6-2020

Goods consumed during transit in split delivery vehicle routing problems: Modeling and solution

Wenzhe YANG

Di WANG

Wei PANG

Ah-Hwee TAN

You ZHOU

Citation

YANG, Wenzhe; WANG, Di; PANG, Wei; TAN, Ah-Hwee; and ZHOU, You. Goods consumed during transit in split delivery vehicle routing problems: Modeling and solution. (2020). *IEEE Access*. 8, 110336-110350.
Available at: https://ink.library.smu.edu.sg/sis_research/5263

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 1, 2020, accepted June 4, 2020, date of publication June 11, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001590

Goods Consumed During Transit in Split Delivery Vehicle Routing Problems: Modeling and Solution

WENZHE YANG[1,2], DI WANG[3,4], (Senior Member, IEEE), WEI PANG[5,6], AH-HWEE TAN[3,7], (Senior Member, IEEE), AND YOU ZHOU[1,2]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore 639798
[4] Joint NTU-WeBank Research Centre on FinTech, Nanyang Technological University, Singapore 639798
[5] School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, U.K.
[6] School of Natural and Computing Sciences, University of Aberdeen, Aberdeen AB24 3UE, U.K.
[7] School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: You Zhou (zyou@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772227, Grant 61972174, and Grant 61972175, in part by the Science and Technology Development Foundation of Jilin Province under Grant 20180201045GX, Grant 20200201300JC, and Grant 20180414012GH, in part by the Jilin Development and Reform Commission Fund under Grant 2020C020-2, and in part by the Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore, under Award NWJ-2019-002.

ABSTRACT This article presents the modeling and solution of an extended type of split delivery vehicle routing problem (SDVRP). In SDVRP, the demands of customers need to be met by efficiently routing a given number of capacitated vehicles, wherein each customer may be served multiple times by more than one vehicle. Furthermore, in many real-world scenarios, consumption of vehicles en route is the same as the goods being delivered to customers, such as food, water and fuel in rescue or replenishment missions in harsh environments. Moreover, the consumption may also be in virtual forms, such as time spent in constrained tasks. We name such a real-world SDVRP as Split Delivery Vehicle Routing Problem with Goods Consumed during Transit (SDVRP-GCT). In this paper, we give mathematical formulas to model SDVRP-GCT and provide solutions by extending three ant colony algorithms. To the best of our knowledge, this is the first research work specifically focussing on the SDVRP-GCT problem and its solutions. To assess the effectiveness of our proposed ant colony algorithms, we first apply them on widely adopted SDVRP benchmarking instances on different scales and their correspondingly extended SDVRP-GCT instances. Then, we formulate a real-world SDVRP-GCT instance for further assessment. Based on the extensive experimental results, we discuss the pros and cons of our proposed solutions and subsequently suggest their preferable application scenarios. In summary, our proposed solutions are shown as highly efficient in solving SDVRP-GCT instances.

INDEX TERMS Split delivery vehicle routing problem (SDVRP), goods consumed during transit, optimization, ant colony algorithms.

I. INTRODUCTION

The Vehicle Routing Problem (VRP) has always been an active and important topic in logistics. In VRP, a set of routes for a fleet of vehicles starting from one or more depots must serve certain geographically scattered locations or customers, under a variety of constraints [1]. The main objective of VRP is to minimize the total travel distance of vehicles. Ever since 1959 [2], VRPs have been widely studied due to their practical relevance to real-world applications and considerable difficulty in objective optimizations. There have been a number of noteworthy exact and heuristic algorithms developed for VRPs in recent decades (e.g., models reviewed in [3]). In VRPs, each customer is served by a vehicle only once. However, in many practical situations, when the customer's demand is less than the capacity of the vehicle, the constraint that each customer is served by a vehicle only once may result in high empty loading rate and wasting of vehicle resources. When the customer's demand exceeds the capacity of the vehicle, the traditional VRP which only

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

provides one delivery service for each customer cannot meet the customer's demand. Therefore, Dror and Trudeau [4] first proposed split delivery VRP (SDVRP) in 1989, wherein each customer may be served more than once (by multiple vehicles or multiple times by the same vehicle). At first, it was thought that the routing costs would be increased by split deliveries, but this relaxation has resulted in a larger set of solutions and reduced costs [5]. The study of Archetti *et al.* [6] showed that reasonable split deliveries are able to not only effectively reduce the number of vehicles, but also save the total routing costs, which can reach up to 50%.

However, to the best of our knowledge, the scope of all prior SDVRP studies have not taken into account the situation that the goods ordered or required by the customers are consumed during transit. A typical example of such scenarios is rescue or replenishment in harsh environments, such as deserts and areas affected by snow or thunderstorms. In such scenarios, the long distance travel (including return) will consume a significant amount of critical resources such as water, food and fuel that may be impossible to get refills. It is worth noting that the route consumption and customer demand are not necessarily actual goods, but also may be in virtual forms. The ''goods'' is a general term that could be extended to represent time consumed or transportation cost, etc. Therefore, another scenario, which may be modeled as a type of traveler's problem, is depicted as follows: a traveler plans to travel through a finite set of tourism attractions within a limited time period, then not only the time to be spent at each tourism attraction should be included as part of the travel cost, but also the time to be spent on the road should be taken into consideration. These practical problems that have to consider the Goods Consumed during Transit (GCT) are regarded as an extension to SDVRP. In such circumstances, we have to ensure that each vehicle has enough resources to be consumed on its way back to the depot. Therefore, the optimization objective of GCT problems extends that of the conventional VRP or SDVRP by adding in the consideration of goods consumed during transit. We name this type of under-investigated problems as SDVRP with Goods Consumed during Transit (SDVRP-GCT).

In this research, we adopt and extend ant colony optimization (ACO) [7] algorithms to solve SDVRP-GCT as they have been shown as performing well in solving SDVRPs. ACO is a nature-inspired optimization method, which simulates the food-foraging behaviors of ant colonies when they explore the surroundings of the anthill and leave pheromone along the trace to guide others in food-finding. ACO has been successfully applied to various classical combinatorial optimization problems, e.g., telecommunication routing [8], travelling salesman problem (TSP) [9], product design [10], virtual machine deployment [11], bi-objective pathfinding [12], and vehicle routing problem [2]. The motivations of our work are mainly two folds: to investigate whether SDVRP-GCT can be efficiently solved by extending existing solutions to SDVRP, and to further identify the pros and cons of the proposed solutions.

Our main contributions in this paper are as follows:

1) We formally formulate SDVRP-GCT in the form of equations and inequalities.
2) We propose three ACO algorithms to solve SDVRPs, focusing on SDVRP-GCT, wherein vehicle en route consumption is the same as goods to be delivered to customers.
3) We apply the three proposed ACO algorithms on eighteen widely adopted SDVRP instances and their correspondingly extended SDVRP-GCT instances as well as an own collected real-world SDVRP-GCT instance for thorough assessments.

The rest of the paper is organized as follows. In Section II, we introduce relevant literature on VRP, SDVRP and ACO algorithms. In Section III, we model SDVRP-GCT by making relevant assumptions and presenting mathematical formulations. In Section IV, we introduce our proposed ACO algorithms with details. In Section V, we first apply the proposed ACO algorithms on eighteen benchmarking instances and a real-world SDVRP-GCT instance, and then present and discuss the experimental results. Finally, in Section VI, we conclude this paper.

II. RELATED WORK

Firstly, based on the chronological order, we review several classical VRP variants. Capacitated VRP (CVRP) [2] takes into account the fact that every vehicle has a limited capacity and no customer demand exceeds its capacity. Periodic VRP (PVRP) [13] studies the time-efficient routing problem that customers require different delivery time slots and aims to find a feasible solution that minimizes the total cost of the route in a certain time frame. The study of Dynamic VRP (DVRP) [14] involves dynamic operation, which is characterized by constantly changing information, such as vehicle location, customer orders, etc. Then the appropriate vehicle is assigned according to the customer's request. In VRP with Time Windows (VRPTW) [15], each customer has a delivery deadline and the earliest delivery time. With the constraint of the given time windows, the total cost not only includes the distance or time cost during the trip, but also includes the waiting time cost such as the vehicle arriving at the customer's location too early. Split-delivery VRP (SDVRP) [4] considers certain realistic cases where customer demand exceeds vehicle capacity and allows each customer to be served by more than one vehicle. Time-dependent VRP (TDVRP) [16] extends the classical static shortest path problem to further consider time variation between nodes. It assumes that the cost variability in relation to time and travel time between any pair of nodes depends on distance, road conditions and weather conditions, etc. Stochastic VRP (SVRP) [17] considers the occurrence of stochastic factors, such as stochastic customer, stochastic demand and stochastic delivery time. Multi-depot VRP (MDVRP) [18] has multiple depots and each customer is served by vehicles from one of them and each route must start and end at the same depot.

Open VRP (OVRP) [19] differs from the traditional vehicle routing problem in that the vehicles travel on an open route, rather than a closed route, and they do not have to return to the depot, or if required, they can visit the customer in the reverse order. In VRP with Simultaneous Pickups and Deliveries (VRPSPD) [20], vehicles deliver the goods to each customer and meet their pickup requirements at the same time. Multi-echelon VRP (MEVRP) [21] studies logistics transportation by multiple echelons, such that the delivery of goods from the depot to the customers is decomposed into from the depot to the intermediate depots and then from the intermediate depots to the customers. Green VRP (GVRP) [22] considers the environmental and ecological impacts when finding the shortest route based on minimizing the fuel consumption or reducing carbon emissions.

