Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 5052897, 20 pages
https://doi.org/10.1155/2022/5052897

Research Article

The Value of Preemptive Pick-Up Services in Dynamic Vehicle Routing for Last-Mile Delivery: Space-Time Network-Based Formulation and Solution Algorithms

Weiran Meng,[1] **Lingyun Meng**,[1] **Guoshuai Han**,[2] **Xiaotian Zhuang**,[3] **Lu Carol Tong**,[4,5] **and Shengnan Wu**[3]

[1]*School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China*
[2]*School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*
[3]*Department of Intelligent Supply Chain, Beijing Jingdong Zhenshi Information Technology Co., Ltd., Beijing 100176, China*
[4]*Research Institute of Frontier Science, Beihang University, Beijing 100191, China*
[5]*National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, Beijing 100191, China*

Correspondence should be addressed to Lu Carol Tong; ltong@buaa.edu.cn and Shengnan Wu; wushengnan1@jd.com

Received 10 March 2022; Accepted 12 August 2022; Published 24 September 2022

Academic Editor: Eneko Osaba

Copyright © 2022 Weiran Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, with the increase of emerging pick-up requests during service, logistics companies have been driven to integrate delivery and pick-up service in a dynamic environment. To provide a balanced and robust approach to cope with delivery requests and emerging pick-up requests, this article aims at considering and modeling a practically useful service principle as preemptive services. To our knowledge, most existing studies assume that the dynamically arriving requests are handled in a non-preemptive processing sequence; that is, once the delivery person is allocated to a task, the process is noninterruptible till it gets completed. In the preemptive service, a service suspension of the delivery process (with low service utility) is allowed to satisfy the pick-up requests (with high service utility) first. To provide a systematic assessment on the value of preemptive service for evolving urban logistics systems, a dynamic vehicle routing problem with preemptive pick-up service (VRPPS) is proposed to systematically describe the problem with potentially complex dynamic priorities among different tasks. Based on a dynamically constructed space-time network, this study formulates a multicommodity flow model that aims at optimizing the generalized service utility and the operating cost simultaneously. To provide a fast value approximation, we present a solution framework deploying the augmented Lagrangian relaxation approach with embedded dynamic programming algorithms. This framework jointly integrates the processes of updating request information and obtaining optimal routes. Finally, the validity and effectiveness of the proposed methods are evaluated on an illustrative network and a real-world last-mile delivery network operated by a logistics company.

1. Introduction

Over the past few decades, delivery and pick-up services have been developed and operated, respectively. Unlike deliveries, which are usually required by customers before service, the majority of pick-up requests appear during service, with tight time windows and high priority. In order to cope with emerging pick-up requests and enhance service efficiency, it is necessary to explore an effective way to combine delivery and pick-up service together. We consider the process of handling delivery and pick-up requests in a multitask scheduling environment. That is, delivery and pick-up service can be regarded as two different types of tasks. The principle of preemptive service is introduced from the field of queueing models and further considered in this study, assuming that delivery people could interrupt their delivery service (with low service utility) to process the newly generated pick-up orders (with high service utility). It is worth studying because delivery people could cooperate with each other to handle emerging pick-ups. In addition, more overall profits could be earned. However, the case of preemptive service has not been fully examined in the

existing dynamic vehicle routing literature. Accordingly, the introduction of preemptive service in this article aims at providing a more flexible service principle to find the service sequence with the maximum profits (the difference between service utility and operating cost) in the last-mile delivery process.

The last-mile delivery process is typically formulated as a vehicle routing problem with time window (VRPTW). Researchers have contributed significantly on modeling methods and solution algorithms for VRPTW. From a modeling perspective, general considerations of VRPTW are summarized as capacity constraints [1, 2] and routing time constraints [3]. With regard to solution algorithms, typical methods are divided into heuristics and exact algorithms. Considering that VRPTW is a NP-hard problem, heuristics are often used as a fast and effective method to achieve optimized solutions. A vehicle routing problem with a soft time window was proposed by Taillard et al. [2] and solved by implementing the tabu search algorithm. Ombuki et al. [4] presented a genetic algorithm to simultaneously minimize the total number of vehicles and total cost. On the other hand, though converging much slower than heuristics, exact algorithms demonstrate advantages on quantifying the optimality gap and evaluating solution equality. Based on the column-generation technique, Desrochers, Desrosiers, and Solomon [1] proposed a branch-and-bound algorithm to generate lower bound and a dynamic programming algorithm to obtain feasible columns. Moreover, the Lagrangian relaxation (LR) technique has been widely implemented [5, 6]. As an augmented form of LR, the alternating direction method of multipliers (ADMM) framework was applied and developed by Yao et al. [7] to decompose VRPTW under space-time-state network. The computational results demonstrated a better quality of solution than the LR-based method. As an extension of VRPTW, the vehicle routing problem with pick-up and delivery (VRPPD) was proposed to consider delivery person's behavior. Mahmoudi and Zhou [8] formulated VRPPD as a multicommodity flow problem in a space-time-state network without adding complex constraints. Tong et al. [9] presented a customized bus service network design problem and utilized a Lagrangian-based algorithm to acquire the optimal route. Zhao et al. [10] studied a pick-up and delivery problem that considers vehicle routing plan and ride-share matching strategy simultaneously, and developed a Lagrangian relaxation-based approach to solve the problem.

When the information cannot be revealed in advance, VRPTW turns into a dynamic scheduling problem (DVRPTW). Chen and Xu [11] proposed a DVRP with hard time windows to minimize total distance along the route and acquire an optimized path at each decision epoch using column generation. Montemanni et al. [12] conducted a tabu search strategy under an ant colony system to solve a dynamic vehicle routing problem with deterministic time windows. For stochastic DVRPTW, information is revealed over time following given probability distributions. Yang, Jaillet, and Mahmassani [13] formally introduced a real-time multivehicle truckload pick-up and delivery problem and proposed several reoptimization policies, with the consideration of varying traffic intensities, request information, and flexibility for request-rejection decisions. Given that additional knowledge becomes available as time evolves, algorithms like stochastic modeling [14–16] are widely utilized to capture uncertainty.

It is important to highlight the principle of dynamic scheduling in a multitasking environment. In preemptive scheduling, processing of high-priority tasks is done first by interrupting low-priority tasks. In non-preemptive scheduling, if subsequent high-priority task arrives, it will wait until the execution of earlier task. It should be noted, there are still different scheduling rules such as FCFS, EDT, to handle the dynamic changes in the task priority.

To our knowledge, the paper by Ulmer, Thomas, and Mattfeld [17] first studied a dynamic vehicle routing problem with preemptive depot returns caused by new delivery requests received during service. In this article, we study preemptive pick-up services and corresponding customer returns.

Table 1 compares the key elements of DVRPTW models in detail. From the literature mentioned above, we can see that researchers have contributed substantially to DVRP, but few studies consider the service in a real-world multitasking environment. Thus, we aim at maximizing the benefits of serving both the delivery and pick-up requests while minimizing total operating cost.

In this study, we focus on handling delivery and pick-up requests efficiently in a multitask environment by incorporating preemptive pick-up service into the vehicle routing problem. A solution framework, which contains the iterative process between updating request information and obtaining optimal routes at each decision epoch, is presented to formulate and solve the proposed problem .

The potential contribution of this article is listed as follows:

(1) We formally introduce the concept of preemptive pick-up service into vehicle routing problem (VRPPS) to jointly capture the characteristics of both delivery and pick-up requests. Specifically, according to the detailed DVRP taxonomy proposed by Psaraftis, Wen, and Kontovas [18], VRPPS under consideration is an integration of dynamic and deterministic DVRP, which considers customer order rejection and pick-up requests and, in particular, allows preemptive service, as shown in Figure 1.

(2) We present a discretized space-time network-based solution framework for VRPPS, which systematically integrates request information updates and optimal route acquisition. The use of the space-time networks can allow exact modeling of the return-to-current customer behavior after a preemptive request. The linear integer programming optimization model is solved over a time horizon due to the dynamic nature of pick-up requests during service, under the augmented Lagrangian relaxation decomposition approach. The proposed method can be well applied to handle delivery and pick-up requests at the same time.

(3) We demonstrate the impact of preemptive service and delivery people cooperation through an

Table 1: Comparison of key elements in DVRPTW.

Objective	Service principles	Decision variables	Main constraints	Solution algorithm	Publication
Minimizing the long-run average cost per requested job	Non-preemptive	Time-dependent truck loading state	TW	Reoptimization policy	Yang et al. [13]
Minimizing total travel time	Non-preemptive	Vehicle route	TW; CP	Ant colony system	Montemanni et al. [12]
Minimizing the expected total travel time plus lateness	Non-preemptive	Vehicle-customer type assignment	TW	Parallel tabu search	Ichoua, et al. [15]
Minimizing total distance	Non-preemptive	Vehicle route	TW	Dynamic column generation	Chen and xu [11]
Minimizing total expected cost	Non-preemptive	Customer state	TW; CP; SD	Adaptive variable neighborhood search	Pillac et al. [16]
Maximizing total expected rewards	Preemptive (depot return for delivery requests)	Customer state	TW	Approximate dynamic programming	Ulmer et al. [17]
Maximizing the difference between service utility and operating cost	Preemptive (delivery process interruption for pick-up requests)	Space-time route (with customer return)	STW; SDC	Augmented Lagrangian relaxation method	This paper

Constraints. TW: time window constraint; CP: capacity constraint; SD: side constraint; STW: space-time window constraint; SDC: service duration constraint.

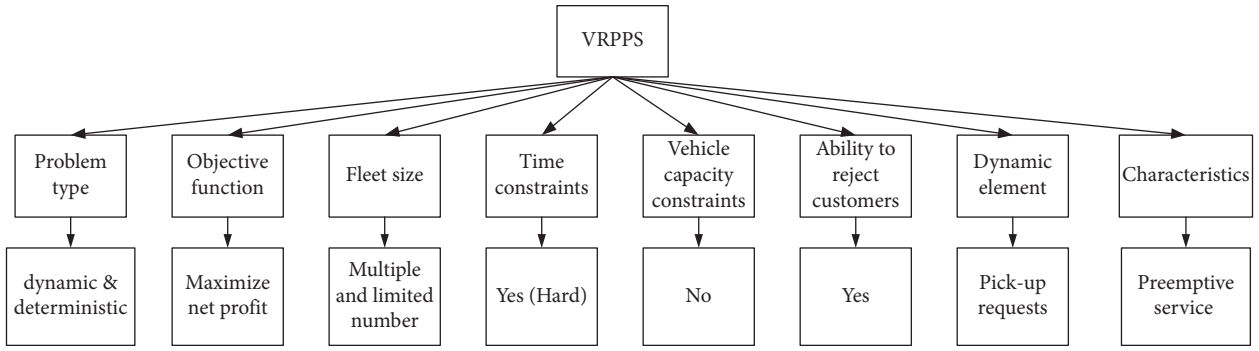


Figure 1: DVRP taxonomy for the proposed VRPPS based on the framework by Psaraftis, Wen and Kontovas [18].

illustrative example, and test the validity and efficiency of the proposed model and algorithm on a real-world network.

