IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 22, 2020, accepted October 13, 2020, date of publication October 19, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032018

A Disruption Recovery Model for Time-Dependent Vehicle Routing Problem With Time Windows in Delivering Perishable Goods

YAO WU[1,2], BIN ZHENG[3], AND XUELIANG ZHOU[2]
[1]School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Industrial Engineering, School of Mechanical Engineering, Hubei University of Automotive Technology, Shiyan 442002, China
[3]School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China

Corresponding author: Yao Wu (yaowumail@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71672154 and Grant 52075107, in part by the Humanities and Social Sciences Foundation of Ministry of Education of China under Grant 16YJA630038 and Grant 17YJC630229, in part by the China Postdoctoral Science Foundation under Grant 2018M640912, in part by the Natural Science Foundation of Hubei Province under Grant 2016CFB402, and in part by the Humanities and Social Sciences Research Project of Hubei Provincial Department of Education under Grant 19Q130.

ABSTRACT In delivering perishable goods, certain unexpected disruptive events may cause the initial routing scheme to be infeasible. A new routing scheme must be generated quickly to alleviate delivery disturbances. According to the idea of disruption management, a disruption recovery model with a distinctive type of split delivery is developed for inter-route recourse based on an initial time-dependent vehicle routing model with time windows, which synthesizes the perishable nature of delivered goods and dynamic travel route choice in urban road networks. Then, a tabu search algorithm is proposed to solve the initial routing problem and further extended to generate the disruption recovery plan. Three computational experiments on the instances adapted from Solomon's and Gehring and Homberger's benchmark problems are conducted to illustrate the effectiveness of the proposed model and algorithm. Supplementary data associated with this article can be found at https://www.amazon.com/clouddrive/share/foL0Vfo5C0GUjTrdO2UwljhZ9j6nNnHcD2Q0bKCTXJS.

INDEX TERMS Vehicle routing, disruption management, time-dependent, time windows, split delivery.

I. INTRODUCTION

In practice, many life necessities are highly perishable goods, such as fast food, delicatessen products, fresh aquatic products, fresh meat, and fresh milk, as well as fresh-cut fruits, vegetables and flowers. These goods' quality deteriorates continually during the distribution process due to their highly perishable nature. For example, the shelf life and keeping quality of meat are influenced by many factors, such as holding temperature, atmospheric oxygen, endogenous enzymes, moisture, light and, most importantly, microorganisms. All these factors can result in detrimental changes in the color, odor, texture and flavor of meat [1]. Therefore, the value of delivered goods is affected by their freshness. It is practical to characterize the perishability to improve the efficiency of delivery activities.

As a core issue of delivery process, the optimal scheduling of vehicle routing problem (VRP) is a challenge task, which not only takes into account of customers' demand but also reduces the total relevant costs. To achieve effectiveness, several customers would be coordinated in a same route originating and terminating at a central depot. Especially considering the perishability of goods, orders are expected to be delivered at desired time intervals. To avoid big losses, cancellation of placed orders is usually not allowed. When customers' time windows are imposed, VRP is extended to vehicle routing problem with time windows (VRPTW), which has been studied extensively in the literature [2]. However, the real problem is that, particularly in urban areas, traffic flow is not static and always fluctuates over time. For example,

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

during the morning and evening rush hours, huge traffic load slows the vehicle speed. Such regular phenomenon can be predicted by analyzing daily traffic data. By means of valid stochastic information, a vehicle routing scheme considering varying travelling times is devised to avert unnecessary traffic jams, which refers to time-dependent vehicle routing problem (TDVRP) [3]. Still and all, certain unexpected disruptive events may cause troubles for vehicle travelling en route, such as terrible traffic block, vehicle breakdown, which would result in enormous transportation delay leading the initial routing plan no longer optimal or even infeasible. When such disruptions occur, a disruption recovery scheme must be generated rapidly based on the instant situation [4]. Fortunately, resorting to advanced equipment and communication technologies, including GPS/GIS/GPRS, site data related to traffic and vehicles can be acquired in real-time [5], and efficient vehicle rerouting decisions can be provided for drivers in time.

Consequently, this work is intended to study a disruption recovery approach for transportation delay in delivering perishable goods. As detailed in the next section, although there have been several studies on VRP in delivering perishable products in time-dependent road networks, minimal literature is available on disruption recovery in such a process. Since the quality of perishable products deteriorates with time during the delivery process, it is critical to select the shortest path from multiple accesses to shorten the travelling time. In addition, the vehicle is advisable to be loaded with only the required amount; however, it will result in no more products being provided to the unserved customers in the disruption recovery process when no extra vehicles are dispatched. The disruption recovery is sequentially confined to inter-route recourse and regeneration of each route. Split delivery is inevitable because of inequality in customers' demand. In general, to improve the overall satisfaction of customers, the urgent requirements in disrupted routes are expected to be served with higher priority by an available vehicle nearby, while the customers with wider time windows are served later. Although split delivery has been widely studied in the extant literature [6], [7], it is only considered in the initial routing plan, and route generation is constricted by the vehicle capacity, which are different from the split delivery implemented for disruption recovery in this article. Therefore, in contrast to the extant literature, we present a time-dependent vehicle routing problem with time windows (TDVRPTW) on a multigraph in the initial routing plan of perishable product delivery, which characterizes the perishability of products and elaborates the real traffic network by considering multiple paths between each couple of nodes. Then, in the initial routing scheme executing stage, the idea of disruption management is adopted, and a disruption recovery model with split delivery is developed to diminish the negative effects of transportation delay.

The contributions of this work are threefold. First, we develop a comprehensive time-dependent vehicle routing model with time windows for the initial routing plan in delivering perishable goods, which simultaneously considers the perishability of goods, time-dependent traffic flow and alternative path selection in traffic networks. Second, we develop a disruption recovery model with a distinctive type of split delivery, which considers the situation that the total remaining supply is only equal to the total remaining demand. Third, according to the problem characteristics, an effective tabu search (TS) algorithm is proposed to solve the initial routing problem and further extended to address the disruption recovery problem with new elaborate neighborhood structures for the specific split delivery.

The remainder of this article is organized as follows. A brief literature review related to our research is performed in the next section. In Section III, a mathematical model for the initial routing scheme is proposed in addition to the problem description, followed by a disruption recovery model for delivery delay. In Section IV, TS algorithms are presented with detailed differences in their applications to the two models. Section V presents the computational results and the effect analysis of disruption recovery scheme. Finally, conclusions are provided, as are hints for future research.

II. LITERATURE REVIEW

Since Dantzig and Ramser [8] first introduced the basic VRP model, vehicle scheduling in goods distribution system has garnered much attention. The literature is rich in studies on variant models and algorithms of VRP, such as the location routing problem [9] and the production routing problem [10]. In contrast, this study focuses on the disruption recovery of unpredictable events in perishable goods. Thus, only the papers related to our study are reviewed instead of a detailed overview. Interested readers can please refer to [2] and [11] for a comprehensive knowledge on general VRPs.

In the early literature, although there are explicit concerns on perishable product delivery, the perishable nature has not been incorporated into VRP models. For example, Adenso-Diaz *et al.* [12] considered the distribution of dairy products in an integrated distribution network, which strives to minimize total distribution costs when clients are allocated fairly among vendors. Tarantilis and Kiranoudis [13], [14] investigated the real-world distribution of fresh milk and fresh meat based on VRP and solved them using meta-heuristic algorithms. Belenguer *et al.* [15] modeled the distribution of meat as a multiobjective VRPTW to simultaneously minimize the lateness in servicing customers and the total distance travelled.

Considering the significance of perishability in vehicle routing decisions, elaborate models incorporating the perishable nature were developed in most later works. Under the consideration of time-dependent traffic conditions and the perishability of products, Osvald and Stirn [16] modeled the delivery problem of fresh vegetables as a VRPTW. A TS-based algorithm was devised to minimize the weighted sum of travel distance, travelling time, delay penalties and perishability costs. Hsu *et al.* [17] formulated the distribution of perishable foods as a stochastic VRPTW with the objective to minimize total relevant costs and then extended the model to allow for time-dependent temperature and time-dependent vehicle travelling times. The calculation of perishability costs is different from that of Osvald and Stirn [16], which employed probability density functions to determine the quantity of spoiled products in the travelling process instead of a linear decay function. Considering the same factor of time-dependent environment temperature, Hu *et al.* [18] studied a refrigerator car scheduling problem from the energy consumption perspective. They developed a time-dependent mixed integer programming model to reduce the total operation cost and then solved the problem by an adaptive heuristic method combining a variable neighborhood search with particle swarm optimization. To demonstrate the trade-off between delivery cost and customer service related to the freshness aspect, Amorim and Almada-Lobo [19] presented a biobjective model for the delivery of highly perishable food products, where three types of geographical scenarios of requests' locations were examined to explain the cost-freshness relationship. Combing prevalent concerns such as the traffic congestion, limited working hours, and carbon emissions caused by the fuel consumption, Zulvia *et al.* [20] also proposed a many-objective green VRP for perishable products delivery which optimizes the operational cost, deterioration cost, carbon emissions and customer satisfaction. The problem was successfully tackled by an improved gradient evolution algorithm with discretization, non-dominated sorting, and crowding distance approaches. In contrast to the models addressing the delivery of all received orders, Song and Ko [21] developed a nonlinear mathematical model to deliver a part of ordered food products using limited number of refrigerated vehicles and general-type vehicles for on-line shopping stores, which is to maximize the total customers' satisfaction with the freshness of delivered food products. Ma *et al.* [22] further considered order acceptance of high perishable goods delivery from the revenue maximizing perspective in a time-dependent network, and integrated order selection and TDVRP as a mixed integer programming model.

Another stream of literature concerns VRP with disruptive events. The idea of disruption management provides a practical approach to address real-time and unpredictable events with objectives to minimize deviations of actual operations from the intended plans at minimum costs [23]. When unpredictable events hinder or disrupt the initial routing plan of the vehicle travelling on the path, a revised schedule should be created to reduce the negative effects on all involved parties; this matters to the tradeoff among multiple conflicting objectives. The idea of disruption management simply accords with these situations.

Until recently, the idea of disruption management has been applied broadly in flight scheduling, machine scheduling, supply chain management, and so on [24]. The concept has also been incorporated into the field of delivery systems. Li *et al.* [25] introduced a real-time vehicle rerouting problem with time windows to address vehicle breakdown, whose objective was to minimize a weighted sum of operation, service cancellation and route disruption costs. The proposed model was solved by a lagrangian relaxation heuristic with an insertion procedure embedded in it. Mu *et al.* [26] investigated a similar problem addressing vehicle breakdown but without considering customers' time windows. To minimize the number of vehicles used and the total travel distance, two TS algorithms were proposed. Nikolić and Teodorović [27] studied a scenario in which unexpected high demand in certain nodes makes one or more planned routes infeasible, where goods are distributed to the customers in the same order every day. A mathematical model was formulated to minimize the negative consequences of these disturbances, followed by a bee colony optimization algorithm to solve the problem in lexicography. In contrast to discussions about disruptive events in the delivery process, Mu and Eglese [28] introduced a new situation that delayed supply causes insufficient commodities available for loading on all vehicles at the start of the delivery period in the just-in-time system. A model was developed to reduce the impact of supply delay on the distribution company, and two TS algorithms were proposed to generate disruption recovery plan. To address a variety and a combination of delivery disruptive events, Wang *et al.* [29] developed a combinational disruption recovery model for VRPTW. An approach was suggested to transform various delivery disruptions into new-adding customer disruption, and the effect of disruptions on real-world participators was measured. The problem was solved by a nested partition method with optimal starting times of rescue vehicles from the depot. In addition, by considering the uncertainty of human behaviors and adopting hierarchical cluster analysis to segment customers, Ding *et al.* [30] formed a disruption management model with multiple stages and multiple objectives for solving delivery delay. The method is verified by a case study on fast food delivery but without considering perishable nature of products. Similar to above vehicle rerouting models, Yuan and Jiang [31] introduced disruption management to the real-time home caregiver scheduling and routing problem. A mathematical model was constructed which minimizes the weighted sum of deviation measurements on customers, caregivers, and companies. Then a TS heuristic was developed to efficiently solve the problem with a cost recorded mechanism to strengthen its performance.