SDVRP plays a significant role in VRP research. Archetti *et al.* [6] showed that compared with VRP, SDVRP can significantly reduce travel distance and fleet size. Several new variants of the conventional SDVRP have been explored in the literature. For example, SDVRP with Time Windows (SDVRPTW) [23] takes into account that each customer can only be visited by one or more vehicles within a specified time interval. Split Delivery Weighted VRP (SDWVRP) [24] considers the relationship between the weight of goods and the distance cost, where ‘‘weight’’ can also represent the priority of customers. Multi-depot SDVRP (MDSDVRP) [25] determines the best delivery route and location of multiple depots when serving customer demands if the depots are not defined in advance. In Commodity-constrained SDVRP (C-SDVRP) [26], each customer has multiple commodity demands and when a vehicle delivers a commodity to a customer, the full amount requested by the customer must be provided. SDVRP with Minimum Delivery Amounts (SDVRP-MDA) [27] allows a customer's demand to be delivered by multiple vehicles, but each visit requires a minimum delivery quantity. SDVRP with 3D Loading Constraints [28] considers the 3D loading of cargoes by using the least number of 3D containers to load the goods to minimize the cost. SDVRP with uncertain travel time and demands [29] plans vehicle routes when delivering critical supplies to people in need after a disaster.

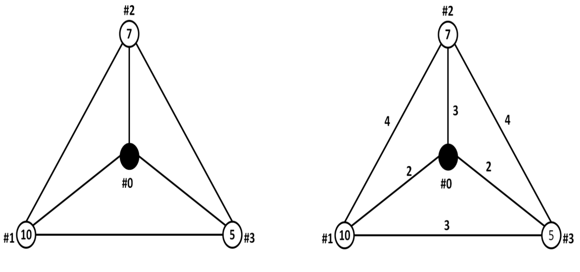
Since the formal formulation of SDVRP [4], many scholars have studied its difficulty, solution characteristics, upper and lower bounds of solution, solution algorithms, etc. Methods to solve SDVRP may be generally divided into two categories: exact algorithms and heuristic ones. The pioneer exact algorithm was proposed by Dror *et al.* [30] in 1994. They provided a mixed-integer programming formulation of SDVRP and developed an arc-column equation that combines new effective inequality classes. Jin *et al.* [31] proposed a two-stage algorithm with valid inequality constraints to solve SDVRP. Archetti *et al.* [32] proposed two exact branch-and-cut algorithms to solve SDVRP based on two relaxed formulations that provide lower bounds to the optimum. Although the exact algorithms may find the optimal solution to the problem, its computational time increases exponentially with the expansion of problem size. Therefore, exact algorithms may only be suitable for solving small-scale instances. To efficiently deal with large-scale instances, heuristic algorithms are often applied to obtain ‘‘sub-optimal solutions’’ or ‘‘satisfactory solutions’’. At present, the most commonly adopted heuristic algorithms are Simulated Annealing (SA) [33], Genetic Algorithm (GA) [34], Tabu Search (TS) [35], Ant colony optimization (ACO) [7], Artificial Neural Network (ANN) [36], Particle Swarm Optimization (PSO) [37], etc.

Ant colony optimization algorithm (ACO) is a kind of heuristic algorithm. It summarizes AS (Ant System) [7], ACS (Ant Colony System) [9], [38], MMAS (Max-Min Ant System) [39] and other ant algorithms under a unified framework. Although they are collectively known as ACO, they are different in the implementation process and have their own characteristics. Because we extended the existing ACO algorithms to solve SDVRP-GCT instances, we follow their original names in this paper. The pioneer AS was proposed by Dorigo *et al.* [7], as an efficient solution to combinatorial optimization problems. In order to search for better results in a relatively larger solution space, a modified ACS was proposed by Gambardella and Dorigo [9], [38]. ACS uses both global and local update rules to adjust pheromone concentration, which makes the solution of ACS more generic. Moreover, ACS applies the pseudorandom-proportional transition rule, which is conducive to transitions towards nodes connected by short edges and with a large amount of pheromone. To better deal with the stagnation phenomena often observed in AS, MMAS [39] was proposed by Stutzle and Hoos, by incorporating three effective mechanisms, namely to update the pheromone of the best solution path in each iteration, to limit the amount of the pheromone, and to initialize all pheromones to large values as a conducive way to explore the large solution space. These mechanisms try to prevent MMAS from premature convergence to non-global optimal solutions and better utilize pheromones. AS, ACS and MMAS were all initially designed to solve traveling salesman problems (TSPs) [40]. TSP can be easily described as the problem of finding the shortest path to visit each node once and only once. Up till now, these three algorithms have been applied to many other combinatorial optimization problems such as vehicle routing problems and quadratic assignment problems. AS can quickly find suitable solutions, and has been successfully applied to solve many problems such as finding the optimal path on a graph. ACS is more generic and has higher efficiency when compared to other algorithms in solving symmetric and asymmetric TSPs. MMAS is one of the best ACO algorithms in solving TSPs because pheromones and updating rules are carefully regulated.

Due to the novelty of the SDVRP-GCT instances, wherein the consumption during transit is considered, existing solutions to SDVRP that do not consider such route consumption cannot be straightforwardly applied. New algorithms are in need to solve SDVRP-GCT instances. Considering the effectiveness of the aforementioned ACO algorithms, we propose three algorithms by extending the existing ACO algorithms, i.e., AS, ACS and MMAS. Moreover, the extended algorithms are more generic than their original versions and can be used to solve both SDVRP and SDVRP-GCT instances. Last but not least, our extended algorithms have their own pros and cons on different instances, and there is no single algorithm outperforms the others on all instances (see Section V). In this research, we assess their capability by conducting extensive experiments, and further suggest their preferable application scenarios.

III. MODELING SDVRP-GCT

In this section, we formulate SDVRP-GCT mathematically. We first provide an intuitive example of SDVRP-GCT. We then present the assumptions of SDVRP-GCT. Finally, we present its mathematical formulations.



(a) (b)

FIGURE 1. Differences between SDVRP and SDVRP-GCT. (a) Standard SDVRP. (b) SDVRP-GCT, wherein goods consumption along the route is labeled accordingly.

SDVRP-GCT inherits the components involved in SDVRP, namely a fleet of vehicles, one depot node, a specific number of nodes of customers known *a prior*, and a network connecting all the nodes including the depot, wherein each edge of the network represents the route between two nodes. Figure 1a shows a simple example of a standard SDVRP with four nodes (inclusive of the depot node with index #0). Each customer (#1, #2 and #3) needs one or more vehicles to deliver the ordered goods. Moreover, every vehicle must start from and return to the only depot. The numbers labelled within the customer nodes indicate the customer's demand. If all available vehicles have the same capacity of 15, for SDVRP, one vehicle can first serve Node #1 and then serve Node #3 because the sum of demand of these two nodes is exactly 15. However, for SDVRP-GCT, as shown in Figure 1b, the numbers labelled along each edge denote the vehicle consumption associated with the respective routes (e.g., the vehicle consumption and the customer ordered goods are the same type of fuel). Therefore, a vehicle cannot serve both Nodes #1 and #3 in a single delivery trip, because the total amount of goods required is $2+10+3+5+2 = 22$, which exceeds the vehicle's capacity.

A. ASSUMPTIONS IN SDVRP-GCT

Before we present the mathematical formulations of SDVRP-GCT, we first introduce the assumptions made in SDVRP-GCT as follows:

1) The properties of all entities, namely the involved vehicles, the location of the depot, the customers' location and demand, and the consumption associated with each route, will remain constant during the entire servicing period.
2) There is only one edge between any two nodes and the cost of travel is the same for either direction.
3) The costs only consist of the customer demand and the route consumption. All the other costs, such as vehicle's maintenance, the staff's salaries, the service time at customers' locations, etc., are not considered.
4) The route consumption during transit is proportional to the distance of the route.
5) Any customer ordered goods or transportation demand is served directly from the depot. Furthermore, all involved customers do not interact with one another.
6) Vehicles can reload or replenish only at the depot when being assigned for delivery.
7) All vehicles have the same capacity and the customers do not have preferences of delivery vehicles.

B. FORMULATION OF SDVRP-GCT

TABLE 1. List of parameters and variable used in SDVRP-GCT.

Symbol	Definition
N	Number of customers
M	Number of vehicles
q_i	Demand of customer i
r_{ij}	Route consumption between nodes i and j
d_{ij}	Distance between nodes i and j
x_{ijk}	Binary indicator of whether edge (i, j) has been traversed by vehicle k
y_{ik}	Amount of customer i's demand served by vehicle k
Q	Capacity of each vehicle
Q_c	Customer demand capacity of each vehicle
Q_r	Route consumption capacity of each vehicle

The definitions of all parameters and variables used in SDVRP-GCT are given in Table 1. We assume that a network consists of N customers (nodes), M vehicles and one depot, which is represented by Node #0. Each edge in the network representing the connection between two nodes is associated with certain amount of route consumption. Each customer i with demand q_i must be satisfied by one or more vehicles.