The remainder of this article is organized as follows. The next section presents the vehicle routing problem considering preemptive pick-up service within a space-time network. In Section 3, we propose a solution framework integrating the augmented Lagrangian relaxation method and dynamic programming algorithm. In Section 4, we make a comparison between the proposed augmented Lagrangian relaxation method and other approaches. Section 5 performs numerical experiments based on an illustrative network and a real-world network. Section 6 provides conclusions and discusses future research.

2. Vehicle Routing Problem with Preemptive Pick-Up Service (VRPPS) and Service-Dependent Utility Considerations

2.1. Problem Statement. We incorporate preemptive service into a last-mile delivery problem in a multitask environment. In this problem, delivery service (with low utility and loose time window) and pick-up service (with high utility and tight time window) can be viewed as two different tasks. Preemptive service means that delivery process can be interrupted by a to-be-served pick-up requests, while non-preemptive service means that delivery process is consecutive. The problem is designed for logistics companies to manage delivery and pick-up service simultaneously.

We depict the problem in an illustrative way, as shown in Figure 2. Here, we compare two service modes as "non-preemptive service" and "preemptive service." In these two modes, the vehicle can carry delivery packages (blue packages, with low service utility and loose time window) and pick-up packages (red packages, with high service utility and tight time window) at the same time. The delivery service begins and ends at depot within a loose time window [*EDT, LAT*], where *EDT* and *LAT* denote the earliest departure time from depot and latest arrival time to depot respectively. Travel cost is considered between customers. The delivery request for each customer is two packages. The vehicle follows an initial routing plan (depot-1-2-3-depot), until a pick-up request occurs at customer 3 when the vehicle is delivering goods at customer 2. Under "non-preemptive service," the delivery person *reneges* on the pick-up request and gives up its utility because he/she cannot serve it within time window; under "preemptive service," considering the

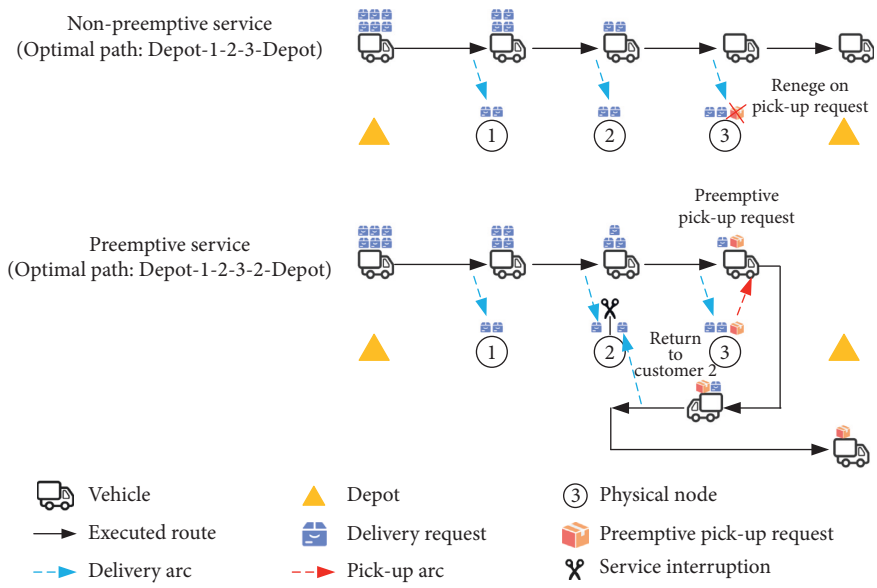


Figure 2: Illustration of non-preemptive and preemptive services.

principle of preemptive scheduling, the vehicle interrupts the delivery process at customer 2 to pick up the emerging request at customer 3 and then returns to customer 2 to complete the remaining delivery requests. This problem aims at finding a service sequence with a maximum profit (the difference between total utility and total operating cost) once the to-be-served pick-up requests occur during service. The value of total profit should be compared very systematically. The "preemptive service" can be better than "non-preemptive service," only if the utility of serving pick-up request is much higher than the additional travel cost due to service interruption and customer return. As a result, a well-defined formulation and efficient solution algorithm are critically needed to offer a solid assessment and optimization of different service plans.

Inspired by the phenomenon that the daily delivery requests are often interrupted by the urgent pick-up orders, we make distinguished assumptions for different types of requests to depict the real-world situation. In addition to practical conditions, classical assumptions should be taken into account. The characteristics of last-mile delivery are illustrated by [19], containing customer service levels, security and delivery types, geographical area and market density/penetration, vehicle fleet, and the environmental factor. For simplicity, we consider delivery types, vehicle fleet, and time window in the problem. The following assumptions are adopted in the underlining mathematical model:

(1) Only delivery process can be interrupted
(2) Each customer is assumed to be a physical node
(3) Utility is defined as the service profit per time unit at physical nodes
(4) Utility of serving pick-up requests is higher than that of delivery requests
(5) Travel cost between customers is much lower than the value of service utility
(6) Vehicles do not have capacity restrictions
(7) Service duration at physical nodes is considered in both delivery and pick-up process
(8) Each physical node can be served more than once
(9) Delivery requests are required to be served within a loose time window [*EDT*, *LAT*], while pick-up requests have much tighter time window

We establish an optimization model under the space-time network to describe VRPPS. We use a directed graph $G = (N, E)$ to represent a physical network, including a set of physical nodes N and a set of arcs E. The set N consists of a depot node O and a set of physical nodes P. The arcs in set E represent road segments. Meanwhile, a space-time network is extended from the physical network G by extending physical nodes to request nodes, and adding a time dimension. The space-time network contains a set of space-time vertices V and a set of space-time arcs A, where vertex (i,t) represent both time t and request node i, and arc (i,j,t,t') denotes a space-time arc starting from request node i at time t and arriving at request node j at time t'.

Given a finite set of vehicles K, delivery requests, pick-up requests, and corresponding time windows, the VRPPS aims at maximizing the difference between total service utility and operating cost, and obtain optimal space-time routes for each vehicle k when a pick-up request occurs at time τ. Table 2 lists the sets, indexes, and parameters used in the article.

2.2. Conceptual Illustration for Systematic Value Calculations. In this section, we explain the process of *preemptive service* in detail through a three-node illustrative example. We set the utility as \$10 per time unit for serving a delivery request and \$15 per time unit for a pick-up request. A physical network is given with 1 depot, 3 physical nodes, and 6 directed transportation links (marked with the travel cost), as shown in Figure 3. One vehicle is available to serve the

Table 2: Sets, indexes, and parameters for model establishment.

Symbol	Definition
O	Index of depot
P	Set of physical nodes in physical network
N	Set of nodes in physical network, $N = P \cup \{O\}$
E	Set of directed links in physical network
T	Set of time stamps
K	Set of vehicles
V	Set of space-time vertexes
A	Set of space-time arcs
A_k	Set of space-time arcs for vehicle $k \in K$
A^T	Set of space-time traveling arcs
A^S	Set of space-time serving arcs
τ	Index of decision epoch
k	Index of vehicles
p	Index of physical nodes
P_τ	Set of physical nodes that still have to-be-served requests until time τ
R_τ	Set of to-be-served request nodes until time τ
$R_{p,\tau}$	Set of to-be-served request nodes generated from physical node p until time τ
$\Psi_{r,k}$	Set of arriving arcs that end at request node r in vehicle k's network
$o_{k,\tau}$	Origin of vehicle k at time τ
d_k	Destination of vehicle k
$q_{p,\tau,\text{num}}$	The num$^{\text{th}}$ request node generated at physical node p until time τ
EDT	The earliest departure time from depot
LAT	The latest arrival time back to depot
SL_r	Required service duration for request node r
SL_p	Required service duration for physical node p
$T_{p,\tau}$	Remaining service duration for physical node p until time τ
i, j, r	Indices of request nodes
t, t'	Indices of time stamps
(i,t), (j,t')	Indices of space-time vertexes
(i, j, t, t')	Index of space-time arc
$c_{i,j,t,t'}$	Transportation cost on arc (i, j, t, t')
$u_{i,j,t,t'}$	Utility on arc (i, j, t, t')

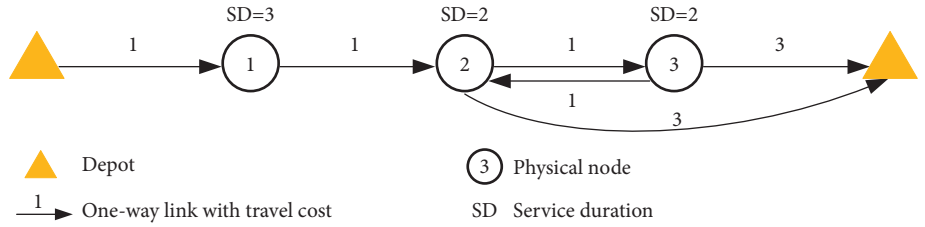


Figure 3: Physical network for a 3-node example.

Table 3: Basic information of requests in the 3-node example.

Request type	Corresponding physical node	Required service duration	Preferred time window
Delivery	1	3	[0,15]
Delivery	2	2	[0,15]
Delivery	3	2	[0,15]
Pick-up	**3**	**1**	[6, 7]

requests. The required service duration and preferred time window for each physical node are shown in Table 3. Note that only the delivery process could be suspended.

To emphasize the advantage of preemptive service, we compare the optimal solution results under non-preemptive (Scenario 1) and preemptive service (Scenario 2) in Table 4. As shown in the table, we use decision tree to explain the process of obtaining optimal path. In Scenario 1, following the optimal path, the vehicle arrives at physical node 3 at $\tau = 8$, so the pick-up request cannot be served. In Scenario 2, the delivery service at physical node 2 can be interrupted, so the decision tree extends a new branch compared with Scenario 1. In the optimal path, the vehicle firstly serves physical node 2 for one time unit, then heads to physical

Table 4: Comparison between non-preemptive and preemptive service.

	Scenario 1: Non-preemptive service	Scenario 2: Preemptive service
Similarities	(i) Bellman equation-based dynamic programming procedure is used	
Differences	(i) Delivery service is uninterruptible (ii) Classical Bellman equation is used	(i) delivery service is interruptible (ii) Bellman equation is modified due to the division of service duration

Optimal path	Depot-1-2-3-depot	Depot-1-2-3-2-depot
Interrupt service at physical node 2?	No	Yes
Travel cost	6	7
Delivery utility	$10^*7 = 70$	$10^*7 = 70$
Pick-up utility	0	15
Objective function value	$70 + 0 - 6 = 64$	$70 + 15 - 7 = 78 > 64$

node 3 for satisfying the pick-up request at $\tau = 7$, and finally returns to physical node 2 at $\tau = 11$. It turns out that the solution result in Scenario 2 incurs a higher travel cost but earns more utility than Scenario 1.

2.3. Space-Time Network Construction for Preemptive Service with Possible Interruptions and Return-to-Customer Requirements. In this section, the process of handling delivery and pick-up tasks is described using a space-time network representation. The reason why we use space-time network is that it can model preemptive service spatially and temporally. Under preemptive service, multiple visits of a customer can be described by generating requests nodes from a physical node.

