9-2021

A learning and optimization framework for collaborative urban delivery problems with alliances

Jingfeng YANG
Singapore Management University, fyang.2018@phdcs.smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

A Learning and Optimization Framework for Collaborative Urban Delivery Problems with Alliances

Jingfeng Yang[1] and Hoong Chuin Lau[1]

School of Computing and Information Systems, Singapore Management University,
80 Stamford Rd, Singapore 178902
jfyang.2018@phdcs.smu.edu.sg, hclau@smu.edu.sg

Abstract. The emergence of e-Commerce imposes a tremendous strain on urban logistics which in turn raises concerns on environmental sustainability if not performed efficiently. While large logistics service providers (LSPs) can perform fulfillment sustainably as they operate extensive logistic networks, last-mile logistics are typically performed by small LSPs who need to form alliances to reduce delivery costs and improve efficiency, and to compete with large players. In this paper, we consider a multi-alliance multi-depot pickup and delivery problem with time windows (MAD-PDPTW) and formulate it as a mixed-integer programming (MIP) model. To cope with large-scale problem instances, we propose a two-stage approach of deciding how LSP requests are distributed to alliances, followed by vehicle routing within each alliance. For the former, we propose machine learning models to learn the values of delivery costs from past delivery data, which serve as a surrogate for deciding how requests are assigned. For the latter, we propose a tabu search heuristic. Experimental results on a standard dataset and a real case in Singapore show that our proposed learning-based optimization framework is efficient and effective in outperforming the direct use of tabu search in most instances. Using our approach, we demonstrate that substantial savings in costs and hence improvement in sustainability can be achieved when these LSPs form alliances and requests are optimally assigned to these alliances.

Keywords: Alliances · Collaboration · Machine Learning · Pickup-and-delivery · Tabu Search.

1 Introduction

With rapid urbanization, urban delivery systems need to be optimized for capacity and efficiency. High delivery demands not only bring challenges to large LSPs such as Amazon and Cainiao, but also create more intense competition among small and medium-sized LSPs. Due to the high uncertainty of demands and locations in daily delivery, LSPs face operational issues from one end of the spectrum (idle capacity) to the other hand (vehicle and manpower shortage).

To overcome these issues, one approach is to establish collaboration with fellow logistics players. As described by Savelsbergh and Woensel [19], collaboration or cooperation is often regarded as a useful path to consolidating freight volumes, leading to a higher and efficient utilization of resources. An alliance by two or more companies offers opportunities for sharing of information and resources to jointly handle delivery tasks. Collaboration in city logistics systems has been widely studied during past few years.

In this paper, we study the pickup and delivery routing problem in a collaborative setting. In particular, we consider the problem that frequently occurs in urban delivery: LSPs perform their daily operations to pickup goods from one location and deliver to another location, and each request has a delivery time window. In an uncooperative setting, each LSP make route plans with their respective requests. For collaborative routing, we assume there exists multiple alliances in the market, and LSPs in the same alliance can share requests and execute the joint routing decision. For simplicity, we assume LSPs in a given alliance will share the same depot to locate their vehicles. Furthermore, an LSP may participate in more than one alliances (perhaps to service different types of goods). Note that this paper is not concerned with the coalition structure generation problem, which focuses on partitioning the set of agents into mutually disjoint coalitions so that the overall total reward is maximized in the long haul. Rather, we assume that the structure of the alliances (composition of LSPs in each alliance) is given as input parameters for our model, and deal with the operational problem of efficient deliveries in an environment where an LSP may belong to multiple alliances.

From the sustainability perspective, it would be ideal to consider the setting where the LSPs are co-operative, and the problem of how planning can be performed on an existing alliance structure that maximizes the system wide objective of total travel cost. We formulate this problem as a multi-alliance multi-depot pickup and delivery problem with time windows (or MAD-PDPTW).

The main contributions of this work are summarized as follows: (1) We propose a MIP model to formulate the MAD-PDPTW; (2) We develop a tabu search based heuristic method to solve the problem on large instances; (3) To solve the problem more efficiently, we decompose MAD-PDPTW to a two-stage problem, which first learns the delivery cost from data and then optimizes the request reassignment and vehicle routing; (4) We demonstrate the significance of the proposed learning and optimization framework (achieve lower delivery cost with less computational time) and obtain managerial insights for LSPs.

2 Related Work

This section provides a summary of existing studies which focus on collaboration in logistics and distance approximation in vehicle routing problems (VRP). Collaboration in logistics industry has been a prevalent topic in urban logistics studies which normally can be achieved in two ways: vertically and horizontally [19]. In this paper, we focus on the horizontal collaboration which involves logistic service providers (LSPs) at the same level in supply chains. A comprehensive description about the opportunities and impediments of horizontal collaborative logistic service was conducted by [3]. They did a survey include 1537 LSPs in Belgium, and the results shows that most of LSPs believe collaboration will increase their profits and improve service quality.