In comparison with SDVRP, when modeling SDVRP-GCT, an important factor is that the consumption en route has to be considered. In each delivery, each vehicle's capacity is divided into customer demand capacity and route consumption capacity. The capacity of each vehicle must be greater than the total route consumption from the depot to any customer and back to depot. For each routine of a vehicle (starting from the depot → serving a number of customers → finally returning to the depot), the sum of all customer demands served or partially served by the vehicle must be equal to or less than the customer demand capacity Q_c, and the total route consumption must be equal to or less than the route consumption capacity Q_r. Moreover, the sum of route consumption and customer demand must be equal to or less than the vehicle capacity, i.e., $Q_c + Q_r \leq Q$.

The solutions of an SDVRP-GCT can be obtained by optimizing the following objective function:

$$Z = \min \sum_{i=0}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{M} x_{ijk} \cdot d_{ij}, \tag{1}$$

subject to the following constraints:

$$\sum_{i=0}^{N} \sum_{k=1}^{M} x_{ijk} \geq 1, \quad j \in \{0, 1, \ldots, N\}, \tag{2}$$

$$\sum_{j=0}^{N} \sum_{k=1}^{M} x_{0jk} \leq M, \tag{3}$$

$$\sum_{j=1}^{N} x_{0jk} = \sum_{j=1}^{N} x_{j0k} \in \{0, 1\}, \quad k \in \{1, \ldots, M\}, \tag{4}$$

$$\sum_{i=0}^{N} x_{ipk} - \sum_{j=0}^{N} x_{pjk} = 0, \; p \in \{0, \ldots, N\}, \quad k \in \{1, \ldots, M\}, \tag{5}$$

$$\sum_{k=1}^{M} y_{ik} = q_i, \quad i \in \{0, 1, \ldots, N\}, \tag{6}$$

$$0 \leq y_{ik} \leq q_i, \; i \in \{0, 1, \ldots, N\}, \quad k \in \{1, \ldots, M\}, \tag{7}$$

$$\sum_{i=1}^{N} y_{ik} + \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} r_{ij} \leq Q, \quad k \in \{1, \ldots, M\}, \tag{8}$$

$$Q - (r_{0i} + r_{i0}) \geq 0, \quad i \in \{0, 1, \ldots, N\}. \tag{9}$$

The objective function defined in (1) tries to minimize the total travel distance of all vehicle routes. Since route consumption is proportional to route distance, the objective function is equivalent to minimize the total route consumption. Equation (2) guarantees that each node is visited at least once. Equation (3) indicates that a maximum number of M vehicles are allowed to departure from the depot at the same time. Equation (4) ensures that each vehicle leaving the depot must return to the depot. Equation (III-B) ensures that the number of vehicles arriving at a node equal to the number of vehicles of leaving the same node. Equations (6) and (7) ensure that the demand of all nodes is satisfied.

In SDVRP-GCT, we divide the capacity of vehicles Q into customer demand capacity Q_c and route consumption capacity Q_r, which correspond to the demand requirement at each node and consumption en route, respectively. Furthermore, we consider two specific scenarios in SDVRP-GCT based on whether Q_r is considered when formulating the problem. The two scenarios are described as follows:

(i) The route consumption capacity Q_r is not considered. In this scenario, the only constraint on vehicles is customer demand capacity Q_c, whereby the problem turns into a standard SDVRP that (9) should be removed and (8) should be changed to the following:

$$\sum_{i=1}^{N} y_{ik} \leq Q_c, \quad k \in \{1, \ldots, M\}. \tag{10}$$

(ii) The route consumption capacity Q_r is considered. This scenario is the major focus of this paper, i.e., SDVRP-GCT, which has been described in the earlier sections. In SDVRP-GCT, all vehicles need to consider both Q_r and Q_c as formulated in (8) and (9).

In brief, Scenario (i) can be regarded as standard SDVRP, which does not consider route consumption. Scenario (ii) considers route consumption as the same type of resource as customers demand. We focus on Scenario (ii) in this paper, because it can be used to solve an under-investigated type of practical problem, such as rescue in harsh environments wherein a significant amount of critical resources for the trapped people will be consumed en route.

It is well known that the classical SDVRP is an NP-hard problem [41], [42]. SDVRP-GCT is more complicated than SDVRP because the route consumption during transit is taken into consideration as well. Therefore, SDVRP-GCT is also an NP-hard problem. To effectively solve the NP-hard problem, three extended ACO algorithms are proposed in this paper.

IV. EXTENDING ANT COLONY OPTIMIZATION (ACO) ALGORITHMS TO SOLVE SDVRP-GCT

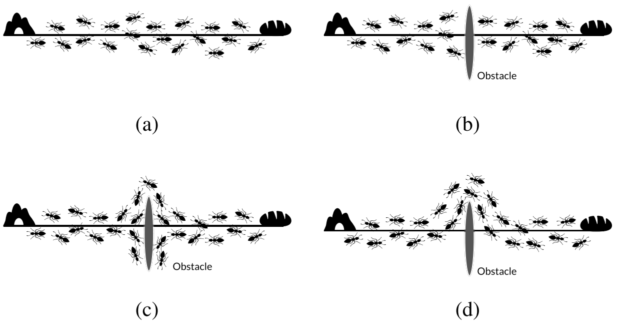


FIGURE 2. Illustrations of the nature-inspired ant colony optimization. (a) Ants in a straight path. (b) An obstacle is added in a straight path. (c) Ants randomly selects alternative paths. (d) Ants aggregate to the shorter path [43].

The solutions of SDVRP-GCT can be obtained by finding a set of minimum cost vehicle routes to facilitate the delivery plans from the depot to a number of customer locations. The behaviors of the vehicles are deemed similar to the food-seeking behavior of ant colonies (see Figure 2). Regarding the depot in SDVRP-GCT as the nest in ACO and customers in SDVRP-GCT as food in ACO, we can solve SDVRP-GCT by applying three extended ACO algorithms, which we name Ant System for SDVRP-GCT (ASGCT), Ant Colony System for SDVRP-GCT (ACGCT) and Max-Min Ant System for SDVRP-GCT (MMGCT), respectively. These three algorithms are extended based on the respective classical ACO algorithms to solve the SDVRP-GCT instances.

When a vehicle plans to serve the next customer, if the sum of next customer's demand and route consumption (from the current customer → the next customer → returning to depot) exceeds the remaining vehicle capacity, the vehicle serves part of the next customer's demand and the remaining part will be served by other vehicles. We will describe the detailed processes of these three extended ACO algorithms in the following subsections.

A. ANT SYSTEM FOR SDVRP-GCT (ASGCT)

Conforming to the generality of ACO-based algorithms, we use the term "vehicle" interchangeably with "ant". In Ant System for SDVRP-GCT (ASGCT), every ant always starts from the depot, then it chooses the next node according to the amount of pheromone in all available paths based on the Ant Movement Rule (see Section IV-A1). Once all ants have explored their routes, the Global Pheromone Matrix Updating Rule (see Section IV-A2) is invoked.

1) ANT MOVEMENT RULE

Suppose there are M ants in an ASGCT. If Ant k is currently located at Node i, the next Node j to be visited is chosen based on the probability defined as follows:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{u\in \mathrm{allowed_k}}[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}}, & j \in \mathrm{allowed_k}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $\mathrm{allowed_k}$ denotes the set of all feasible unvisited nodes that can be visited by Ant k in this iteration such that Ant k's remaining load is enough to go to that node and return to depot afterwards, τ_{ij} denotes the amount of pheromone in the path connecting Nodes i and j, $\eta_{ij} = 1/d_{ij}$ denotes the heuristic information of objective, α denotes the intensity parameter of residual pheromone, and β denotes the intensity parameter of heuristic information.

2) GLOBAL PHEROMONE MATRIX UPDATING RULE

The pheromones gradually decay over time. When ants complete a cycle, the pheromone matrix τ_{ij} is updated as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1), \tag{12}$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^{M} \Delta\tau_{ij}^{k}(t, t+1), \tag{13}$$

$$\Delta\tau_{ij}^{k}(t, t+1) = \begin{cases} z/C_k, & i, j \in R_k, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where ρ denotes the pheromone evaporation rate, which enhances the global search capability of the ant system and t denotes the index of the timestamp (iteration number). Term $\Delta\tau_{ij}^{k}(t, t+1)$ denotes the amount of pheromone left by Ant k along the route from Node i to Node j in this iteration. Moreover, $\Delta\tau_{ij}(t, t+1)$ denotes the increment of pheromone on the road between Node i and Node j in this iteration, z denotes a constant, which affects the convergence speed of the algorithm, C_k denotes the total cost of Ant k traversed in this iteration, and R_k denotes the routes that Ant k has traversed.

