Order consolidation for the last-mile split delivery in online retailing

Article in Transportation Research Part E Logistics and Transportation Review - January 2019

2 authors:

Yuankai Zhang
Dalian University of Technology

Lijun Sun
Dalian University of Technology

Some of the authors of this publication are also working on these related projects:

Multi-item order fulfillment in large-scale online retailing View project
Order consolidation for the last-mile split delivery in online retailing

Yuankai Zhang, Lijun Sun, Xiangpei Hu, Chen Zhao

Abstract

This paper addresses the last-mile split delivery problem emerging in online retailing, in which multiple shipments are delivered to the same customer multiple times. We propose an innovative order consolidation approach which consolidates same customers’ shipments in the delivery station and delivers them in fewer trips. An integer programming model is formulated to make a cost tradeoff between splitting and consolidating shipments, which is solved by a new three-phase heuristic algorithm. Numerical experiments are conducted to demonstrate the superiority of the order consolidation approach over the First-In-First-Out approach on various parameters and generate managerial insights for applying it in practice.

1. Introduction

The last-mile delivery under e-commerce environment is currently regarded as the most expensive, least efficient and most polluting sections of the entire logistics chain (Gevaers et al., 2014). The booming e-commerce industry has contributed to the expansion of express delivery markets. However, it has also presented major challenges to the last-mile delivery. In the world’s largest and fastest-growing e-commerce market, Chinese shoppers made $1.1 trillion in online purchases in 2017 at a growth rate of 32.2% (National Bureau of Statistics of China, 2018), which resulted in more than 40 percent of the world’s total delivery volume (a total of 40.1 billion parcels) (Xinhua, 2018). One of the biggest challenging problems is the last-mile split delivery problem, in which multiple shipments are delivered to the same customer multiple times in a short time period. The issue has led to an inefficient operation (delivery operations including making a delivery schedule with customers, transportation, waiting for customers’ signatures, etc.) and a high shipping cost, especially for online retailers which normally deliver a large number of shipments in their day-to-day operations. Such inefficiency in the last-mile delivery has contributed to the bottleneck of online retailing development.

Online retailers (especially large-scale online retailers) usually adopt a two-echelon logistics distribution system (warehouses → delivery stations → customers) to serve customers in the urban area with a high population density (Zhou et al., 2018). Here, the warehouse serves the purpose of packaging orders, and delivering shipments to corresponding delivery stations, while the delivery station functions to receive and consolidate shipments from different warehouses/distribution centers, and then deliver the shipments to customers in its distribution zone. The last-mile delivery refers to the second stage which focuses on the package distribution from delivery stations to customers. It has two features, which are multiple shipments to the same customer arriving at a delivery station in different times, and multiple delivery times from the delivery station to the same customer. These two features are both happening in
a short time period, e.g., 1 day. For the first feature, there are three practical scenarios. The first scenario is order splitting. It is one of the most serious phenomena for online retailers because multiple Stock Keeping Units (SKU’s) in an order are fulfilled separately by multiple warehouses (Zhang et al., 2018). The multi-item order feature in online retailers, e.g., an average of 16.7 items in an order in Yihaodian.com, greatly increases the possibility of splitting orders. Order splitting brings multiple shipments to the same customer. For instance, Xu et al. (2009) found that the number of shipments has increased by 3.75–6% due to 10–15% of order splitting in a large online retailer. For a specific delivery station, these shipments often arrive at different times due to various order processing time among warehouses and different shipping time from warehouses to the delivery station. The second scenario is that customers may place multiple orders from the same online retailer at different time periods, which results in multiple deliveries. Some online retailers have explored order consolidation to reduce the number of shipments, e.g., Amazon’s “Combined Shipments” and “1-Click Order Consolidation” services. However, multiple shipments to the same customer still exist because these consolidation services are generally restricted to a single warehouse, and the consolidation time period is often very short (30 min on Amazon). The third scenario is that customers may place several orders from different e-commerce platforms but the shipments will arrive at the same delivery station and the last-mile delivery will be fulfilled by the same distribution company. For the second feature, multiple delivery times have to be made from delivery stations due to a large number of shipments being delivered while meeting various delivery time requirements. A variety of same-day and next-day delivery choices provided by online retailers, e.g., Three-Hour Delivery, and “211 Program” by JD.com, also need drivers to deliver multiple times on a daily basis.

Fig. 1 presents a visualization of the whole order fulfillment process in online retailing, in which the last-mile split delivery problem under discussion is the most important part. The multiple shipments to the same customer may arrive at the delivery station at different time periods, which may also have different delivery deadlines. Through consolidating same customers' shipments, this research aims to reduce delivery times to same customers and thus reduce the cost with satisfying the deadline of each shipment. To minimize the total cost, this paper will therefore address the following research questions:

(1) Whether to consolidate multiple shipments for the same customer?
(2) If consolidation should be performed, how many time periods should be postponed for each shipment?

Consolidation of shipment deliveries is challenging from a city logistics perspective (Savelsbergh and Van Woensel, 2016). First, making delivery time decisions for multiple shipments of a single customer can be difficult enough since each shipment has a specific arrival time and shipping deadline. Second, consolidating shipments for all customers requires using the delivery station's limited capacity, but the capacity availability varies with the time periods and the consolidation decision. Third, the piecewise-integer structure in the proposed model, which is brought by the shipping cost measurement function observed in practice, is difficult to solve by directly using existing algorithms.

In summary, the contributions of this research include:

(1) We propose a new time-based consolidation policy to solve the emerging last-mile split delivery problem in online retailing, which consolidates same customers' shipments while satisfying the delivery deadline of each shipment;
(2) To the best of our knowledge, the developed integer programming model is the first work to consider the relationship among multiple shipments to same customers in the last-mile delivery and determine the shipping time for each shipment;
(3) The proposed three-phase heuristic algorithm can handle the piecewise-integer structure in the objective function, which performs close to the lower bound of the optimal solution;
(4) The superiority of the proposed order consolidation approach over the First-In-First-Out (FIFO) approach is demonstrated on various parameters and settings.

The rest of the paper is organized as follows. The next section reviews the related literature. Section 3 describes the problem details and compares the FIFO with the order consolidation approach. In Section 4, we present the notations and the mathematical formulation. A three-phase heuristic algorithm is presented in Section 5. Experimental results and managerial insights are presented in Section 6. Finally, conclusions and future research are presented in Section 7.

2. Literature review

In this section we review the literature in the related areas of split order fulfillment in online retailing, last-mile delivery, and shipment consolidation.

(1) Split order fulfillment in online retailing

Split order fulfillment has raised extensive attention in recent years with the booming development of online retailing. Relevant research can be classified into order allocation, assortment allocation, and inventory optimization. Order allocation mainly focuses on how to better allocate multi-item orders to multiple warehouses to reduce order splitting. Xu et al. (2009) consider to re-allocate the not-yet-picked customer orders with their real-time warehouse assignments. While incorporating demand forecasts, Jasim and Sinha (2016) develop a correlated rounding scheme that utilizes the solution of a deterministic linear program to minimize total shipping cost of multi-item orders. Inspired by the airline network revenue management, Acimovic and Graves (2015) propose a linear programming based approach that takes into account current inventory levels and future demand when making fulfillment decisions. Other representative studies include Mahar et al. (2009), Torabi et al. (2015), Ramakrishna et al. (2015), Lei et al. (2017), and Ardjmand et al. (2018). Order allocation can reduce shipping costs for the situation of overlapping storage of SKUs among warehouses. However, in practice, many online retailers (e.g., JD.com) normally have multiple category warehouses with non-overlapping SKUs, which brings order splitting even with a sound order allocation strategy. Assortment allocation and inventory optimization aim to avoid splitting orders from the warehouse perspective. Assortment allocation decides how to assign each SKU to each warehouse (Co et al., 2007) while inventory optimization decides the inventory level of each SKU in each warehouse (Govindarajan et al., 2017). For instance, Co et al. (2007) develop a two-step heuristic method to allocate SKUs in each warehouse by clustering SKUs that ordered frequently. Acimovic and Graves (2017) examine how to allocate inventory to multiple fulfillment centers under a periodic-review joint-replenishment policy, with an objective to minimize outbound shipping costs. Assortment allocation and inventory optimization still leave a large number of split orders because any single warehouse cannot hold all the categories.