Horizontal Collaboration: Various studies for horizontal collaboration in logistic systems have been published in last decades. Readers can refer to [6], [21] for more details. Two main themes can be further summarized: (1) develop optimization models and mechanisms for collaborative network planning and design problems to help LSPs increase profits or decrease costs; and (2) propose cooperative and non-cooperative game theory methods for cost/gain sharing to establish and keep better collaborations. This study will focuses on the optimization models for collaborative multi-LSPs delivery problem, the literature review is conducted accordingly. Berger et al. [1] proposed a decentralized control and auction exchange mechanisms to maximize total profits through collaboration among individuals carriers. Similar research has been conducted by Lai et al. [11], which focus on a centralized control with iterative auction to minimize empty traveling miles. Dahl and Derigs [4] studied a pickup and delivery vehicle routing problem with time windows (PDVRPTW) to minimize total delivery cost. Li et al. [13] also studied the pickup and delivery problem with requests exchange to maximize total profits. [18] solved a multi-depot vehicle routing problem to minimize the total distance traveled with a local search method. Unlike exchange requests or vehicle sharing, [5] introduced a new vehicle routing problem that customers can be served by more than one carrier. It aims to minimize overall operational cost by such collaboration. For more various vehicle routing problem in a collaborative setting, readers can refer to the survey investigated by Gansterer and Hartl [8].

Approximations of Routes: The VRP has been well studied from the last decade. Many exact and heuristic algorithms have been investigated to solve it optimally or in a short computational time. Different from the optimization algorithms which aims to get the optimal or good solutions, the continuum approximation (CA) models are used to approximate the travel distance of routes without solve the complex routing problem. Those CA models can provide faster and good approximation of route distance, which are developed and applied for many applications, such as terminal design problem [17], supply chain distribution network design [14] and collaboration mechanisms design [7]. While, in the face of large-scale complex problems, most CA approaches hold a low accuracy performance. Recently, few studies [15, 16] use machine learning approaches to direct estimate the total travel distance of routes. In this paper, we develop a machine learning approach to estimate the delivery cost for pickup and delivery problem with time windows (PDPTW). And with the help of the learned cost, we can further integrate it in requests assignment procedure, and decide which alliance the order should be allocate to.

As discussed above, most studies have devoted to optimize collaborative planning and operation problems from a perspective of entire coalition, whereas all

LSPs take part in a single coalition. Zhang et al. [22] investigated the less-than-truck collaboration decision making problem for the e-commerce logistic network, which objective is maximize the total profit of the entire alliance. To our best of knowledge, Guajardo et al. [10] is the first work that studied the coalition configuration problem which allows company can collaborate in more than one coalition (we prefer to use the term 'alliance' in this paper) in collaborative transport. They developed optimization model to help finding the best coalition configuration. Hence, research gaps are identified form the review of extant literature. In our paper, alliances have been established as inputs in our model, and LSPs in one alliance can share requests and do centralized planning for urban delivery services. More specifically, we focus on optimizing collaborative urban delivery service, with some LSPs can collaborate in more than one alliance.

3 Problem Formulation

In this section, we present our collaborative urban logistic delivery problem in the context of multiple LSPs and multiple alliances. Since each LSP may specialize in fulfilling different types of goods (e.g. groceries and electronics) which may or may not be loaded in the same vehicle, and each may have their own trusted partners, it is plausible to have multiple alliances with overlapping participants.

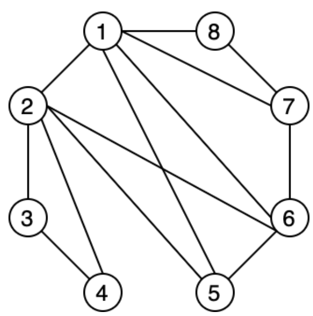

Fig. 1. Multiple alliances with overlapping LSPs

Figure 1 gives an example comprising 8 LSPs and 3 alliances. Each node is defined as a LSP, and if two nodes are connected with an edge, it represents those two LSPs that can share requests. So the alliance is defined as a complete sub-graph, in which a unique edge connects every pair of distinct vertices. Here, we have alliances $[2, 3, 4], [1, 2, 5, 6]$ and $[1, 7, 8]$ in this example. This study aims to assess the potential benefits of collaborative routing among LSPs by sharing requests and joint planning, which means a centralized platform will decide the optimal assignment of requests among each alliance with the constraint that requests cannot be shared between different alliances.

