances with low demand

Strategy	Policy	Criteria	I5($\chi=100$)	I6($\chi=200$)	I7($\chi=300$)	I8($\chi=400$)
CL strategy	T-policy	Optimal $\|M\|$	2	4	4	4
		C	122437	196702	258448	320193
		ARC (km)	1.34	0.95	0.94	0.94
		CPU time (s)	127	83	84	81
	S-policy	Optimal Δt	30	50	60	100
		C	101148	176347	230977	286349
		ARC (km)	1.05	0.88	0.80	0.63
		CPU time (s)	172	128	113	98
Route-based delivery strategy	Scenario-based planning approach	C	92889	166900	228885	306522
		ARC (km)	1.31	1.29	1.24	1.12
		CPU time (s)	9145	9672	10048	10387

Table 7. Strategy comparison results in the instances with normal demand

Strategy	Policy	Criteria	I9($\chi=100$)	I10($\chi=200$)	I11($\chi=300$)	I12($\chi=400$)
CL strategy	T-policy	Optimal $\|M\|$	4	4	4	4
		C	181253	301187	421121	541055
		ARC (km)	0.96	0.93	0.93	0.95
		CPU time (s)	225	220	218	224
	S-policy	Optimal Δt	30	30	50	50
		C	154798	268699	368103	466602
		ARC (km)	0.83	0.82	0.64	0.65
		CPU time (s)	310	314	230	233
Route-based delivery strategy	Scenario-based planning approach	C	140256	249961	352736	469036
		ARC (km)	1.10	1.08	1.06	1.05
		CPU time (s)	21739	23622	25925	27047

6. Discussions and implications

Based on the results of the simulation experiments conducted in this study, we extract and generalize several key characteristics found in the real delivery environment, which are deemed to best fit the deployment of the proposed community logistic strategy.

6.1 Common features of delivery contexts where community logistics is applicable

Results from the simulation experiments reveal the feasibility of the proposed community logistics framework in various demand scenarios. To ease future research and applications on Community Logistics, we generalize three key features of the delivery environment that are best suitable for applying the Community Logistics: (i) Dynamic and frequent delivery arrivals, (ii) tight schedule for decision-making, and (iii) high consumer density. These features are explained in detail in Appendix II.

In addition, there are two existing delivery sectors that exhibit the above features, they are: (i) same-day or next-day delivery sector, and (ii) Instant grocery delivery. Therefore, they are particularly suitable for community logistics strategy deployment. A detailed discussion of the practical relevance of Community Logistics towards these sectors are provided in Appendix III.

6.2 Dynamic policy selection

Fluctuating order arrival rates in e-commerce delivery environment is a common phenomenon. Therefore, this suggests the complexity of determining an optimal policy under such dynamic environment. However, based on the experimental results, different demand scenarios clearly pose an effect towards the duration of order delivery delay, route compactness and traveling distance. To justify an optimal number of communities and decision epochs, practitioners are required to understand their order arrival patterns (the demand side) and the availability of their vehicle fleets (the supply side) to perform last-mile deliveries. After all, at what order arrival rate is deemed to be "high" or "low" depends on the capacity and resources available for the practitioners. Therefore, the sensitivity of community partitioning towards various demand periods performed in this study offers a general, practical implication of the need to make trade-offs between postponing the delivery of orders and the route compactness, thus determining the optimal community sizes and departure times at each decision cycle. As a remark, optimal policy is one of the future studies to strike a balance between delaying order delivery and partitioning communities under changing demand scenarios.

6.3 CL strategy advantages over route-based delivery strategy

Unlike traditional delivery strategies tackling e-commerce last mile delivery problems by dynamically optimizing vehicle traveling routes, the CL strategy solves it through dynamically generating compact delivery communities. Such an alteration of objective brings following three significant benefits, as validated in the benchmarking analysis against a static dispatch policy in Section 4.6 and a DVRP in Section 5. Firstly, the route compactness is substantially improved which potentially reduces the delivery cost during practical delivery fulfillment. Secondly, there is no obvious conflict between route compactness and vehicle traveling distance. With the CL strategy, a short vehicle traveling distance can also be guaranteed even we merely optimize the cell compactness. Thirdly, the solution time is remarkably reduced because solving the SRGP and TSP is very timesaving than directly solving the VRP, especially in large-scale delivery instances. Combining these benefits, the general CL strategy provides simplicity for practitioners' deployment in real business environment.