In Table 1, we compare our work with the relevant literature in terms of the type of disruptive event, the type of product, and the modeling characteristics. The fifth column shows the recourse approaches before and after executing routes, which are divided by lines.

In summary, although there is minimal literature regarding disruption recovery of TDVRP, dispatching new vehicles or loading vehicles fully before they leave the depot are main recourse approaches to disruptive events in the delivery process. However, these actions are not practical for perishable product delivery due to its perishability, which in practice restricts a delivery vehicle leaving the depot with a load equal to the total demand of its customers. As an effective

TABLE 1. Comparison of the related works dealing with disruptive events.

Author	Disruption-type	Product-type	Base model	Recourse	Objective
Li et al. [25]	vehicle breakdown	general	VRPTW	Full load Extra vehicle Service cancellation	Travel cost Service cancellation cost
Mu et al.[26]	vehicle breakdown	general	CVRP	Full load Extra vehicle	Vehicle cost Travel cost
Mu and Eglese [28]	supply delay	general	CVRP	--- Waiting Multiple trips Delayed service Overtime working	Travel cost Labor cost Delay cost
Nikolić and Teodorović [27]	high demand	general	VRPTW	Full load Service cancellation	Service cancellation cost Customer-route deviation Travel cost
Wang et al. [29]	combinational disruptions	general	VRPTW	Full load Extra vehicle	Service time deviation Delivery sequence deviation Travel cost
Ding et al. [30]	delivery delay	perishable	VRPTW	Full load Extra vehicle	More severed customers Vehicle cost Delay cost
Yuan and Jiang [31]	new service request service cancellation	---	VRPTW	--- Extra caregiver Delayed service	Service time/ consistency deviation Route duration/ segment deviation New caregiver/ travel cost Lateness penalty
This study	delivery delay	perishable	TDVRPTW	--- Split delivery Delayed service	Service delay and frequency Travel cost Delivery sequence deviation

CVRP: capacitated VRP; ---: non-full load without help for disruptive events

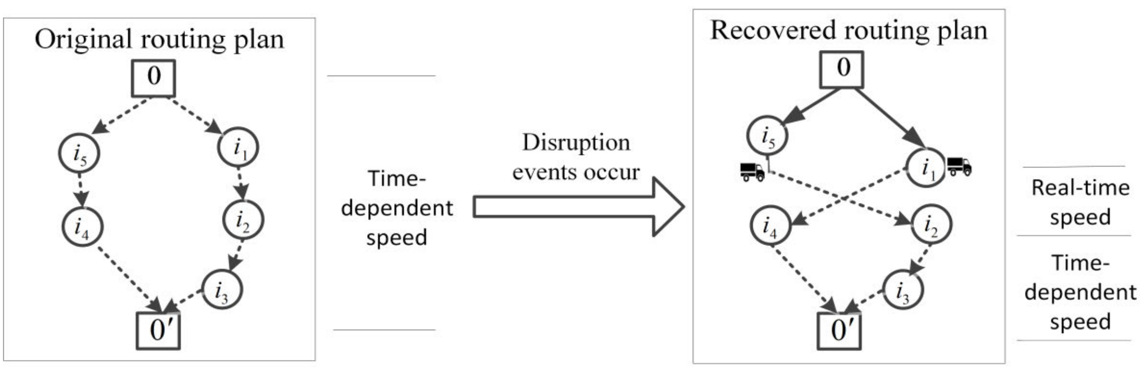


FIGURE 1. Computation framework proposed for responding to disruptive events.

inter-route recourse approach, split delivery is adopted in this study to decrease the dissatisfaction caused by delivery delay, which is distinct from the traditional split delivery studied in the literature [6] and [7]. Therefore, the disruption recovery model with a distinctive type of split delivery presented in this article is a new and practical delivery scheduling problem.

III. MODEL FORMULATION

Before dispatching vehicles for the delivery of assigned orders, an initial routing scheme should be generated to guide the vehicle travelling. At the execution stage, when transportation delay is discovered, a disruption recovery plan must be constructed quickly to respond to it. The following visits of vehicles in transit are associated with the current traffic condition. Therefore, a computation framework for responding to disruptive events is suggested in Fig. 1. To obtain an efficient disruption recovery scheme, the arc travelling times are computed by using combinational traffic information. Specifically, the travelling time from its current site to the next visit node is calculated based on real-time speed, and other travelling times continue to be calculated based on time-dependent speed.

In this section, we first introduce the properties of the time-dependent road network and the calculation of time-dependent travelling time. Then, the initial routing plan and disruption recovery plan are modeled successively.

A. TIME-DEPENDENT ROAD NETWORKS

Under the stable traffic condition, each path has a constant travelling time. In addition, the shortest path between two road network nodes is definite and associated with the least distance. However, when the dynamic nature of traffic flow is significant, particularly in busy urban areas, the travelling time of any path changes as time passes. Thus, the optimal path with the least travelling time depends on the specific time of the day. To elaborate the time-dependent characteristics, we model the road network with a multigraph. The similar application is provided in the work of Setak *et al.* [32]. In contrast to the simple graph, the multigraph has more than one edge between each couple of nodes. To clarify different directed edges, we represent each arc with a triple (i, j, h), where the first and second numbers correspond respectively to the origin and destination nodes, while the third number indicates the edge identifier. The set of arcs from i to j is denoted by H_{ij}.

Then, the calculation of arc travelling times is considered. When a vehicle traverses an arc, an earlier departure from the origin constantly ensures an earlier arrival at the destination, which refers to the first-in-first-out (FIFO) principle. In this paper, we apply the method provided by Ichoua *et al.* [33] to derive a continuous travelling time function from a discrete travel speed function, which overwhelms the noncompliance with FIFO policy when using the speed function directly.

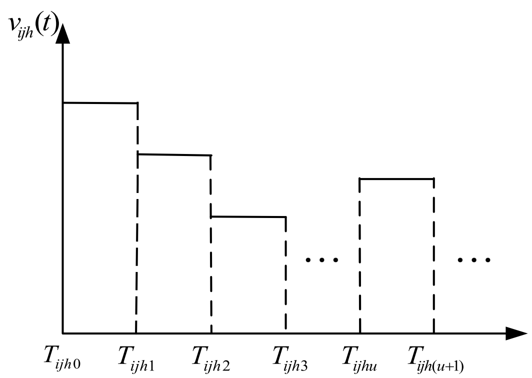


FIGURE 2. Discrete speed function of an arc in time-dependent network.

As shown in Fig. 2, the planning horizon is divided into a specified number of periods, and each has an approximately equal travel speed; thus, the travel speed on arc (i, j, h) can be described as a pricewise linear function. In other words, the speed v_{ijhu} is stable during the time period $[T_{ijhu}, T_{ijh(u+1)})$ when a vehicle travelling through arc (i, j, h), where time points T_{ijhu} and $T_{ijh(u+1)}$ represent the start and end of the time period u, respectively. When a vehicle departs later, or the arc is sufficiently long, the travelling may cover several time periods, and the correct travelling time can be calculated recursively by formula (1). Given a departure time t_i' from point i and the length d_{ijh} of arc (i, j, h), the arrival time at node j is equal to $t_j = t_i' + \tau_{ijh}(d_{ijh}, t_i')$,

$$\tau_{ijh}(\ell_j, t_{curr}) = \begin{cases} t_{res} + \tau_{ijh}(\ell_j - \ell_{res}, t_{curr} + t_{res}), & \ell_{res} < \ell_j \\ \ell_j / v_{ijhu}, & \ell_{res} \geq \ell_j \end{cases} \quad (1)$$

where t_{curr} and t_{res} are, respectively, the current time and the remaining time of current period, i.e., $t_{res} = T_{ijh(u+1)} - t_{curr}$ and $t_{curr} \in [T_{ijhu}, T_{ijh(u+1)})$; ℓ_j and ℓ_{res} denote respectively the length from the present site to vertex j and the distance that can be covered within t_{res}, i.e., $\ell_{res} = t_{res} v_{ijhu}$. The recursive function $\tau_{ijh}(\ell_j, t_{curr})$ is used to compute the travelling time of traversing the remaining length ℓ_j from the current time point t_{curr}. This recursive formula can be implemented by an iterative computation process as described in the work of Ichoua etc. [33].

B. INITIAL ROUTING PLAN

1) PROBLEM DESCRIPTION

A distribution network is described by a complete directed graph $G = (N, E)$, where N represents the nodes set, and E corresponds to the edge set. Each customer i has a demand D_i, which should be fulfilled during a specified time window $[e_i, l_i]$. A homogenous fleet located at the depot are responsible for delivering one kind of perishable product to customers. Although occasionally a few kinds of products are required to be delivered simultaneously, each kind is addressed separately in case of interaction influence leading to increasing deterioration. Similar to the work of Osvald and Stirn [16], the perishability of delivered products is characterized as a constant value-loss for the unit product in the unit delivery time. Although there may be material loss during the delivery process, it can be transformed into value loss, i.e., loss in revenue. It is assumed that the decayed product continues to be delivered to customers, and lost sales are not considered, which guarantees that the customer demand is entirely satisfied. To decrease the deterioration loss during the delivery process, the initial load of dispatched vehicles is simply equal to the total demand of its customers instead of a full load. All the dispatched vehicles leave the depot simultaneously and return to the depot after finishing their delivery tasks. For the convenience of receiving goods, each customer is assigned to one vehicle route, and its demand is delivered entirely. The objective is to find a set of routes such that the sum of the total travelling cost and the deterioration loss is minimized. The decision variables not only involve vehicle assignment and node visit sequence, but also path selection in the time-dependent road network.

2) INITIAL ROUTING MODELING

The notations used in the initial routing model are listed as follows:

Sets

C : a set of vertices representing n customer nodes, $|C| = n$.

O : a set of vertices representing copies of the depot as return centers of vehicles.

N : a set consisting of the depot 0 and elements in set C and O, $N = C \cup O \cup \{0\}$.

K : a set of trucks for delivering products.

H_{ij} : a set of arcs from node i to node j, $H_{ij} \in E$.

Indexes

k : index of vehicles.

i, j : index of nodes.

$n + k$: a copy of the depot representing the return center of vehicle k, $n + k \in O$, $k \in K$.

Parameters

Q : capacity of vehicle.

D_i : product demand of customer i.

s_i : service duration at customer i.

e_i, l_i: lower and upper limit of time window specified by customer i, respectively.

$\tau_{ijh}(d_{ijh}, t_i')$: travelling time needed from node i to node j along h^{th} path at the departure time t_i'.

θ_1, θ_2: variable cost derived from transportation operations and deterioration loss per unit product per unit travelling time, respectively.

M : a sufficiently large number.

Decision variables

t_i : service start time at node i by the assigned vehicle or return time to the depot when $i \in O$.

x_{ijh}^k : a binary variable that equals 1 when arc (i, j, h) is traversed by vehicle k; 0 otherwise.