Note that if this problem were to be treated as a whole, one will need to simultaneously decide how LSPs' own requests are distributed to different alliances,

Notation	Description
G	A complete direct graph
N	Set of all LSPs
A	Set of all alliances as well as depots
K	Set of vehicles
R	Set of all requests, each request r has a pickup node and delivery node
P	Set of pickup nodes
D	Set of delivery nodes
V	Set of all nodes in graph G
R_a	Set of requests only belong to alliance a
K_a	Set of vehicles only belong to alliance a
d_a	Depot node for alliance a
$[e_i, l_i]$	Time windows for node i, earliest pickup time and latest delivery time
s_i	Service time at location i
q_i	Weight of goods to pickup or delivery at node i
c_{ij}	Travel cost between node i and node j
t_{ik}	Time node i served by vehicle k
w_{ik}	Weight of vehicle k after visit node i
Q	Vehicle capacity
y_{ijk}	Binary variable, 1 if the vehicle k visited node j directly after visited node i, 0 otherwise

Table 1. Notations

and how routing is performed on the assigned requests within each alliance. It is worth noting that even for a small-scale problem instance, a straightforward meta-heuristic approach such as Tabu Search may not be computationally efficient and may not provide an effective solution, as our experiment would show.

Before presenting our mathematical programming model, we first introduce notations in Table 1.

Given the above notations, we formulate the multi-alliances multi-depots vehicle routing problem with pickup and delivery (MAD-PDPTW) as follows:

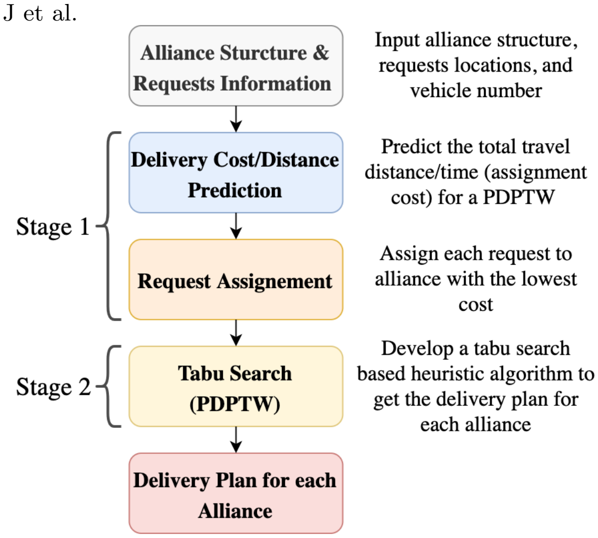

Fig. 2. A two-stage learning and optimization framework to solve the MAD-PDPTW

$$\textbf{minimize} \quad \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}y_{ijk} \tag{1}$$

$$\textbf{subject to} \quad \sum_{i\in V}\sum_{k\in K} y_{ijk} = 1 \quad \forall j \in P\cup D \tag{2}$$

$$\sum_{i\in V} y_{ijk} - \sum_{i\in V} y_{jik} = 0 \quad \forall j \in P\cup D, \forall k \in K \tag{3}$$

$$\sum_{i\in V} y_{id_ak} = \sum_{j\in P\cup D} y_{d_ajk} \leq 1 \quad \forall a \in A, \forall k \in K_a \tag{4}$$

$$\sum_{j\in P\cup D} y_{ijk} - \sum_{j\in P\cup D} y_{(i+r)jk} = 0 \quad \forall i \in R, \forall k \in K \tag{5}$$

$$t_{ik} + s_i + c_{ij} - M(1-y_{ijk}) \leq t_{jk} \quad \forall i,j \in P\cup D \tag{6}$$

$$e_i \leq t_{ik} \leq l_i \quad \forall i \in P\cup D, \forall k \in K \tag{7}$$

$$t_{ik} \leq t_{(i+r)k} \quad \forall i \in P \tag{8}$$

$$t_{ak} = 0 \quad \forall a \in A, \forall k \in K \tag{9}$$

$$w_{jk} \leq w_{ik} + q_j + M(1-y_{ijk}) \; \forall i,j \in V, \forall k \in K \tag{10}$$

$$w_{jk} \geq w_{ik} + q_j - M(1-y_{ijk}) \; \forall i,j \in V, \forall k \in K \tag{11}$$

$$w_{ik} \leq Q \quad \forall i \in V, \forall k \in K \tag{12}$$

$$y_{ijk} = 0 \quad \forall i \notin K_a, \forall j \notin K_a, \forall k \in K_a \tag{13}$$

We divide constraints into four groups. The first group of constraints deals with the in and out flow between each pickup and delivery node. Constraint (2) guarantees that each pickup or delivery node will be visited exactly once. Constraint (3) ensures that each pickup or delivery node, it must be served by the same vehicle k. Constraint (4) imposes constraints on each depot and ensures that each vehicle k belongs to depot K_l will start and back to depot d with at most once. Constraint (5) guarantees the pickup node and delivery node belonging to one request will be served within the same tour.

The second group of constraints deals with visiting precedence of pickup nodes, delivery nodes and time windows. And constraint (6) is the Miller-Tucker-Zemlin (MTZ) sub-tour elimination constraint. If $y_{ijk} = 1$, then we have $t_{ik} + s_i + c_{ij} \leq t_{jk}$, otherwise we have a constraint with right hand side (RHS) is a enough big positive value. Constraint (7) is time windows constraints, which guarantee the delivery time for each request must in the time window. Constraint (8) is precedence constraint that ensure each request is serviced at its pick up node first before the delivery. Constraint (9) denotes the arriving time for each vehicle at the depots equals to 0.