7. Conclusion and future work

A Community Logistics Strategy is proposed as a new logistic support tool that integrates "dynamic community departure time" as one of the decision variables in generating dynamic delivery dispatching solutions, in addition to "dynamic community space" which allows formulations of "communities" with flexible sizes and locations. The Temporal and Spatial policies are proposed to examine the spatiotemporal relativity under various demand patterns. Experiments reveal the potentials of both the Spatial and Temporal Policies under the Community Logistics strategic framework against a static policy and a DVRP solution method in managing the dynamic arrivals of e-commerce orders in distribution centre. This implies that dynamic departure time and space as proposed in the Community Logistics framework is not only a feasible, but a better scheme. While this study demonstrates the feasibility of relaxing the departure times of vehicle fleets, future studies could be performed to develop optimal policies under the spectrum of community logistics. Furthermore, a systematic and effective method of partitioning a serving region into communities remains to be a research gap.

References

Banerjee, D., Erera, A. L., & Toriello, A. (2022). Fleet sizing and service region partitioning for same-day delivery systems. Transportation Science.

Bender, M., Kalcsics, J., & Meyer, A. (2020). Districting for parcel delivery services–A two-Stage solution approach and a real-World case study. *Omega*, 96, 102283.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300-313.

Carlsson, J. G. (2012). Dividing a territory among several vehicles. *INFORMS Journal on Computing*, 24(4), 565-577.

Carlsson, J. G., & Delage, E. (2013). Robust partitioning for stochastic multivehicle routing. Operations research, 61(3), 727-744.

Jia, Y. H., Chen, W. N., Gu, T., Zhang, H., Yuan, H., Lin, Y., ... & Zhang, J. (2017). A dynamic logistic dispatching system with set-based particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1607-1621.

Glover, F., & Laguna, M. (1998). Tabu search. In Handbook of combinatorial optimization (pp. 2093-2229). Springer, Boston, MA.

Gendreau, M., Guertin, F., Potvin, J. Y., & Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transportation Research Part C:

Emerging Technologies, 14(3), 157-174.

González-Ramírez, R. G., Smith, N. R., Askin, R. G., Miranda, P. A., & Sánchez, J. M. (2011). A hybrid metaheuristic approach to optimize the districting design of a parcel company. *Journal of applied research and technology*, 9(1), 19-35.

Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

Hintsch, T., & Irnich, S. (2018). Large multiple neighborhood search for the clustered vehicle-routing problem. European Journal of Operational Research, 270(1), 118-131.

Huang, Y., Savelsbergh, M., & Zhao, L. (2018). Designing logistics systems for home delivery in densely populated urban areas. *Transportation Research Part B: Methodological*, 115, 95-125.

Kalcsics, J., & Ríos-Mercado, R. Z. (2019). Districting problems. In *Location science* (pp. 705-743). Springer, Cham.

Kim, B. S., & Lee, W. S. (2011). An adaptive genetic algorithm for a dynamic lot-sizing and dispatching problem with multiple vehicle types and delivery time windows. *Journal of Korean Institute of Industrial Engineers*, 37(4), 331-341.

Kim, G., Ong, Y. S., Cheong, T., & Tan, P. S. (2016). Solving the dynamic vehicle routing problem under traffic congestion. *IEEE Transactions on Intelligent Transportation Systems*, 17(8), 2367-2380.

Minkoff, A.S. (1993) A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. *Operations Research*, 41(1), pp. 77–90.

Mitrović-Minić, S., & Laporte, G. (2004). Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38(7), 635-655.

Mourão, M. C., Nunes, A. C., & Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196(3), 856-868.

Lei, H., Laporte, G., & Guo, B. (2012). Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1-2), 67-85.

Lei, H., Wang, R., & Laporte, G. (2016). Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. Computers & Operations Research, 67, 12-24.

Lin, Y., Qu, T., Zhang, K., & Huang, G. Q. (2020). Cloud-based production logistics synchronisation service infrastructure for customised production processes. *IET Collaborative Intelligent Manufacturing*, 2(3), 115-122.