In conclusion, existing studies from the order perspective (order allocation) and warehouse perspective (assortment allocation and inventory optimization) provide different ideas to deal with splitting orders in online retailing. However, as Zhang et al. (2018) mentioned, order splitting is inevitable for online retailers because multi-item orders have to be fulfilled by multiple warehouses. As multiple split shipments from different warehouses to one customer will go through the same delivery station, we propose to consolidate shipments in the delivery stations to reduce shipping costs, which would provide a novel perspective to optimize split-order fulfillment in the logistic operations for online retailing.

(2) Last-mile delivery

The rapid growth of e-commerce has brought a great challenge in delivering billions of packages to customers efficiently. In recent years, there have been many efforts to solve the last-mile delivery problem. Previous researchers mainly focus on distribution modes (Goethals et al., 2012), cost analysis and pricing (Asdemir et al., 2009; Yang et al., 2014), crowdsourcing delivery (Wang et al., 2016; Devari et al., 2017) and vehicle routing problems (Stenger et al., 2013; Zhou et al., 2018), etc. For example, Goethals et al. (2012) introduce a concept of the unattended delivery model of e-retailing and study consumers' perceptions. Inspired by the development of crowdsourcing logistics, Devari et al. (2017) study an effective large-scale mobile crowd-tasking model in which the last-mile delivery can be crowdsourced to a large pool of citizen workers. In addition, Zhou et al. (2015) consider a multi-depot two-echelon vehicle routing problem with delivery options in e-commerce, which allows customers to pick up packages at a pickup facility close to their locations. Moreover, consolidating shipments in last-mile delivery has shown its great potential in reducing the high costs and improving the delivery efficiency. For instance, Urban Consolidation Center (UCC), which has been implemented in Europe and other countries (Cherrett et al., 2017), is intended to reduce truck trips into urban centers and therefore truck vehicle-miles of travel, energy consumption and emissions (Janjic and Ndaiye, 2017). However, UCC is more like a freight transfer station which only consolidates lot-size shipments from different logistics companies, rather than providing consolidation for split shipments to same customers.

Different from the well-known split delivery problem in vehicle routing problems, last-mile split delivery brings additional shipping costs instead of potential cost savings. This is because the cost saving by splitting a customer's demand among several vehicles in vehicle routing problems is mostly for large shipments, and this saving is not significant for small demands (up to 10% of vehicle's capacity) (Dror and Trudeau, 1990). But small demands (small shipments) are actually the normal situation in the last-mile delivery within the e-commerce environment. In addition, the shipping cost measurement for the last-mile delivery observed in practice is based on each customer delivery, which makes the last-mile split delivery costlier. Different from previous literature, to the
best of our knowledge, this research is the first work to consider the relationship among multiple shipments to same customers in the last-mile delivery, which can reduce delivery times to individual customers and thus reduce the total cost.

(3) Shipment consolidation

Shipment consolidation is a logistic strategy which is used to reduce transportation costs by combining two or more orders or shipments into a larger one to achieve economies of scale in transportation and better transportation operation (Dror and Hartman, 2007). Meanwhile, shipment consolidation is also an environmentally friendly strategy by reducing carbon emission (Ülkü, 2012).

Most previous researchers in this area focus on the coordination of inventory replenishment and shipment decisions (Nguyen et al., 2014), and the coordination of material flows of semi-finished and finished products from single/multiple vendors to single/multiple retailers (Çapar, 2013; Glock and Kim, 2014; Chen et al., 2018). This kind of consolidation often has the characteristics of a small number of customers with large quantities of freight in each order, and a long transportation cycle via long-haul vehicles. However, with the booming development of e-commerce in recent years, several researchers have noticed the great consolidation potentials for the increasingly less-efficient small-package flows through small truckloads in the last-mile delivery, in which a great number of small packages have to be delivered to individual customers within a short time period. Inspired by non-profit organizations, Hewitt et al. (2015) introduce a consolidation strategy for meals home-delivery by making fewer deliveries for multiple (frozen) meals, which can minimize operational disruptions and maintain client satisfaction. Considering strict order deadlines and expedited shipping options in the e-retailing environment, Wei et al. (2017) present optimal consolidation policies and their structures in settings with up to two warehouses. We summarize the related literature on shipment consolidation and this paper in Table 1. Previous shipment consolidation research put more emphasis on the scale economy for multiple customers, while this paper focuses on the scale economy for same customers’ shipments in the last-mile split delivery problem.

Shipment consolidation may lengthen the shipment cycle and thus increase inventory costs. Therefore, the trade-off for shipment consolidation is mainly between the decreased shipping costs and the increased inventory costs. There are three classical consolidation policies: time-based (Stenius et al., 2016), quantity-based (Çetinkaya and Bookbinder, 2003; Satır et al., 2018) and hybrid (time-and-quantity) policies (Mutlu and Çetinkaya, 2010).

- **Time-based policy.** It pre-determines a waiting time to await the accumulation of items, and one shipment is dispatched at the end of the interval. In the online retailing environment, timely delivery service is a basic requirement for the customers. To ensure a delivery-time guarantee in the last-mile delivery (i.e., the required time period that a particular shipment will be shipped to the customer), distribution companies employ a time-based shipment consolidation policy (e.g. the FIFO policy).
- **Quantity-based policy.** Under this policy, a shipment is dispatched when a threshold quantity of orders is accumulated. For some scenarios (e.g., transportation is performed by private carriage and demands arrival is Poisson distributed), the quantity-based policy is the most cost-effective solution (Çetinkaya et al., 2006). However, for a quantity-based policy, a specific delivery time cannot be guaranteed (Çetinkaya et al., 2006), which is not acceptable by online retailers.
- **Hybrid policy.** It releases a shipment either when a threshold quantity is achieved or a pre-determined time is reached. The hybrid policy is usually cost-wise superior to time-based policies and service-wise superior to quantity-based policies (Çetinkaya et al., 2006). It assumes that vehicles can be dispatched at any point in time. However, for the last-mile delivery problem, delivery stations usually dispatch vehicles at fixed times, which is restricted by the related sorting and loading operations in practice.

In a word, both the quantity-based policy and the hybrid policy are not suitable for the problem under discussion. Our proposed order consolidation approach contributes to the time-based consolidation policy by further consolidating same customers’ shipments and determining the shipping time for each shipment while satisfying its delivery deadline.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Most traditional shipment consolidation research (Dror and Hartman, 2007; Ülkü, 2012; Nguyen et al., 2014; Stenius et al., 2016; ...)</td>
</tr>
<tr>
<td></td>
<td>This paper</td>
</tr>
<tr>
<td>The number of customers</td>
<td>Small</td>
</tr>
<tr>
<td>Shipment size</td>
<td>Large quantities of freight</td>
</tr>
<tr>
<td>Split delivery</td>
<td>Seldom</td>
</tr>
<tr>
<td>Vehicle</td>
<td>Long-haul vehicles</td>
</tr>
<tr>
<td>Transportation cycle</td>
<td>Long (in days/weeks)</td>
</tr>
<tr>
<td>Scale economy</td>
<td>Consolidation for multiple customers</td>
</tr>
</tbody>
</table>
3. Problem statement and analysis

3.1. Problem statement

Considering an online retailer which adopts a two-echelon logistics distribution system (warehouses → delivery stations → customers) to fulfill customer orders, this research concerns the last-mile stage from delivery stations to customers. The distribution area served by multiple delivery stations in a city is generally divided into multiple subareas according to the distribution of customer locations. This research focuses on one distribution subarea served by a specific delivery station. Multiple shipments for one customer may arrive at the delivery station in different time periods since shipments fulfilled by warehouses usually have different operating times and the shipping time of shipments from warehouses to delivery stations may vary. At the beginning of the decision time periods, we assume that the delivery station knows all upcoming orders since the processing time between customers placing orders and the shipments arriving at the delivery station is usually longer than the decision time period in the last-mile delivery.