The third group of constraints are the capacity constraints. Constraints (10) to (11) calculate the vehicle weight after visiting each node. In addition, we have $q_i = -q_{i+r}$ for $i \in R_p$. And constraint (12) means for each vehicle k after serve node i, the weight of it cannot exceed the capacity.

The final constraint is the request assignment constraint. It ensures that vehicles belonging to one alliance cannot deliver a request belonging to other alliance. In other words, each alliance is responsible for its own requests.

4 Two-Stage Learning and Optimization Framework

The above section introduces a MIP model to determine the optimal request assignment as well as routing of multiple alliances. In the MIP model, the decision variable y_{ijk} not only decide the delivery sequence from node i to node j, but also make decision for LSPs participating in multiple alliances on request assignment (choose the alliance to share requests). However, the underlying problem is NP-hard, which is computationally intractable to cope with larger instances. In this section, we propose a learning and optimization framework consisting of two stages from requests assignment to vehicle routing. Specifically, the first stage makes decisions for LSPs participating in multiple alliances, which alliance each request should be assigned to (Section 4.1). The second stage adopts a tabu search based heuristic algorithm to solve the PDPTW for each alliance with the assigned requests (Section 4.2). The whole framework is depicted in Figure 2.

4.1 Delivery Cost Prediction, Request Assignment

In this subsection, we first discuss the prediction model for the delivery cost for each alliance. Second, we use the estimated delivery cost as input parameters for requests assignment.

Cost prediction: Previous research has proposed approximate analytic formulas for TSP and VRP under various application scenarios as described in the literature review. However, those analytic based methods always have poor performance on larger problems or complex real-world constraints (e.g., capacity vehicle routing problem with time windows). Here, we use machine learning models to predict the delivery cost for PDPTW (in this paper, we take the delivery cost as the total travel distance). We first generate the promising features for the total travel distance. The general classes of predictors are based on the number of locations, visiting area, the distance between nodes, node dispersion, time windows, and the number of routes. Table 1 lists all the features included in our learning model, and there is a total of 19 features used in our prediction model.

Features	**Definitions**
f_1	Number of locations need to be visited
f_2, f_3	Min/max distance between customers and depots
f_4, f_5	Min/max x distance between customers and depots
f_6, f_7	Min/max y distance between customers and depots
f_8	Average distance between customers and depots
f_9	Average x distance between customers and depots
f_{10}	Average y distance between customers and depots
f_{11}	Standard deviation of distance between customers (and depots)
f_{12}	Area of the smallest rectangle covering customer locations
f_{13}	Area of the smallest rectangle covering customer and depot locations
f_{14}	Sum of the length of time windows
f_{15}	Standard deviation of the length of time windows
f_{16}	Sum of the length of overlap time windows
f_{17}	Standard deviation of the length of overlap time windows
f_{18}	Total demand/Vehicle capacity ratio
f_{19}	Vehicle capacity/Average demand ratio

Table 2. Features for total travel distance prediction

After extracting the features of the PDPTW, the second step is to get the actual solutions to the problem instances as labeled data. Since PDPTW is an NP-hard problem, it would be computationally challenging to generate a large number of exact solutions to be used as label data. In this paper, we find a proxy for the best solution by applying our tabu search algorithm (presented in the next section) instead. Experimental results show that our algorithm comes within a 5% gap on average compared with the best known solutions, and this gives the assurance that the labeled data generated by this approach is accurate and precise. The next step is to select the appropriate machine learning model compatible with the request assignment optimization. We tried a wide range of machine learning regression models in this work, including linear models, such as ordinary least square, LASSO and ridge regression, and nonlinear models, e.g., decision trees and random forest. In summary, we want to identify prediction models that can achieve both good performance and interpretability. In the numerical experiments of Section 5, we show the selection details considering the above criteria.

Request assignment: In the MAD-PDPTW, one request can only be assigned to one alliance, but the assignment for request belongs to LSPs participate in more than one alliance (e.g., LSP 1 in Figure 1) has multiple options. Therefore, we can divide the whole requests into two categories: (1) requests waiting for assignment; and (2) alliance base requests which only belong to one specific alliance. Assume there are $I = \{1, 2, ..., |I|\}$ requests waiting for assign to $J = \{1, 2, ..., |J|\}$ alliances. Each request must be served and there is no limit on how many requests each alliance can have. Then the request assignment problem is to find a partition of the requests waiting for assignment with the minimum total cost. As a result, the request assignment problem can be cast as the set partitioning problem:

$$\min \quad \sum_{j \in Z} c_j v_j \tag{14}$$

$$\text{s.t.} \quad \sum_{j \in Z} \delta_{ij} v_j = 1 \quad \forall i \in I \tag{15}$$