Poot, A., Kant, G., & Wagelmans, A. P. M. (2002). A savings based method for real-life vehicle routing problems. *Journal of the Operational Research Society*, 53(1), 57-68.

Rios, B. H. O., Xavier, E. C., Miyazawa, F. K., Amorim, P., Curcio, E., & Santos, M. J. (2021). Recent dynamic vehicle routing problems: A survey. *Computers & Industrial Engineering*, 160, 107604.

Rossit, D. G., Vigo, D., Tohmé, F., & Frutos, M. (2019). Visual attractiveness in routing problems: A review. *Computers & Operations Research*, 103, 13-34.

Sarasola, B., Doerner, K. F., Schmid, V., & Alba, E. (2016). Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*, 236(2), 425-461.

Savelsbergh, M., & Van Woensel, T. (2016). 50th anniversary invited article—city logistics:

Challenges and opportunities. *Transportation Science*, 50(2), 579-590.

Sevaux, M., & Sörensen, K. (2008, October). Hamiltonian paths in large clustered routing problems. In *Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME*, vol. 8, pp. 411-417).

Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

Ulmer, M. W., Mattfeld, D. C., & Köster, F. (2018). Budgeting time for dynamic vehicle routing with stochastic customer requests. *Transportation Science*, 52(1), 20-37.

Ulmer, M. W., & Streng, S. (2019). Same-Day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*, 108, 1-19.

van Heeswijk, W. J., Mes, M. R., & Schutten, J. M. (2019). The delivery dispatching problem with time windows for urban consolidation centers. *Transportation science*, 53(1), 203-221.

Voccia, S. A., Campbell, A. M., & Thomas, B. W. (2019). The same-day delivery problem for online purchases. *Transportation Science*, 53(1), 167-184.

Wassan, N. A., & Nagy, G. (2014). Vehicle routing problem with deliveries and pickups: modelling issues and meta-heuristics solution approaches. *International Journal of Transportation*, 2(1).

Winkenbach, M., Kleindorfer, P. R., & Spinler, S. (2016). Enabling urban logistics services at La Poste through multi-echelon location-routing. *Transportation Science*, 50(2), 520-540.

World Economic Forum (2020). The future of the last-mile ecosystem. [online] United States: The World Economic Forum, p.7. Available at: <http://www.weforum.org/docs/WEF_Future_of_the_last_mile_ecosystem.pdf> [Accessed 21 March 2021].

Yu, G., & Yang, Y. (2019). Dynamic routing with real-time traffic information. *Operational Research*, 19(4), 1033-1058.

Zhou, L., Zhen, L., Baldacci, R., Boschetti, M., Dai, Y., & Lim, A. (2021). A Heuristic Algorithm for solving a large-scale real-world territory design problem. Omega (United Kingdom), 103, 102442. https://doi.org/10.1016/j.omega.2021.102442

Appendices

Appendix I – DVRP variants in the pick-up context

Most of the existing dynamic vehicle routing literature has limited applicability to the delivery context as they focus on the accommodating dynamic arrivals of pick-up requests into the real-time routing formulations, but not delivery requests. For example, Gendreau et al. (2006) applied neighborhood search heuristics to optimize the planned routes of vehicles in a context where new requests, with a pick-up and a delivery location, occur in real-time. Similarly, Sarasola et al. (2015) developed a variable neighborhood search algorithm for the stochastic and dynamic vehicle routing problem, a problem that considers the dynamic arrivals of customer requests with stochastic demands only revealed when the vehicle arrives at the customer location. This is a context where couriers depart from a depot to pick up customer's parcels from virous locations. As planning horizon is fixed, both Gendreau et al. (2006) and Sarasola et al. (2015) deal with (1) and (3). Yu and Yang (2017) consider a VRP with real-time traffic information, where stochastic intermediate times, i.e. travel times and service times, are assumed to be realized with probability distributions at the end of each customer's service and before determining the next customer to visit. They proposed a DVRP model addressing the varying intermediate times to determine the next visit. This model can be applied to both pickup and delivery context. Yet, the set of customers and planning horizon are known and fixed, so the model is not to decide (1) and (2), but to optimize results of (3). Ulmer er al. (2018) studied pickup requests occur dynamically during the day and are unknown before their actual request. They presented an anticipatory time budgeting heuristic which allows dispatchers to budget their time effectively by anticipating future requests. This work has a relevance to managing (2).