Let \( n \) be the length of time horizon (e.g., 3 time periods in a day) and \( t \) be the index for all time periods \( T = \{0, ..., t, ..., n\} \). There are \( m \) shipments to be delivered. For each shipment \( i \), the arrival time \( t^\text{start}_i \in T \) at the delivery station is known since the delivery status of each shipment can be monitored in real-time by the Internet of Things (IoT), such as GPS and other sensors. For example, Qinglong system applied in JD.com provides a strong guarantee for every delivery station to monitor the status of each order/shipment from warehouses to customers (Jingdong, 2016). Other IoT based logistics systems can refer to Reaidy et al. (2015) and Venkatadri et al. (2016). Every shipment has a shipping deadline \( t^\text{end}_i \in T \) which is requested by the customer or guaranteed by the retailer. Dispatch time for each shipment \( t^\text{ship}_i \in T \) has to be determined. The condition \( t^\text{start}_i < t^\text{ship}_i \leq t^\text{end}_i \) has to be satisfied because it takes time to do unloading, sorting, and loading operations before shipments are dispatched, and all shipments have to be delivered to customers before deadlines. The delivery station has a storage capacity \( Q^\text{station} \) to store postponed shipments, which leads to additional inventory costs. In addition, homogeneous vehicles in the delivery station have a weight loading capacity \( W^\text{vehicle} \). The number of dispatched vehicles in each time period is determined by the total loading weight, which ignores the shipments-to-vehicle loading decision due to its little influence on the consolidation decision. In the online retailing environment, the objective is to minimize the total cost since the last-mile delivery is the most expensive section of the entire logistics chain (Gevaers et al., 2014) and there is a tradeoff between the shipping cost and the inventory cost if the consolidation of shipments is conducted. The shipping cost is calculated at each customer visit, which is coming from practice and will be introduced in “(2) Cost analysis” of §3.2 Problem analysis. To minimize the total costs of the vehicle-dispatching cost, the shipping cost and the inventory cost, the problem is how to make decisions of the delivery time period for each shipment.

3.2. Problem analysis

(1) Shipment fulfillment approaches

We first show the present time-based shipment fulfillment approach in the last-mile delivery, and then introduce the proposed order consolidation approach. The details of these two approaches are as follows:

First in First out (FIFO): It is the present time-based shipment fulfillment approach, in which all arrived shipments are delivered from the delivery station at the next time period. It is a time-based shipment consolidation approach since shipments are consolidated and delivered in fixed time points instead of the whole time horizon. Some shipments for a same customer can be consolidated and delivered to the customer at one time if they arrive at the delivery station at a same time period. The FIFO is a widely used shipment fulfillment strategy in the last-mile delivery in practice. However, with the order splitting problem emerging in online retailing, the FIFO is myopic because multiple shipments may be delivered to a same customer at multiple times, which lowers the delivery efficiency and brings the high shipping cost. Accordingly, we propose a new shipment fulfillment approach incorporating the consolidation of multiple shipments to same customers.

Order Consolidation: It is a new shipment fulfillment approach based on the FIFO. In this approach, some shipments are delivered as the FIFO, while other shipments have the opportunities to be postponed at the delivery station. The postponed shipments will be consolidated with upcoming shipments and shipped to the customer at fewer times. Compared with the FIFO, order consolidation has the potential of decreasing the shipping cost due to a fewer number of deliveries by consolidating multiple shipments. However, consolidation will bring the inventory cost at the delivery station. Therefore, in the order consolidation approach, we have to make the tradeoff between increased inventory costs and reduced shipping costs.

(2) Cost analysis

The cost of last-mile fulfillment mainly consists of vehicle-dispatching cost, shipping cost, and inventory cost, which are based on the practices of several online retailers.

Vehicle-dispatching cost refers to the sum of the fixed payment for the drivers, the cost of loading operations of shipments into vehicles, and other related dispatching costs for vehicles. The total dispatching cost is calculated by the cost of dispatching a vehicle multiplied by the number of dispatched vehicles. If there is no shipment to be delivered within a certain time period, then the vehicle-dispatching cost is zero.

Shipping cost consists of a fixed shipping cost plus a variable shipping cost. Each delivery for each customer incurs the fixed shipping cost brought by the delivery operations, such as making a delivery schedule with customers, transportation (vehicle
routing), waiting for customers’ signatures, etc. The variable shipping cost occurs only when the weight of shipments in one delivery is higher than the critical weight, and it is measured by the unit variable shipping cost multiplied by the overweight of shipments. The total shipping cost is the sum of the cost of each delivery in which a delivery stands for a customer visit. If multiple shipments are delivered to a same customer at one time, we treat them as a single consolidated shipment since the related delivery operations for the customer only be executed once. Thus, we use the following formula to measure the shipping cost:

Shipping cost per customer visit = fixed shipping cost + variable shipping cost = fixed shipping cost + unit variable shipping cost \* overweight of shipments.

**Inventory cost** is caused by postponed shipments in the delivery station. It cannot be negligible since it represents the landed estate renting costs for the delivery stations. Also, it is well accepted that the trade-off between the shipping cost and the inventory holding cost is the main focus in the consolidation literature (Venkatadri et al., 2016; Wei et al., 2017). Based on our interviews with several operation analysts in an online retailer, the inventory cost is measured by unit shipment inventory cost per unit time period multiplied by the postponed number of shipments and the postponed time periods. This measurement is also widely accepted in the literature (Çetinkaya and Bookbinder, 2003).

(3) An example

We use a specific example which is illustrated in Fig. 2 to explain the two different delivery strategies, FIFO and order consolidation.

In this example, we assume \( n = 4 \), \( m = 5 \) and there are 3 customers. The vehicle-dispatching cost is ignored. Other cost parameters are set as follows: the fixed shipping cost is 1.5; the critical weight per delivery is 20; the variable shipping cost is 0.5 and the unit inventory cost is 0.2. The shipment weights of P1, P2-1, P2-2, P3-1, and P3-2 are respectively 5, 3, 10, 8, and 6. Assume that the arrival times and shipping deadlines for shipments P1, P2-1, P2-2, P3-1, and P3-2 are \((0, 1), (0, 2), (1, 3), (2, 3)\) and \((1, 4)\) respectively. As shown in Fig. 2(a), there are five shipments delivered separately and no inventory costs occurred under the FIFO approach. The total cost of FIFO is \(1.5 \times 5 = 7.5\). If the shipments are delivered with the order consolidation approach, as shown in Fig. 2(b), P2-1 and P3-1 are postponed for one time period. Then P2-1 and P2-2 are consolidated and delivered to customer 2 at time period 2, and P3-1 and P3-2 are consolidated and delivered to customer 3 at time period 3. The total cost of the order consolidation is: \(1.5 \times 3 + 0.2 \times 2 = 4.9\). As mentioned above, the cost savings of the order consolidation are achieved by consolidating shipments for the same customer.

4. Problem formulation

4.1. Notations

\[ J = \{1, \ldots, J, \ldots \} \], customer set;
\[ M = \{1, \ldots, M, \ldots \} \], shipment set;
\[ T = \{0, \ldots, T, \ldots \} \], delivery time period set;
\[ w_i \], the weight of shipment \( i \);
\[ h_{ij} = \{0, 1\} \], shipment \( i \) is for customer \( j \) if \( h_{ij} = 1 \), and otherwise, \( h_{ij} = 0 \);
\[ W_{\text{vehicle}} \], the loading weight capacity of one vehicle;
\[ Q_{\text{station}} \], the inventory capacity for shipments in the delivery station;
\[ c_{\text{vehicle}} \], vehicle-dispatching cost;
\[ c_{\text{ship}} \], fixed shipping cost per delivery;
\[ c_{\text{ship}} \], variable shipping cost per delivery;
\[ c_{\text{hold}} \], the shipping cost for customer \( j \) in time period \( t \);
\[ W_{i} \], the weight threshold for each delivery;
\[ c_{\text{hold}} \], unit inventory cost for a postponed shipment per unit time period;
\[ t_{i}^{\text{start}} \in T \], shipment \( i \) arrived at the delivery station during the time period \( t_{i}^{\text{start}} \), \( t_{i}^{\text{start}} < n \);
\[ t_{i}^{\text{end}} \in T \], the deadline of shipment \( i \) shipped to the customer, \( t_{i}^{\text{end}} > t_{i}^{\text{start}} \);
Decision variables:
\[ x_{it} = \{0, 1\} \], shipment \( i \) is delivered in time period \( t \) if \( x_{it} = 1 \), and otherwise, \( x_{it} = 0 \);
\[ t_{i}^{\text{ship}} \in T \], the delivery time of shipment \( i \), \( t_{i}^{\text{ship}} = \sum x_{it} t_{i} \);
\[ y_{it} > 0 \], the number of dispatched vehicles in time period \( t \), \( y_{it} \in N \);
\[ z_{jt} = \{0, 1\} \], the customer \( j \) is served in time period \( t \) if \( z_{jt} = 1 \), and otherwise, \( z_{jt} = 0 \).