$$\sum_{j \in J} v_j = |A| \tag{16}$$

$$v_j \in \{0, 1\} \tag{17}$$

where Z is the set of all possible partition of requests. δ_{ij} equals to 1 if request i belongs to subset j, and 0 otherwise. Constraints (15) ensure that every request is assigned to a alliance and constraint (16) ensures the number of selected subsets equal to the number of alliances $|A|$. The problem involves an exponential number of variables (columns) since the number of possible subsets grows exponentially in the number of requests waiting for assignment. And predict the cost c_j of all possible partition of request is also very time-consuming. Instead of enumerating all the possible partitions, we provide a simply greedy heuristic approach to solve the request assignment iteratively. We randomly rank requests sequence of unassigned requests, and assign one request to one alliance at each iteration. Here, variable a_{ij} denotes the cost for request i assigned to alliance j. The value of assignment cost a_{ij} is predicted by the machine learning model introduced in **cost prediction**. In this case, the problem becomes a simple facility location problem and we can simply assign each request i to the alliance j with lowest cost a_{ij}. Then the total cost equals to $\sum_{i \in I, j \in J} a_{ij}$.

4.2 Tabu search algorithm

In this subsection, we first develop an efficient tabu search algorithm to solve the PDPTW for each alliance. Furthermore, with minor adjustments by including constraint (12) into the algorithm, we can use it to solve the MAD-PDPTW, which is used as a baseline method in our numerical experiments in Section 5.

Tabu search [9] is one of the well-known meta-heuristics. It takes a potential solution and search its neighborhood iteratively to find improved solutions. It has been applied successfully to various routing problems [20, 2]. In what follows, we introduce the full framework of our algorithm, including initial solution construction and the tabu search algorithm.

The procedure to construct an initial solution s_0 is described here. We construct the initial solution s_0 where not all the constraints defined in PDPTW need be satisfied. Given requests set R, pickup node set P, delivery node set D, and available vehicles K as inputs, for each vehicle $k \in K$, we iteratively select request c from the pickup set P, and check whether it satisfy the earliest pickup time constraint $e_i \leq e_c \leq e_{i+1}$. If yes, we add the both the pickup node and delivery node of requests c to vehicle k, otherwise, we put it in a new vehicle $k+1$. When there are no requests in set P, we end up with the initial solution s_0.

Algorithm 1 Tabu search algorithm

Input: s_0, best solution $s^* = s_0$, tabu list $\mathcal{L} = \emptyset$
Output: Best solution s^*

```
Let current solution s_c = s_0
while i ≤ I_max do
    Do insertion and removal operation
    Get the neighborhood solution N_s of s_c
    for s_i ∈ N_s do
        Calculate fitness function f(s_i)
        if s_i ∉ L and f(s_i) ≤ f(s_c) then
            s_c = s_0
        end if
    end for
    if f(s_c) ≤ f(s*) then
        s* = s_c
    end if
    if Size of L ≥ L_min then
        Update L
    end if
end while
```

Based on the initial solution found, the tabu search based heuristic algorithm is described in Algorithm 1. In our paper, the termination condition is that the maximum number of iterations I_{max} is reached. And the fitness function is described as $f(s) = C(s) + \alpha \cdot Q(s) + \beta \cdot T(s)$, where $C(s)$ is the value of objective function (1), $Q(s)$ denotes the total amount of weights that exceed the vehicle capacity and $T(s)$ represents the total unit of times that violate the time windows constraint. As can be seen, the fitness function consists of two parts: the original objective function and the penalty cost. Parameters α and β are both positive penalty terms that make the solution s become more likely to meet the capacity and time windows constraints, respectively. To achieve this, we introduce a new parameter θ with small value (e.g., 0.1) as step size to adjust the value of α and β. If either $Q(s)$ or $T(s)$ not equals to 0, we multiply it by $(1+\theta)$ in the next iteration. Another important part is the tabu list, which represents a set of solutions that have been visited in the recent past. In this paper, we define the maximum length of tabu list is L_{max} and use it to memorize the insertion operations when we insert the pickup node i and delivery node $i+r$ to route k. In order to solve the MAD-PDPTW, we only need to add constraint (12) before doing insert and remove operation, to ensure that nodes i and $i+r$ can only be added to or removed from route $k \in K_a$ that belongs to the same alliance.

5 Numerical Experiments

This section presents experimental setup for problem instance generation, delivery cost prediction and compares our learning and optimization framework against tabu search in solving the MAD-PDPTW. Computational experiments are conducted to validate the developed framework's performance for multiple alliances under different kinds of settings. All computational experiments are conducted on a desktop computer with Intel Core i5 2.3 GHz with 16GB RAM. The tabu search algorithm are implemented in Java, while the machine learning models are coded in Python 3.7.