Algorithm 1 ASGCT

```
 1: Initialize parameters $\alpha, \beta, \rho, M, z, Q, maxiter$;
 2: Initialize $\tau_{ij}(0) = \tau_0$ for all node pairs;
 3: Set iterate index $t = 0$, current position $now = 0$;
 4: for $t = 1$ to $maxiter$ do
 5:   for each ant $k = 1$ to $M$ do
 6:     Set $demand$ = demand of nodes;
 7:     Set capacity $v_k = Q$;
 8:     while $\sum_{j=1}^{N} demand_j \neq 0$ do
 9:       Put nodes required constraints that $demand_j > 0$
          and $v_k - (c_{now,j} + c_{j,0}) > 0$ into $allowed_k$;
10:       if $allowed_k$ is not empty then
11:         Compute transition probability according
            to (11);
12:         Select the subsequent node $j$ according to
            roulette wheel selection [44], then add $j$ into
            the path;
13:         $v_k = v_k - c_{now,j}$, $now = j$;
14:         if $v_k - demand_j \geq c_{j,0}$ then
15:           $demand_j = 0$; $v_k = v_k - demand_j$;
16:         else
17:           $demand_j = demand_j - (v_k - c_{j,0})$;
18:           $v_k = c_{j,0}$;
19:         end if
20:       else
21:         Return to depot 0; Reload set $v_k = Q$;
22:       end if
23:     end while
24:     if ant $k$ has not returned to the depot then
25:       Add 0 to the path;
26:     end if
27:   end for
28:   Save the shortest paths of all $M$ ants in this iteration;
29:   Update pheromone according to (12), (13) and (14);
30:   $t = t + 1$;
31: end for
32: Output the shortest paths;
```

The overall procedure of using Ant System for SDVRP-GCT (ASGCT) is summarized in Algorithm 1. Specifically, to satisfy Constraint (8), Algorithm 1 checks whether a Node k is included in $\mathrm{allowed_k}$ (see Line 10). Moreover, Line 10 ensures that an ant can successfully return to the depot with enough supply. In addition, Algorithm 1 also automatically determines the maximum amount of demand required by the selected Node j can be met to ensure the ant can successfully return to the depot. Lines 14 to 18 consider two different serving behaviors: (i) if the vehicle can return to the depot after serving the full extent of demand required by the customer being served, it will serve the customer to the full extent, otherwise (ii) the vehicle reserves the necessary amount of supply and returns to the depot afterwards. In the latter case, the demand of the last visited customer is partially served by this vehicle. The best solution, which leads to the minimum overall traveled distance, is then obtained over iterations.

B. ANT COLONY SYSTEM FOR SDVRP-GCT (ACGCT)

In order to improve the effectiveness of ants in route selection, Ant Colony System (ACS) was first proposed in [38]. ACS is a well-known meta-heuristic algorithm in the ACO family. In ACS, a group of artificial ants collaboratively search for the optimal solutions with shared information.

Comparing to ASGCT, ACGCT introduces the rules of local update, global update, and a modified version of pseudo-random-proportional transition. In ACGCT, only the best solution from the beginning of the trial updates the pheromone trial according to the global update rule. Specifically, in ACGCT, the global pheromone matrix τ_{ij} is updated as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t, t+1), \tag{15}$$

$$\Delta\tau_{ij}(t, t+1) = 1/\mathrm{Cost}^{\mathrm{gb}}, \tag{16}$$

where the superscript gb denotes the best solution which found until iteration t (global-best solution) and $\mathrm{Cost}^{\mathrm{gb}}$ denotes the cost of the global-best solution.

When an ant chooses a Node j as the next node to visit, the pheromone of the current Node i to Node j is updated according to the following local update rule:

$$\tau_{ij}(t+1) = (1-\epsilon)\tau_{ij}(t) + \epsilon\tau_0, \tag{17}$$

where ϵ denotes the pheromone evaporation in local update process and τ_0 denotes the initial pheromone value.

Lastly, the pseudo-random-proportional transition rule is defines as follows:

$$j = \begin{cases} \arg\max_{j \in \mathrm{allowed_k}} \{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}\}, & \text{if } q \le q_0, \\ P_{ij}^{k}(i,j), & \text{otherwise,} \end{cases} \tag{18}$$

where q_0 is a predefined parameter ($q_0 \in [0, 1]$) and q is a random number uniformly distributed in the [0, 1] interval. The definitions of $\eta, \alpha, \beta, \rho, \Delta\tau_{ij}(t, t+1)$ and $P_{ij}^{k}(i, j)$ remain the same as those introduced in ASGCT.

The ACGCT algorithm for solving SDVRP-GCT is shown in Algorithm 2. It is worth noting that the local update rule is applied when each ant selects a new node (see Line 14), and the global update rule is performed on the global optimal path after all ants have travelled through all nodes (see Line 30).

Algorithm 2 ACGCT

```
1:  Initialize parameters $\alpha, \beta, \rho, \epsilon, M, Q, q_0, maxiter$;
2:  Initialize $\tau_{ij}(0) = \tau_0$ for all node pairs;
3:  Set iterate index $t = 0$, current position $now = 0$;
4:  for $t = 1$ to $maxiter$ do
5:    for each ant $k = 1$ to $M$ do
6:      Set $demand$ = demand of nodes;
7:      Set capacity $v_k = Q$;
8:      while $\sum_{j=1}^{N} demand_j \neq 0$ do
9:        Put nodes required constraints that $demand_j > 0$
          and $v_k - (c_{now,j} + c_{j,0}) > 0$ into $allowed_k$;
10:       if $allowed_k$ is not empty then
11:         Compute transition probability according
            to (18);
12:         Select the next node $j$ according to roulette
            wheel selection, then add $j$ into the path;
13:         $v_k = v_k - c_{now,j}$; $now = j$;
14:         local update pheromone according to (17);
15:         if $v_k - demand_j \geq c_{j,0}$ then
16:           $demand_j = 0$; $v_k = v_k - demand_j$;
17:         else
18:           $demand_j = demand_j - (v_k - c_{j,0})$;
19:           $v_k = c_{j,0}$;
20:         end if
21:       else
22:         Return to depot 0; Reload set $v_k = Q$;
23:       end if
24:     end while
25:     if ant $k$ has not returned to the depot then
26:       Add 0 to the path;
27:     end if
28:   end for
29:   Save the shortest paths of all $M$ ants in this iteration;
30:   Update global pheromone according to (15) and (16)
      for the global-best solution;
31:   $t = t + 1$;
32: end for
33: Output the shortest paths;
```

C. MAX-MIN ANT SYSTEM FOR SDVRP-GCT (MMGCT)

Min-Max Ant System (MMAS) [39] was proposed to better deal with the potential premature convergence problem often observed in AS. Several MMAS-based variations for VRP were proposed and have shown convincing performance [45]–[47]. The MMGCT algorithm extends the ASGCT algorithm, wherein the main improvements are delineated as follows. Firstly, in the setting of MMGCT, only the best solution in each iteration updates the pheromone trail according to the global update rule (see (19) and (20)). Secondly, the pheromone of all edges is initialized to a maximum value $\tau_{\max}$ to increase the probability of being searched in the initial phase. Thirdly, MMGCT restricts the value range of pheromone of all edges to be within the $[\tau_{\min}, \tau_{\max}]$ interval, i.e., $\tau_{\min} \le \tau_{ij}(t) \le \tau_{\max}, \forall i, j, t$.

In MMGCT, the pheromone trail update rule is given by:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1), \tag{19}$$

$$\Delta\tau_{ij}(t, t+1) = z/\mathrm{Cost}^{\mathrm{ib}}, \tag{20}$$

where the superscript ib denotes the best solution in the current iteration t (iteration-best solution) and $\mathrm{Cost}^{\mathrm{ib}}$ denotes the cost of the iteration-best solution. The definitions of

$\Delta\tau_{ij}(t, t+1)$ remain the same as introduced in ASGCT. The global-best solution is not used in MMGCT as it may limit the search space and be easier to get trapped into local optimal solutions.

MMGCT restricts the minimum and maximum pheromone trails for all pheromone trails $\tau_{ij}(t)$ according to the following rules:

$$\tau_{\max} = 1/(\rho \cdot \text{Cost}^{\text{ib}}), \tag{21}$$

$$\tau_{\min} = \tau_{\max} \cdot (1 - \sqrt[n]{P_{\text{best}}})/((N/2 - 1) \cdot \sqrt[n]{P_{\text{best}}}), \tag{22}$$

where P_{best} is a constant. If P_{best} is too small, it may lead to $\tau_{\min} > \tau_{\max}$. To better regulate the influence of the lower trail limits, P_{best} is set to 0.05 heuristically. The MMGCT algorithm for SDVRP-GCT is shown in Algorithm 3.

V. EXPERIMENTAL RESULTS

We conduct experiments on fourteen benchmarking instances obtained from SDVRPLIB,[1] four large-scale benchmarking instances named kelly[2] and an own collected real-world instance to assess the three extended algorithms. The benchmarking instances are of SDVRP type, which do not consider goods consumed during transit (see discussions on Scenario (i) in Section III-B). To extend the benchmarking instances to SDVRP-GCT type, which is the main focus in this research, we use a proportion parameter h to regulate the amount of consumption en route. It is reasonable to assume that the amount of resources to be consumed en route (i.e., consumption during transit) is proportional to the distance of the route (assuming travel cost is not affected by other factors).

TABLE 2. Details of fourteen benchmarking instances.