2.3.1. Generating Request Nodes in Space-Time Network to Model Service Interruptions. To distinguish different requests generating at the same physical node and model service interruptions, we introduce **request nodes** into space-time network. The number of request nodes depends on how many time units the physical node has remained. For example, physical node 1 has one delivery request (with a service duration of three time units) and one pick-up request (with a service duration of one time unit). Therefore, physical node 1 has four corresponding request nodes (1a, 1b, 1c, 1d), each of which has a service duration of one time unit, as shown in Figure 4. It ensures that the delivery process can be interrupted.

2.3.2. Dynamic Construction of Space-Time Network with Preemptive Pick-Up Requests. Since the pick-up request can occur at any time during service, we generate the space-time network dynamically to capture the time-dependent request information. Assume that a pick-up request, with time window $[st, et]$ and service duration l, occurs at time $\tau \in (\text{EDT}, \text{LAT}]$, requested by a physical node $p \in P_\tau$. The corresponding space-time network is updated following the steps:

Step 1: Update existing space-time vertices and arcs until time τ

$***$ **For** each space-time vertex $(i, t) \in V$

If $i \in R_\tau$ **and** $t \geq \tau$

then reserve (i, t) in set V and reserve its corresponding space-time arcs from set A

else delete (i, t) from set V and delete its corresponding space-time arcs from set A

Step 2: Generate request nodes corresponding to the pick-up request

If there are already m (where $m = \|R_{p,\tau}\|$) request nodes generated at physical node p before time τ

then add the request node $q_{p,\tau,m+1}$, which corresponds to the preemptive request, to set R_τ and $R_{p,\tau}$

Step 3: Add space-time vertices and arcs of the preemptive pick-up request

Step 3.1: Add space-time vertices to existing space-time network

For each time $t \in [st, et + l]$

Add $(q_{p,\tau,m+1}, t)$ to set V

End for

Step 3.2: Add space-time arcs to existing space-time network

 # Add serving arcs at $q_{p,\tau,m+1}$

For each time $t \in [st, et]$

Add $(q_{p,\tau,m+1}, q_{p,\tau,m+1}, t, t + l)$ to set A

End for

 # Add inflow and outflow arcs from/to $q_{p,\tau,m+1}$

Add outflow arcs from $q_{p,\tau,m+1}$

For each time $t \in [st, et]$

Add $(q_{p,\tau,m+1}, j, t, t\prime)$ to set A

End for

Add inflow arcs to $q_{p,\tau,m+1}$

For each time $t' \in [st, et]$

Add $(i, q_{p,\tau,m+1}, t, t')$ to set A

End for

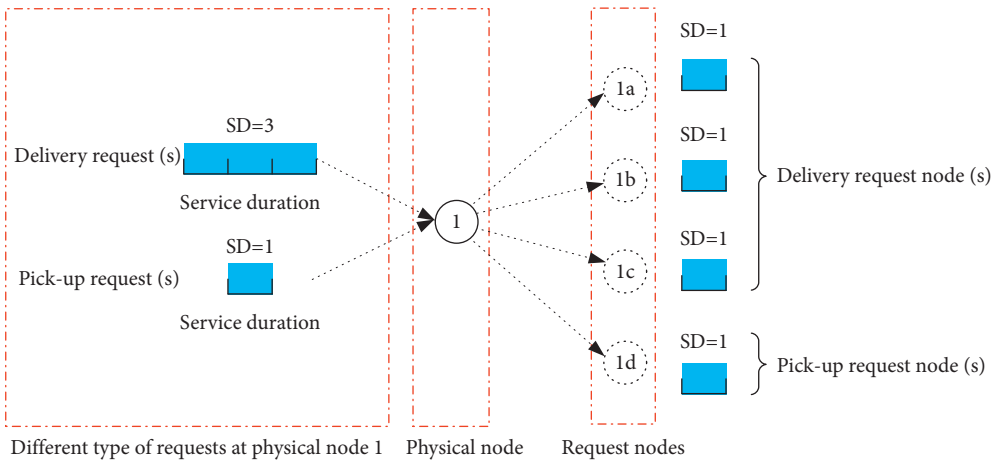
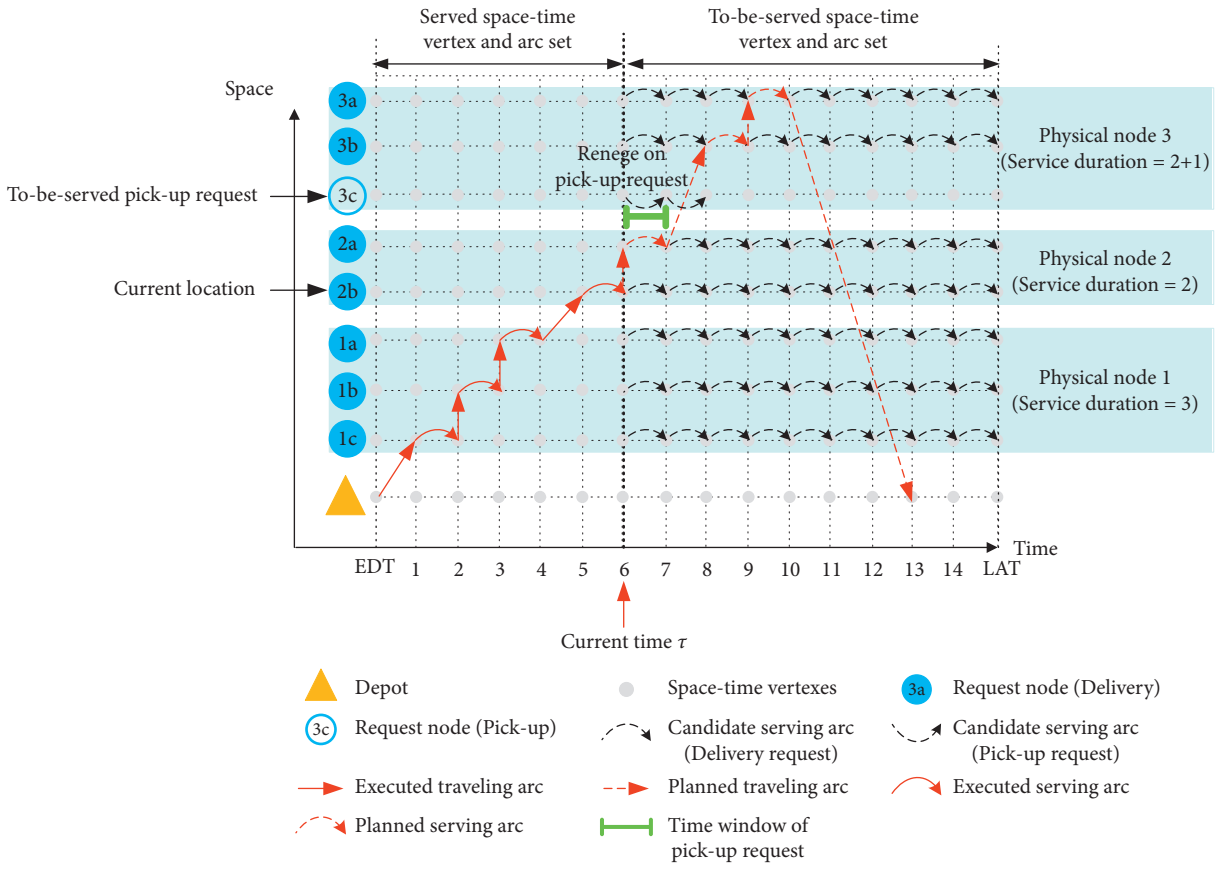


Figure 4: The process of generating request nodes.



(a)

Figure 5: Continued.

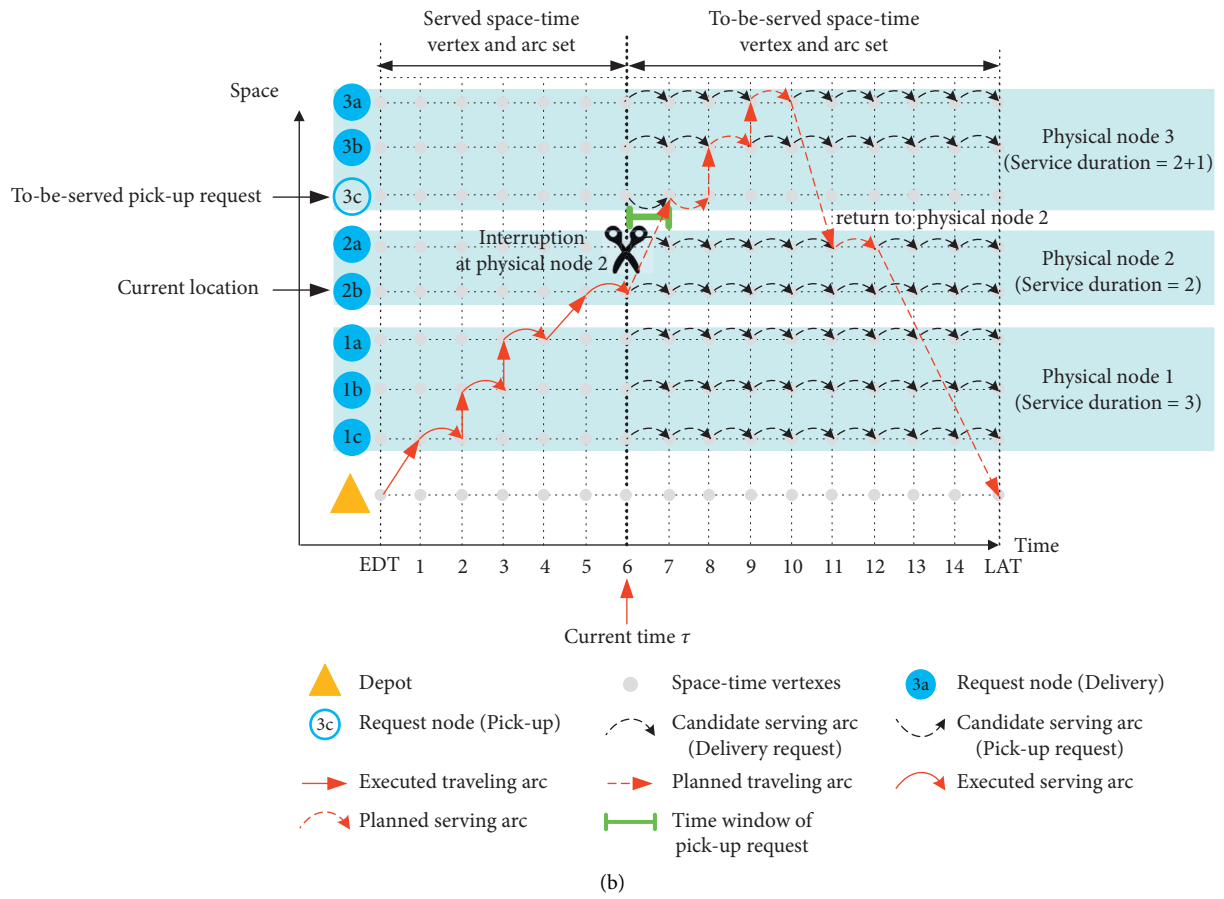


FIGURE 5: Space-time network in a 3-node example. (a) Space-time network for 3-node example (non-preemptive service) and (b) space-time network for 3-node example (preemptive service).