The initial routing model is formulated as follows:

$$\min\ F = \theta_1 \sum_{i\in O} t_i + \theta_2 \sum_{i \in C} t_i D_i \tag{2}$$

Subject to

$$\sum_{j\in C}\sum_{h\in H_{0j}}\sum_{k\in K} x_{0jh}^k \le |K| \tag{3}$$

$$\sum_{j\in C}\sum_{h\in H_{0j}} x_{0jh}^k = \sum_{i\in C}\sum_{h\in H_{i0}} x_{i,n+k,h}^k, \forall k \in K \tag{4}$$

$$\sum_{i\in C\cup\{0\}}\sum_{h\in H_{ij}}\sum_{k\in K} x_{ijh}^k = 1, \quad \forall j \in C \tag{5}$$

$$\sum_{i\in C\cup\{0\}}\sum_{h\in H_{ij}} x_{ijh}^k = \sum_{i\in C\cup O}\sum_{h\in H_{ji}} x_{jih}^k, \quad \forall j \in C,\ k \in K \tag{6}$$

$$\sum_{i\in C}\sum_{j\in C\cup O}\sum_{h\in H_{ij}} D_i x_{ijh}^k \le Q, \quad \forall k \in K \tag{7}$$

$$t_j \ge t_i + s_i + \sum_{h\in H_{ij}} \tau_{ijh}(d_{ijh}, t_i + s_i) - M \sum_{h\in H_{ij}}\sum_{k\in K}(1 - x_{ijh}^k),$$
$$\forall i \in N\backslash O,\ j \in N \tag{8}$$

$$t_i \ge e_i, \quad \forall i \in C \tag{9}$$

$$t_i \le l_i, \quad \forall i \in C \tag{10}$$

$$t_0 = 0 \tag{11}$$

$$x_{ijh}^k \in \{0, 1\}, \quad t_i \ge 0,\ \forall i,\ j \in N,\ k \in K,\ h \in H_{ij} \tag{12}$$

The objective function (2) is to minimize the weighted sum of transportation cost and deterioration loss. In the first term, vehicle travel time is employed to calculate its corresponding transportation cost, while product losses are evaluated by order delivery time in the second term. Constraints (3) to (6) are vehicle flow conservation constraints. Constraint (3) restricts the available vehicle quantity, while constraint (4) forces the dispatched vehicles to return to the depot after serving customers. Constraint (5) states that each customer is served by exactly one vehicle. Constraint (6) ensures that the vehicle leaves a customer after the completion of its service. Constraint (7) requires the total delivery quantity of a vehicle to be no more than its capacity. Constraint (8) expresses that the service start time of a customer is in accordance with the vehicle visiting sequence, in which the travelling time on an arc is calculated by formula (1). To establish a feasible node visit sequence, service start time of each node is also required to be determined. Constraints (9) and (10) impose that the start time of serving a node cannot be earlier than its lower time window or later than its upper time window. Constraint (11) initiates the start time of each route with zero. Constraint (12) states the binary and nonnegative variables.

C. DISRUPTION RECOVERY PLAN WITH SPLIT DELIVERY

When transportation delay in one or more routes causes the initial routing plan to be infeasible, vehicles in transit should be rescheduled rapidly to reduce the loss from the failure of serving the remaining customers on time. The idea of disruption management provides a practical approach to address this situation, which attempts to recover the initial routing plan and diminish the negative effects of the recovered routing plan on the main participators.

1) DISRUPTION MEASUREMENTS

Given a disruption recovery scheme, three key participants (customers, drivers and delivery company) will be mainly affected by the deviations of the new routing plan, and corresponding disturbance are described quantitatively as disruption measurements [34]. For the convenience of further explanation, certain new notations regarding the situation when a disruptive event occurs are illustrated as follows.

$DisT$: occurrence time of a disruptive event.

IK : a set of vehicles in transit.

Q_k' : remaining load of vehicle $k \in IK$.

p_k : a virtual node representing the site of vehicle k in transit.

P : a set of virtual nodes, $P = \{p_1, p_2, \cdots, p_k \cdots, p_{|IK|}\}$.

C' : a set of vertices representing unserved customers.

N' : a set consisting of the elements in set C' and O, $N' = C' \cup O$, where O denotes a set of vertices representing copies of the depot as return centers of vehicles.

x_{ijh}^k : a binary number determined by the delivery sequence of the initial routing scheme, which equals 1 when arc (i, j, h) is traversed by vehicle k; 0 otherwise.

$\bar{x}_{ijh}^k$: a binary variable indicating the delivery sequence of vehicle k in the disruption recovery plan, which equals 1 when arc (i, j, h) is traversed by vehicle k; 0 otherwise.

t_{ik} : time of vehicle k arrives at node i.

q_{ik} : delivery quantity for node i by vehicle k.

The disruption measurements of customers, the delivery company and the drivers are sequentially analyzed as follows.

First, split delivery is an appropriate means to provide products for customers in time. However, each customer expects that his/her demand is satisfied by minimal visits to reduce the trouble of receiving goods. Hence, the visit frequency and the service start time should be traded off in the disruption recovery plan for each unserved customer. In addition, each customer is not of equal importance to the decision maker; thus, discrepant concerns will be assigned to them. For instance, long-term customers providing great benefits should be taken care of to avoid damage, while occasional customers can be serviced at the second position. Without loss of justice, the delivery company usually claims that unexpected arrival delay is guaranteed within a maximum limit to improve their business competition power. Therefore, in addition to highlighting the justice by establishing an allowable maximum delay time for each customer, the disruption recovery scheme should minimize the sum of weighted service dissatisfaction.

$$F_1 = \sum_{i \in C'} w_i \left(\mu_1 \Big(\sum_{j \in N'} \sum_{h \in H_{ij}} \sum_{k \in IK} \bar{x}_{ijh}^k - 1 \Big) + \mu_2 \sum_{k \in IK} \frac{\max\{t_{ik} - l_i, 0\} \cdot q_{ik}}{D_i(l_i - e_i)} \right)$$
$$\textit{Subject to } t_{ik} \le l_i + L,\ \forall i \in C',\ k \in IK \tag{13}$$

where w_i is the importance degree of customer i. Coefficients μ_1 and μ_2 are relative weights of visit frequency and service start time, respectively. Factor L in the constraint is defined as a commitment of the maximum tolerable delay limit.

Second, the delivery company cares particularly about reducing transportation cost and product deterioration loss. In the disruption recovery plan, some alternative routes may induce charge changes. Regardless of the constant cost of the initial routing plan, the delivery company will look forward to minimizing the delivery cost of new routing plan under the traffic conditions at that time. Therefore, the deviation of total relevant cost is expressed as follows:

$$F_2 = \theta_1 \sum_{i \in O} \sum_{k \in IK} t_{ik} + \theta_2 \sum_{i \in C'} \sum_{k \in IK} t_{ik} q_{ik} \tag{14}$$

Third, drivers are usually well primed with the assigned delivery routes. As the real-time disruption recovery plan is transmitted, route adjustment will trouble the drivers and make them feel tired of new paths. To diminish the disturbance on drivers, the new routing scheme should seek to maintain the initial routing scheme as much as possible. The deviation of total driving paths is described as the number of newly added or alternative paths.

$$F_3 = \sum_{i \in C' \cup P} \sum_{j \in N'} \sum_{h \in H_{ij}} \sum_{k \in IK} \max\left\{ \bar{x}_{ijh}^k - x_{ijh}^k, 0 \right\} \tag{15}$$

2) DISRUPTION RECOVERY MODELING

According to the assumption of product perishability, lost sales are not considered in the disruption recovery model. The recourse process is to split the total remaining load to fulfill the demand of unserved customers. Given the analysis above, a triple-objective mathematic model of disruption recovery scheme is formulated as follows.

$$\min\ \{P_1 : F_1,\ P_2 : F_2,\ P_3 : F_3\} \tag{16}$$

Subject to

$$P_1 \gg P_2 \gg P_3 \tag{17}$$

$$\sum_{i \in C'} \sum_{h \in H_{i,n+k}} \bar{x}_{i,n+k,h}^k = 1, \quad \forall k \in IK \tag{18}$$

$$\sum_{j \in N'} \sum_{h \in H_{ij}} \bar{x}_{p_k jh}^k = 1, \quad \forall k \in IK \tag{19}$$

$$\sum_{i \in C' \cup P} \sum_{h \in H_{ij}} \sum_{k \in IK} \bar{x}_{ijh}^k \ge 1, \quad \forall j \in C' \tag{20}$$

$$\sum_{i \in C' \cup P} \sum_{h \in H_{ij}} \bar{x}_{ijh}^k = \sum_{i \in N'} \sum_{h \in H_{ji}} \bar{x}_{jih}^k, \quad \forall j \in C',\ k \in IK \tag{21}$$

$$\sum_{k \in IK} q_{ik} \ge D_i, \quad \forall i \in C' \tag{22}$$

$$\sum_{i \in C'} q_{ik} \le Q_k', \quad \forall k \in IK \tag{23}$$

$$q_{ik} \le Q_k' \sum_{j \in N'} \sum_{h \in H_{ij}} \bar{x}_{ijh}^k, \quad \forall i \in C',\ k \in IK \tag{24}$$

$$t_{jk} \ge t_{ik} + s_i + \sum_{h \in H_{ij}} \tau_{ijh}(d_{ijh}, t_{ik} + s_i) - M \sum_{h \in H_{ij}} (1 - \bar{x}_{ijh}^k), \quad \forall i, j \in N',\ k \in IK \tag{25}$$

$$t_{ik} \ge e_i, \quad \forall i \in N',\ k \in IK \tag{26}$$

$$t_{ik} \le l_i + L, \quad \forall i \in N', \tag{27}$$

$$t_{p_k,k} = DisT \tag{28}$$

$$\bar{x}_{ijh}^k \in \{0, 1\},\ t_{ik} > 0, \quad \forall i, j \in N',\ k \in IK,\ h \in H_{ij} \tag{29}$$

In addition to the notations defined specially in this part, other symbols in the disruption recovery model retain the same meaning as the initial routing model. Objective function (16) strives to minimize triple disturbances from deviations between the disruption recovery plan and the initial routing scheme. The series of objective functions F_1, F_2 and F_3 represent the disturbances to customers, the delivery company and drivers, respectively; these are correspondently assigned to different preemptive priority levels P_1, P_2 and P_3. Constraint (17) indicates the priority order of different objectives, which can be adjusted according to the preference of the decision maker and the practical experiences. Constraints (18) to (21) ensure the flow conservation of vehicles in transit. Constraint (18) restricts the dispatched vehicles finally returning to the depot. Constraint (19) forces the vehicles at virtual nodes to move ahead to unserved customers or the depot. Constraint (20) states that each customer can be visited more than once. Constraint (21) ensures that the vehicle must leave a customer after finishing its service. Constraint (22) guarantees that the remaining demand of each customer is fulfilled. Constraint (23) restricts the total delivery quantity of a vehicle en-route is no more than its remaining load. Constraint (24) ensures that the delivery quantity for a customer is offered by a vehicle visiting it. Constraint (25) requires that the service start time of a customer respects the vehicle visiting sequence. Constraints (26) and (27) impose that customer service must be begun between its earliest allowable time and maximum limitation with tolerable delay. Constraint (28) realizes the start time of recovery routes with the occurrence time of a disruptive event. Constraint (29) introduces the involved binary and nonnegative variables.

In addition to nonlinear multi-objective functions, the disruption recovery model is a variant of the dynamic vehicle rerouting problem, which is an NP-hard problem. It is impossible to find optimal solutions to actual problems in a reasonable computation time; therefore, it is better to develop meta-heuristic methods to solve the model.

IV. TABU SEARCH ALGORITHM

The disruption recovery model is developed from the initial routing model. Except for several special characteristics, such as multiple criteria in objectives, virtual depots and split delivery of each route, constraints on customer requirements and service sequences are the same in both models. To facilitate an ongoing optimization, we attempt to adopt a unified computation framework to solve the two models. The computation process is introduced based on solving the initial routing model, and the disruption recovery model is resolved in the same manner, unless otherwise specified.