4.2. Model

The model can be represented as follows:

\[
\text{Min} \sum_{t} y_{it} c_{\text{vehicle}} + \sum_{j,t} c_{\text{ship}} x_{jt} + \sum_{t} c_{\text{hold}} (t_{i}^{\text{ship}} - t_{i}^{\text{start}} - 1) \tag{1}
\]

s.t.
\[ c_{ship}^i = \begin{cases} c_{ship}^{L,i}, & \text{if } \sum_j x_{ij} w_j \leq W, \quad \forall j, \ t \\ c_{ship}^{L,i} + c_{ship}^{R,i} \left( \sum_j x_{ij} w_j - W \right), & \text{else, } \forall j, \ t \end{cases} \]

\[
\sum_i x_{it} = 1, \quad \forall i
\]  

\[
t_{i}^{\text{start}} + 1 \leq t_{i}^{\text{ship}} \leq t_{i}^{\text{end}}, \quad \forall i
\]

\[
\sum_i x_{it} w_i \leq \lambda_i W_{\text{vehicle}}, \quad \forall t
\]

\[
\sum_{i, t_i^{\text{start}} + 1 \leq t_i^{\text{ship}}} x_{it} \leq Q_{\text{station}}, \quad \forall t
\]

Fig. 2. Illustration of FIFO and order consolidation.
The objective is to minimize the total cost, which is the sum of vehicle-dispatching cost \( \sum x_i c_{vehicle} \), shipping cost \( \sum c_{ship} x_i t_i \), and inventory cost \( \sum c_{hold} x_i t_i \), and the latter happens only when the delivery weight is over the weight threshold \( \sum x_i h_i w_i > W \). Constraint (3) indicates that each shipment should be assigned within a single time period. Constraint (4) ensures that the delivery time of each shipment should satisfy the requirements by its arrival time and shipping deadline. Constraint (5) and (6) represent the constraints of a vehicle’s loading capacity and a delivery station’s capacity in each delivery time period. Constraint (7) represents that a customer is served only when at least one shipment for the customer is delivered. Constraint (8) ensures that a vehicle is dispatched only when at least one shipment is delivered. Constraint (9) represents the relationship between variables \( x_i \) and \( t_{i,ship} \). Constraint (10) represents the variable requirements.

The complexity of solving this model lies in two aspects: the huge solution space and the piecewise-integer structure. First, the solution space grows exponentially as the increasing of the number of shipments and total time periods since this problem is an NP-hard problem, which is proved as follows.

**Proposition 1.** The order consolidation problem for last-mile split delivery in online retailing is NP-hard.

**Proof.** We prove it by showing that an already-proven NP-hard problem is a special case of this problem. If \( c_{vehicle} = 1 \), \( c_{ship} = 0 \), \( c_{hold} = 0 \), \( c_{station} = +\infty \), \( \sum x_i h_i w_i \), \( \gamma_i = 1 \), then the model is transformed as follows:

\[
\min \sum_{i,t} w_i x_{it} \\
\text{s.t. } \sum_{t} x_{it} = 1, \forall i \\
\sum_{i} x_{it} w_i \leq W_{vehicle}, \forall t \\
x_{it} = [0,1], \forall i,t.
\]

It means to find a feasible packing of the items into the bins that minimizes the total cost. Assume \( n \) bins, a capacity \( W_{vehicle} \) for each bin \( t \), and \( m \) items such that each item \( i \) has size \( w_i \) and yields cost \( w_i \) when packed into bin \( t \). It belongs to the generalized assignment problem, which is a well-known NP-hard problem (Savelsbergh, 1997, pp. 831-832). Also, the transformation can be done in polynomial time. Therefore, order consolidation for last-mile split delivery is also an NP-hard problem.

Second, the piecewise-integer structure in the objective function greatly increases the complexity of this integer combination optimization model. Let \( f(x_{i1},...,x_{im}) \) be the shipment weight of one delivery to the customer. Then \( f(x_{i1},...,x_{im}) \) varies as any of integer combinations of decision variables \( x_{it}, \forall i,t \). As shown in Fig. 3, the shipping cost stays at \( c_{ship} \) if \( f(x_{i1},...,x_{im}) \) is not larger than \( W \), and the shipping cost increases linearly when \( f(x_{i1},...,x_{im}) \) is larger than \( W \).

Previous studies, e.g., Croxton et al. (2003); Keha et al. (2006), have explored how to solve piecewise-linear optimization model by adding additional variables. However, existing methods cannot solve the piecewise-integer structure due to the combination nature of multiple integer decision variables. The piecewise-integer structure is also hard to solve directly by commercial software (e.g., CPLEX). We also have tried the Big M method to translate the inequality constraints in (2), however, it turns out that the transformed model is a nonlinear model, which is still not easy to solve. Because of the higher complexity of the model, we intend to propose a new heuristic algorithm to solve it.

Before proposing the algorithm, we find some preliminary properties through the analysis of the proposed model with \( c_{vehicle} = 0 \), which is helpful to solving the model.

**Property 1.** Assume \( \delta = \left\lceil \frac{c_{ship}}{t_{i,ship}} \right\rceil \), then we have \( t_{i,ship} \leq \min\{t_{i,start} + 1 + \delta, t_{i,end}\} \).

**Proof.** The maximum cost saving of any two consolidated shipments is \( c_{ship} \) (ignoring the vehicle-dispatching costs) and the possible increased costs for one postponed time period is \( c_{hold} \), then the maximum postponement time is \( \delta = \left\lceil \frac{c_{ship}}{t_{i,ship}} \right\rceil \). We can get \( t_{i,ship} \leq t_{i,start} + \delta \). Recalling \( t_{i,ship} \leq t_{i,end} \), we have \( t_{i,ship} \leq \min\{t_{i,start} + 1 + \delta, t_{i,end}\} \).

Property 1 is used to provide the initial searching solution space for the second phase of the proposed algorithm, which can reduce the whole solution space and improve the efficiency of the algorithm.

**Property 2.** The shipments can be postponed and consolidated only when \( c_{hold} \leq c_{ship} \).

Property 2 can be gotten by the constraint \( t_{i,start} + 1 \leq t_{i,ship} \) and property 1, \( t_{i,ship} \leq \min\{t_{i,start} + 1 + \delta, t_{i,end}\} \). It is a necessary
condition for consolidating shipments, which acts as the premise of applying the shipment consolidation approach in practice. That is, the order consolidation can only be applied when the unit inventory cost is lower than the fixed shipping cost. From property 2, we can also infer that the optimal policy is the FIFO policy if \( c_{\text{hold}} > c_{\text{ship}} \), since the consolidation is beneficial only when \( c_{\text{hold}} \leq c_{\text{ship}} \).

**Property 3.** If \( t_i^{\text{end}} = t_i^{\text{start}} + 1 \), then \( t_i^{\text{ship}} = t_i^{\text{end}} \).

Property 3 can be gotten because of \( t_i^{\text{start}} + 1 \leq t_i^{\text{ship}} \leq t_i^{\text{end}} \). Shipments satisfying this criteria should be sent to the customer instantly without consolidation. It can exclude some shipments in the iteration of the Breadth-first algorithm, which contributes to the efficiency of the proposed algorithm.

### 5. Solution procedure

In this section, we first present a three-phase algorithm to solve the proposed model. Then, an enumeration algorithm is proposed to obtain a lower bound, which can verify the accuracy and efficiency of the three-phase algorithm.

#### 5.1. A three-phase algorithm

As shown in Fig. 4, the algorithm contains three parts: Phase I-Initial split delivery solution generation, Phase II-Breadth-first search for each customer and Phase III-Solution improvement. Based on the present split delivery schemes, the initial generation of order consolidation schemes is obtained in Phase I. We then generate the best order consolidation scheme for each customer using a breadth-first search based algorithm in Phase II. The breadth-first search based algorithm incorporates three strategies, the checking duplicate shipment strategy, dropping same path set strategy and cutting node strategy, which will greatly reduce the searching space and improve the searching efficiency. In Phase III, two improvements, which refer to the checking delivery station's capacity feasibility and improving with dispatched vehicles, are included to optimize the order consolidation schemes for all customers.