2.3.3. An Illustrative Example for Handling Preemptive Pick-Up Requests in Space-Time Network. Here, we use the 3-node example mentioned in Section 2.2 to demonstrate how the space-time network is constructed for describing preemptive service. In this case, a pick-up request appears at physical node 3 (denoted as request node 3c) when $\tau = 6$, with a very tight time window [6, 7], while delivery requests have loose time window [EDT, LAT]. The corresponding serving arcs of pick-up request can be selected as (3c,3c,6,7) or (3c,3c,7,8). The service duration is set to be one time unit.

Under non-preemptive service, the delivery process is consecutive, as shown in Figure 5(a). From the planned path (red dotted line), we can see that the delivery person reneges on the pick-up request because he/she cannot arrive at physical node 3 within time window [6, 7], after fully complete the delivery task at physical node 2.

Under preemptive service, the delivery process can be interrupted by pick-up request, as depicted in Figure 5(b). From the newly generated planned path (red dotted line), we can see that the delivery person will interrupt the delivery service at physical node 2 and head to request node 3c at $\tau = 7$. The service process of this pick-up request lasts for one time unit, represented with a space-time arc (3c,3c,7,8). To accomplish the remaining work at physical node 2, the delivery person will revisit physical node 2 at $\tau = 11$, denoted as a space-time traveling arc (10,11,3a,2a).

2.4. Dynamic Space-Time Network-Based Optimization Model. The vehicle routing problem with preemptive service can be formulated as a multicommodity flow model with time window constraints based on the constructed space-time network. The travel cost on a space-time arc is defined as $c_{i,j,t,t'}$, the utility is denoted as $u_{i,j,t,t'}$, and both of them are set to be constant in the model. It is worth noting that our model is used for optimizing the vehicles' route, which is determined by decision variable $x^k_{i,j,t,t'}(\tau)$ whenever pick-up requests occur at time τ, as shown in Table 5. The objective function is expressed as formula (1) and is subject to constraints (2), (4), and (5). The pick-up and delivery time windows are represented in the constructed space-time network, without adding extra constraints.

Objective Function. The objective function is to maximize the difference between total service utility and total travel cost. We transform this into a *minimization problem* in expression (1). The penalty for early/late arrival is not considered as we demand that each customer to be served within their time windows. The first item in the expression indicates the total

Table 5: Decision variable for model establishment.

Symbol	Definition
$x^k_{i,j,t,t'}(\tau)$	Vehicle routing variable (=1, if space-time arc (i,j,t,t') is selected by vehicle k at time τ; = 0, otherwise)

travel cost, and the second item shows the total service utility.

In the model, $u_{i,j,t,t'}$ is the utility of selecting space-time serving arc (i,j,t,t'). If $(i,j,t,t')\epsilon A^S$, then $u_{i,j,t,t'}$ is set to be a positive value; otherwise, set $u_{i,j,t,t'} = 0$. The utility of pick-up requests is considered to be higher than the utility of delivery requests, but we do not have to particularly identify them in formula (1). $c_{i,j,t,t'}$ represents the travel cost of arc (i,j,t,t'). Note that if arc (i,j,t,t') is not in set A^T,$c_{i,j,t,t'} = 0$.

$$L = \min \sum_{k\in K} \sum_{(i,j,t,t')\in A_k} c_{i,j,t,t'} x^k_{i,j,t,t'}(\tau) - \sum_{k\in K} \sum_{(i,j,t,t')\in A_k} u_{i,j,t,t'} x^k_{i,j,t,t'}(\tau). \tag{1}$$

Flow Balance Constraint. Constraint (2) ensures the balance of inflow and outflow for each space-time vertex (i,t). $o_{k,\tau}$ and d_k represent the origin and destination of vehicle k until time τ:

$$\sum_{(i,j,t,t')\in A_k} x^k_{i,j,t,t'}(\tau) - \sum_{(j,i,t',t)\in A_k} x^k_{j,i,t',t}(\tau) = \begin{cases} 1, & i = o_{k,\tau}, t = \tau, \\ -1, & i = d_k, t = \mathrm{LAT}, \quad \forall k \in K. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Request Satisfaction Constraint. To describe the interruption of delivery tasks, each physical node $p \in P_\tau$ can be visited more than once to satisfy the delivery requests. To achieve this, the remaining service duration at p should be fully satisfied, as shown in the following expression:

$$\sum_{k\in K} \sum_{r\in R_{p,\tau}} \sum_{(i,j,t,t')\in\Psi_{r,k}} \mathrm{SL}_r \cdot x^k_{i,j,t,t'}(\tau) = T_{p,\tau}, \quad \forall p \in P_\tau. \tag{3}$$

However, it is hard to cope with expression (3) mathematically. We have to formulate an equivalent constraint that is easy to be solved. Since the requests for each physical node are represented by request nodes in space-time network, we only need to ensure that each space-time vertex is served only once. We use the following constraint to replace expression (3):

$$\sum_{k\in K} \sum_{(i,j,t,t')\in\Psi_{r,k}} x^k_{i,j,t,t'}(\tau) = 1, \quad \forall r \in R_{p,\tau}. \tag{4}$$

Decision Variable. As shown in expression (5), decision variable $x^k_{i,j,t,t'}(\tau)$ is equal to 1 if space-time arc (i,j,t,t') is selected by vehicle k at time τ; otherwise, it is equal to 0:

$$x^k_{i,j,t,t'}(\tau) = \begin{cases} 0, & \text{otherwise,} \\ 1, & arc(i,j,t,t') \text{ is selected by vehicle } k \text{ at time } \tau. \end{cases} \tag{5}$$

3. Augmented Lagrangian Relaxation-Based Solution Framework

3.1. Overall Solution Framework. We propose a solution framework, which integrates the optimization of the routing plan and updates of pick-up request information to handle the VRPPS problem systematically, as shown in Figure 6.

3.2. Reformulation of the VRPPS Model. In the proposed VRPPS model, we have to handle the coupling constraint (4). Using the Lagrangian relaxation method, we can add constraint (4) to the original objective function L with Lagrangian multipliers π_r for each request node and transform it into equation L_{1-a}.

3.2.1. Lagrangian Relaxation Problem (Model 1-a). Objective function:

$$L_{1-a} = \min \sum_{k\in K} \sum_{(i,j,t,t')\in A_k} c_{i,j,t,t'} x^k_{i,j,t,t'}(\tau) - \sum_{k\in K} \sum_{(i,j,t,t')\in A_k} u_{i,j,t,t'} x^k_{i,j,t,t'}(\tau) + \sum_{r\in R_{p,\tau}} \pi_r \left(\sum_{k\in K} \sum_{(i,j,t,t_1)\in\Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right), \tag{6}$$

s.t. constraints (2) and (5).

Lagrangian relaxation problem for each vehicle (*Model 1-b*).

In order to optimize each individual vehicle's route in sequence, the Lagrangian problem (denoted as *Model 1-a*) is further decomposed for each vehicle (denoted as *Model 1-b*):

$$L_{1-b} = \min \sum_{(i,j,t,t')\in A_k} \left(c_{i,j,t,t_1} - u_{i,j,t,t'} \right) x^k_{i,j,t,t'}(\tau) + \sum_{r\in R_{p,\tau}} \pi_r \left(\sum_{(i,j,t,t_1)\in\Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right), \tag{7}$$

s.t. constraints (2) and (5).

To avoid solution symmetry in the fundamental Lagrangian-based approach as mentioned by [20], *Model 1-b* is converted into augmented Lagrangian form by implementing the augmented Lagrangian relaxation technique by Yao et al. [7]. Therefore, we can obtain *Model 2-a*.

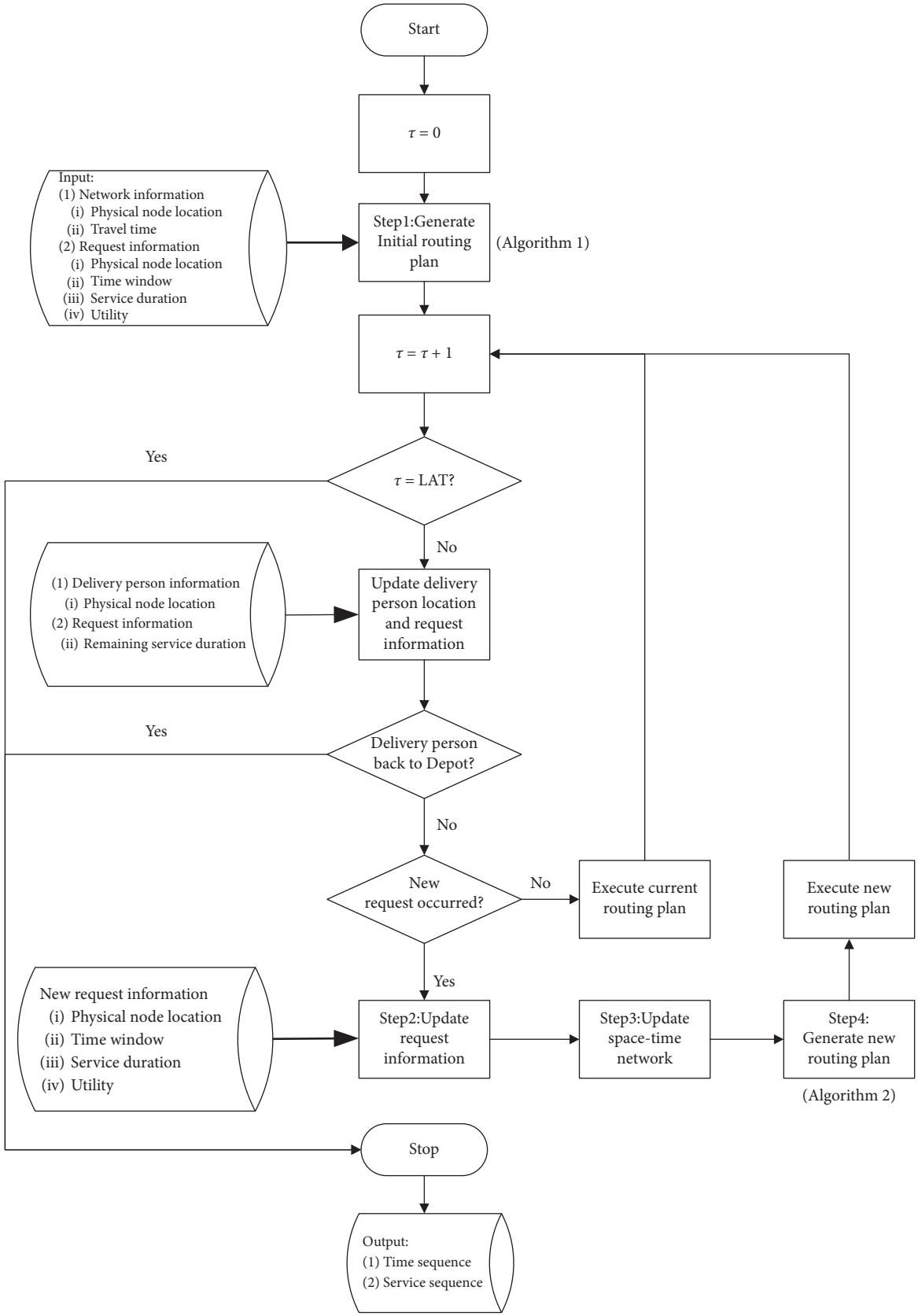


Figure 6: Flowchart of the overall solution framework.