In this section, we introduce a tabu search (TS) heuristic to solve the above proposed models. TS provides an adaptive search mechanism that begins with an initial solution for solving optimization problems. The solution space is explored iteratively by simple local modifications to the current solution. To escape the trap of local optimality, the best neighboring solution is accepted as the incumbent event if its objective value deteriorates. In turn, the movement back to the newly visited solution is declared tabu for a certain number of iterations to avoid cyclic search. The motivation for using TS in our particular application is based on the fact that this metaheuristic approach has been used in a wide variety of classical and practical problems of a high degree of complexity, including several variants of the vehicle routing problem, such as vehicle routing problem with split delivery [7] and time-dependent vehicle routing problem [35]. On the other hand, among the meta-heuristics proposed for the vehicle routing problem, TS has been shown to be a very effective one, providing a good compromise between solution quality and computation time, which facilitates the solving of the disruption recovery model to provide an effective and timely respond for the vehicles en route. The algorithm has been adopted to solve some dynamic models of routing problems [26], [28].

The proposed approach differs from the TS procedures for the similar vehicle routing problems reported by Ho and Haugland [7] and Mu *et al.* [26] in two aspects. First, the supply of each route is equal to its remaining load in the disruptive recovery model, which requires that the neighborhood structures of the split delivery sustain the balance of supply and delivery. New neighborhood structures are designed for the specialties of the disruption recovery model, which have not been taken into account in the literature. Moreover, based on the above analysis about the difference between the initial routing problem and disruption recovery problem, TS is adjusted to accommodate to both models for the convenience of its practical application. A greedy randomized strategy is also introduced to improve the performance and accelerate the convergence speed of the proposed TS algorithm.

The procedure is outlined in Algorithm 1. First, an initial solution is constructed, and certain related parameters are initialized. Then, inter-route operators are employed to generate neighborhoods of the incumbent solution when it comprises a few routes. After evaluating all generated neighborhood solutions, the incumbent solution for the next iteration is replaced and then improved by intra-route operators. Finally, the best-thus-far solution is updated if a better feasible solution is found, followed by updating related parameters and tabu tenures. To provide a detailed description of the algorithm, a few critical components are explained, including the initial solution construction, neighborhood structures, objective evaluation, tabu list and stopping criterion, as well as the complexity analysis of the proposed algorithm.

A. INITIAL SOLUTION

In Algorithm 1, the initial solution generation process *GenerateInit*() is realized with a greedy randomized procedure, which is used for the initial routing model and described as Algorithm 2. Each route is from the depot and sequentially constructed by inserting one delivery node at the last place at a time. For each unvisited node j, the insertion cost into the partial route $R_{m,k} = (0, i_1, \cdots, i_k, 0)$ is evaluated by formula $c = \max\{t_{i_k} + s_{i_k} + \tau_{i_k jh}, e_j\} - t_{i_k}$, where $\tau_{i_k jh}$ is computed by the embedded iterative procedure of the time-dependent travelling time. To embody greed and the randomness property in the insertion process, a restricted candidate list (*RCL*) is constructed, which only includes feasible nodes with inserting

Algorithm 1 Pseudocode of the Proposed TS Algorithm

function TS( )
  % generate an initial solution and update the best-thus-far solution
  $S \leftarrow GenerateInit(\,), S^* \leftarrow S$
  $TabuTenure \leftarrow 0$*% initialize tabu tenures of operators*
  $\alpha \leftarrow 1, \beta \leftarrow 1$, choose $\gamma \in [0, 1]$ randomly *% initialize parameters*
 while terminal criterion is not meet **do**
  $N(S) \leftarrow Neighborhood(S)$*% generate inter-route neighborhoods of S*
  $S \leftarrow UpdateIncumbent(N(S))$*% update the incumbent solution S*
  $S \leftarrow IntraImprove(S)$*% generate intra-route neighborhoods of S*
  % update tabu tenures, penalty parameters and the best-thus-far solution
  $Update(TabuTenure, \alpha, \beta, \gamma, S^*)$
  end while
  return the best solution S^* found in the search process
end function

Algorithm 2 Pseudocode of Greedy Randomized Procedure

$m \leftarrow 1$*% initialize the number of solutions*
for $m \leq Psize$ **do**
  $k \leftarrow 1$*% initialize the number of routes*
  $R_{m,k} \leftarrow \{0, 0\}, i \leftarrow 0, \tilde{C} \leftarrow C$
  while the set $\tilde{C}$ is not empty **do**
   $c_{min} \leftarrow \min_{h \in H_{ij}, j \in \tilde{C}}\{\max\{t_i + s_i + \tau_{ijh}, e_j\} - t_i\}$
   $RCL \leftarrow \left\{ j \in \tilde{C} \,\middle|\, \begin{array}{l} c \leq (1+\alpha)c_{min},\ t_{n+k} \leq l_0, \\ Quantity(R_{m,k}) \leq Q) \end{array} \right\}$
   if RCL is not empty randomly select $j \in RCL$,
    $R_{m,k} \leftarrow \{0, \cdots, i, j, 0\}$, $C = C \backslash \{j\}$ and $i \leftarrow j$
   else
    $k \leftarrow k+1, R_{m,k} \leftarrow \{0, 0\}, i \leftarrow 0$ when $k+1 < |K|$
    Insert all nodes in C into the current route $R_{m,k}$ when $k + 1 = |K|$
   end if
  end while
 $S_m = \{R_{m,1}, R_{m,2}, \cdots, R_{m,k}\}$
end for
Return the optimal solution S^* with the minimum objective value

cost less than the percentage $(1+\alpha)$ of the minimum insertion cost; then, the node to be inserted is selected randomly from *RCL*. It is worth noting that, between every two consecutive nodes, the arc with the minimum travelling time is selected for travelling, since the optimal solution always possesses the feature that the arrival time at each customer is as early as possible for the given customer sequence, which complies with the property of objective function (2).

The insertion process is implemented until a node could no longer be inserted into the current route, and then a new route is created if available vehicles remain. If the current vehicle is the last one, all unvisited nodes will be continuously inserted into its route without considering the constraints of vehicle capacity ($Quantity(R_{m,k}) \leq Q$) and customers' time windows ($t_{n+k} \leq l_0$).

For the disruption recovery model, each initial solution contains routes with the number of used vehicles. Each route is generated from a virtual node, which represents a site of vehicles on the way, and then constructed sequentially using the same approach as in the solution construction of the initial routing problem. When the supply of the current route is insufficient to meet the demand of the last inserted node, the demand is split, and only the amount equal to the left supply is met by the current route.

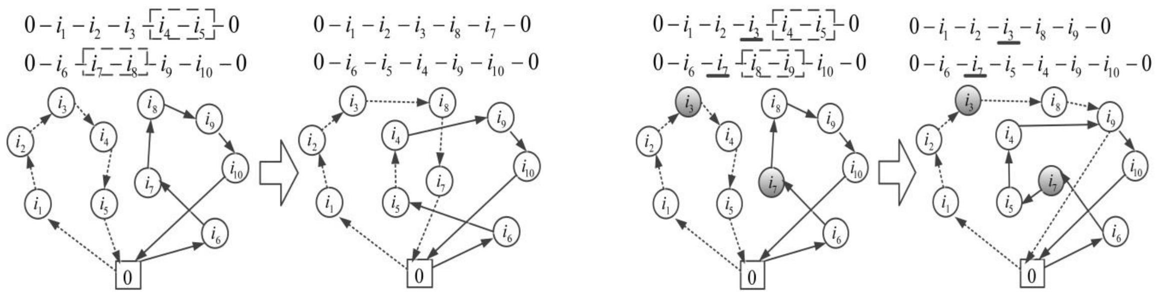
B. SOLUTION EVALUATION

To expand the search space, certain infeasible solutions are also accepted, which helps the incumbent solution to escape from being trapped in the local minima and enables it to move to promising regions. In this article, the infeasible solutions are evaluated by adding a penalty cost in the objective function. That is, the evaluation function is defined as $O(S) = F + \beta_1 \cdot \Delta C + \beta_2 \cdot \Delta T$, where ΔT is the total time excess, and ΔC is the total capacity excess. For a solution represented as $S = \{R_1, R_2, \cdots R_k\}$, the total capacity excess is calculated by $\Delta C = \sum_{R \in S} \max\{\sum_{i \in R} D_i - Q, 0\}$, while the total time excess is computed by $\Delta T = \sum_{R \in S} \sum_{i \in R} \max\{t_i - l_i, 0\}$. The departure time from node i is noted as $l_i + s_i$ if its service start time meets the condition that $t_i > l_i$; otherwise, equals $t_i + s_i$. Both penalty coefficients β_1 and β_2 are initially set as 1, and after each iteration, their value is divided by $1+\gamma$ ($\gamma \in (0, 1]$) if the corresponding constraint is respected; otherwise, they are multiplied by $1+\gamma$. The specific operations enable the search process to oscillate between feasible and infeasible solutions.

In the disruption recovery model, the capacity constraint of each neighborhood solution is always satisfied according to neighborhood structures described in Section 4.3. The penalty cost of the time window violation is only added in the objective function F_1, i.e., $O_1(S) = F_1 + \beta_2 \Delta T$. When two solutions have the same value of $O_1(S)$, the better solution is identified by the hierarchical comparison of the values of the subordinate objective functions F_2 and F_3.

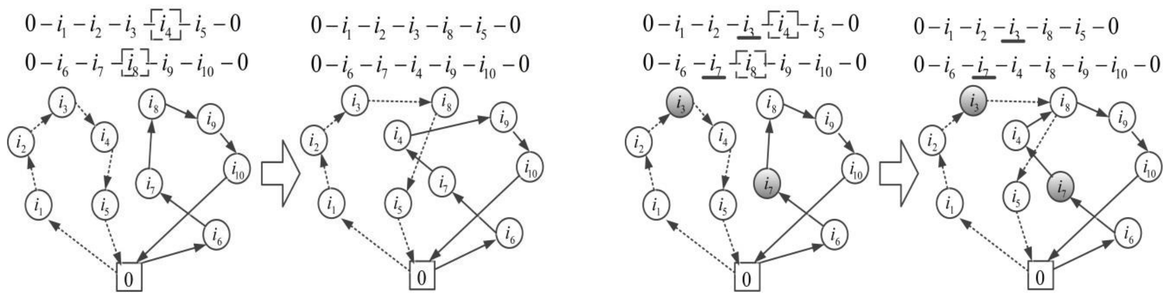
C. NEIGHBORHOOD STRUCTURES

In this study, three inter-route operators and one intra-route operator are implemented to explore a more extensive solution space. The former includes cross exchange, node exchange and inter-route relocation, while the latter is intra-route relocation. The inter-move and intra-move cooperate for exploration and exploitation. In each iteration, one of inter-route operators is randomly selected to be executed, and then the updated incumbent solution is improved by the intra-route operator. Particularly when the cross-exchange operator moves a string of consecutive visit nodes, reversing its sequence is also carried out to enrich the



(a) Cross exchange operation without split delivery (b) Cross exchange operation with split delivery

FIGURE 3. **Cross exchange operations for the initial routing plan and the disruption recovery scheme.**



(a) Node exchange operation without split delivery (b) Node exchange operation without with split delivery

FIGURE 4. **Node exchange operations for the initial routing plan and the disruption recovery scheme.**

neighborhood solutions. Since the restrictions of the disruption recovery model are different from those of the initial routing model, particularly the limitation on delivery capacity and the allowance of split delivery, the neighborhoods of their solutions have different characteristics. In the following, the details of the four neighborhoods are illustrated given the initial routing model, and certain differences for the disruption recovery model are supplemented later. Similar to the process of generating initial solution, the arc with minimum travelling time between any couple consecutive nodes should be selected for each route. Note that dashed boxes were used to select the subsequence of nodes to execute neighborhood operations in Fig. 3 to Fig. 5, and the nodes being visited or just visited by the travelling vehicles on the way at the time of disruption were underlined in these figures for split delivery.