##### 5.1.1. Phase I: Initial split delivery solution generation

Initial split delivery schemes are the schemes generated by FIFO in practice, which are also the basis of order consolidation schemes. Assume \( I_0 \) is the initial shipment scheme for customer \( j \). \( I_0 \) is generated by setting \( t_i^{\text{ship}} = t_i^{\text{start}} + 1 \) for shipment \( i \). According to Property 3, all shipments are divided into two sets, the set of shipments that can be postponed, and the set of shipments that cannot be postponed. The shipments in the second set only have one solution and would not be changed. We only need to optimize the shipment assignment in the first set with the following two algorithm phases. For the example in §3.2, the first set is \( \{P2-1, P2-2, P3-2\} \) and the second set is \( \{P1, P3-1\} \). The following steps mainly improve the solution for the shipments \( \{P2-1, P2-2, P3-2\} \). The initial solution by the FIFO is \( \{P1-1-1, P2-1-1, P2-2-2, P3-1-3, P3-2-2\} \).

##### 5.1.2. Phase II: Breadth-first search for each customer

We found that it is difficult to generate order consolidation schemes even for a single customer because of the following three reasons. (1) The best delivery scheme for one customer may be multiple shipments being delivered in several delivery time periods; (2) Every shipment has its specific arrival time and shipping deadline; (3) The calculation of shipping costs in piecewise-integer
structure varies with combination of shipments.

To generate the order consolidation scheme, we propose a breadth-first search based algorithm which is widely used in state-space search and is an efficient technique for solving combinatorial optimization problems (Sewell and Jacobson, 2012; De Carvalho and Soma, 2014). An order consolidation scheme is a set of decision variables of shipments in delivery time periods. We observe that an order consolidation scheme forms a state-space, in which one shipment with one delivery time period \((x_{it})\) is a search node. The initial state is corresponding with the initial split delivery scheme generated in the first step. The best scheme is a set of nodes which can minimize the sum of the shipping cost and the inventory cost. The algorithm can easily handle the piecewise-integer structure in the objective function since the shipping cost will be calculated for extended nodes during the searching process, in which the combination of shipments is already known. Possible extending nodes for each shipment vary with its arrival time and shipping deadline, referring to its state space \(t_{i_{\text{start}}} + 1 \leq t_{i_{\text{ship}}} \leq \min(t_{i_{\text{start}}} + 1 + \delta, t_{i_{\text{end}}})\). Therefore, we first generate the set of possible extending nodes for each customer \(j\), and the nodes are sorted in ascending order of \(i\) and \(t\). This set will be adjusted during the procedure of the algorithm.

The breadth-first search may generate duplicate nodes and extend to nodes with higher costs, which consumes memory space and execution time. Therefore, we propose the following three strategies to reduce the running time of this algorithm.

A. Checking duplicate shipment strategy

According to constraint (3) in the proposed model, only one delivery time period can be assigned for one shipment. Thus, if one node (one shipment with one delivery time) has been extended, then the other nodes which have the same shipment and different delivery times will not be extended in the same path.

B. Dropping same node set strategy

The breadth-first search algorithm may generate different extending paths but with the same node set, which wastes the searching time. This strategy is proposed to prevent the same set of nodes being visited again if the set is already extended.

C. Cutting node strategy

For each extending node, we will calculate the cost saving. If the negative cost saving is generated by the node, the cutting node strategy will remove the node from the expanding path. Assume \(I_{i_0}\) and \(I_{i_h}\) represent the original scheme and the new scheme with the extending node respectively. With the objective function \(g(x_{it}, \forall i, t) = \sum_{i \in I_{i_0}} c_{i_{\text{ship}}} + \sum_{i \in I_{i_h}} c_{i_{\text{hold}}} (t_{i_{\text{ship}}} - t_{i_{\text{start}}}) - 1) d_{ih}\), the formula for

\[
g(x_{it}, \forall i, t) = \sum_{i \in I_{i_0}} c_{i_{\text{ship}}} + \sum_{i \in I_{i_h}} c_{i_{\text{hold}}} (t_{i_{\text{ship}}} - t_{i_{\text{start}}}) - 1)
\]
calculating the cost saving is \( \text{costsaving} = g(x_t \in I_{j0}) - g(x_t \in I_{j0}) \).

The steps of the algorithm are as follows:

Step 1: For customer \( j \), generate the possible node list \( \Omega = \{x_t | h_j = 1, \ t_{\text{start}} + 2 \leq t_{\text{ship}} \leq \min[t_{\text{start}} + 1 + \delta, \ t_{\text{end}}]\} \) based on properties 1 and 2, initialize the set list \( \Omega_2 \) and add the split delivery scheme to \( \Omega_2 \);

Step 2: Extend each possible node from list \( \Omega \). If the nodes in the extended path violate the constraint of one delivery time period for one shipment, then the node will not be extended; if the set of nodes in the extended path is as same as any set in the list \( \Omega_2 \), then the node will not be extended;

Step 3: If the cost saving of the new scheme with nodes in the extended path is negative, then the node will not be extended; else, the node will be extended and the nodes in this path will be added to list \( \Omega_2 \);

Step 4: Repeat the extending procedure with steps 2–3 until no possible solution exists;

Step 5: Find the maximum cost saving of the path in \( \Omega_2 \) as the best solution for customer \( j \);

Step 6: Assume \( I_{j1} \) is the node list in the best path, then the solution list for customer \( j \) is generated by combining the solution \( I_{j1} \) and the initial solution \( I_{j0} \).

We continue to use the example in §3.2 to illustrate how the proposed Breadth-first search algorithm (Phase II) works, as shown in Fig. 5. First, we generate the set of possible nodes \( \{P2-1-2, P2-2-3, P3-2-3, P3-2-4\} \), in which, for example, the node P2-1-2 represents the shipment P2-1 delivered at the time period 2. The algorithm expands the searching tree to the first layer with all possible nodes as shown in Fig. 5a. According to the cutting node strategy, the branches of P2-2-3 and P3-2-4 will be cut down due to the negative cost saving compared with the initial split delivery scheme. Then we expand to the next layer (as shown in Fig. 5b), the node of P2-2-3 will be cut down because of its negative cost saving. In the branch of P3-2-3, the node P2-1-2 will not be expanded according to the dropping same path set strategy, and the node P3-2-4 will not be expanded due to the checking duplicate shipment strategy. Next, continue expanding the nodes to the last layer (in Fig. 5c). Finally, we will compare the cost savings of all expanded paths, which are \{P2-1-2, P3-2-3\}, \{P2-1-2, P3-2-3\}, \{P2-1-2, P3-2-4\}, \{P3-2-3, P2-2-3\} and \{P2-1-2, P2-2-3, P3-2-3\}. \{P2-1-2, P3-2-3\} has the minimal cost saving, then combining the initial solution \{P1-1-1, P2-1-1, P2-2-2, P3-1-3, P3-2-2\} by the FIFO in section 5.1.1, the best result is \{P1-1-1, P2-1-2, P2-2-2, P3-1-3, P3-2-3\}.

5.1.3. Phase III: Solution improvement

The solution generated above does not consider the coordination among shipments to different customers, which mainly lies in the capacities of the delivery station and a vehicle. Thus, two improvements, including Checking delivery station’s capacity feasibility and Improving with dispatched vehicles, are designed to deal with the constraints of capacities for shipments to multiple customers. For those two improvement parts, we implement the sequence “Capacity → Dispatched vehicle → Capacity” in Phase III. Our basis intuition is: The “Capacity” is used first to make the previous solution feasible since the solution gotten by Phase II may be infeasible because of exceeding the delivery station’s capacity; Then the “Dispatched vehicle” can reduce the number of dispatched vehicles and improve the vehicles’ loading rate based on the feasible solution; Third, the “Capacity” is used again to ensure the feasibility of the solution, since the “Dispatched vehicle” may cause the solution infeasible against the delivery station’s capacity. However, one can also directly implement the sequence “Dispatched vehicles → Capacity”. The comparison results are shown in “(1) The algorithm sequence optimization” of §6.2.