3.2.2. Augmented Lagrangian Problem (Model 2-A). Objective function:

$$\begin{aligned} L_{2-a} = \min & \sum_{k\in K}\sum_{(i,j,t,t')\in A_k} c_{i,j,t,t'}x^k_{i,j,t,t'}(\tau) \\ & - \sum_{k\in K}\sum_{(i,j,t,t')\in A_k} u_{i,j,t,t'}x^k_{i,j,t,t'}(\tau) \\ & + \sum_{r\in R_{p,\tau}} \pi_r \left(\sum_{k\in K}\sum_{(i,j,t,t_1)\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right) \\ & + \frac{\rho}{2}\sum_{r\in R_{p,\tau}} \left(\sum_{k\in K}\sum_{(i,j,t,t_1)\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right)^2, \end{aligned} \tag{8}$$

s.t. constraints (2) and (5).

Assuming that

$$\alpha^k_r = \sum_{k_1\in K\mid\{k\}}\sum_{(i,j,t,t_1)\in \Psi_{r,k_1}} x^k_{i,j,t,t'}(\tau) \quad \forall r \in R_{p,\tau}, \tag{9}$$

then L_{2-a} can be reformulated as

$$\begin{aligned} L_{2-a} = \min & \sum_{k\in K}\sum_{(i,j,t,t')\in A_k} c_{i,j,t,t'}x^k_{i,j,t,t'}(\tau) \\ & - \sum_{k\in K}\sum_{(i,j,t,t')\in A_k} u_{i,j,t,t'}x^k_{i,j,t,t'}(\tau) \\ & + \sum_{r\in R_{p,\tau}} \pi_r \left(\sum_{k\in K}\sum_{(i,j,t,t_1)\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right) \\ & + \frac{\rho}{2}\sum_{r\in R_{p,\tau}} \left(\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) + \alpha^k_r - 1 \right)^2. \end{aligned} \tag{10}$$

To eliminate quadratic terms in the objective function L_{2-a}, we conduct the following calculation:

$$\begin{aligned} & \left(\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) + \alpha^k_r - 1 \right)^2 = \left(\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) \right)^2 \\ & + 2\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau)\left(\alpha^k_r - 1\right) + \left(\alpha^k_r - 1\right)^2 \\ & = \sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) + 2\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau)\left(\alpha^k_r - 1\right) + \left(\alpha^k_r - 1\right)^2. \end{aligned} \tag{11}$$

Subproblems are constructed for each vehicle as *Model 2-b*.

Augmented Lagrangian problem for each vehicle (*Model 2-b*):

$$\begin{aligned} L_{2-b} = \min & \sum_{(i,j,t,t')\in A_k} \left(c_{i,j,t,t_1} - u_{i,j,t,t'} \right) x^k_{i,j,t,t'}(\tau) \\ & + \sum_{r\in R_{p,\tau}} \pi_r \left(\sum_{(i,j,t,t_1)\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) - 1 \right) \\ & + \frac{\rho}{2}\sum_{r\in R_{p,\tau}} \left[\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau) + 2\sum_{(i,j,t,t')\in \Psi_{r,k}} x^k_{i,j,t,t'}(\tau)\left(\alpha^k_r - 1\right) + \left(\alpha^k_r - 1\right)^2 \right] \\ & = \min \sum_{(i,j,t,t')\in A_k} a_{i,j,t,t_1}x^k_{i,j,t,t'}(\tau) \\ & + \sum_{r\in R_{p,\tau}}\sum_{(i,j,t,t_1)\in \Psi_{r,k}} b_r x^k_{i,j,t,t'}(\tau) + C, \end{aligned} \tag{12}$$

s.t. constraints (2) and (5).

Here, parameters a_{i,j,t,t_1}, b_r, and C are calculated as follows:

$$\begin{aligned} a_{i,j,t,t_1} &= c_{i,j,t,t_1} - u_{i,j,t,t'}, \\ b_r &= \pi_r + \frac{\rho}{2}\left(2\alpha^k_r - 1\right), \\ C &= \sum_{r\in R_{p,\tau}} \left[\frac{\rho}{2}\left(\alpha^k_r - 1\right)^2 - \pi_r \right]. \end{aligned} \tag{13}$$

3.3. Solution Algorithms Using the Augmented Lagrangian Decomposition Method. Firstly, we propose a method to generate an initial routing plan by using a dynamic programming algorithm for solving each subproblem (namely Algorithm 1). Later, we present an algorithm to update the space-time network whenever pick-up requests appear (namely Algorithm 2). The details of the algorithms are demonstrated as follows:

4. Discussion

In the proposed model, constraint (4) makes the problem difficult to solve and therefore leads to computational challenges. To address this problem, we deploy the augmented Lagrangian relaxation method for problem decomposition. Compared to the Lagrangian relaxation method, the augmented Lagrangian relaxation introduces an extra quadratic penalty term into the objective function and can break solution symmetry. Meanwhile, compared to other exact solution methods (e.g., big-M method, branch and bound, and column generation), the solution method based on augmented Lagrangian relaxation has the advantage of solving large-scale problems and maintaining

Step 1: Initialization
Initialize time $\tau = 0$
Initialize iteration number $n = 0$
Initialize Lagrangian multipliers $\pi_r^0 = 0$ for iteration 0
Initialize penalty factor $\rho^0 = 1$
Set the best lower bound estimate $\mathrm{LB}^* = -\infty$
Set the best upper bound estimate $\mathrm{UB}^* = +\infty$
Set the origin node $o_{k,\tau}$ for each vehicle k, $o_{k,\tau}$ =*depot*
Set the destination node d_k for each vehicle k, d_k =*depot*
Step 2: Solve problem 2-b
Use dynamic programming algorithm to solve problem 2-b for each vehicle k sequentially. During this process, update the cost for each space-time arc before optimizing each vehicle's route.
Step 3: Generate upper bound estimate
Step 3.1: Obtain a feasible solution X_{UB}^n for problem 2-b
Adopt the optimal routing solution in Step 2
For each request node
If the request node is not served by any vehicle
then designate a virtual vehicle to this request node and add penalty.
If the request node is served more than once
then designate one specific vehicle to this request node.
End for
After the above adjustment, the optimal solution in Step 2 now becomes a feasible solution X_{UB}^n and then local upper bound estimate UB^n can be computed at iteration n.
Step 3.2: Generate best upper bound estimate UB^*
$\mathrm{UB}^* = \min \mathrm{UB}^*, \mathrm{UB}^n$
Step 4: Generate lower bound estimate
Step 4.1: Solve problem 1-b
Use dynamic programming algorithm to solve problem 1-b for each vehicle k in sequence and obtain the optimal solution X_{LB}^n.
Step 4.2: Generate best lower bound estimate LB^*
Calculate local lower bound estimate at iteration n as LB^n
Generate best lower bound estimate $\mathrm{LB}^* = \max \mathrm{LB}^*, \mathrm{LB}^n$
Compute the relative gap between UB^* and LB^*, $\mathrm{gap} = \mathrm{UB}^* - \mathrm{LB}^*/\mathrm{UB}^* \times 100\%$
Step 5: Update Lagrangian multipliers and penalty factors
Step 5.1: Update Lagrangian multipliers

$$\pi_r^{n+1} = \pi_r^n + \rho^n \left(\sum_{k \in K} \sum_{(i,j,t,t') \in \Psi_{r,k}} x_{i,j,t,t'}^k (\tau) - 1 \right) \quad \forall r \in R_{p,\tau}$$

Step 5.2: Update penalty factor

$$\rho^{n+1} = \begin{cases} \rho^n + 2, & \text{if some request nodes are served more than once,} \\ 1, & \text{if all request nodes are served only once,} \\ \rho^n, & \text{otherwise.} \end{cases}$$

Step 6: Terminal Condition
If the iteration number $n >$ maximum iteration number $n_{\max}$ **or** the value of the relative gap gap $<$ *a* boundary toleration value ε, **then** terminate the algorithm procedure
else
$n = n + 1$ and go back to Step 2.

Algorithm 1: Generate initial routing plan

Assume that a pick-up request occurs at time τ, requested by a physical node $p \in P_\tau$. The steps for handling pick-up requests and regenerating the routing plan are demonstrated below.
Step 1: Update vehicle and request information
Step 1.1: Update origin node for each vehicle k
For each vehicle $k \in K$
Update origin node for each vehicle k at time τ as $o_{k,\tau}$
End for
Step 1.2: Modify space-time network using the idea from Section 2.3.2
Step 2: Implement Algorithm 1 on modified space-time network

Algorithm 2: Network updating for handling pick-up requests and rearranging vehicle routes

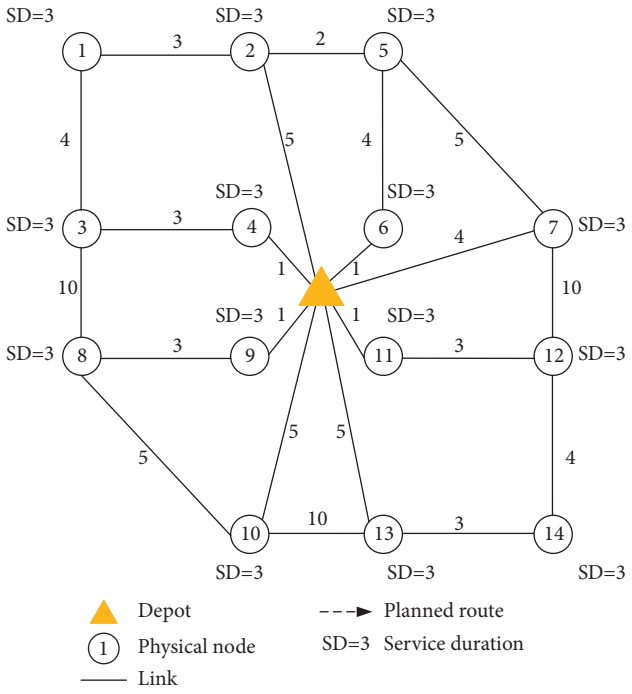


Figure 7: Physical network for the illustrative example.

an acceptable convergence speed. Moreover, although most of the solution algorithms for solving DVRP are heuristic, we still deploy augmented Lagrangian relaxation-based solution method because we have already converted the dynamic problem into the equivalent formulation for static VRP, via the problem reformulation in Section 3.2. Consequently, the augmented Lagrangian relaxation with dynamic programming algorithm has the advantage in improving solution equality as compared to most heuristics.

5. Numerical Experiments

In this section, we perform two numerical experiments to verify the effectiveness of the proposed model and algorithms. The algorithms are coded in *Python*, run in PyPy, and evaluated on a personal computer with 2.4 GHz CPU and 8 GB RAM.