Name	MNV	Dimension	Q	D	Optimal_value
S51D1	3	51	160	$\lceil 0.01Q, 0.1Q \rfloor$	458
S51D2	9	51	160	$\lceil 0.1Q, 0.3Q \rfloor$	726
S51D3	15	51	160	$\lceil 0.1Q, 0.5Q \rfloor$	972
S51D4	27	51	160	$\lceil 0.1Q, 0.9Q \rfloor$	1677
S51D5	23	51	160	$\lceil 0.3Q, 0.7Q \rfloor$	1440
S51D6	41	51	160	$\lceil 0.7Q, 0.9Q \rfloor$	2327
S76D1	4	76	160	$\lceil 0.01Q, 0.1Q \rfloor$	594
S76D2	15	76	160	$\lceil 0.1Q, 0.3Q \rfloor$	1147
S76D3	23	76	160	$\lceil 0.1Q, 0.5Q \rfloor$	1474
S76D4	37	76	160	$\lceil 0.1Q, 0.9Q \rfloor$	2257
S101D1	5	101	160	$\lceil 0.01Q, 0.1Q \rfloor$	716
S101D2	20	101	160	$\lceil 0.1Q, 0.3Q \rfloor$	1393
S101D3	31	101	160	$\lceil 0.1Q, 0.5Q \rfloor$	1975
S101D5	48	101	160	$\lceil 0.3Q, 0.7Q \rfloor$	2915

* “Name” denotes the instances’ name; “MNV” denotes the minimum number of vehicles used in each instance; “Dimension” denotes the number of nodes (customers); “Q” denotes the capacity of vehicles; “D” denotes the scope of node (customer) demand; “Optimal_value” denotes the value of the optimal solution suggested by the dataset contributors in [49].

The properties of the fourteen selected SDVRPLIB benchmarking instances are shown in Table 2. Their names follow the same convention that the number appended to S refers to the number of customers and the number appended to D refers to the level of customer demand (the higher the number is, the more amount of demands the customers require). For example, the S51D1 instance’s name suggests that there are 51 customers in this instance and their amount of demands is the lowest among all instances involving 51 customers. As shown in Table 2, the customer demands in S51D1 range from 1% to 10% of the vehicle’s capacity. The kelly datasets comprise 20 large-scale

Algorithm 3 MMGCT

```
Initialize parameters $\alpha, \beta, \rho, M, z, Q, q_0, maxiter$;
Initialize $\tau_{ij}(0) = \tau_{max}$ for all node pairs;
Set iterate index $t = 0$, current position $now = 0$;
for $t = 1$ to $maxiter$ do
    for each ant $k = 1$ to $M$ do
        Set $demand$ = demand of nodes;
        Set capacity $v_k = Q$;
        while $\sum_{j=1}^{N} demand_j \neq 0$ do
            Put nodes required constraints that $demand_j > 0$
            and $v_k - (c_{now,j} + c_{j,0}) > 0$ into $allowed_k$;
            if $allowed_k$ is not empty then
                Compute transition probability according
                to (18);
                Select the next node $j$ according to roulette
                wheel selection, then add $j$ into the path;
                $v_k = v_k - c_{now,j}$; $now = j$;
                if $v_k - demand_j \geq c_{j,0}$ then
                    $demand_j = 0$; $v_k = v_k - demand_j$;
                else
                    $demand_j = demand_j - (v_k - c_{j,0})$;
                    $v_k = c_{j,0}$;
                end if
            else
                Return to depot 0; Reload set $v_k = Q$;
            end if
        end while
        if ant $k$ has not returned to the depot then
            Add 0 to the path;
        end if
    end for
    Save the shortest paths of all $M$ ants in this iteration;
    Update pheromone according to (19) and (20) for the
    iteration-best solution;
    Compute $\tau_{max}$ and $\tau_{min}$ according to (21) and (22);
    if $\tau_{ij}(t) < \tau_{min}(t)$ then
        $\tau_{ij}(t) = \tau_{min}(t)$;
    else if $\tau_{ij}(t) > \tau_{max}(t)$ then
        $\tau_{ij}(t) = \tau_{max}(t)$;
    end if
    $t = t + 1$;
end for
Output the shortest paths;
```

[1] URL: http://www.uv.es/belengue/sdvrp.html

[2] URL: http://www.bernabe.dorronsoro.es/vrp/

TABLE 3. Experimental results of three extended ACO algorithms on fourteen benchmarking SDVRP instances.

Dataset	Optimal_ value	ASGCT			ACGCT			MMGCT		
		Len	Avg (std)	Gap (%)	Len	Avg (std)	Gap (%)	Len	Avg (std)	Gap (%)
S51D1	458	478.44	484.98 (3.72)	4.53	476.30	484.63 (6.86)	3.99	466.25	476.84 (5.99)	1.80
S51D2	726	775.40	785.75 (4.42)	6.80	753.12	768.75 (6.39)	3.74	**724.07**	741.93 (8.88)	**-0.30**
S51D3	972	1039.49	1055.46 (7.93)	6.94	1031.07	1041.03 (3.79)	6.08	996.81	1019.16 (12.51)	2.55
S51D4	1677	1687.84	1697.40 (5.05)	0.65	**1667.97**	1681.02 (5.32)	**-0.54**	**1620.41**	1640.93 (10.42)	**-3.37**
S51D5	1440	**1428.66**	1456.13 (8.09)	**-0.79**	**1423.40**	1452.42 (11.53)	**-1.15**	**1380.21**	1402.86 (10.06)	**-4.15**
S51D6	2327	2395.19	2411.54 (8.14)	2.93	2387.82	2410.10 (8.11)	2.61	2348.00	2359.17 (6.52)	0.90
S76D1	594	649.42	656.43 (4.39)	9.33	623.62	638.40 (5.44)	4.99	612.70	622.98 (6.19)	3.15
S76D2	1147	1199.38	1205.93 (3.11)	4.57	**1143.87**	1173.25 (12.11)	**-0.27**	**1130.36**	1151.30 (8.98)	**-1.45**
S76D3	1474	1594.18	1614.96 (7.35)	8.15	1574.8	1597.18 (9.90)	6.84	1532.84	1551.91 (13.70)	3.99
S76D4	2257	2304.28	2334.19 (11.70)	2.10	2270.84	2280.77 (7.19)	0.61	**2232.96**	2259.44 (23.69)	**-1.07**
S101D1	716	777.25	787.52 (4.93)	8.55	756.71	776.48 (9.30)	5.69	744.46	762.31 (7.27)	3.98
S101D2	1393	1524.47	1539.41 (5.76)	9.44	1515.06	1523.71 (6.34)	8.76	1467.68	1491.00 (11.72)	5.36
S101D3	1975	2020.34	2052.15 (11.44)	2.30	2008.19	2035.46 (11.15)	1.68	1997.26	2030.69 (15.59)	1.13
S101D5	2915	3010.29	3029.02 (8.15)	3.27	2967.30	2984.10 (9.40)	1.79	2942.52	2962.11 (16.51)	0.94
Avg. gap (%)	-	-	-	4.91	-	-	3.20	-	-	0.96

* "Len" denotes the shortest route length of 20 trials obtained by the respective algorithm; "Avg" denotes the average route length of 20 trials obtained by corresponding extended ACO algorithms; "std" denotes the standard deviation of the corresponding results. "Avg. gap" denotes the average gap among all instances.

instances. We choose the first four large-scale single-depot instances for experiments in this paper. The real-world instance is taken from Question F of the 2015 National Mathematical Modeling Competition held in China. Details of the kelly datasets and the real-world instance are given in Section V-C and Section V-D, respectively. All our proposed algorithms were implemented using Matlab 2016a and all experiments were conducted on the same personal computer equipped with an Intel I5-4460 @ 3.2 GHz CPU and 16GB RAM.

A. EXTENDED ACO ALGORITHMS FOR SDVRP

We first directly assess the effectiveness of the three extended ACO algorithms on the fourteen benchmarking SDVRP instances adopted from SDVRPLIB. Specifically, for this set of SDVRP instances, we set $h = 0$ (i.e., no goods consumed during transit) and compare the results with the optimal values [48] (reported by the dataset contributors, see Table 3). We followed the specified settings in the dataset, namely the depot in each instance is set to Node #1 and the capacity of vehicles Q is set to 160. The average and best of experimental results are shown in Table 3.