5.1 Problem Instance Generation

The dataset proposed by [12] is a popular standard dataset in the study of PDPTW, and is used to generate sampled PDPTW instances in our paper. We need to construct two types of instances, which are synthesized from the PDPTW benchmark dataset, with the first one used as a training and testing dataset for delivery cost prediction, while the second one is prepared for running MAD-PDPTW. For the first type of instances, we randomly sample with the total number of requests of each instance are in the range of 100 to 200. The labeled data of each PDPTW instance is computed by the tabu search algorithm described in Section 4. We obtain 500 instances in total, of which 400 are randomly selected as the training set, and the remaining 100 serve as the test set.

Notations	Description
x	The x coordinate of the pickup/delivery locations
y	The y coordinate of the pickup/delivery locations
q_i	Demand of node i
e_i	Earliest pickup/delivery time of node i
l_i	Latest pickup/delivery time of node i
s_i	Service time of node i
p_i	Pickup (index to sibling) of node i
d_i	Delivery (index to sibling) of node i
L_i	LSP index of node i

Table 3. Instances generated from the PDPTW benchmark dataset

To set up the multiple alliance structures, we construct a second type instance

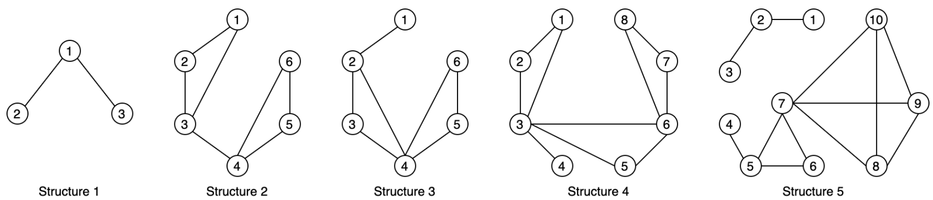

Fig. 3. Alliances structure setting for the case study

by sampling from the original data and randomly reallocating the requests to LSPs and alliances. Compared to the first type of instances, the second type has one more column with request ownership information. Table 2 gives a brief description of the second type of sampled instances. Table 3 and Figure 3 list the detailed parameters and shows the alliance structures for all second type test instances.

Parameters	Settings
Number of instances	15
Number of LSPs	3, 5, 6, 8, 10
Number of alliances	2, 3, 4, 5
Number of requests	18, 60, 65, 75, 105, 120, 135, 150, 180, 185

Table 4. Main parameter settings for the case study

Model	5-CV R^2	5-CV MAPE	Test R^2	Test MAPE
LR	**0.969**	**0.067**	0.904	0.140
LASSO	0.966	0.072	**0.972**	**0.066**
Ridge	0.967	0.071	0.953	0.095
Elastic Net	0.947	0.101	0.939	0.099
Decision Tree	0.937	0.089	0.961	0.085
Random Forest	0.965	0.068	0.966	0.069

Table 5. Performance evaluation of the machine learning models

5.2 Prediction Model Selection

We test 5 different machine learning models: linear regression, LASSO regression, ridge regression, elastic net, decision trees, and random forest. To achieve the best performance, we implement 5-fold cross-validation (5-CV) to select the best hyper-parameters (e.g., coefficient value for the regulation term, maximum depth of the tree) for all models. All the training and validation procedures are

No.	Structure	Alliance	LSPs	Requests	Request Configuration
1	1	2	3	18	[6, 6, 6]
2	1	2	3	65	[30, 10, 25]
3	1	2	3	65	[20, 15, 30]
4	1	2	3	65	[25, 5, 35]
5	2	2	6	60	[10, 10, 10, 10, 10, 10]
6	3	3	6	105	[30, 10, 20, 10, 20, 15]
7	3	3	6	105	[40, 5, 25, 10, 15, 10]
8	3	3	6	120	[30, 15, 30, 20, 15, 10]
9	4	4	8	135	[20, 10, 5, 30, 15, 10, 15, 30]
10	4	4	8	135	[15, 15, 5, 20, 25, 15, 20, 20]
11	4	4	8	135	[30, 5, 15, 25, 10, 20, 10, 20]
12	5	5	10	150	[20, 10, 20, 5, 15, 10, 15, 15, 20, 20]
13	5	5	10	180	[30, 15, 25, 20, 15, 30, 10, 10, 10, 15]
14	5	5	10	185	[25, 5, 25, 30, 10, 20, 15, 20, 10, 25]

Table 6. Detail parameters setting for all test instances

implemented in Python 3.7. Table 4 summarizes the average cross-validation R^2 value and the mean absolute percentage value (MAPE). Let l_s denotes the best solutions we get by tabu search, $\hat{l}_s$ denotes the predicted delivery cost for a sample s in each fold S of the training set. The MAPE is defined as: $\frac{1}{|S|}\sum_{t\in S}\frac{|l_s-\hat{l}_s|}{l_s}$.

Based on the evaluation results, all the above machine learning models achieve reasonably good performance on delivery cost prediction. In particular, the LASSO regression model has the lowest test error and highest R^2 score. Besides, LASSO estimates sparse coefficients that reduce the number of features in the model and maintain good interpretability. Hence, we decide to use LASSO as the prediction model in our framework.