First, an illustrative example is displayed based on a 14-node physical network with five vehicles. Then, we conduct a medium-scale numerical experiment on a real-world last-mile delivery network to demonstrate the validity of the augmented Lagrangian relaxation decomposition framework and the efficiency of the proposed algorithms.

5.1. An Illustrative Example. We chose to examine an illustrative network with 1 depot, 14 physical nodes, and 22 two-way transportation links as our numerical example, as shown in Figure 7. Five vehicles are available. The numbers on each link represent the corresponding travel time, and the figures beside the physical nodes denote the required service duration. Basic information for this illustrative example is shown in Table 6. The preferred time window [0,28] denotes the earliest departure time (EDT $= 0$) and latest arrival time (LAT $= 28$) to the depot for each vehicle. We set the fixed cost of using a vehicle as \$300 per day, travel cost as \$1 per time unit, utility for serving a delivery request as \$100 per time unit, and utility for serving a pick-up request as \$300 per time unit. The initial value of the Lagrangian relaxation multiplier π_p^0 and penalty factor ρ^0 are set as 0 and 1, respectively. The number of iterations is 300.

Table 6: Basic information of delivery requests in the illustrative example.

Corresponding physical node ID	Required service duration	Preferred time window
1	3	[0,28]
2	3	[0,28]
3	3	[0,28]
4	3	[0,28]
5	3	[0,28]
6	3	[0,28]
7	3	[0,28]
8	3	[0,28]
9	3	[0,28]
10	3	[0,28]
11	3	[0,28]
12	3	[0,28]
13	3	[0,28]
14	3	[0,28]

5.1.1. Generating Initial Routing Plan. In this illustrative example, the vehicles are originally located at depot. We first initialize the vehicle routes to ensure that each vehicle is assigned to several physical nodes. The optimal solution is obtained by implementing the algorithm and is intuitively displayed in Figure 8, where the optimal route for each vehicle is marked with dotted lines.

The visiting sequence and time sequence for each vehicle are presented in Table 7. The computational results demonstrate that 14 physical nodes can be fully served by exactly four vehicles within the time window, with one vehicle left unused. Each vehicle is assigned to several physical nodes and will follow the visiting sequence until a new request occurs.

The convergence process is shown in Figure 9. We can see that the optimal solution is obtained at the 43^{th} iteration and the value of the objective function is -2940. Total running time is 1.40s.

5.1.2. Optimal Solution for Handling Pick-Up Requests. We assume that when $\tau = 9$, a pick-up request with a time window [11, 12] and total service duration of one time unit appears at physical node 2, as shown in Table 8. The delivery and pick-up request node are generated at physical node 2, as shown in Figure 10. Before the optimization process, vehicle positions and the remaining service duration for each physical node should be updated. For example, when $\tau = 9$, vehicle 1 is

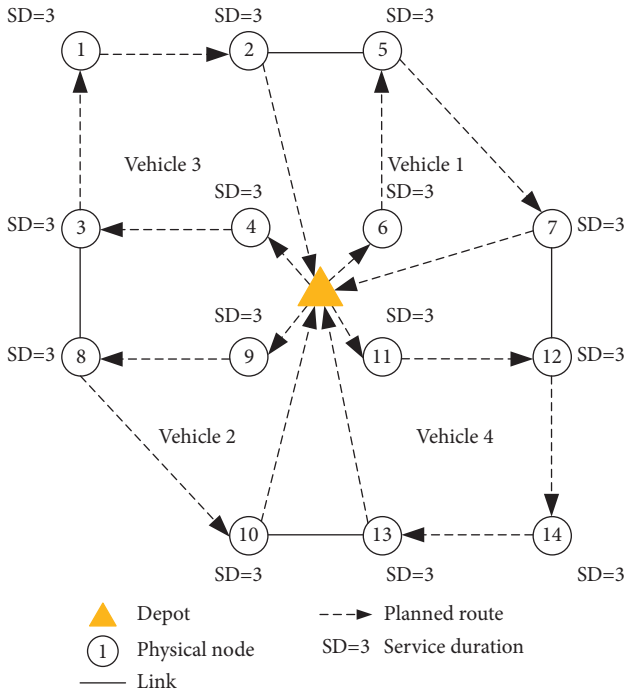


Figure 8: Optimal solution for each vehicle when initializing original routing plan ($\tau = 0$).

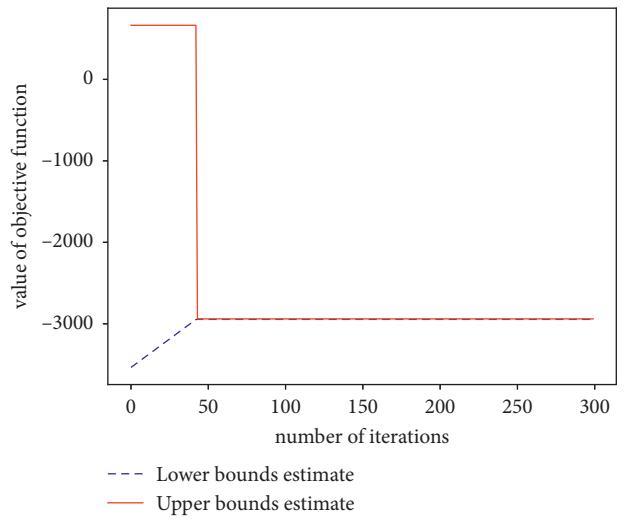


Figure 9: Evolution process of LB and UB estimate.

Table 8: Basic information on the pick-up request.

Corresponding physical node ID	Required service duration	Preferred time window
2	1	[11,12]

Table 7: Optimal solution demonstration for initial routing plan.

Vehicle ID	Visiting sequence	Time sequence
1	[Depot-6-5-7-depot]	[0,0]-[1,4]-[8,11]-[16,19]-[23,28]
2	[Depot-9-8-10-depot]	[0,0]-[1,4]-[7,10]-[15,18]-[23,28]
3	[Depot-4-3-1-2-depot]	[0,0]-[1,4]-[7,10]-[14,17]-[20,23]-[28,28]
4	[Depot-11-12-14-13-depot]	[0,0]-[1,4]-[7,10]-[14,17]-[20,23]-[28,28]

Note. The first and second elements in [_, _] denote the arrival and departure time at customer/depot, respectively.

serving physical node 5 and therefore will originate from physical node 5 in the following optimization process. Until now, physical node 6 has been fully served, physical node 7 is yet to be served at all, and physical node 5 has been served for one time unit. By conducting a similar analysis, information of other vehicles is updated, respectively, and each vehicle has a new origin.

The optimized results are depicted in Figure 11. The result displays that vehicle 1 interrupts its service at physical node 5 in order to pick up the request at physical node 2 (which should be served by vehicle 3 if following initial routing plan) within the time window and then returns to physical node 5 to accomplish the remaining orders. Although vehicle 1 spends a higher travel cost ($4) on the road, additional utility ($300) is obtained simultaneously during the detour process, which means that a net profit ($296) has been obtained.

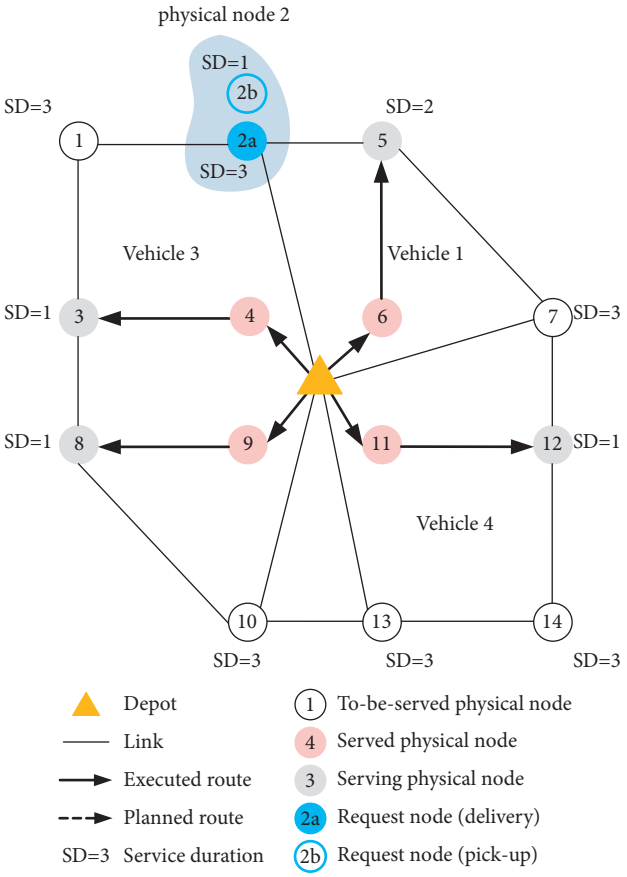


Figure 10: Updated physical network when $\tau = 9$.

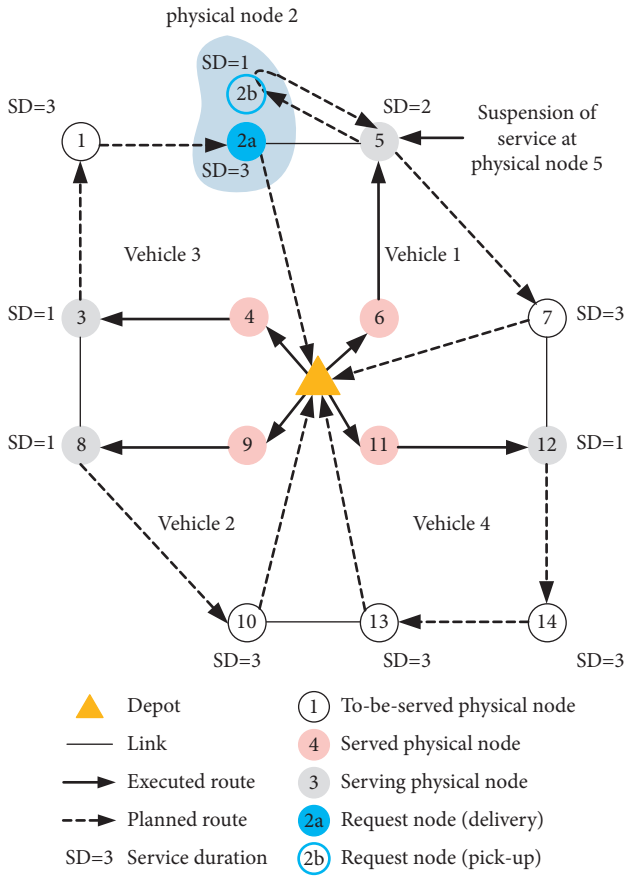


Figure 11: Optimal solution for each vehicle when handling pick-up request at $\tau = 9$.

The corresponding visiting and time sequence are displayed in Table 9. The value of the objective function is −953. We can see that although each vehicle has a different origin, they arrive at depot in time without missing any requests, which can be attributed to the cooperation between vehicles 1 and 3.