1) CROSS EXCHANGE

From two randomly selected routes, two strings of consecutive visit nodes are extracted. Then, they are exchanged such that two new routes are reconstructed. Detailed operation is illustrated in Fig. 3(a).

For split delivery, the total delivery quantity of the two strings is compared first. While the string with less quantity is directly shifted into the other route, the string with more delivery quantity leaves the redundancy in its source route and only shifts a substring with an equal amount into the other route. Fig. 3(b) depicts the way this operator works. Nodes i_3 and i_7 are just visited by vehicles in two different routes. Strings $(i_8 - i_9)$ and $(i_4 - i_5)$ are selected to exchange. Then, string $(i_8 - i_9)$ only exchanges the demand equal to that of string $(i_4 - i_5)$, and the extra demand (a part of demand of node i_9) is left in its original route.

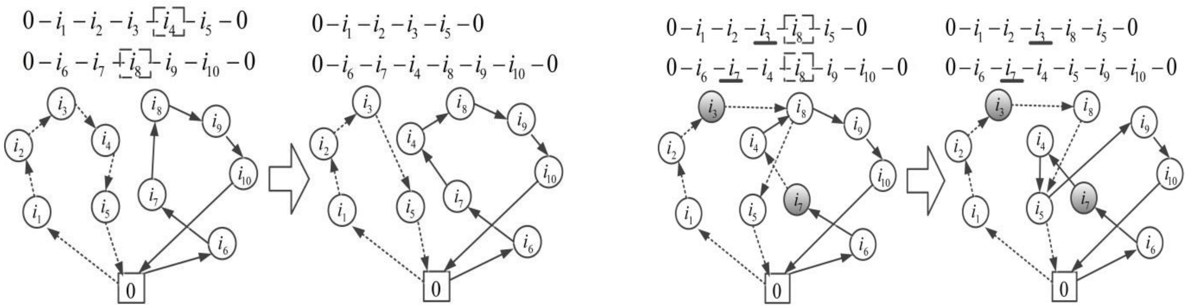
2) NODE EXCHANGE

As a special case of cross exchange operator, only one node is extracted from each selected route, and then the two nodes of different routes are exchanged. Through this operation, a smaller neighborhood is explored, as shown in Fig. 4(a).

For split delivery, the node with more delivery quality is essential to be divided into two same visit nodes. The delivery quantity of exchanged nodes remains equal to ensure the total delivery quantity of each vehicle coincide its load. Fig. 4(b) describes the process in detail. Nodes i_3 and i_7 are just visited by vehicles in two different routes. Nodes i_4 and i_8 are selected to exchange. Then, node i_8 leaves the extra demand in its original route and only the equal demand is exchanged with node i_4.

3) INTER-ROUTE RELOCATION

Two routes are selected randomly, and then a node in one route is removed into the other route at an optional position. However, in the case of split delivery, this operation is confined into two routes with more than one same visit node. A split visit node in one route is combined into the same visit node in the other route, and the route with delivery increase transfers a part of visit nodes with equal delivery quantity back to the former route. Detailed process is depicted



(a) Inter-route relocation operation without split delivery

(b) Inter-route relocation operation with split delivery

FIGURE 5. **Inter-route relocation operations for the initial routing plan and the disruption recovery scheme.**

in Fig. 5. Nodes i_3 and i_7 are just visited by vehicles in two different routes. Node i_8 is a split node visited by the two routes. Then, node i_8 in the second route is removed and combined with the demand of node i_8 in the first route, and a part of demand of another node i_5, which is equal to the demand increase in the first route, is transferred to the second route.

4) INTRA-ROUTE RELOCATION

Within a route, a node is randomly removed from its primary site into another site. Thus, the visit sequence of the route is reordered to help exploit a large solution space.

D. TABU LIST AND STOPPING CRITERION

In each iteration, the accepted neighborhood solution meets either of the following conditions: (i) the solution with the best evaluation value is non-tabu; and (ii) the tabu status of the solution is overridden by an aspiration criterion, which is feasible and has the best objective value thus far. When a move is employed, its inverse operation is set tabu for the next δ iterations until the tabu status is expired or overridden by the aspiration criterion, where tabu tenure δ is chosen randomly in $[1, \sqrt{n}]$ in each iteration. In this article, for the node exchange and the inter-route relocation operator, moving the operated visit node back to its source route is tabu; for the cross-exchange operator, moving the end of the exchanged string to its source route is set tabu. For split delivery, only the non-split node can be set tabu, which strives to decrease the frequency of split.

Regarding the above two models, if the best solution thus far, S^*, is not improved for consecutive *NoImp* iterations, the search stops and returns S^*.

E. COMPUTATIONAL COMPLEXITY

The time complexity of the proposed TS depends on some critical steps. When choosing the optimal path between two different nodes, all alternative paths are evaluated giving a time complexity of $O(|H| \cdot |U|)$, where $|H|$ is the maximum number of alternative paths between any two nodes, and $|U|$ is the maximum number of time periods segmented from the work horizon. To avoid high computational burden associated with excessive insertion trials, nodes that can be visited directly after each ahead point with the ideal travelling speed are first calculated in $O(|C|^2)$ time. In the process of generating a candidate initial solution, at most $|C|$ nodes are attempted to be inserted following the last element of the just constructed route. Then in worst case, the time complexity of evaluating all insertion operations for one solution is $O(|C| \cdot (|K| + |C|) \cdot |H| \cdot |U|)$. When *Psize* candidate initial solutions are generated, the overall complexity of the greedy randomized procedure is $O(|C| \cdot (|K| + |C|) \cdot |H| \cdot |U| \cdot Psize + |C|^2) = O(|C|^2 \cdot |H| \cdot |U| \cdot Psize)$.

To improve the performance of neighborhood operations, only nodes with similar visit time could be exchanged, and removed nodes are confined to be inserted into positions without excess waiting time. In the worst case, $|C|$-1 possible nodes to be exchanged or $|C|$ positions need to be evaluated for an neighborhood operation, and the computation complexity of determining an optimal position is $O(|C| \cdot |H| \cdot |U|)$. Then for a specific neighborhood solution, the complexity of checking time feasibility is $O(|C| \cdot |H| \cdot |U|)$. So, an appropriate neighborhood solution can be obtained and evaluated in $O(2|C| \cdot |H| \cdot |U|) = O(|C| \cdot |H| \cdot |U|)$ time. To mitigate expensive computation of evaluating too many neighborhoods, the proposed algorithm is improved by executing intra-route relocation operator every q iterations. Suppose Ng neighborhood solutions are generated in each iteration and total *Iter* iterations are processed. The time complexity of neighborhood search is $O(|C| \cdot |H| \cdot |U| \cdot Ng \cdot (Iter + \left\lceil \frac{Iter}{q} \right\rceil)) = O(|C| \cdot |H| \cdot |U| \cdot Ng \cdot Iter)$.

Apart from the above analysis, the complexity of other computation process can be neglected. Although the three objectives of the disruption discovery model need to be evaluated, it does not affect the computation complexity. Therefore, the computation complexity of the proposed algorithm is $O(|C|^2 \cdot |H| \cdot |U| \cdot Psize + |C| \cdot |H| \cdot |U| \cdot Ng \cdot Iter)$, which is directly associated with the problem scale $|C|$ and the complexity coefficient $|H|$ and $|U|$ of the road network, and also affected by the number *Psize* of initial candidate solutions and the number *Iter* of computation iterations. As the initial routing plan is executed, the disruption recovery model is awakened with a relatively small-scale problem, but its computation iteration maybe enlarged for lexicographical optimization.

V. EXPERIMENTS

In this section, the effectiveness of the presented models and TS algorithms are demonstrated by sample problems. Three computational experiments are conducted. First, the process of vehicle rerouting is exemplified by two different approaches on a small-size instance, and the results are compared to interpret the implication of disruption recovery approach. Second, to validate the effectiveness of the proposed algorithm, we compared the results of certain instances against the methods suggested for similar problems. Third, the effect of recovering disruptive events at different occurrence times are compared to illustrate the feature of disruption recovery plan with split delivery.

A. TEST PROBLEMS

Since there is no standard testing dataset appropriate for our problems, two sets of problems are created by modifying the benchmark problems from the literatures. For the first set of medium-scale instances, eighteen of Solomon's VRPTW benchmark problems [36] with 100 customers are randomly selected as the base data to generate 100-customer instances. According to the characteristics of customers' geographical locations, these instances are divided into three categories: C-type (clustered customers), R-type (uniformly distributed customers) and RC-type (a mix of R and C types), which can be further classified by the time window widths. To be fair, herein we construct six instances for each geography type, three with narrow time windows and three with large time windows. Additionally, the second set of test instances is composed of forty-eight large-scale instances (200-customer instances, 400-customer instances, 600-customer instances and 800-customer instances), which are generated from Gehring and Homberger's data sets [37] for the VRPTW problem. This benchmark problems were constructed based on Solomon's data, and consequently, they are also classed into six groups by customers' locations and time windows. Hence, two instances are sampled randomly from each group with any scale of customers mentioned above. In total, sixty-six instances derived from the above two benchmark sets were tested using the tabu search algorithm. Each instance is coded as A_Bx_id, where A is the abbreviation of the author name, B denotes the geography property, and x and id are the customer scale and identifier of the instance, respectively.

For each specific instance, all data related to customers remain dimensionless. Some essential corrections are made to supplement necessary information, such as arcs between couples of nodes and time-dependent travel speeds on arcs. Each couple of road network nodes has 2∼3 arcs with the length of a percent λ ($\lambda \in [0.7, 1.3]$) of the Euclidean distance between them. The importance degree of each customer follows a discrete uniform distribution $U(1, 3)$. For simplicity, the planning horizon is divided into three equal intervals, and then five types of time-dependent speed functions are devised, each with the pattern $(1-\varepsilon, 1+\varepsilon, 1-\varepsilon)$. Each value of $\varepsilon(\varepsilon \in \{0.0, 0.1, 0.2, 0.3, 0.4\})$ corresponds to one speed profile. To construct a time-dependent network, each arc is randomly assigned a speed profile. When disruptive events occur, the real-time speed on an arc is set as a percentage λ of its current time-dependent speed.

All proposed procedures are implemented in MATLAB R2014a and run on a PC with a Core i3, 2.13 GHz CPU and 4 GB of memory. These parameters are best set by tuning experiments as follows: $Psize = 10$, $\alpha = 0.15$, $NoImp = 50$ and $q = \lfloor \sqrt{n} \rfloor$.

B. ILLUSTRATION OF DISRUPTION RECOVERY APPROACH

An example is conceived by the first 25 customers of instance S_R100_1 (as shown in Table 5) modified from R101 of Solomon benchmark, since the requirement of customers is randomly distributed. To reflect the importance of both items in function (2), the cost coefficients are set as $\theta_1 = 1$ and $\theta_2 = 0.015$. Assume that the depot possesses 8 vehicles and solve the initial routing model, the optimal result is shown in Table 2 and illustrated in Fig. 6. The squares in Fig. 6 represent the customer locations.

TABLE 2. Details of the initial routing plan of an illustrated example.