As shown in Table 3, although most route lengths obtained by the three extended algorithms are larger than the respective optimal values, ACGCT obtains the best results on S51D4, S51D5 and MMGCT obtains the best results on S51D2, S51D4, S51D5, S76D2 and S76D4. To further quantify the performance difference between our proposed algorithms and the optimal values, we devise the following equation to measure the relative difference:

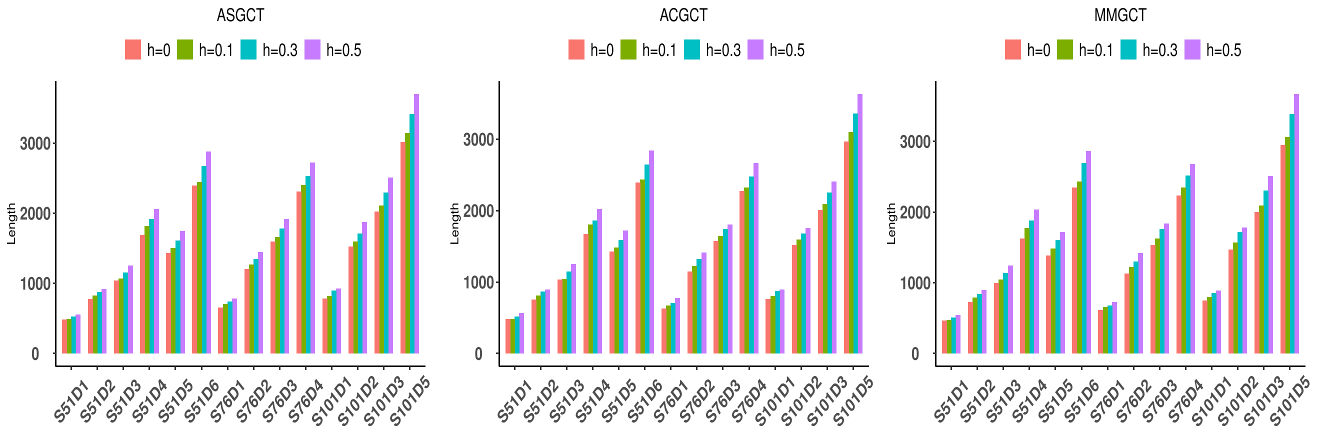
$$\text{gap} = \frac{\text{Len} - \text{Optimal\_value}}{\text{Optimal\_value}}. \tag{23}$$

The average gaps of ASGCT, ACGCT and MMGCT are computed as 4.91%, 3.24% and 0.96%, respectively. Although our proposed three algorithms do not all reach or surpass the optimal values, the experimental results obtained by our proposed algorithms are close to the optimal values. In particular, MMGCT obtains better results for five out of fourteen instances and the average gap of MMGCT is as small as 0.96%. The relatively small gap suggests that the extended algorithms are capable of finding competitive solutions or only a bit inferior to the optimal values, while being able to solve the challenging and practical SDVRP-GCT instances. In the following subsection, we further assess their performance on SDVRP-GCT instances.

B. EXTENDED ACO ALGORITHMS FOR SDVRP-GCT

In order to transform the fourteen SDVRPLIB benchmarking SDVRP instances (see Section V-A) to SDVRP-GCT ones, we first investigate how the parameter value of h affects the results of the three extended ACO algorithms. Specifically, we set $h = 0$, 0.1, 0.3 and 0.5 in each set of experiments and show the best results obtained from 20 respective trials in Figure 3. Note that for $h = 0$, we adopt the results reported in Section V-A. As expected, with the increasing value of h, which means the vehicles consume more goods during transit, longer total travel costs are required.

To further assess the performance of the three extended ACO algorithms, we fix h as 0.1. Table 4 shows the results of the three algorithms being applied to solve the fourteen benchmarking instances after they have been transformed to SDVRP-GCT ones. The numbers highlighted in bold show the shortest route length obtained from 20 independent runs using the corresponding algorithms for each instance. We first analyze in terms of the shortest route length. It is clearly



(a) (b) (c)

FIGURE 3. Performance on the shortest route length among of 20 trials based on different values of *h*.

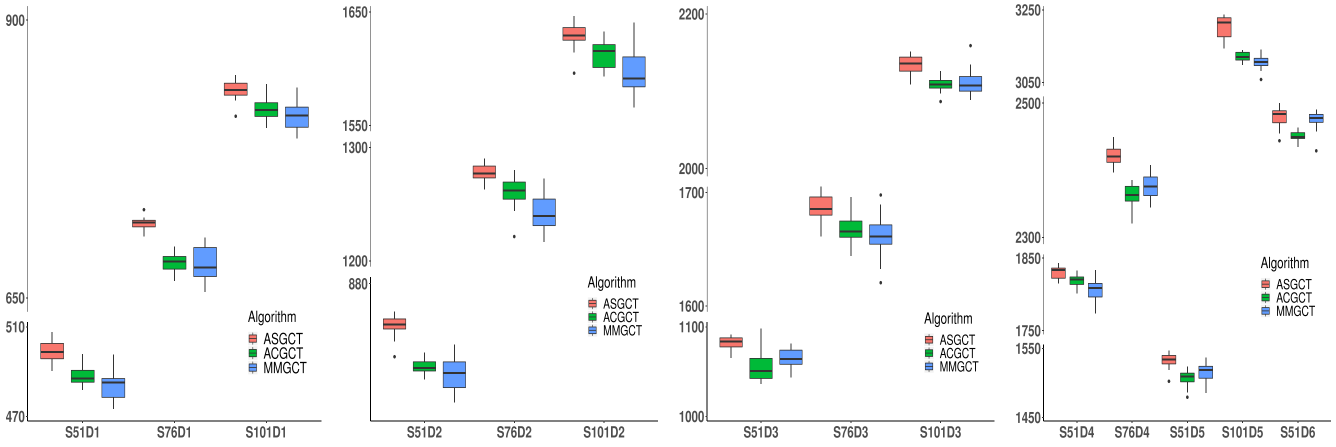
TABLE 4. Experimental results of the three extended ACO algorithms on fourteen transformed SDVRP-GCT instances.

Dataset	ASGCT				ACGCT				MMGCT			
	NV	Len	T (s)	Avg (std)	NV	Len	T (s)	Avg (std)	NV	Len	T (s)	Avg (std)
S51D1	3	490.21	14.28	499.16 (5.20)	3	481.85	17.49	487.98 (4.61)	3	**473.34**	17.76	483.90 (6.46)
S51D2	10	824.08	16.82	847.30 (8.39)	10	806.78	20.27	816.91 (6.33)	10	**789.19**	19.96	811.26 (12.52)
S51D3	15	1065.51	18.37	1082.15 (6.98)	15	**1036.60**	21.99	1056.46 (17.97)	15	1043.19	23.53	1064.91 (11.17)
S51D4	29	1815.09	23.90	1830.01 (8.92)	29	1801.44	29.40	1819.24 (7.67)	29	**1773.70**	29.02	1805.31 (15.31)
S51D5	24	1502.25	23.10	1532.60 (10.21)	24	**1479.22**	27.19	1506.42 (11.26)	24	1485.07	27.00	1515.06 (13.23)
S51D6	42	2443.97	33.97	2479.82 (14.47)	42	2434.71	34.93	2450.68 (7.92)	42	**2428.95**	34.83	2474.72 (13.99)
S76D1	5	705.44	34.33	716.60 (5.77)	5	665.23	36.01	681.37 (8.79)	5	**655.56**	37.42	680.21 (16.07)
S76D2	16	1263.20	39.44	1278.18 (7.89)	16	1221.40	40.63	1260.07 (13.66)	16	**1216.70**	43.15	1243.99 (17.10)
S76D3	24	1661.41	43.41	1685.88 (12.53)	24	1643.98	46.23	1668.23 (11.63)	24	**1620.52**	47.30	1662.03 (17.92)
S76D4	38	2396.63	51.09	2421.87 (12.43)	38	**2321.08**	53.96	2361.50 (16.82)	38	2344.62	52.42	2377.14 (18.50)
S101D1	6	813.45	60.74	837.38 (8.66)	6	803.16	60.74	820.23 (9.38)	6	**793.60**	61.04	813.49 (12.41)
S101D2	21	1596.01	70.96	1629.53 (11.26)	21	1593.21	75.56	1613.29 (11.81)	21	**1565.73**	70.03	1598.78 (21.42)
S101D3	32	2109.01	74.91	2134.06 (11.93)	32	**2086.62**	78.78	2108.90 (9.85)	32	2088.62	76.88	2111.64 (17.35)
S101D5	50	3142.76	90.96	3204.34 (30.51)	50	3098.82	92.55	3121.55 (11.31)	50	**3057.73**	88.72	3105.59 (19.53)

* "NV" denotes the number of vehicles obtained in the best solution of 20 trials; "T" denotes the algorithm running time in the best solution of 20 trials;

shown in Table 4 that the results of ACGCT and MMGCT are always better than those of ASGCT while ASGCT uses lesser amount of time in most cases. Furthermore, we may observe that MMGCT performs better than ACGCT on S51D1, S51D2, S76D1, S76D2, S101D1 and S101D2 with lower demand. However, ACGCT performs better than MMGCT on cases with higher demand, such as S51D3, S51D5, S76D4, S101D3. The averaged route length follows a similar pattern as observed in the shortest route length.

To further investigate the difference in performance among the three algorithms, we provide relevant visualizations in the form of boxplot. The boxplots show the distribution of experimental results on each instance. Figures 4a and 4b show the distribution of route length obtained by the three extended ACO algorithms on instances with lower demand, while Figures 4c and 4d show the distribution of those with higher demand. Boxplot is useful in determining where the majority of the experimental results lie. As shown in Figure 4, the route length obtained by ASGCT is obviously larger than that obtained by ACGCT and MMGCT, indicating that ACGCT and MMGCT always perform better than ASGCT in all the fourteen instances. In order to further analyze the difference in performance, we use single factor ANOVA to obtain the corresponding *p*-value, as shown in Table 5. The observations from both Figure 4 and Table 5 are consistent that for instances S76D1, S101D1, S51D2, S51D3, S76D3 and S101D3, the performance of ACGCT and MMGCT has no significant difference. For other instances with lower demand namely S51D1, S76D2 and S101D2, the location of MMGCT's boxplots are obviously lower than those of ACGCT. Moreover, the *p*-values between ACGCT and MMGCT are less than 0.05, i.e., MMGCT performs significantly better than ACGCT. Among the five remaining instances with higher demand, ACGCT performs



(a) (b) (c) (d)

FIGURE 4. The boxplots of experimental results on fourteen transformed SDVRP-GCT instances. Note that for better visualization, the y-axis of each subfigure takes different value range.