To further demonstrate the advantage of preemptive service, we run the experiment under non-preemptive service and compare the solution results under two service modes, as shown in Table 10. Under preemptive service, the value of objective function is 45.1% lower than non-preemptive service. As the value of objection function is calculated by subtracting total utility from total operating cost, we can see that preemptive service achieves a better trade-off between utility and cost in this case.

5.2. Medium-Scale Experiments. To test the proposed model and algorithms, we further perform a medium-scale experiment on a real-world last-mile delivery network. To generate more customers, we download a digital map covering the service area in the OpenStreetMap format and then generate POIs using a light-weight tool called OSM2GMNS. Readers can refer to the link https://github.com/asu-trans-ai-lab/OSM2GMNS for further details. We regarded those POIs as physical nodes, and the distance between them was measured using their latitude and longitude information. Given the travel speed (15 km/h), we can compute the travel time on each link.

Table 9: Optimal solution demonstration when $\tau = 9$.

Vehicle ID	Visiting sequence	Time sequence
1	[5-2-5-7-Depot]	[9,10]-[12,13]-[15,16]-[21,24]-[28,28]
2	[8-10-Depot]	[9,10]-[15,18]-[23,28]
3	[3-1-2-Depot]	[9,10]-[14,17]-[20,23]-[28,28]
4	[12-14-13-Depot]	[9,10]-[14,17]-[20,23]-[28,28]

Note. The first and second elements in [_, _] denote the arrival and departure time at physical node/depot, respectively.

Table 10: Comparison of the solution results under two service modes.

	Non-preemptive service	Preemptive service
The value of objective function (unit: $)	−657	−953 (−45.1%)
How to handle the pick-up request	Renege on the pick-up request at physical node 2	Serve the pick-up request at physical node 2

In order to test the effectiveness of the proposed algorithm for handling medium-scale cases, we present three cases where 45, 70, and 100 physical nodes are considered. Basic assumptions are listed as follows:

(1) Each delivery person starts their work at 8 : 00 and returns to the depot before 12 : 00

(2) Vehicle usage cost is $100/(veh · day) and transportation cost is $1/(veh · min)

(3) Delivery requests should be satisfied before 12 : 00. Besides, all pick-up requests have a specific and tight time window

(4) Service duration for each delivery request and pick-up request is 40 min and 2 min, respectively

(5) Utility for serving a pick-up request is $10/(veh · min) and utility for serving a delivery request is $2.5/(veh · min)

5.2.1. Generating Initial Routing Plan. We set the initial value of parameter π as 0, parameter ρ as 1 and the iterative number as 100 iterations. The optimized results for generating the initial routing plan in each case are displayed in Table 11, and the convergence curves are shown in Figure 12.

As shown in Table 11, computational challenges for obtaining an initial routing plan are mainly attributed to the number of physical nodes. Meanwhile, solution equality, which is measured by relative gap, does not change dramatically with the expansion of case scale, meaning that the

Table 11: Optimized results for generating initial routing plan in different case scales.

Case number	Number of physical nodes	Number of used vehicles	Running time (s)	Best lower bound estimate	Best upper bound estimate	Relative gap (%)
1	45	9	106	−3517	−3478	1.12
2	70	14	511	−5159	−5021	2.75
3	100	20	1603	−7805	−7479	4.36

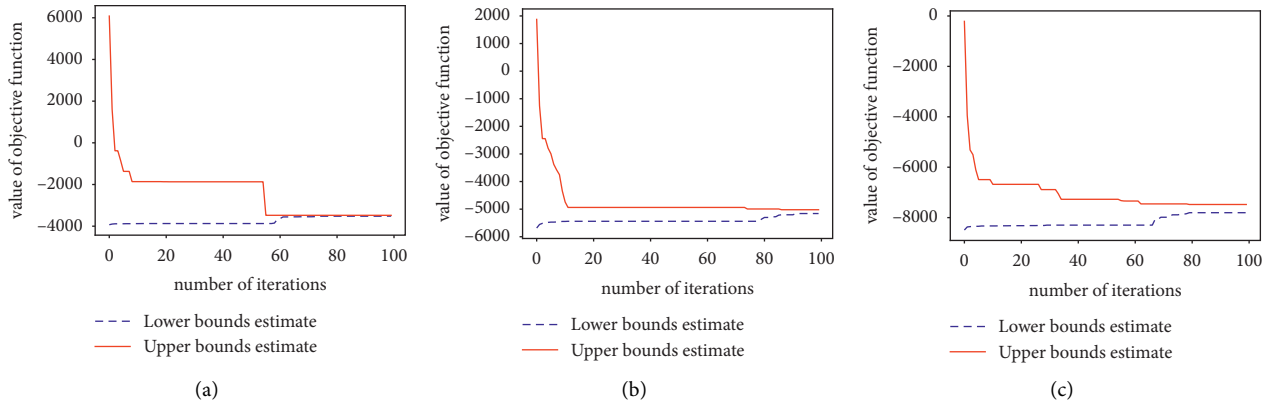


Figure 12: Iterative curve for cases with different case scales. (a) Iterative process for Case 1(45 physical nodes). (b) Iterative process for Case 2(70 physical nodes). (c) Iterative process for Case 3 (100 physical nodes).

proposed algorithm performs well in terms of addressing medium-scale cases within reasonable iteration times or solution time.

To demonstrate the illustrative results more directly, we take Case 1 as an example to demonstrate graphical results (as shown in Figure 13) and detailed solution information (as shown in Table 12).

5.2.2. Testing Preemptive Service under Case 1. In this section, we introduce pick-up requests into Case 1 in order to demonstrate the flexibility of preemptive service. The time unit is assumed to be 5 min. For example, if a physical node should be served for 40 min, then it can be extended to 8 request nodes. Assume that when $\tau = 80$, there are five pick-up requests that occur at physical nodes 3, 5, 34, 35, and 45. Basic information of the pick-up requests is shown in Table 13.

The optimal solution is derived in 110s using 9 vehicles, and the space-time route is presented in Figure 14. We obtain the best upper bound estimate as −1861, the best lower bound estimate as −2054, and the relative gap as 10.37%. Five pick-up requests are all satisfied within their time windows.

We compare the solution results for handling pick-up requests under two service modes, as shown in Table 14. The results indicate that the value of objective function is 18.1% lower than non-preemptive service and more pick-up requests are satisfied under preemptive service.

From the illustrative results in Figure 15, we can see that the pick-up requests that occur at physical nodes 34 and 35 cause interruptions to other physical nodes. The

Figure 13: Initial routing plan for Case 1 (45 physical nodes).

first interruption happens at physical node 30 because vehicle 1 has to satisfy the pick-up request at physical node 34 before completing all of the delivery requests at physical node 30. The second interruption happens at physical node 27 because vehicle 4 has to satisfy the pick-up requests at physical node 35 before completing all of the delivery requests at physical node 27.

5.3. Sensitivity Analysis. In real-world scenarios, the route plan can be significantly affected by the values of service utilities and the time window of pick-up service. Based on the illustrative example in Section 5.1, we conduct the

Table 12: Initial routing plan demonstration for Case 1 (45 physical nodes).

Vehicle ID	Visiting sequence	Time sequence
1	[Depot-34-30-37-24-8-depot]	[0,0]- [8,48]-[51,91]-[96,136]-[138,178]-[183,223]-[229,240]
2	[Depot-41-13-3-1-19-depot]	[0,0]-[2,42]-[44,84]-[85,125]-[126,166]- [168,208]-[210,240]
3	[Depot-44-12-45-38-15-depot]	[0,0]-[2,42]-[44,84]-[85,125]-[126,166]- [168,208]-[211,240]
4	[Depot-35-27-36-31-25-depot]	[0,0]-[7,47]-[48,88]-[89,129]-[131,171]- [173,213]-[220,240]
5	[Depot-33-21-23-39-22-depot]	[0,0]-[1,41]-[42,82]-[83,123]-[124,164]- [165,205]-[207,240]
6	[Depot-40-2-6-26-14-depot]	[0,0]-[5,45]-[47,87]-[89,129]-[130,170]- [171,211]-[214,240]
7	[Depot-42-10-28-11-9-depot]	[0,0]-[1,41]-[42,82]-[83,123]-[125,165]- [166,206]-[208,240]
8	[Depot-18-4-43-5-17-depot]	[0,0]-[1,41]-[42,82]-[83,123]-[126,166]- [169,209]-[212,240]
9	[Depot-32-29-20-7-16-depot]	[0,0]-[2,42]-[43,83]-[85,125]-[128,168]- [169,209]-[211,240]

Note. The first and second elements in [_, _] denote the arrival and departure time at physical node/depot, respectively.

Table 13: Basic information of pick-up requests when $\tau = 80$.

Corresponding physical node ID	Required service duration	Preferred time window
3	2	[101,102]
5	2	[103,104]
34	2	[85,86]
35	2	[85,86]
45	2	[105,106]

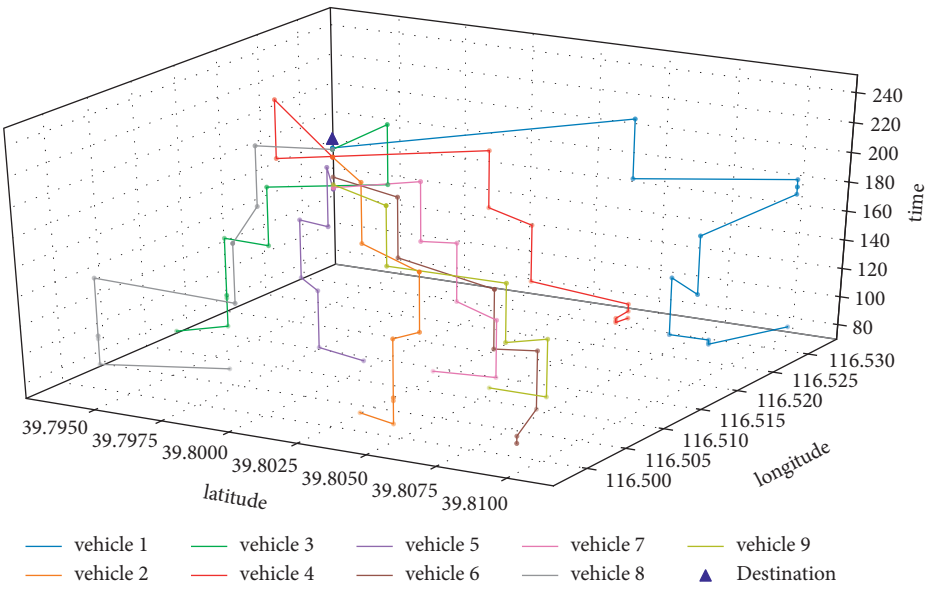


Figure 14: Optimal space-time route when $\tau = 80$.

Table 14: Comparison of the solution results under two service modes.

	Non-preemptive service	Preemptive service
The value of objective function (unit: $)	−1576	−1861 (−18.1%)
How to handle the pick-up request	Renege on the pick-up request at physical nodes 34,35; Serve the pick-up request at physical nodes 3,5,45	Serve the pick-up request at physical nodes 3,5,34,35,45

following sensitivity analysis to examine the impact of the ratio of pick-up and delivery utility and the time window of pick-up service on the solution results. We use the reduction percentage of objective function between preemptive service and non-preemptive service to decide which service mode to take. If the reduction percentage is a negative value, preemptive service performs better than non-preemptive service; otherwise, non-preemptive service is more profitable.