5.3 Performance Comparison

This subsection, we compare the results on delivery costs obtained by (1) self-routing by LSPs without collaboration, (2) collaborative routing with alliances solving by tabu search heuristic alone, (3) collaborative routing with alliances solving by proposed learning-based optimization framework and (4) collaborative routing with fully collaboration, which means each LSP can cooperate with each other and exchange requests from both the computational and management perspectives. Table 5 gives the detail configurations for all test instances. That include the instances, alliances structures shown in Figure 3, number of alliances, number of requests and LSPs for the instance. The last column shows the request configuration information which indicates the number of requests belongs to each LSP. And experiments results of all instances are shown in Table 6.

Columns $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{L}$ denote the delivery costs obtained by self-routing without collaboration, collaborative routing with fully collaboration and collaborative routing with alliances solving by our learning-based approach, respectively. Columns $\mathcal{A}_{min}$, $\mathcal{A}_{max}$ denotes the minimal and maximal delivery cost obtained for collaborative routing with alliances after run the tabu search alone 5 times.

No.	$\mathcal{G}$	$\mathcal{I}$	$\mathcal{F}$	$\mathcal{L}$	$\mathcal{A}_{min}$	$\mathcal{A}_{max}$	$\mathcal{S}_1$ (%)	$\mathcal{S}_2$ (%)	$\mathcal{S}_{min}$ (%)	$\mathcal{S}_{max}$ (%)	$\mathcal{T}_1$ (s)	$\mathcal{T}_2$ (s)
1	1609	1894	1582	1642	1609	1666	13.31	16.47	-2.05	1.44	1.0	1.5
2	Fail	5699	4165	4660	4915	5096	18.23	26.92	5.18	8.50	20	50
3	Fail	5263	3958	4706	4587	4784	10.58	24.80	-2.59	1.63	21	51
4	Fail	5067	3855	4804	4747	4887	5.19	23.92	-0.64	1.67	19	47
5	Fail	5659	3705	4550	4678	4785	18.45	34.53	2.81	5.16	18	49
6	-	8990	5685	7337	7805	7935	18.39	36.76	5.99	7.17	43	57
7	-	8499	5711	7342	7613	7763	13.61	32.80	3.79	5.65	40	62
8	-	12857	8955	11058	11286	11886	13.99	30.35	2.02	6.96	59	94
9	-	14978	9472	12856	13434	13628	14.17	36.76	4.30	5.66	65	93
10	-	14969	10187	12711	13205	13676	15.08	31.95	3.74	7.06	64	102
11	-	14416	10536	12302	13313	13700	14.66	26.91	**7.59**	10.20	55	104
12	-	17060	10572	14005	14986	15241	17.91	38.03	6.55	8.11	81	134
13	-	21029	11375	16802	17512	20297	**20.10**	45.91	4.05	**17.22**	106	226
14	-	19850	13645	16550	17166	17576	16.62	31.26	3.58	5.83	87	223

Table 7. Experimental results for all test instances

Columns $\mathcal{S}_1$, $\mathcal{S}_2$ are the cost savings in percentage achieved by collaborative routing with alliance and fully collaboration, compare to self-routing.

For small and medium size instances (instances No.1 to No.5), we also implement the exact method in Gurobi. It find our tabu search method can obtain the optimal solution for Instance 1. And for instances No.2 to No.5, Gurobi fails to give feasible solutions in 3600 seconds. The results obtained are depicted in column $\mathcal{G}$. It shows that our tabu search method can achieve optimal solution as same as Gurobi for instance No.1. However, for medium size instances (instance No.2 to No.5), Gurobi fails to find feasible solutions in 3600 seconds, and both our tabu search and learning-based framework can find good solutions in less than 1 minutes. As shown in columns $\mathcal{S}_1$ and $\mathcal{S}_2$, we can find that both collaboration with alliance and fully collaboration always lead to fewer delivery costs compares to self-routing. Column $\mathcal{S}_{min}$ and $\mathcal{S}_{max}$ are the minimum and maximum savings that the learning framework can achieve compare to the direct use heuristic method (tabu search) alone. We find that for the small and medium size of instances (No.1 to No.5), our learning and optimization framework can obtain solutions as good as tabu search. While for moderate or larger test instances with denser alliance structure graph (No.2 to No.14), our learning framework is about 2% to 10% better than use heuristic method (tabu search) alone, and it can achieve up to 17% cost savings. We also compare the running times of our proposed learning and optimization framework and directly using heuristic method (tabu search), as shown in column $\mathcal{T}_1$ and $\mathcal{T}_2$. It shows that the our new approach needs less computing resources compare to the heuristic method (tabu search), especially in large scale cases.

At last, for a two-stage approach, prediction error will always exists in stage one. Here, to better evaluate the benefits of our learning-based approach, we also investigate the influence of error cascade. We incorporate the errors occurs in prediction stage via an error term $\widetilde{e}$ into request assignment stage. Column $\mathcal{E}$ shows the estimate error of assignment cost will reduce the quality of our learning-based approach, but still better than using tabu search alone.