TABLE 5. The *p*-value of ACGCT vs MMGCT on fourteen transformed SDVRP-GCT instances.

Dataset	S51D1	S51D2	S51D3	S51D4	S51D5	S51D6	S76D1	S76D2	S76D3	S76D4	S101D1	S101D2	S101D3	S101D5
p-value	**<0.05**	0.08	0.08	**<0.05**	**<0.05**	**<0.05**	0.78	**<0.05**	0.20	**<0.05**	0.06	**<0.05**	0.54	**<0.05**

significantly better than MMGCT on three of them, namely S51D5, S51D6 and S76D4, because the boxplots formed by ACGCT are obviously downwardly skewed than those of MMGCT in these three instances with the *p*-values less than 0.05. Thus, ACGCT is shown to perform better than MMGCT on instances with higher demand.

In summary, we may draw a conclusion that ACGCT and MMGCT outperform ASGCT but ASGCT uses lesser amount of computational time. MMGCT is shown to be more suitable to solve instances with lower demand and ACGCT is shown to be slightly suitable than MMGCT for instances with higher demand.

C. EXTENDED ACO ALGORITHMS FOR LARGE-SCALE INSTANCES

In this subsection, we further assess the extended algorithms' capability of solving large-scale SDVRP-GCT instances. Specifically, we select four large-scale instances from the VRP web. Out of the 20 instances given in this kelly datasets, we select the first four instances as they involve single depot, which is consistent with our problem statement. The characteristics of the selected instances are shown in Table 6. The scale of these large-scale instances ranges from 240 to 480 number of customers and each instance has different vehicle capacity. The average of customer demand is 20 and the standard deviation is 10, which means the customer demand in each instance is relatively small. The customers' location distribution is depot-centered and spreads outward akin to a magnetic field. Firstly, we evaluate the performance of the extended ACO algorithms on these four large-scale SDVRP benchmarking instances. The comparison between the optimal values provided alongside the dataset and the shortest route length and the average route length obtained by the extended algorithms among 20 trials is shown in Table 7. The relative performance difference gap is computed according to (23) that the average gap of ASGCT, ACGCT and MMGCT are computed as 2.93%, 1.68% and 0.41%, respectively. The results obtained by the three extended algorithms all surpass the optimal value on kelly03. In addition, the result obtained by MMGCT on kelly04 also surpasses the optimal value. Such improved performance suggests that our extended ACO algorithms can well handle large-scale SDVRP instances with improved performance.

TABLE 6. Characteristics of the four selected large-scale benchmarking instances.

Name	n	Capacity	Avg. demand	Std. demand
kelly01	240	550	20	10
kelly02	320	700	20	10
kelly03	400	900	20	10
kelly04	480	1000	20	10

* "Name" denotes the instances' given name; "n" denotes the number of nodes (customers); "Capacity" denotes the capacity of vehicles; "Avg. demand" denotes the average of all customer demands in each instance; and "Std. demand" denotes the standard deviation of customer demands distribution in each instance.

Secondly, we further assess the performance of the three extended algorithms on these four large-scale SDVRP-GCT instances. Similar to Section V-B, we set h to 0.1 to transform the SDVRP benchmarking instances into SDVRP-GCT ones and the experimental parameters remain as the same as the

TABLE 7. Experimental results of the three extended ACO algorithms on four large-scale SDVRP instances.

Dataset	Optimal_value	ASGCT			ACGCT			MMGCT		
		Len	Avg (std)	Gap (%)	Len	Avg (std)	Gap (%)	Len	Avg (std)	Gap (%)
kelly01	5646.46	5733.41	5778.96 (33.73)	1.53	5721.39	5778.01 (39.39)	1.33	5671.24	5736.00 (31.25)	0.44
kelly02	8566.04	9066.69	9179.71 (71.88)	5.84	8998.67	9110.60 (85.31)	5.05	8832.60	9025.45 (117.72)	3.11
kelly03	11649.06	**11647.51**	12072.32 (171.40)	**-0.01**	**11633.56**	11988.83 (167.90)	**-0.13**	**11551.52**	11753.85 (106.57)	**-0.84**
kelly04	14639.32	15273.38	15619.42 (176.26)	4.33	14709.22	15351.09 (295.16)	0.48	**14479.30**	14909.72 (216.12)	**-1.09**
Avg. gap (%)	-	-	-	2.93	-	-	1.68	-	-	0.41

* "Optimal_value" denotes the value of the optimal solution provided alongside the dataset on the website.

TABLE 8. Experimental results of the three extended ACO algorithms on four large-scale SDVRP-GCT instances.

Dataset	ASGCT				ACGCT				MMGCT			
	NV	Len	T (s)	Avg (std)	NV	Len	T (s)	Avg (std)	NV	Len	T (s)	Avg (std)
kelly01	10	6131.12	483.96	6184.94 (28.88)	10	**5972.07**	496.38	6069.62 (45.37)	10	5983.21	505.15	6072.57 (48.58)
kelly02	11	9762.81	892.02	9852.42 (42.30)	11	9563.83	903.74	9685.41 (64.93)	11	**9440.57**	916.93	9596.86 (82.61)
kelly03	11	13089.33	1432.38	13237.35 (75.77)	11	12806.85	1439.28	13071.63 (115.72)	11	**12520.46**	1498.54	12717.87 (123.14)
kelly04	12	16412.48	2075.00	16795.47 (153.29)	12	16196.34	2190.33	16615.24 (219.82)	12	**15711.38**	2174.68	16086.33 (201.66)

previous settings. We conducted 20 independent trials on each instance and presented the results in Table 8. In terms of computational time, we can see that ASGCT runs faster than both ACGCT and MMGCT. In terms of the shortest route length and average route length, the results obtained by ACGCT and MMGCT are always shorter than ASGCT. Furthermore, the results obtained by MMGCT are better than ACGCT in most cases, except in kelly01, where the shortest and average route length obtained by ACGCT is slightly shorter than that of MMGCT. The experimental results suggest that for large-scale SDVRP-GCT instances, MMGCT generally performs better than ACGCT.