No.	Route	No.	Route
1	0-12-9-20-1-0	5	0-5-7-18-10-0
2	0-14-16-6-13-0	6	0-2-15-22-24-0
3	0-23-3-4-25-0	7	0-21-0
4	0-11-19-8-17-0		

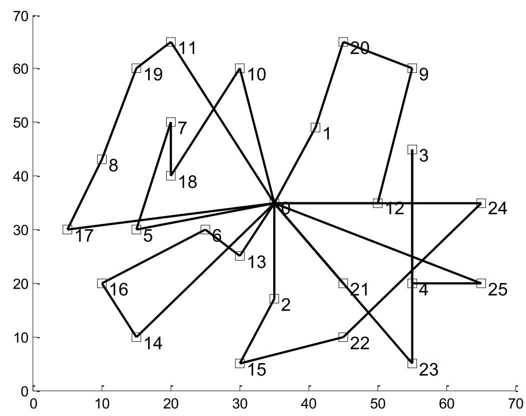


FIGURE 6. The initial routing plan of an illustrated example with 25 customers.

According to the predetermined routes, 7 vehicles are dispatched to deliver the signed orders. Suppose that all the dispatched vehicles were travelling in their specific sites (i.e., virtual nodes in the set P) at time 67, which are marked with solid circles in Fig. 7 and Fig. 8. It was found that vehicles 3 and 5 were both affected by traffic accidents; their transportation delay times are 30.5 and 26, respectively. It is evident that the initial routing plan is not applicable by the subsequent evaluation.

TABLE 3. Details of the recovered routing scheme for delivery delay.

No.	Route	No.	Route
1	0-12- p_1 -9-20-1-0	5	0-5- p_5 -18-***8***-10-0
2	0-14- p_2 -16-6-13-0	6	0-2- p_6 -15-3-24-***25***-1
3	0- p_3 -23-22-4-***25***-0	7	0-21- p_7 -0
4	0-11(p_4)-19-7-***8***-17-0		

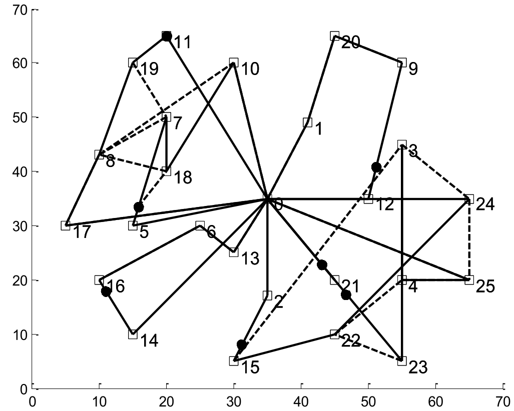


FIGURE 7. The recovered routing scheme produced by the disruption recovery model for delivery delay.

According to the disruption recovery approach proposed in this article, the best disruption recovery plan is revealed in Table 3 and illustrated in Fig. 7, which is the best result among 5 repeat computations with average consuming times of 38.5 seconds. The objective function values of disruption recovery model are $F_1 = 1.4272$, $F_2 = 1478.2$, and $F_3 = 11$. Here, coefficients μ_1, μ_2 and L are respectively set as 0.1, 0.9 and 30. Furthermore, another vehicle rerouting scheme is calculated by a global rescheduling method, which considers a re-execution of the initial routing model. Thus, only the interest of the delivery company is focused on instead of the effects of the recovery scheme on all participators. The computation results are provided in Table 4 and illustrated in Fig. 8. The objective function values of the global rescheduling method are $F_1 = 10.4491$, $F_2 = 1398.2$, and $F_3 = 13$. In both the two figures, full lines and dotted lines represent the initial routes and recovery routes, respectively.

TABLE 4. Details of the global rescheduling routing scheme for delivery delay.

No.	Route	No.	Route
1	0-12- p_1 -9-20-1-0	5	0-5- p_5 -18-7-10-0
2	0-14- p_2 -16-***6***-17-13-0	6	0-2- p_6 -15-3-24-***4***-1
3	0- p_3 -23-22-***4***-25-0	7	0-21- p_7 -0
4	0-11(p_4)-19-8-***6***-0		

In Tables 3 and 4, the split delivery nodes are marked in bold. There are two split nodes in both solutions. By comparing the objective function values, the disruption

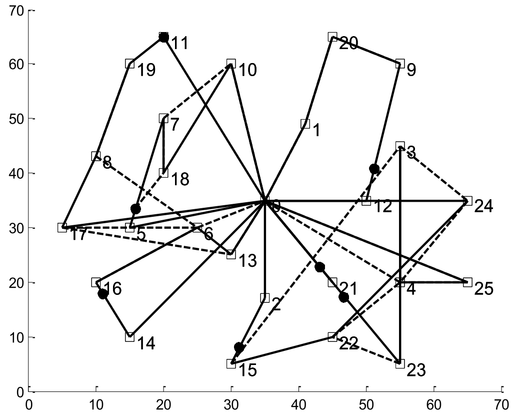


FIGURE 8. The global rescheduling routing scheme produced by reoptimizing the initial routing model for delivery delay.

recovery scheme has fewer negative effects on customers with approximately equal cost, while the global rescheduling scheme leads to a larger influence on customers with minimal cost saving. It can be inferred that the disruption recovery model is more appropriate in real applications.

C. COMPARISON OF ALGORITHM PERFORMANCE

Suitable performances of the adapted ant colony optimization (ACO) algorithm have been achieved on disruption management models [38], [39] and SDVRP [40], which are closest to our study. Hence, the ACO algorithm is fit for solving our problem with certain modifications. To meet the rigid constraint that the supply quantity of each vehicle is equal to the left load, initial solutions of the ACO algorithm are constructed with the method that the unmet demand of the last inserted customer is split and left to other routes when the supply of the current route is not sufficient. The search termination criteria are the same as that of the TS algorithm. Based on parameters tuning experiences, the size of ant solutions is set as 20 and importance levels of heuristic information and pheromone trail are set at 1.5 and 4, respectively. Similar to the work of Ding *et al.* [38] the ACO algorithm procedure is described briefly as follows.

- Step 1: generating initial ant solutions sequentially according to the pheromone trail and heuristic information. Each ant solution consists of several routes with the number of vehicles in transit. Each route begins from a virtual node indicating the vehicle site. When the supply of a route is insufficient to meet the demand of the last inserted node, the unmet amount is split as a new node.
- Step 2: evaluating the generated solutions by the methods introduced in the part B of Section IV. The best local solution is the recorder as S_{local}, and then the neighborhood operators introduced in the part C of Section IV are performed on it. If S_{local} is improved, S_{local} is updated. If S_{local} is better than the recoded global best solution, S_{global}, S_{global} is also updated.
- Step 3: updating the pheromone trail on the routes of solution S_{local}. Repeating Step 1 if the termination criteria are not met.

TABLE 5. Results of TS and ACO algorithms on the instances modified from Solomon benchmark.

Instances	Original problem	ACO			TS			$Rate_{TS}$
		F1	F2	F3	F1	F2	F3	
S_C100_1	C102	**0**	**9389.7**	18	0	9396.1	8	-0.07%
S_C100_2	C103	0	7896.6	13	**0**	**7889.6**	14	0.09%
S_C100_3	C105	1.1	9020.1	16	**0.7**	8632	16	36.4%
S_C100_4	C201	**0**	**8118**	12	0	8204.3	16	-1.1%
S_C100_5	C204	0.9	7239.5	9	**0.6**	7081.4	12	33.3%
S_C100_6	C205	4.2	7824.1	14	**1.3**	6817.1	18	69.0%
S_R100_1	R101	5.9	3508.9	6	**1.6**	3605.9	4	72.9%
S_R100_2	R102	2.1	4819.2	9	**1.1**	5007.3	13	47.6%
S_R100_3	R104	3.6	6391.3	17	**2.3**	7694.3	19	36.1%
S_R100_4	R201	1.5	4711	10	**0**	4637.5	13	100%
S_R100_5	R203	10.4	5463.9	9	**8.7**	5396.3	11	16.3%
S_R100_6	R206	9.1	4913.2	17	**5.2**	4615.8	19	42.9%
S_RC100_1	RC102	3.7	2941.4	10	**1.9**	3362	4	48.6%
S_RC100_2	RC103	0	2312.5	3	**0**	**2300.5**	3	0.5%
S_RC100_3	RC105	9.3	3199.8	5	**5.4**	3261.3	8	41.9%
S_RC100_4	RC201	7.1	9010.1	9	**6.5**	**8520.3**	14	8.5%
S_RC100_5	RC203	4.7	5869.2	11	**3.1**	5691.6	5	34.0%
S_RC100_6	RC205	8.2	8220.3	9	**5.8**	6813.6	12	29.3%

The effectiveness of the TS algorithm is investigated by an experiment test conducted on different scale instances in comparison with ACO. For equitable performance appraisals within acceptable disruption reaction time, the 100-customer instances are resolved within 2 minutes while the calculation time limit of other larger instances is 5 minutes. All the computation results are shown in Tables 5 and 6. Columns F_1, F_2 and F_3 list the best objective values among 10 repeated calculations. According to the objective priority, the computation results of the two algorithms are compared hierarchically. For each instance, when the objective values obtained by the two algorithms are different, index $Rate_{TS}$ is calculated to reflect the superiority of the TS algorithm. $Rate_{TS}$ is defined as $Rate_{TS} = \frac{F_{*,\,ACO} - F_{*,\,TS}}{\max(F_{*,\,ACO}, F_{*,\,TS})}$, where "$*$" represents the priority level.

From the last column in Table 5, we can observe that the results of these 100-customer instances show a significant superiority of the TS algorithm over the ACO algorithm. The former achieves better outcomes in sixteen of total eighteen instances, except two instances with slightly worse solutions. As the scale of the instances expands, the advantage of the TS algorithm begins to fade, which is discovered from Table 6. In every twelve instances for different scales, with 200, 400, 600 and 800 customers, there are seven, five, six and five instances with better solutions, respectively. The best results in Tables 5 and 6 are bold-faced. Statistical analysis of the comparative improvement rates of the TS algorithm on different scale instances are reported in Fig. 9. We can see that the average improvement rates approach 40% on the 100- and 200- customer instances. Although slightly worse solution results were achieved with the TS algorithm on the half of the larger scale instances (with 400, 600 and 800 customers), the comparative weakness rate is less than 25% on average.

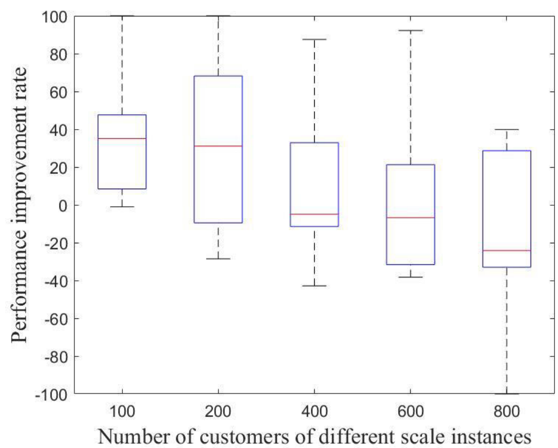


FIGURE 9. Performance statistic of the proposed TS by comparing with the ACO.

Given the analysis above, the TS algorithm, with a time-consuming process of examining neighborhood-based changes for better solution, overall, is appropriate for the general scale problem investigated in this article. It is asserted that the solutions of large-scale instances can be improved when the computation time is prolonged. In addition, the TS algorithm is easy to be realized and convenient to address the proposed two models in the same algorithm framework. In contrast, the ACO algorithm has parallel search capabilities to obtain satisfactory solutions; however, its shortcomings of the local search ability impede the performance improvement.

D. EFFECT OF DISRUPTION MANAGEMENT

To evaluate the effectiveness of the disruption recovery model for a different number of disruptive events at different times,

TABLE 6. Results of TS and ACO algorithms on the larger instances modified from Gehring & Homberger benchmark.