(1) Checking delivery station’s capacity feasibility

The capacity constraint of the delivery station in one time period may be violated if the shipping scheme of shipments for
customers is generated one by one. It means that some shipments cannot be postponed in the delivery station in that time period. Accordingly, the improvement mainly involves which shipments should be adjusted and which time periods these shipments should be adjusted. The detailed steps of the improvement are as follows:

Step 1: For time period \( t \), if the number of postponed shipments \( \sum_{i \in I} q_{i, \text{ship}} + 1\) is larger than the capacity \( Q_{\text{station}} \), go to Step 2; else, \( t \leftarrow t + 1 \), return to Step 1;

Step 2: Sort the postponed shipments by the sum of added shipping cost and inventory cost in ascending order. The added cost is calculated with the formula \( \text{adding cost} = g(x_q \in I_{\text{ship}}) - g(x_q \in I_{\text{ship}}) \), where \( I_{\text{ship}} \) represent the present scheme and \( I_{\text{ship}}' \) represent the new adjusted scheme with the postponed shipment \( i \) assigned to \( t_{\text{start}} + 1 \);

Step 3: Select first \( \left( \sum_{i \in I} q_{i, \text{start}} + 1\right) < q_{i, \text{ship}} \) number of shipments in the sorted list, and adjust them with \( t_{i, \text{ship}} = t_{i, \text{start}} + 1 \);

Step 4: Set \( t \leftarrow t + 1 \), and return to Step 1 until all time periods have been visited.

(2) Improving with dispatched vehicles

This part aims to further reduce the number of dispatched vehicles and improve vehicles’ loading rates in each time period. The loading rate in the time period \( t \) refers to \( r_t = \left( \sum x_i w_i - (y_t - 1) W_{\text{vehicle}} \right) / W_{\text{vehicle}} \), in which \( \sum x_i w_i - (y_t - 1) W_{\text{vehicle}} \) is the actual loading weight while \( W_{\text{vehicle}} \) is the vehicle loading capacity. We will focus on optimizing the time periods with low vehicle loading rates. The detailed steps are as follows:

Step 1: Calculate the vehicle’s loading rate \( r_t \) in the time period \( t \). If \( 0 < r_t \leq \varepsilon \) (a threshold), then go to Step 2; else, \( t \leftarrow t + 1 \), return to Step 1;

Step 2: For each node in the time period \( t \) (present scheme \( I_{\text{ship}}' \)), adjust the shipment to a different time period \( t' \) (the new scheme \( I_{\text{ship}}' \)), and calculate the sum of added shipping cost and inventory cost with formula \( g(x_q \in I_{\text{ship}}') - g(x_q \in I_{\text{ship}}') \). If the corresponding shipment satisfies \( t_{i, \text{ship}}^t > t_{i, \text{start}} + 1 \), then add the node to the backup node list, which is a list containing strong candidate nodes for further improvement;

Step 3: Sort the backup nodes by the added costs in ascending order;

Step 4: Remove the shipment node from the time period \( t \) one by one in the backup node list until the total removed weights are lower than \( \sum x_i w_i - (y_t - 1) W_{\text{vehicle}} \). Record the added weights in each time period \( W_t \) and the responsible node list \( L \) of the removed nodes, and sum the added costs for these nodes. For each time period \( t' \neq t \), if \( \left( \sum x_i w_i - (y_t - 1) W_{\text{vehicle}} \right) + W_t > W_{\text{vehicle}} \), then \( t \leftarrow t + 1 \), and return to Step 1;

Step 5: If the sum of added costs is higher than the reduced vehicle-dispatching costs, then \( t \leftarrow t + 1 \), and return to Step 1; else, adjust the scheme with the node list \( L \);

Step 6: Set \( t \leftarrow t + 1 \) and return to Step 1 until all time periods have been visited.

5.2. Lower bound: An enumeration algorithm

As the proposed model cannot be solved directly by existing exact algorithms or by commercial software (e.g., CPLEX) to get the optimal solution, we get a lower bound to verify the accuracy of the three-phase algorithm. The lower bound is generated by a model of relaxing constraints (5) and (6) in the original model. The relaxed model can be solved by considering each customer individually since there are no connections for different customers in the relaxed model (the details of the model is shown in Proposition 2). For each customer, the delivery scheme of multiple shipments is generated by a proposed enumeration algorithm. The number of dispatched vehicles equals to the sum of the shipments’ weight dividing the vehicle capacity. The detailed steps of the algorithm are as follows:

Step 1: Set \( Q_{\text{station}} = +\infty \) and the number of dispatched vehicles \( \sum y_j = \left\lfloor \sum w_i / W_{\text{vehicle}} \right\rfloor \);

Step 2: For the customer \( j \), generate the possible node list with the number \( \delta \) of nodes;

Step 3: Enumerate schemes for customer \( j \) with the number of nodes in the scheme varying from 1 to \( \delta \);

Step 4: Set the scheme for customer \( j \) with the lowest sum costs of inventory costs and shipping costs;

Step 5: Iterate Steps 2–4 for other customers and output the solutions.

Proposition 2. We have \( \phi_2 \leq \phi_1 \leq \phi_0 \), where \( \phi_1, \phi_2, \phi_0 \) represent the best result, the lower bound by the enumeration algorithm, and the result obtained by the three-phase algorithm, respectively.

Proof. We first prove that \( \phi_2 \) is the lower bound of the problem \( \phi_2 \leq \phi_1 \) by a relaxation of the original model. We relax the constraint (6) in the proposed model by assuming that the delivery station’s capacity is unlimited, since the delivery station’s capacity limits the consolidation of shipments for multiple customers. The vehicle-dispatching cost cannot be less than \( \sum y_j / W_{\text{vehicle}} \) no matter whether we conduct the order consolidation or not. In this way, constraint (5) can also be relaxed. The relaxed model is as follows:

\[
\begin{align*}
\text{Min } & \sum y_j / W_{\text{vehicle}} + \sum c_{ij}^\text{ship} + \sum c_{i,j}^\text{hold} (t_{i, \text{ship}} - t_{i, \text{start}} - 1) \\
\text{s.t. } & (2), (3), (4), (7), (9)
\end{align*}
\]
Consequently, we have the lower bound $\phi_2$ and $\phi_2 \leq \phi_1$. It always has $\phi_1 \leq \phi$, because $\phi$ is feasible with all constraints. Then we get $\phi_2 \leq \phi_1 \leq \phi$.

Proposition 2 is used to verify the accuracy of the proposed three-phase algorithm since we cannot get the best result easily. We use $\text{Gap} = (\phi_1 - \phi_2)/\phi_2$ to verify it in the following numerical experiments.

6. Numerical experiments

Numerical experiments are carried out to assess and analyze the performance of the proposed three-phase algorithm and the order consolidation approach. The instances are randomly generated based on the existing practice of last-mile delivery in China. The proposed algorithm is implemented with C#. Experiments are performed on an Intel Core i5-5200U 2.2GHz processor with 4 GB RAM under Windows 10 system.

6.1. Generation of instances

For a distribution area served by a delivery station, we set the number of customers $n = 300$, which will be extended to 2000 later. Considering that the number of shipments per customer and the total time periods are the two factors influencing the complexity of the problem, twelve instances (see Table 2) are generated in terms of increasing the number of shipments per customer (choosing from Poisson distribution $\lambda_1 = 2$ to $\lambda_1 = 5$), and total time periods (from 3 to 5). Most of the parameters are based on our investigations made with several distribution companies serving the last-mile delivery in China. The vehicle holding capacity is given on the basis of electric, three-wheeled vehicles, which deliver most of the packages in the last mile delivery of China. Vehicle-dispatching cost is estimated on the total dispatching cost (including the fixed payment for drivers, the loading costs, etc.) and the dispatched times of vehicles in a delivery station in a month. It is a little difficult to estimate the holding cost, and hence we refer to the holding cost of the intelligent parcel lockers in the last-mile delivery. The same estimation is made for the delivery station’s holding capacity. The unit holding cost and the holding capacity may vary a lot with the varied size of the delivery station, which are further explored by the analysis in §6.3 (1)–(2). The shipping costs, including the fixed shipping cost and the variable shipping cost, refer to the piecework payment for drivers in practice. When the driver delivers an order, the fixed shipping cost is about 1.5 RMB, and the variable shipping cost (about 0.5RMB/kg) is calculated based on the extra weights of shipments over 20kgs. Since the shipping costs may vary slightly among different companies, we did the sensitivity analysis of these parameters in §6.3 (3). Consequently, the default values of the parameters are set to $W_{\text{vehicle}} = 50$, $Q_{\text{station}} = 50$, $c_{\text{vehicle}} = 20$, $c_{\text{ship}} = 1.5$, $c_{\text{ship}} = 0.5$, $W = 20$ and $c_{\text{hold}} = 0.2$ respectively. Sensitivity analyses of these parameters were done further. $t_i^{\text{start}}$ and $t_i^{\text{end}}$ are randomly generated from $U(0, n)$ and satisfy $t_i^{\text{end}} > t_i^{\text{start}}$. The weight of each shipment is randomly determined according to Poisson distribution $\lambda_1 = 5$.