Appendix II – Key features of the delivery environment suitable for Community Logistics deployment

Based on the results from the simulation experiments, we generalize three key features of the delivery environment that are best suitable for applying the Community Logistics, they are: (i) Dynamic and frequent delivery arrivals, (ii) tight schedule for decision-making, and (iii) high consumer density:

(i) *Dynamic and frequent delivery arrivals* – The fluctuating and discrete arrivals of delivery orders in distribution centres increases the complexity for practitioners to perform delivery scheduling. To wait for more potential delivery requests to arrive at the distribution centres, the design of the mechanism for solution iterations must present the ability of updating the solution. In the family of dynamic VRP, solutions are updated by assessing whether or not new requests are able to be fitted into the current route. However, this approach is suitable only for managing pick-up requests because routing decisions could no longer be reverted as vehicles have departed from the facility for actual delivery. It is not possible for the departed vehicles to re-visit the facility and load new delivery parcels to fulfil the corresponding delivery requests added to the existing route. The framework proposed in this study provides a new approach of how dispatching solutions can be updated with the introduction of intended delivery postponement delaying the actual delivery of pending delivery requests at distribution centres.

(ii) *Tight schedule for decision-making* – Unlike conventional scheduled non-e-commerce delivery,

in which practitioners have plenty of time ahead of actual delivery to make delivery dispatching decisions, last-mile delivery environment in e-commerce has a tight schedule for such decision-making, which requires an agile, light-weight and rapid approach to determine the right batch of delivery requests to be served by a delivery vehicle on a minute-to-minute basis. The proposed postponement-first, route-optional framework, which eliminate the need to perform routing decisions during solution iterations, hugely reduce the required computational power to generate solution in a short computational time. The reduced computational requirement implies that a large number of arriving delivery requests can be managed in bulk for delivery dispatching solution generation. Therefore, the framework is able to deal with dynamic arrivals of delivery requests by generating feasible, quality solutions with a high degree of route compactness on an hourly or even half-hourly basis.

(iii) *High consumer density* – Given a high level of digital maturity in retailing in urban areas, densely populated cities and megacities with skyscrapers and apartment buildings, infer that a large number of spatially compact delivery requests would arrive along the day. Distribution hub’s serving region that is situated at high consumer density areas is best suitable for deployment of the framework, which enables practitioners to enjoy the economies of scale in consolidating delivery requests into a community for batch delivery.

Appendix III – Existing delivery sectors suitable for Community Logistics deployment

(i) *Same-day or next-day guaranteed delivery* – the same-day and next-day delivery has become a new normal in last-mile e-commerce delivery. E-commerce customers are provided with a variety of delivery options, such as three-hour instant delivery, same-day delivery, and same-day or next-day delivery guarantee, at additional costs. With a considerable number of same-day or next-day customer orders, logistics service providers have their flexibility of introducing waiting times to consolidate spatially compact customer orders to be delivered within the same-day or next-day delivery time window, while ensuring meeting the delivery promise to avoid customer dissatisfaction. As e-commerce orders are arriving dynamically and continuously, community logistics strategy could play a role in generating real-time delivery dispatching solutions, taking “dynamic community departure times” into account for the sake of optimizing delivery efficiencies.

(ii) *Instant grocery delivery* – the instant grocery delivery sector is getting popular in many countries and cities. It enables end customers to receive their grocery orders just a few hours upon their online purchases. Grocery retailers with their own delivery fleets are required not only to consolidate the items of an order, but also dispatch the dynamically arriving orders at a right timing. In the face of high customer demand, it is noticeable that some grocery retailers begin to offer a wider delivery time window like up to six hours upon an order is placed. Given such wider time windows, the proposed CL strategy, which can generate a community-based delivery solution at an interval less than 30 minutes in higher demand scenarios, is applicable for consolidating more newly arrived grocery orders for bulk community deliveries. Potentially, more last-minute arrived grocery orders could be added into the community if a tolerable waiting time is introduced to regions destined to have high demands.