Instances	Original problem	ACO			TS			$Rate_{TS}$
		F1	F2	F3	F1	F2	F3	
GH_C200_1	C1_2_1	**0**	**9389.7**	18	0	10376	23	-9.5%
GH_C200_2	C1_2_5	0.4	9600.2	15	**0**	9543.7	19	100%
GH_C200_3	C2_2_3	1.1	14152.9	21	**0.7**	16152.9	19	36.4%
GH_C200_4	C2_2_4	**1.9**	15011.7	19	2.1	15716.2	24	-9.5%
GH_R200_1	R1_2_2	2.3	16108.3	35	**1.6**	15909.2	31	30.4%
GH_R200_2	R1_2_4	1.3	13850.2	20	**0**	14218.1	16	100%
GH_R200_3	R2_2_3	8.4	20147.3	24	**5.6**	19741.9	25	33.3%
GH_R200_4	R2_2_5	6.3	15991.6	28	**4.3**	17896.3	19	31.7%
GH_RC200_1	RC1_2_2	3	14719.3	15	**0**	14311.2	21	100%
GH_RC200_2	RC1_2_5	**7.2**	17210.1	21	8.1	17981.6	16	-11.1%
GH_RC200_3	RC2_2_1	**3.1**	**15989.2**	19	3.1	16833.9	23	-5.0%
GH_RC200_4	RC2_2_4	**4**	21017.9	26	5.6	20213.8	21	-28.6%
GH_C400_1	C1_4_2	**0**	**30240.1**	27	0	31519	29	-4.1%
GH_C400_2	C1_4_5	**4.8**	37024.3	30	5.1	36138.1	21	-5.9%
GH_C400_3	C2_4_2	10.9	40916.4	24	**9.3**	40148.2	31	14.7%
GH_C400_4	C2_4_7	**5.3**	30109.2	33	5.8	29461.6	28	-8.6%
GH_R400_1	R1_4_2	**7.4**	35413	24	8.6	31891.1	20	-13.9%
GH_R400_2	R1_4_5	**7.1**	38220.7	20	7.8	36671.2	17	-8.9%
GH_R400_3	R2_4_3	9	38421.4	23	**5.4**	41081.2	32	40%
GH_R400_4	R2_4_6	3.2	41020.5	31	**0.4**	38091	28	87.5%
GH_RC400_1	RC1_4_2	**7.6**	30896	30	9.6	31209.2	35	-20.8%
GH_RC400_2	RC1_4_4	11.8	35712	26	**8.3**	33461.1	30	29.7%
GH_RC400_3	RC2_4_3	**3.2**	40037.5	19	5.6	40120.7	21	-42.8%
GH_RC400_4	RC2_4_6	4.7	31216.2	34	**3**	31241.6	32	36.2%
GH_C600_1	C1_6_1	**7.9**	40813.7	29	10.1	39401.3	38	-21.8%
GH_C600_2	C1_6_5	6.4	45271.3	35	**0.5**	42132.6	31	92.2%
GH_C600_3	C2_6_3	6.3	51204.5	24	**5.7**	48143.5	26	9.5%
GH_C600_4	C2_6_5	**9.4**	47398.1	35	14.6	50126.3	31	-35.6%
GH_R600_1	R1_6_3	**10.2**	50172.9	27	14.2	5	33	-28.2%
GH_R600_2	R1_6_4	**10.8**	42073.6	41	17.5	43961.6	39	-38.3%
GH_R600_3	R2_6_2	22.9	49133.6	37	**14.8**	50881.3	42	35.4%
GH_R600_4	R2_6_4	11	53170.2	40	**9.9**	54216.3	68	13.6%
GH_RC600_1	RC1_6_1	0	50281	28	**0**	**46190.6**	37	8.1%
GH_RC600_2	RC1_6_4	10.4	43962.9	39	**7.4**	48163.2	53	28.8%
GH_RC200_3	RC2_6_3	**9.2**	43701.6	46	11.8	39157.1	43	-22.0%
GH_RC600_4	RC2_6_5	**6.1**	39910.4	29	9.4	42810.7	41	-35.1%
GH_C800_1	C1_8_2	**0**	40183.7	31	2.2	40215.3	36	-100%
GH_C800_2	C1_8_6	**9.4**	45216.3	37	11.9	43120.6	48	-21.0%
GH_C800_3	C2_8_3	14.4	43100.8	47	**9.2**	42261.1	43	36.1%
GH_C800_4	C2_8_5	26.1	54209.3	40	**15.7**	51380.2	37	39.8%
GH_R800_1	R1_8_1	**12.7**	52937.1	41	17.5	53109.6	52	-27.4%
GH_R800_2	R1_8_3	**7.3**	44819.3	30	11.2	46851.2	39	-34.8%
GH_R800_3	R2_8_4	9.1	49210.6	26	**6.5**	46988.6	49	28.6%
GH_R800_4	R2_8_6	23.2	43190.6	50	**17.3**	37625.8	61	25.4%
GH_RC800_1	RC1_8_2	**3**	49081.3	56	5.6	50720.1	49	-46.4%
GH_RC800_2	RC1_8_4	22	60292.2	52	**15.7**	63104.3	41	28.6%
GH_RC800_3	RC2_8_1	**7.3**	50311.7	37	10.6	45199.8	52	-31.1%
GH_RC800_4	RC2_8_4	**12.9**	52810.3	29	18.4	51326.4	38	-29.8%

nine instances are randomly extracted from the above generated test set. In the light of solving efficiency of the proposed algorithm on different scale problems, they are composed of three instances with 100 customers, three with 200 customers and the other three with 400, 600 and 800 customers, respectively. For each instance, the same disruptive events are set at time t_1 and t_2 to generate two different situations, which correspond to 0.3 and 0.6 of the planning horizon,

TABLE 7. Maximum delay time and total delay time caused by a single disruptive event.

Instance No.	time t_1				time t_2			
	MDT^0	TDT^0	MDT^1	TDT^1	MDT^0	TDT^0	MDT^1	TDT^1
S_C100_2	41.3	233.7	23.6	161.3	57.2	412.6	36.9	293.5
S_R100_5	19.6	73.5	12.7	42.8	32.6	189.2	28	140.8
S_RC100_5	52.3	493.3	41.9	381.8	106.3	896.6	104.2	554.1
GH_C200_2	37.9	139.5	21.6	94.2	49.8	189.3	37.7	150.4
GH_R200_3	42.1	164.8	27.9	105.2	79.2	613.8	63.4	479.3
GH_RC200_1	27.8	141.3	19.5	104.1	103.6	329.7	81	277.1
GH_R400_1	61.5	260.3	43.8	179.6	68.1	1016.2	56.3	741.4
GH_C600_3	29.3	301.8	20.3	184.1	43.4	284.1	35.1	226.8
GH_RC800_2	59.7	213	39.3	142.7	49.2	306.5	37.4	257.6

TABLE 8. Maximum delay time and total delay time caused by double disruptive events.

Instance No.	time t_1				time t_2			
	MDT^0	TDT^0	MDT^1	TDT^1	MDT^0	TDT^0	MDT^1	TDT^1
S_C100_2	73.3	379.6	53.1	252.6	104.6	651.7	83.7	508.4
S_R100_5	25.4	132.2	25.4	74.2	43.6	311.2	44.6	290.4
S_RC100_5	98	968.8	105.7	778.9	183	1381.6	183	1268.1
GH_C200_2	59.1	217.2	39.6	142.6	76.2	325.8	78.2	251.4
GH_R200_3	74.6	274.6	52.4	197.2	133.5	1013.9	121.5	892.6
GH_RC200_1	48.3	241.5	48.3	173.1	154.1	574.3	163.1	471.2
GH_R400_1	106.4	417.3	92.7	306.1	149.4	1833.6	149.4	1531.6
GH_C600_3	41.6	586.8	41.6	410.2	80.2	503.7	70.4	457.2
GH_RC800_2	90.7	338.2	74.2	263.2	76.4	521.4	63.3	423.8

respectively. At each time point, two scenarios are derived, one disruptive event occurring on the longest route and double occurring on the first two longest routes, respectively. For each disruption scenario, the best solution is attained by the TS and ACO, each with ten repeated runs. The corresponding maximum delay time (MDT) and total delay time are calculated and compared with its direct consequence of an uninterrupted initial routing plan. For comparison, we cancel the delay time limits in the disruption recovery model.

All computation results of single and double disruptive events are shown in Tables 7 and 8, respectively. MDT and TDT are marked with superscript "0" and "1", which are used to identify the uninterrupted initial routing plan and the disruption recovery scheme, respectively.

Given the results shown in Tables 7 and 8, two indices could be calculated, i.e., the recovery rate of the maximum delay time (RM) and the recovery rate of the total delay time (RT). RM is defined as RM $= \frac{MDT^0 - MDT^1}{MDT^0}$, while RT is defined as RT $= \frac{TDT^0 - TDT^1}{TDT^0}$. To differentiate the scenarios with single and double disruptive events at a time, the prefixes "S" and "M" are used, respectively, before indices. That is, SRM signifies RM of a single disruptive event, while DRM signifies RM of double disruptive events, and so is RT. The summary statistics of the results related to RM and RT are described in Fig. 10 and Fig. 11.

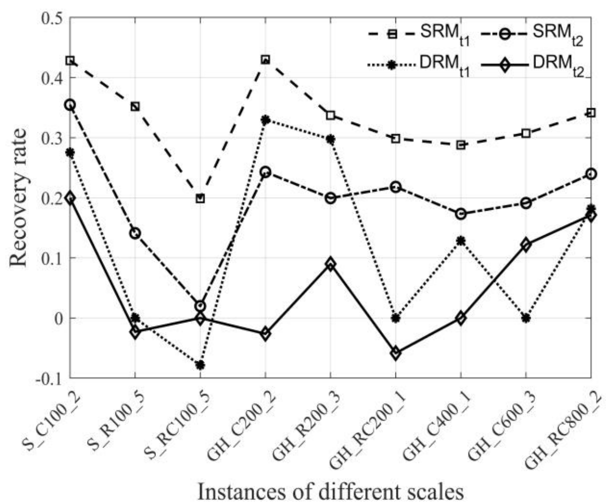


FIGURE 10. The recovery rates of the maximum delay time against the effects of adhering to the initial plan.

From Fig. 10, we can observe that curves SRM$_{t1}$ and SRM$_{t2}$ distribute over DRM$_{t1}$ and DRM$_{t2}$, respectively, and similarly, except for two nodes close to its counterpart, curves SRM$_{t1}$ and DRM$_{t1}$ spread over SRM$_{t2}$ and DRM$_{t2}$, respectively. It is concluded that the disruption recovery scheme can reduce the maximum delay time of earlier disruptive events to a larger extent, and the maximum delay time of a single

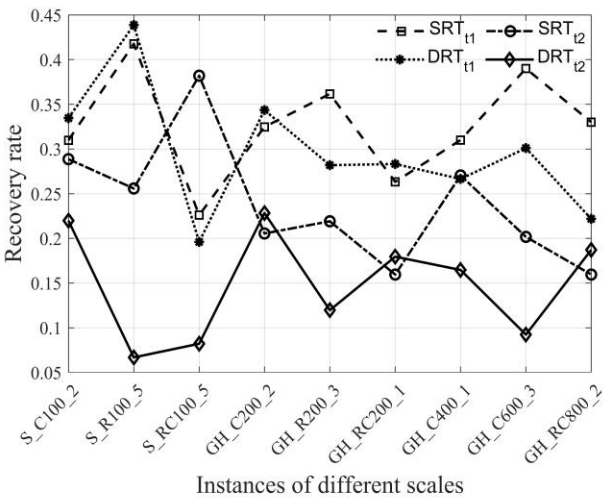


FIGURE 11. The recovery rates of the total delay time against the effects of adhering to the initial plan.

disruption event can be shortened by a higher percentage than that of double disruption events. Fig. 11 reports a similar trend of the curves. A disruption recovery scheme can reduce the total delay time of earlier disruptive events more significantly. However, it is not ensured that a higher percentage of total delay time is saved in a single disruptive event than that of double disruptive events.