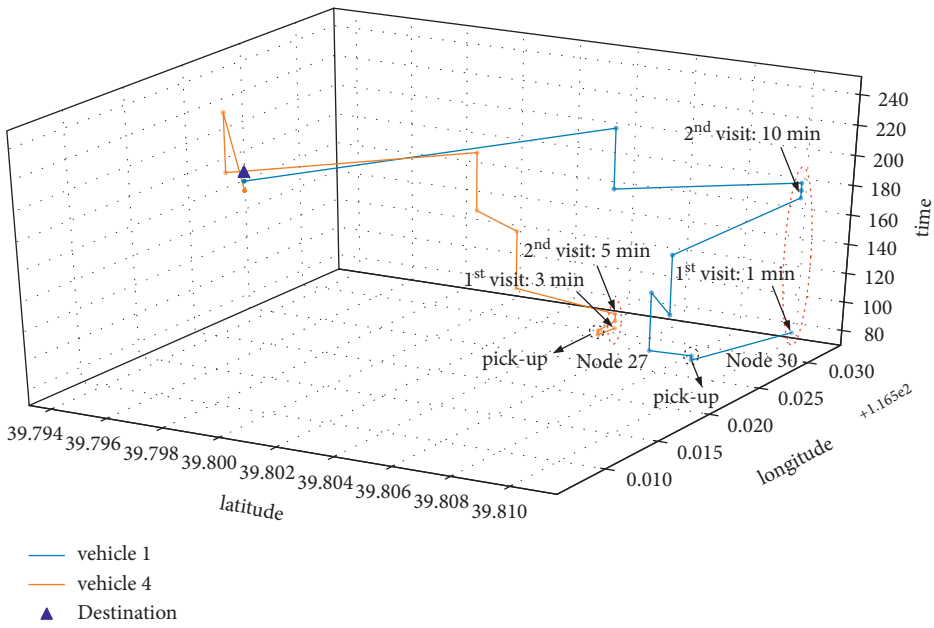


Figure 15: Service interruption in optimal space-time route.

Table 15: The solution results under different ratios of pick-up & delivery utility.

The ratio of pick-up and delivery utility	The value of objective function under preemptive service	The value of objective function under non-preemptive service	The reduction percentage
1	−4753	−4457	−6.64%
1.2	−3803	−3507	−8.44%
1.5	−2853	−2557	−11.58%
2	−1903	−1607	−18.42%
2.25	−878	−657	−33.64%
3	−953	−657	−45.05%
3.75	−1028	−657	−56.47%
4.5	−1103	−657	−67.88%

Table 16: The solution results under different lengths of pick-up time window.

The length of pick-up time window	The value of objective function under preemptive service	The value of objective function under non-preemptive service	The reduction percentage
1	−953	−657	-45.10%
5	−953	−657	-45.10%
9	−843	−957	11.90%
13	−843	−957	11.90%

5.3.1. Sensitivity Analysis on the Ratio of Pick-Up and Delivery Utility. To conduct sensitivity analysis on the ratio of pick-up and delivery utility, the time window is set as fixed values and the ratio of pick-up and delivery utility is variable. The reduction percentage of objective function between preemptive service and non-preemptive service under different ratios of pick-up and delivery utility are shown in Table 15. The computational result demonstrates that the reduction percentage decreases with the increasing ratio of pick-up and delivery utility, indicating that the advantage of preemptive service over non-preemptive service expands.

5.3.2. Sensitivity Analysis on the Time Window of Pick-Up Service. To conduct sensitivity analysis on the time window of pick-up service, the delivery and pick-up utility is set as fixed values and the length of time window is variable. We keep the starting time of the time window fixed and set up different values of ending times. The reduction percentage of objective function between preemptive service and non-preemptive service is shown in Table 16.

As the length of pick-up time window increases, the reduction percentage of objective function between preemptive service and non-preemptive service remains

Table 17: Definitions of some key modeling elements.

Concept	Definition
Decision epoch	The moment when pick-up requests are captured, acceptance/rejection decisions are made, and new routing plan is obtained.
Preemptive service	Delivery people can interrupt delivery process whenever pick-up requests occur. Therefore, they can visit physical nodes for more than once to complete the remaining orders.
Pick-up request	A type of high-utility request that appears during service.
Delivery request	A type of low-utility request that appears before service.
Served physical nodes	A type of physical nodes that all requests are satisfied.
Serving physical nodes	A type of physical nodes that delivery people are working on.
To-be-served physical nodes	A type of physical nodes that no requests or partial requests are satisfied.

unchanged until the length reaches 9 time units. Afterwards, the time window is so wide that there is no need to interrupt the service process. Therefore, the reduction percentage will reach a positive value, meaning that non-preemptive service achieves better results than preemptive service.

6. Conclusions

In this article, we study a dynamic vehicle routing problem with preemptive pick-up service (VRPPS) in a multitasking environment and thoroughly examine the value of this new service mode. Delivery and pick-up service can be regarded as two different types of tasks. Under a preemptive service principle, service suspension of the delivery process is allowed to satisfy the pick-up requests. Focusing on maximizing the difference between the service utility of delivering/picking up packages and the total operating cost, we have constructed a multicommodity flow model with constraints on time windows and service duration, embedded in a space-time network. Overall, the proposed research demonstrates a systematic view on offering rapid responses to pick-up requests, as follows. (1) From the perspective of modeling the dynamic decision process and service interruption, we extend the physical nodes to requests nodes for different types of requests. This ensures that delivery service can be suspended at any time during the service duration. Additionally, we describe a method for updating the space-time network for each decision epoch to capture variations in requests over time. In each decision epoch, our model aims at achieving a trade-off between total service utility and total operating cost. (2) From the perspective of providing an effective optimization approach, the proposed model is reformulated by applying the augmented Lagrangian relaxation technique to decompose the primal problem into several simplified subproblems for each vehicle. A dynamic programming algorithm is then called to decide the optimal vehicle routes in each decision epoch.

To evaluate the effectiveness of our model and algorithm, we firstly conducted an illustrative example to compare the values of preemptive and non-preemptive service on solution equality. It indicates better solutions (with improved values) are obtained under preemptive service. The solution framework was also systematically tested through a real-world experiment based on a delivery network, taking into consideration different number of physical nodes. The computational results show that our approach can handle medium-scale cases effectively and obtain an acceptable gap between the best lower bound estimate and the best upper bound estimate.

In future research, we will take more practical factors into consideration. Firstly, further work may aim at handling stochastic pick-up requests, service time, and travel time. Secondly, we will discuss how sensitive optimized results are to the value of service utility and operating cost. Thirdly, we may consider the possibility for delivery people to hand over delivery packages during service. Eventually, it will also be interesting to study the joint optimization of goods stacking and dynamic vehicle routing in future studies, because, in practice, it is quite time-consuming for delivery people to rummage through their pre-organized goods.

Appendix

Definition of some key modeling elements.

The authors list definitions of some basic concepts that may be used in the article, as shown in Table 17.

Data Availability

The data will be available upon request to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors would like to thank Dr. Xuesong Zhou at Arizona State University and Dr. Yu Yao at Hohai University for their valuable comments.

References

[1] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle-routing problem with time windows," *Operations Research*, vol. 40, no. 2, pp. 342–354, 1992.

[2] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," *Transportation Science*, vol. 31, no. 2, pp. 170–186, 1997.

[3] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.

[4] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, no. 1, pp. 17–30, 2006.

[5] M. L. Fisher, K. O. Jornsten, and O. B. G. Madsen, "Vehicle routing with time windows: two optimization algorithms," *Operations Research*, vol. 45, no. 3, pp. 488–492, 1997.

[6] N. Kohl and O. B. G. Madsen, "An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation," *Operations Research*, vol. 45, no. 3, pp. 395–406, 1997.

[7] Y. Yao, X. N. Zhu, H. Y. Dong et al., "ADMM-based problem decomposition scheme for vehicle routing problem with time windows," *Transportation Research Part B: Methodological*, vol. 129, pp. 156–174, 2019.

[8] M. Mahmoudi and X. S. Zhou, "Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state-space-time network representations," *Transportation Research Part B: Methodological*, vol. 89, pp. 19–42, 2016.

[9] L. Tong, L. S. Zhou, J. T. Liu, and X. S. Zhou, "Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing," *Transportation Research Part C-Emerging Technologies*, vol. 85, pp. 451–475, 2017.

[10] M. Zhao, J. T. Yin, S. An, J. Wang, and D. J. Feng, "Ride-sharing problem with flexible pickup and delivery locations for app-based transportation service: mathematical modeling and decomposition methods," *Journal of Advanced Transportation*, vol. 2018, Article ID 6430950, 21 pages, 2018.

[11] Z. L. Chen and H. Xu, "Dynamic column generation for dynamic vehicle routing with time windows," *Transportation Science*, vol. 40, no. 1, pp. 74–88, 2006.

[12] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, no. 4, pp. 327–343, 2005.

[13] J. Yang, P. Jaillet, and H. Mahmassani, "Real-time multi-vehicle truckload pickup and delivery problems," *Transportation Science*, vol. 38, no. 2, pp. 135–148, 2004.

[14] W. B. Powell, Y. Sheffi, K. S. Nickerson, K. Butterbaugh, and S. Atherton, "Maximizing profits for north-american-van-lines truckload division - a new framework for pricing and operations," *Interfaces*, vol. 18, no. 1, pp. 21–41, 1988.

[15] S. Ichoua, M. Gendreau, and J. Y. Potvin, "Exploiting knowledge about future demands for real-time vehicle dispatching," *Transportation Science*, vol. 40, no. 2, pp. 211–225, 2006.

[16] V. Pillac, C. Gueret, and A. L. Medaglia, "An event-driven optimization framework for dynamic vehicle routing," *Decision Support Systems*, vol. 54, no. 1, pp. 414–423, 2012.

[17] M. W. Ulmer, B. W. Thomas, and D. C. Mattfeld, "Preemptive depot returns for dynamic same-day delivery," *EURO Journal on Transportation and Logistics*, vol. 8, no. 4, pp. 327–361, 2019.

[18] H. N. Psaraftis, M. Wen, and C. A. Kontovas, "Dynamic vehicle routing problems: three decades and counting," *Networks*, vol. 67, no. 1, pp. 3–31, 2016.

[19] R. Gevaers, E. Van de Voorde, and T. Vanelslander, "Cost modelling and simulation of last-mile characteristics in an innovative B2C supply chain environment with implications on urban areas and cities," in *Proceedings of the 8th International Conference on City Logistics*, vol. 125, pp. 398–411, Bali, Indonesia, June 2013.

[20] H. M. Niu, X. S. Zhou, and X. P. Tian, "Coordinating assignment and routing decisions in transit vehicle schedules: a variable-splitting Lagrangian decomposition approach for solution symmetry breaking," *Transportation Research Part B: Methodological*, vol. 107, pp. 70–101, 2018.