6.2. Computational results

(1) The algorithm sequence optimization

To verify the algorithm sequence as stated in §5.1.3, we conduct numerical experiments for two possible algorithm sequences. As shown in Table 3, the first sequence “Capacity → Dispatched vehicles → Capacity” performs a little better than the second sequence “Dispatched vehicles → Capacity” and the average cost gap is $-0.40\%$. The difference between these two sequences is the “Capacity” part. We infer that in the first sequence, the “Dispatched vehicles” may be more effective based on the feasible solutions gotten by “Capacity”, while in the second sequence, the “Dispatched vehicles” may be less effective based on the infeasible solutions.
The effectiveness of the three-phase algorithm

We attempt to verify the optimality of the three-phase algorithm by comparing it with the lower bound obtained by the enumeration algorithm. As shown in Table 4, the gap between the lower bound and the solution obtained from the three-phase algorithm ranges from 0.76% to 2.50%, which demonstrates that the three-phase algorithm has nearly optimal performance. The execution time of the enumeration algorithm grows a lot with the complexity of the instances. The solutions of three instances cannot be obtained by the enumeration algorithm because of out of memory. However, the three-phase algorithm can easily generate nearly optimal results for all twelve instances within 2s, which verify the efficiency of the proposed algorithm.

(3) Order consolidation vs. First in First out

We verify the performance of the proposed order consolidation approach through comparing it with the FIFO approach. As shown in Table 5, order consolidation reduces in an average of 5.09% (from 3.28% to 7.03%) of the total costs in comparison with FIFO. This reduction is mainly caused by the reduction in the shipping costs (an average of 178.67) with a little increase in the inventory costs (an average of 27.25), which shows the superiority of order consolidation.

6.3. Managerial insights

We numerically investigate the impact of various parameters such as the unit inventory cost factor as well as larger problem scales such as the number of customers. The sensitivity analyses will draw some managerial insights for applying the order consolidation approach.

(1) Scaling with the number of customers

As scaling with the number of customers from 100 to 2000, compared with the FIFO, the cost saving by order consolidation increases at first and then maintains a nearly steady line, as shown in Fig. 6. This phenomenon can be explained by the similar trend

### Table 3
The comparison of two algorithm sequences.

<table>
<thead>
<tr>
<th>Instances</th>
<th>Sequence 1: Capacity → Dispatched vehicles → Capacity</th>
<th>Sequence 2: Dispatched vehicles → Capacity</th>
<th>Cost gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2122.30</td>
<td>2140.10</td>
<td>−0.83%</td>
</tr>
<tr>
<td>2</td>
<td>2743.20</td>
<td>2743.20</td>
<td>0.00%</td>
</tr>
<tr>
<td>3</td>
<td>3398.90</td>
<td>3422.30</td>
<td>−0.68%</td>
</tr>
<tr>
<td>4</td>
<td>4143.00</td>
<td>4168.40</td>
<td>−0.61%</td>
</tr>
<tr>
<td>5</td>
<td>2055.10</td>
<td>2055.10</td>
<td>0.00%</td>
</tr>
<tr>
<td>6</td>
<td>2898.30</td>
<td>2919.50</td>
<td>−0.73%</td>
</tr>
<tr>
<td>7</td>
<td>3537.80</td>
<td>3538.00</td>
<td>−0.01%</td>
</tr>
<tr>
<td>8</td>
<td>4349.70</td>
<td>4374.20</td>
<td>−0.56%</td>
</tr>
<tr>
<td>9</td>
<td>2217.10</td>
<td>2217.10</td>
<td>0.00%</td>
</tr>
<tr>
<td>10</td>
<td>2834.60</td>
<td>2834.60</td>
<td>0.00%</td>
</tr>
<tr>
<td>11</td>
<td>3624.60</td>
<td>3651.90</td>
<td>−0.75%</td>
</tr>
<tr>
<td>12</td>
<td>4259.90</td>
<td>4285.70</td>
<td>−0.60%</td>
</tr>
</tbody>
</table>

Cost Gap = (Sequence 1 − Sequence 2)/Sequence 2 * 100%.

### Table 4
The comparison between the lower bound and the results of the three-phase algorithm.

<table>
<thead>
<tr>
<th>Instances</th>
<th>Lower bound</th>
<th>Three-phase algorithm</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total costs</td>
<td>CPU time (s)</td>
<td>Total costs</td>
</tr>
<tr>
<td>1</td>
<td>2106.20</td>
<td>0.01</td>
<td>2122.30</td>
</tr>
<tr>
<td>2</td>
<td>2694.10</td>
<td>0.02</td>
<td>2743.20</td>
</tr>
<tr>
<td>3</td>
<td>3344.90</td>
<td>4.31</td>
<td>3398.90</td>
</tr>
<tr>
<td>4</td>
<td>4102.70</td>
<td>1.00</td>
<td>4143.00</td>
</tr>
<tr>
<td>5</td>
<td>2039.60</td>
<td>0.91</td>
<td>2055.10</td>
</tr>
<tr>
<td>6</td>
<td>2850.80</td>
<td>1118.97</td>
<td>2898.30</td>
</tr>
<tr>
<td>7</td>
<td>3451.40</td>
<td>268.58</td>
<td>3537.80</td>
</tr>
<tr>
<td>8</td>
<td>N/A</td>
<td>N/A</td>
<td>4349.70</td>
</tr>
<tr>
<td>9</td>
<td>2179.00</td>
<td>2.02</td>
<td>2217.10</td>
</tr>
<tr>
<td>10</td>
<td>2802.20</td>
<td>3809.56</td>
<td>2834.60</td>
</tr>
<tr>
<td>11</td>
<td>N/A</td>
<td>N/A</td>
<td>3624.60</td>
</tr>
<tr>
<td>12</td>
<td>N/A</td>
<td>N/A</td>
<td>4259.90</td>
</tr>
</tbody>
</table>

N/A stands for that no results can be gotten because of out of memory.
Table 5
The comparison between the results by Order consolidation and First in First out.

<table>
<thead>
<tr>
<th>Instances</th>
<th>Order consolidation</th>
<th>First in first out</th>
<th>Total cost savings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shipping cost</td>
<td>Inventory cost</td>
<td>Total costs</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>664.50</td>
<td>17.80</td>
<td>2122.30</td>
</tr>
<tr>
<td>2</td>
<td>806.00</td>
<td>17.20</td>
<td>2743.20</td>
</tr>
<tr>
<td>3</td>
<td>959.50</td>
<td>19.40</td>
<td>3398.90</td>
</tr>
<tr>
<td>4</td>
<td>1084.00</td>
<td>19.00</td>
<td>4143.00</td>
</tr>
<tr>
<td>5</td>
<td>669.50</td>
<td>25.60</td>
<td>2055.10</td>
</tr>
<tr>
<td>6</td>
<td>870.50</td>
<td>27.80</td>
<td>2898.30</td>
</tr>
<tr>
<td>7</td>
<td>1030.00</td>
<td>27.80</td>
<td>3537.80</td>
</tr>
<tr>
<td>8</td>
<td>1180.50</td>
<td>29.20</td>
<td>4349.70</td>
</tr>
<tr>
<td>9</td>
<td>702.50</td>
<td>34.60</td>
<td>2217.10</td>
</tr>
<tr>
<td>10</td>
<td>879.00</td>
<td>35.60</td>
<td>2834.60</td>
</tr>
<tr>
<td>11</td>
<td>1089.00</td>
<td>35.60</td>
<td>3624.60</td>
</tr>
<tr>
<td>12</td>
<td>1242.50</td>
<td>37.40</td>
<td>4259.90</td>
</tr>
</tbody>
</table>

Total cost savings = (Total costs of FIFO – Total costs of order consolidation)/Total costs of FIFO.