6 Conclusion

This paper attempted to address an emerging concept in a collaborative urban delivery problem involving multiple alliance structures. Compared to individuals performing optimal planning by LSPs themselves, our experiments show that centralized collaborative routing can potentially reduce the total operating cost by about 20%. Compared to centralized collaborative routing with the direct use of a heuristic algorithm, our experiments show that our learning-based optimization approach can reduce the total operating cost up to 17% with the less computational time required. Furthermore, the learning-based approach is a framework so methodologically, which means we can replace tabu search with any other heuristic methods to improve the results. We observe that (1) more LSPs joining alliances generally produces more cost savings; (2) the alliance structure has a significant impact: the denser the alliance structure is, the more substantial savings we can achieve, which suggests that overlapping alliance structure allows us to perform logistics more sustainably. This saving can be translated into profit-sharing schemes among participating LSPs, thereby incentivizing them to join such an alliance structure. Profit sharing mechanisms are another topic worthy of future works which fall outside the scope of this paper. In the future, we also aims to provide a robust optimization model to handle the errors for cost prediction in the first stage.

References

1. Berger, S., Bierwirth, C.: Solutions to the request reassignment problem in collaborative carrier networks. Transportation Research Part E: Logistics and Transportation Review **46**(5), 627–638 (2010)
2. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society **52**(8), 928–936 (2001)
3. Cruijssen, F., Cools, M., Dullaert, W.: Horizontal cooperation in logistics: opportunities and impediments. Transportation Research Part E: Logistics and Transportation Review **43**(2), 129–142 (2007)
4. Dahl, S., Derigs, U.: Cooperative planning in express carrier networks—an empirical study on the effectiveness of a real-time decision support system. Decision Support Systems **51**(3), 620–626 (2011)
5. Fernández, E., Roca-Riu, M., Speranza, M.G.: The shared customer collaboration vehicle routing problem. European Journal of Operational Research **265**(3), 1078–1093 (2018)
6. Ferrell, W., Ellis, K., Kaminsky, P., Rainwater, C.: Horizontal collaboration: opportunities for improved logistics planning. International Journal of Production Research **58**(14), 4267–4284 (2020)
7. Gansterer, M., Hartl, R.F.: Request evaluation strategies for carriers in auction-based collaborations. OR spectrum **38**(1), 3–23 (2016)
8. Gansterer, M., Hartl, R.F.: Collaborative vehicle routing: a survey. European Journal of Operational Research **268**(1), 1–12 (2018)
9. Glover, F., Laguna, M.: Tabu search. In: Handbook of combinatorial optimization, pp. 2093–2229. Springer (1998)

10. Guajardo, M., Rönnqvist, M., Flisberg, P., Frisk, M.: Collaborative transportation with overlapping coalitions. European Journal of Operational Research **271**(1), 238–249 (2018)
11. Lai, M., Cai, X., Hu, Q.: An iterative auction for carrier collaboration in truckload pickup and delivery. Transportation Research Part E: Logistics and Transportation Review **107**, 60–80 (2017)
12. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. International Journal on Artificial Intelligence Tools **12**(02), 173–186 (2003)
13. Li, J., Rong, G., Feng, Y.: Request selection and exchange approach for carrier collaboration based on auction of a single request. Transportation Research Part E: Logistics and Transportation Review **84**, 23–39 (2015)
14. Lim, M.K., Mak, H.Y., Shen, Z.J.M.: Agility and proximity considerations in supply chain design. Management Science **63**(4), 1026–1041 (2017)
15. Liu, S., He, L., Max Shen, Z.J.: On-time last-mile delivery: Order assignment with travel-time predictors. Management Science (2020)
16. Nicola, D., Vetschera, R., Dragomir, A.: Total distance approximations for routing solutions. Computers & Operations Research **102**, 67–74 (2019)
17. Ouyang, Y., Daganzo, C.F.: Discretization and validation of the continuum approximation scheme for terminal system design. Transportation Science **40**(1), 89–98 (2006)
18. Pérez-Bernabeu, E., Juan, A.A., Faulin, J., Barrios, B.B.: Horizontal cooperation in road transportation: a case illustrating savings in distances and greenhouse gas emissions. International Transactions in Operational Research **22**(3), 585–606 (2015)
19. Savelsbergh, M., Van Woensel, T.: 50th anniversary invited article—city logistics: Challenges and opportunities. Transportation Science **50**(2), 579–590 (2016)
20. Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y.: A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation science **31**(2), 170–186 (1997)
21. Verdonck, L., Caris, A., Ramaekers, K., Janssens, G.K.: Collaborative logistics from the perspective of road transportation companies. Transport Reviews **33**(6), 700–719 (2013)
22. Zhang, M., Pratap, S., Huang, G.Q., Zhao, Z.: Optimal collaborative transportation service trading in b2b e-commerce logistics. International Journal of Production Research **55**(18), 5485–5501 (2017)