This phenomenon can be explained by the observation that a more feasible recovery plan can be found for earlier disruptive events since all dispatched vehicles travel near the depot with short distances between one another and load a large volume of products for inter-route recourse.

Although double disruptive events occurring at an early time can continue to be rescued efficiently, with a mere split delivery-based recourse method, it is difficult to address that occurring at a later time. In this case, other methods may need to be adopted, such as order cancelation.

VI. CONCLUSION

In this article, we study a disruption recovery model for transportation delays in perishable product delivery. To characterize the problem accurately, inherent characteristics are integrated into the proposed models, which embrace the nature of perishable products and the time-dependent travel speed of urban traffic. In addition, to describe the path selection under dynamic traffic conditions in urban areas, more than one edge between each pair of road network nodes is considered. In the initial routing problem, the model's objective is to minimize the total cost including value loss of perishable products and transportation cost during the delivery. The perishability of products enables the load of scheduled vehicles to just meet the total demand of assigned customers. Therefore, in the disruption recovery model for transportation delay, there is no more supply to meet unserved customers, and split delivery may be the only strategy for inter-route recourse. With the disruption measurements of customers, the delivery company and drivers being analyzed, a triple-objective model is formulated to provide a disruption recovery plan. Furthermore, a TS algorithm is proposed to solve the initial routing problem, which is also extended to address the disruption recovery model with certain modifications in the neighborhood search process. Finally, a small-size disruption recovery example demonstrates the advantage of the disruption recovery approach over the traditional global rescheduling approach. The effectiveness of the proposed algorithm is validated by the comparison of the computation results with the ACO algorithm. The effect analysis of disruption occurrence time reveals that an earlier transportation delay can be relieved more effectively.

However, the scope of this work is limited to the split delivery-based method for the recovery of transportation delay. In certain extremely difficult circumstances, solely using this method is not efficient to address an excessive delay occurring in the terminal portion of a route. Future work can be conducted by considering more recourse methods, such as order cancelation, in the disruption recovery model. In addition, more efficient algorithms should be studied to accommodate to the real-time vehicle rescheduling process, particularly for the large-scale real-life problem, and then the performance of actual disruption recovery operation could be further enhanced.

REFERENCES

[1] G. H. Zhou, X. L. Xu, and Y. Liu, "Preservation technologies for fresh meat – a review," *Meat Sci.*, vol. 86, no. 1, pp. 119–128, Sep. 2010.

[2] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, Nov. 2009.

[3] C. Liu, G. Kou, X. Zhou, Y. Peng, H. Sheng, and F. E. Alsaadi, "Time-dependent vehicle routing Problem with time windows of city logistics with a congestion avoidance approach," *Knowl.-Based Syst.*, vol. 188, 104813, pp. 1–13, Jan. 2020.

[4] L. Cadarso and Á. Marín, "Combining robustness and recovery in rapid transit network design," *Transportmetrica A, Transp. Sci.*, vol. 12, no. 3, pp. 203–229, Jan. 2016.

[5] A. Nuzzolo and A. Comi, "Advanced public transport and intelligent transport systems: New modelling challenges," *Transportmetrica A, Transp. Sci.*, vol. 12, no. 8, pp. 674–699, Apr. 2016.

[6] C. Archetti and M. G. Speranza, "Vehicle routing problems with split deliveries," *Int. Trans. Oper. Res.*, vol. 19, nos. 1–2, pp. 3–22, Jan. 2012.

[7] S. C. Ho and D. Haugland, "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries," *Comput. Oper. Res.*, vol. 31, no. 12, pp. 1947–1964, Oct. 2004.

[8] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959.

[9] Y. Zhou, H. Yu, Z. Li, J. Su, and C. Liu, "Robust optimization of a distribution network location-routing problem under carbon trading policies," *IEEE Access*, vol. 8, pp. 46288–46306, 2020.

[10] Y. Qiu, L. Wang, X. Fang, P. M. Pardalos, and B. Goldengorin, "Formulations and branch-and-cut algorithms for production routing problems with time windows," *Transportmetrica A, Transp. Sci.*, vol. 14, no. 8, pp. 669–690, Jan. 2018.

[11] T. Vidal, G. Laporte, and P. Matl, "A concise guide to existing and emerging vehicle routing problem variants," *Eur. J. Oper. Res.*, vol. 286, no. 2, pp. 401–416, Oct. 2020.

[12] B. Adenso-Díaz, M. González, and E. García, "A hierarchical approach to managing dairy routing," *Interfaces*, vol. 28, no. 2, pp. 21–31, Apr. 1998.

[13] C. D. Tarantilis and C. T. Kiranoudis, "A meta-heuristic algorithm for the efficient distribution of perishable foods," *J. Food Eng.*, vol. 50, no. 1, pp. 1–9, Oct. 2001.

[14] C. D. Tarantilis and C. T. Kiranoudis, "Distribution of fresh meat," *J. Food Eng.*, vol. 51, no. 1, pp. 85–91, Jan. 2002.

[15] J. M. Belenguer, E. Benavent, and M. C. Martínez, "RutaRep: A computer package to design dispatching routes in the meat industry," *J. Food Eng.*, vol. 70, no. 3, pp. 435–445, Oct. 2005.

[16] A. Osvald and L. Z. Stirn, "A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food," *J. Food Eng.*, vol. 85, no. 2, pp. 285–295, Mar. 2008.

[17] C.-I. Hsu, S.-F. Hung, and H.-C. Li, "Vehicle routing problem with time-windows for perishable food delivery," *J. Food Eng.*, vol. 80, no. 2, pp. 465–475, May 2007.

[18] H. Hu, Y. Zhang, and L. Zhen, "A two-stage decomposition method on fresh product distribution problem," *Int. J. Prod. Res.*, vol. 55, no. 16, pp. 4729–4752, Feb. 2017.

[19] P. Amorim and B. Almada-Lobo, "The impact of food perishability issues in the vehicle routing problem," *Comput. Ind. Eng.*, vol. 67, pp. 223–233, Jan. 2014.

[20] F. E. Zulvia, R. J. Kuo, and D. Y. Nugroho, "A many-objective gradient evolution algorithm for solving a green vehicle routing problem with time windows and time dependency for perishable products," *J. Clean Prod.*, vol. 242, Jan. 2020, Art. no. 118428.

[21] B. D. Song and Y. D. Ko, "A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery," *J. Food Eng.*, vol. 169, pp. 61–71, Jan. 2016.

[22] Z.-J. Ma, Y. Wu, and Y. Dai, "A combined order selection and time-dependent vehicle routing problem with time widows for perishable product delivery," *Comput. Ind. Eng.*, vol. 114, pp. 101–113, Dec. 2017.

[23] J. Clausen, J. Larsen, A. Larsen, and J. Hansen, "Disruption management-operations research between planning and execution," *OR/MS Today*, vol. 28, no. 5, pp. 40–43, 2001.

[24] G. Yu, and X. T. Qi, *Disruption Management: Framework, Models and Application*. Singapore: World Scientific, 2004.

[25] J.-Q. Li, P. B. Mirchandani, and D. Borenstein, "Real-time vehicle rerouting problems with time windows," *Eur. J. Oper. Res.*, vol. 194, no. 3, pp. 711–727, May 2009.

[26] Q. Mu, Z. Fu, J. Lysgaard, and R. Eglese, "Disruption management of the vehicle routing problem with vehicle breakdown," *J. Oper. Res. Soc.*, vol. 62, no. 4, pp. 742–749, Apr. 2011.

[27] M. Nikolić and D. Teodorović, "Vehicle rerouting in the case of unexpectedly high demand in distribution systems," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 535–545, Jun. 2015.

[28] Q. Mu and R. W. Eglese, "Disrupted capacitated vehicle routing problem with order release delay," *Ann. Oper. Res.*, vol. 207, no. 1, pp. 201–216, Aug. 2013.

[29] X. Wang, J. Ruan, and Y. Shi, "A recovery model for combinational disruptions in logistics delivery: Considering the real-world participators," *Int. J. Prod. Econ.*, vol. 140, no. 1, pp. 508–520, Nov. 2012.

[30] Q. Ding, X. Hu, and Y. Wang, "A model of disruption management for solving delivery delay," in *Advances in Intelligent Decision Technologies. Smart Innovation, Systems and Technologies*, vol. 4. Berlin, Germany: Springer, 2010, pp. 227–237. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-14616-9_22

[31] B. Yuan and Z. Jiang, "Disruption management for the real-time home caregiver scheduling and routing problem," *Sustainability*, vol. 9, no. 12, p. 2178, Nov. 2017.

[32] M. Setak, M. Habibi, H. Karimi, and M. Abedzadeh, "A time-dependent vehicle routing problem in multigraph with FIFO property," *J. Manuf. Syst.*, vol. 35, pp. 37–45, Apr. 2015.

[33] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle dispatching with time-dependent travel times," *Eur. J. Oper. Res.*, vol. 144, no. 2, pp. 379–396, Jan. 2003.

[34] A. C. A. Cauvin, A. F. A. Ferrarini, and E. T. E. Tranvouez, "Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours," *Int. J. Prod. Econ.*, vol. 122, no. 1, pp. 429–439, Nov. 2009.

[35] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, "Tabu search for the time-dependent vehicle routing problem with time windows on a road network," *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 129–140, Jan. 2021.

[36] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987.

[37] H. Gehring and J. Homberger, "A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows," in *Proc. EUROGE*, vol. 2, K. Miettinen, M. Makela, and J. Toivanen, Eds. Berlin, Germen: Springer, 1999, pp. 57–64.

[38] Q. Ding, X. Hu, and Y. Jiang, "A model of disruption management based on prospect theory in logistic distribution," *J. Manage. Sci. China*, vol. 17, no. 11, pp. 1–9, 2014.

[39] J. Eaton, S. Yang, and M. Gongora, "Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2980–2992, Nov. 2017.

[40] J. Tang, Y. Ma, J. Guan, and C. Yan, "A max–min ant system for the split delivery weighted vehicle routing problem," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7468–7477, Dec. 2013.

YAO WU received the M.S. and Ph.D. degrees in management science and engineering from Southwest Jiaotong University, Sichuan, China, in 2011 and 2018, respectively. Since 2014, he has been a Lecturer with the Department of Industrial Engineering, Hubei University of Automotive Technology. Since 2018, he has also been holding postdoctoral position focused on management science and engineering at the University of Electronic Science and Technology of China. His research interests include logistics system optimization and supply chain operations.

BIN ZHENG received the B.S. degree in information and computing science and the Ph.D. degree in logistics engineering from Southwest Jiaotong University, Sichuan, China, in 2006 and 2013, respectively. Since 2013, she has been a Lecturer with the Department of Logistics Engineering, Southwest Jiaotong University. Her research interests include logistics system optimization and humanitarian logistics.

XUELIANG ZHOU was born in Shiyan, Hubei, China, in 1978. He received the B.S. degree in mechanical engineering from the Hubei University of Automotive Technology, Shiyan, in 2001, the M.S. degree in mechanical engineering from Wuhan University, Wuhan, China, in 2006, and the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2015.

From 2012 to 2019, he was an Associate Professor and since 2020, he has been a Professor with the School of Mechanical Engineering, Hubei University of Automotive Technology. He is the author of more than 20 articles. His research interests include CAD/CAM technology and quality control.

Dr. Zhou's awards and honors include the member of the council of Hubei Society of Electromechanical Engineering and the Hubei Province Scientific and Technological Progress Award.