Fig. 6. The costs by order consolidation scaling with the number of customers.

Fig. 7. The costs of order consolidation scaling with the delivery station’s capacity.
of the inventory cost that is restricted by the delivery station’s capacity. When the postponed number of shipments reaches the maximum storage capacity of the delivery station, shipments can only be fulfilled by FIFO, leading to no more cost savings by order consolidation for more shipments. We test additional analysis of the delivery station’s capacity based on the situation of 2000 customers. As shown in Fig. 7, cost savings by order consolidation still can increase with the increase of the delivery station’s capacity. It indicates that the delivery station’s capacity is one of the most important factors that greatly restricts the performance of the order consolidation.

(2) Impact of the unit inventory cost

We conduct the sensitivity analysis of the unit inventory cost based on instance 7. As shown in Fig. 8, the cost savings by order consolidation decrease with an increase of the unit inventory cost in the delivery station. An interesting phenomenon is that the inventory cost of order consolidation increases first and then decreases. When the unit inventory cost is low, the number of postponed shipments stays the same, and the inventory cost increases as the unit inventory cost. However, when the unit inventory cost is high, it will be uneconomical to postpone the shipments, which results in the decreasing of delayed shipments and thus the inventory cost. Therefore, there exists a break-even point of the unit inventory cost to influence the performance of order consolidation.

![Fig. 8. The costs of order consolidation varying with the unit inventory cost.](image)

![Fig. 9. The cost savings by order consolidation varying with the unit shipping cost.](image)
Sensitivity to the shipping cost

Based on instance 7, we set the value of the fixed shipping cost $c_{\text{ship}}$ to change from 1.5 to 3, while the value of the variable shipping cost $c_{\text{vship}}$ varies from 0.5 to 2. Fig. 9 shows that the cost savings by order consolidation increase as the fixed shipping cost increases. The similar trend can be found as the variable shipping cost increases. We can conclude that it is more economical to apply the order consolidation approach if the unit fixed or variable shipping cost in the last-mile delivery increases.

Sensitivity to the number of shipments per customer

We set $n$ to vary from 2 to 8 with $n = 4$. From Fig. 10, we find that the running time of the three-phase algorithm is no more than 40.03 CPU seconds for about 2400 shipments (the number of shipments is large enough for a delivery station). Additionally, order consolidation reduces fewer costs with an increase in the number of shipments per customer. It is caused by two reasons: one is the limitation of the delivery station’s capacity as we stated above; the other is that a higher average number of shipments in one time period may have a higher possibility to achieve the lower costs by split delivery rather than order consolidation.

Robustness on multiple sets of parameters

Here, we test the algorithm on several different sets of parameters to show the robustness of our numerical results and the sensitivity analysis results. Different from the parameter set at the beginning, which can represent the scenarios of regular delivery stations, we present two sets of parameters which can represent the scenarios of delivery stations located in the center of a city and the edge of a city. For the first scenario (the center of a city), the shipping cost parameters and the holding cost are higher, which are $W_{\text{vehicle}} = 50$, $Q_{\text{station}} = 50$, $c_{\text{vehicle}} = 20$, $c_{\text{ship}} = 2$, $c_{\text{vship}} = 0.5$, $W = 15$ and $c_{\text{hold}} = 0.5$. For the second scenario (the edge of a city), the storage capacity is larger and the vehicle dispatching cost and holding cost parameters are lower, which are $W_{\text{vehicle}} = 50$, $Q_{\text{station}} = 80$, $c_{\text{vehicle}} = 15$, $c_{\text{ship}} = 1.5$, $c_{\text{vship}} = 0.5$, $W = 20$ and $c_{\text{hold}} = 0.1$.

The overall numerical results are very encouraging, which draw the same conclusions as the previous parameter set did. Here, we just show two new results to get more insights. The first is that the gap between order consolidation approach and the FIFO approach ranges from 3.78% to 7.03% in the regular scenario (Table 5), while it ranges from 3.46% to 5.81% in the first scenario, and from 5.76% to 9.07% in the second scenario. The three scenarios all demonstrate the superiority of order consolidation over the FIFO. We can find that the saving costs in the first scenario are lower than that in the regular scenario, which reflects that it will be a little difficult to consolidate the shipments in the delivery stations in the center of a city since its holding costs are much higher. In addition, the saving costs in the second scenario are higher than that in the regular scenario, which reflects that it can be easier to conduct order consolidation in the delivery stations in the edge of a city because of its larger storage capacity and lower holding costs. The second is that there is a break-even point for the inventory cost in Fig. 8, and the break-even point increases from 1 to 1.5 in the first scenario. Due to the tradeoff between the shipping costs and the inventory costs, a higher unit shipping cost may result in a higher break-even point of the inventory cost.
7. Conclusions

This paper presents an order consolidation approach for the last-mile split delivery problem in a two-echelon logistics distribution system for online retailers. By exploring the benefits of consolidating multiple shipments, the order consolidation approach postpones some shipments in the delivery stations and consolidates them with other incoming shipments, and then delivers them together to the customers. We propose an integer programming model to make a cost tradeoff between split delivery and the consolidation. Since the model shows the piecewise-integer structure by the shipping cost function coming from practice and the huge solution space by the NP-hardness, we present a three-phase heuristic algorithm to solve it. The numerical experiments show that the sequence of “Capacity → Dispatched vehicles → Capacity” in the algorithm performs better, and verify that only a small gap exists between a lower bound and the solution obtained from the proposed algorithm. Additionally, the order consolidation approach can reduce an average of 5.09% of the total costs compared with the FIFO approach. The extensive sensitivity analyses provide some managerial insights that show promise for applying order consolidation approach in practice. Managerial insights are concluded as follows:

1. The superiority of order consolidation approach is robust on various parameters and settings;
2. The tight capacity of delivery stations could restrict the performance of order consolidation;
3. There is a break-even point for the unit inventory cost to influence the performance of order consolidation;
4. A higher unit fixed or variable shipping cost leads to more cost savings by order consolidation;
5. The company may prefer the FIFO if there is a higher average number of shipments in one time period;
6. It will be more beneficial for the distribution company to conduct order consolidation in the delivery stations in the edge of a city than in the center of a city.

Our work serves as the initial efforts on order consolidation for the last-mile split delivery problem in online retailing. Several promising directions for future research remain. First, it remains open whether the piecewise-integer model can be solved by exact algorithms. One possible way is to solve the nonlinear model transformed by the Big M method. Also, a better lower bound may be further explored for the model. Second, the consolidation for multiple customers lies in aspects of the delivery station’s capacity and the dispatched vehicles, and one can explore deeper in the aspect of the routing decisions in the last-mile delivery. Third, the model assumes that all upcoming orders are known in advance. In reality, the delivery station still faces the uncertainty of orders if there are some urgent orders or for some orders, the processing time between customers placing orders and the shipments arriving at the delivery station are shorter than the decision time period. It will be interesting and challenging to consider the uncertainty of orders in the future model. Additionally, it would be interesting to take the delivery station’s capacity as a decision variable into the problem in an environment of uncertain order arrivals. Moreover, the future model can consider the delivery time uncertainty and customer returns in the last-mile delivery. Furthermore, in this research, our model’s objective is to minimize the total cost. Another interesting and challenging extension would be to extend the objective of minimizing cost to maximizing profit (Ülkü and Bookbinder, 2011, 2012) since customers may prefer to pay a higher price for a better delivery service with the order consolidation. In this way, the price decision variable has to be considered and the major change to our model is to involve the demand function of different consolidation schemes and the corresponding prices. The schemes may vary with the delivery time periods and the number of delivery times.

Acknowledgment

We would like to thank the editors and anonymous reviewers for their valuable comments and suggestions that helped us to improve this paper to a great extent. This research was supported by the program of the National Natural Science Foundation of China (NSFC) (71571067, 71571027), the innovative research group program of the NSFC (71421001), the key project of the NSFC (71531002), and China Scholarship Council (201706060098, 201806065010).

References


Ulku, M.A., 2012. Dare to care: shipment consolidation reduces not only costs, but also environmental damage. Int. J. Prod. Econ. 139 (2), 438–446.