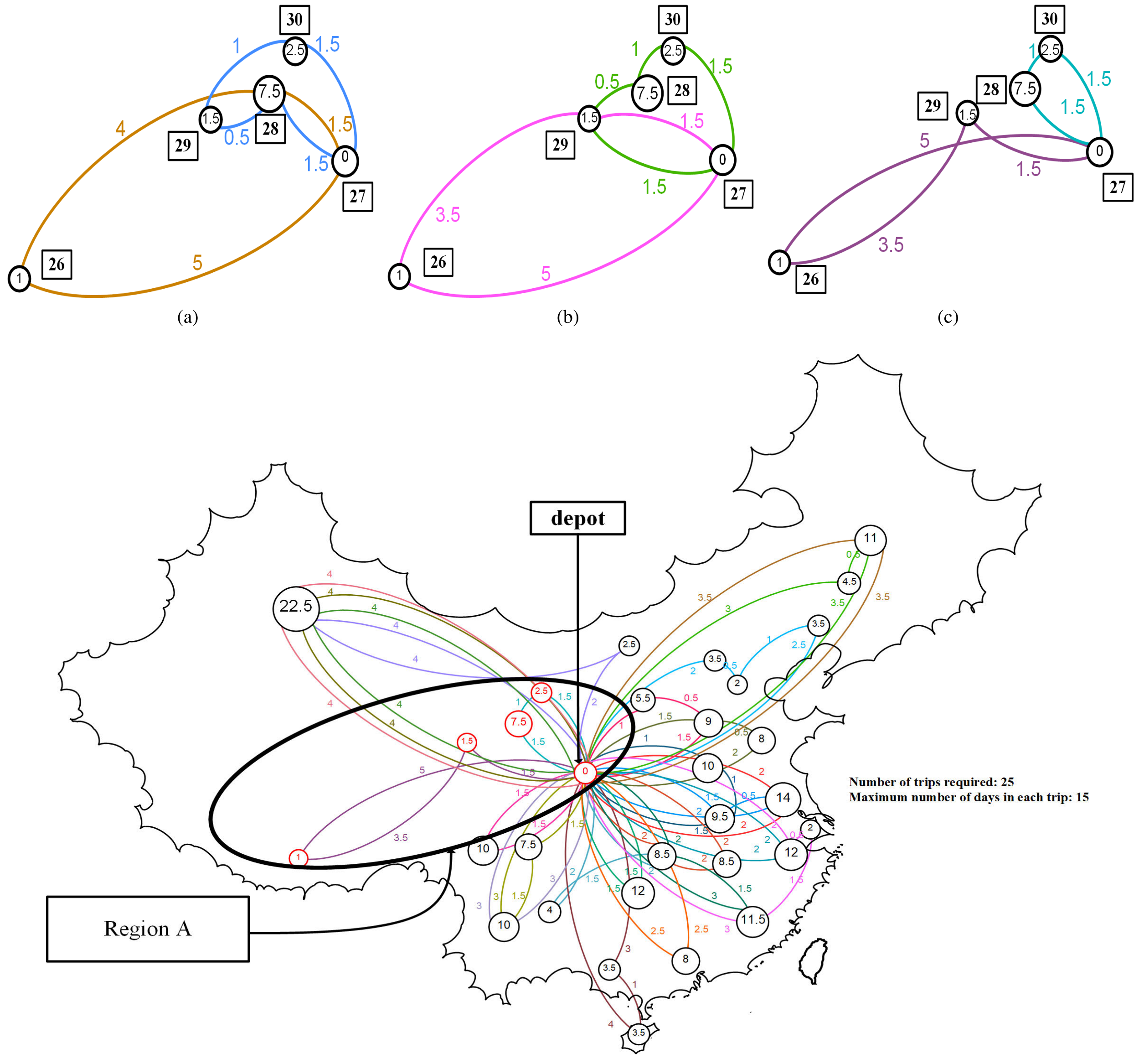
D. EXTENDED ACO ALGORITHMS FOR REAL-WORLD INSTANCE

In this subsection, we present how we evaluate the proposed algorithms on a real-world instance, which we name it "Travel in China with time limits". This instance can be described as follows: a traveller who lives in Xi'an, the capital city of Shaanxi Province, the geographically central part of China and also the depot of the SDVRP-GCT, plans to visit all 5A-graded (the highest grade given in China) tourism attractions across the whole country only by driving a car. Although we select Xi'an as the only depot in this paper, the instance can be varied easily by selecting other cities as the depot. It is noted that when selecting a city as the depot, we assume that the travel time required for the city (depot) is zero. In this instance, the traveler has only a limited time period of fifteen days for each travel (assuming each vacation has the same maximum duration), which is equivalent to the capacity Q of vehicles in SDVRP-GCT, to carry out his travel plan. In this real-world instance, customer demand is the time required to visit each city and route consumption is the time required to drive to the next city. The traveller only has four vacations dedicated to such travels each year. By the end of each travel, the traveller has to come back to Xi'an (depot) for work. The objective of this instance is to figure out the best solution for travelling across all tourism attractions using the least amount of time. To model this instance, we obtained the distance information between provincial capitals and the number of days of staying in each 5A-graded tourism attraction from Question F of the 2015 National Mathematical Modeling Competition held in China.[3] According to the description in Question F, driving time does not exceed eight hours per day, and the average speed of driving on the highway is 90 km/h. We can approximate the time consumed en route (rounded to half-day granularity) according to the distance information between provincial capitals (distance information is taken from Appendix 3 of Question F). Subsequently, the travel time within each province (may have multiple 5A-graded attractions) is computed as the total travelling time spent on the road from the provincial capital to all the scenic spots plus the time of staying in each attraction (suggested time of stay is taken from Appendix 1 of Question F). The computed travel time required in each province is listed in Table 9.

The best solutions obtained by the three algorithms are shown in Table 10. As shown, MMGCT performs better than the other two algorithms. In the best solution obtained by MMGCT, the traveller needs to carry out 25 trips to cover all the attractions, i.e., he needs six and a quarter years' time (assuming four travels per year) to travel to all the most attractive sceneries in China.

To further delve into the experimental results, we provide visualizations of optimal travel routes obtained by the three algorithms in the same region of China denoted as

[3] URL: https://www.shumo.com/home/html/3168.html


30
28
29
27
26
2.5
7.5
1.5
0
1
4
5
0.5
3.5
depot
Region A
22.5
Number of trips required: 25
Maximum number of days in each trip: 15

(a)

(b)

(c)

(d)

FIGURE 5. **Visualization of travel routes. (a) Travel routes in Region A obtained by ASGCT. (b) Travel routes in Region A obtained by ACGCT. (c) Travel routes in Region A obtained by MMGCT. (d) Best travel routes with time limits in China obtained by MMGCT. Note that the number alongside the routes denotes the number of days required to drive en route, the number in the circle denotes the number of days required to visit the province, the number in the box denotes the province ID, and the same color route represents a trip.**

Region A in Figure 5, wherein province IDs are 26, 27, 28, 29 and 30, respectively. The province with ID 27 refers to Shaanxi Province (capital city Xi'an is the depot), while the travel time required for other provinces is shown in Table 9. The travel routes in Region A obtained by the three algorithms are shown in Figures 5a, 5b and 5c, respectively. The travel routes obtained by ASGCT are: 27→30→29→28→27, 27→26→28→27; the travel routes obtained by ACGCT are: 27→30→28→29→27, 27→29→26→27; the travel routes obtained by MMGCT are: 27→28→30→27, 27→29→26→27. The travel routes obtained by the three algorithms show that they all need two trips to visit these provinces. The total number of days required to visit within these provinces (excluding time spent between provinces) are $1 + 7.5 + 1.5 + 2.5 = 12.5$. If we use $R_{[\mathrm{Algorithm}]}$ to denote the total route consumption obtained by the corresponding algorithm, the total number of days required to traverse through Region A are given as follows:

$$R_{\mathrm{ASGCT}} = 1.5 + 1 + 0.5 + 1.5 + 5 + 4 + 1.5 = 15$$
$$R_{\mathrm{ACGCT}} = 1.5 + 1 + 0.5 + 1.5 + 1.5 + 3.5 + 5 = 14.5$$
$$R_{\mathrm{MMGCT}} = 1.5 + 1 + 1.5 + 1.5 + 3.5 + 5 = 14$$

It can be seen that the travel time obtained by MMGCT in Region A is the shortest, ACGCT comes in the second and ASGCT requires the longest time. In general, MMGCT obtains more optimized travel plans in other regions as well.

TABLE 9. Travel time computed for each province.

Province ID	1	2	3	4	5	6	7	8
Demand (day)	3.5	2	9	5.5	2.5	3.5	4.5	11
Province ID	9	10	11	12	13	14	15	16
Demand (day)	2	14	12	9.5	11.5	8.5	8	10
Province ID	17	18	19	20	21	22	23	24
Demand (day)	8.5	12	8	3.5	3.5	7.5	10	4
Province ID	25	26	27	28	29	30	31	-
Demand (day)	10	1	4	7.5	1.5	2.5	22.5	-

* "Province ID" denotes each province in China; "demand" denotes the amount of time required (in terms of days) to travel across all 5A-graded attractions in the corresponding province.

TABLE 10. Results of travel in China with time limits instance.

ASGCT			ACGCT			MMGCT		
NV	Len (day)	T (s)	NV	Len (day)	T (s)	NV	Len (day)	T (s)
25	148.5	11.65	25	148	14.81	25	**145.5**	13.17

* "NV" denotes the number of trips in the best solution; "Len" denotes the number of days required en route in the best solution; "T" denotes the algorithm running time in the best solution.

The best travel routes with time limits in China obtained by MMGCT are shown in Figure 5d.

VI. CONCLUSION

In this paper, we formulate the under-investigated split delivery vehicle routing problem with goods consumed during transit (SDVRP-GCT). Furthermore, we extended three ant colony algorithms, namely ASGCT, ACGCT and MMGCT, to solve SDVRP-GCT. For performance evaluations, we apply the proposed algorithms on eighteen widely adopted SDVRP instances with their respective variations to SDVRP-GCT and a real-world SDVRP-GCT instance. By analyzing the experimental results of the three extended algorithms on different instances, we found that these three algorithms are suitable to solve a large variety of the SDVRP-GCT instances. Specifically, although ASGCT requires lesser computational time, its performance is always worse than the other two. ACGCT may performs slightly better than MMGCT for instances with higher demand and MMGCT not only has superior performance over ACGCT for instances with lower demand, but also performs better than ACGCT in large-scale instances. To the best of our knowledge, this research is the first attempt to solve the SDVRP-GCT problem, which provides baseline solutions and lays the groundwork for future more in-depth research. Going forward, we plan to conduct further research from the following two aspects: 1) using other heuristic algorithms, such as Genetic Algorithm, combined with ACO algorithms to improve the solution of SDVRP-GCT, and 2) devising more algorithms based exact algorithms, such as branch-and-price algorithm, to solve SDVRP-GCT problem.

REFERENCES

[1] G. Laporte, Y. Nobert, and S. Taillefer, "Solving a family of multi-depot vehicle routing and location-routing problems," *Transp. Sci.*, vol. 22, no. 3, pp. 161–172, Aug. 1988.

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

[3] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications* (MOS-SIAM Series on Optimization), no. 18. 2014.

[4] M. Dror and P. Trudeau, "Savings by split delivery routing," *Transp. Sci.*, vol. 23, no. 2, pp. 141–145, May 1989.

[5] M. N. Haddad, R. Martinelli, T. Vidal, S. Martins, L. S. Ochi, M. J. F. Souza, and R. Hartl, "Large neighborhood-based Metaheuristic and branch-and-price for the pickup and delivery problem with split loads," *Eur. J. Oper. Res.*, vol. 270, no. 3, pp. 1014–1027, Nov. 2018.

[6] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza, "Worst-case analysis for split delivery vehicle routing problems," *Transp. Sci.*, vol. 40, no. 2, pp. 226–234, May 2006.

[7] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[8] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, and L. J. M. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adapt. Behav.*, vol. 5, no. 2, pp. 169–207, Jan. 1997.

[9] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, Jul. 1997.

[10] M. D. Albritton and P. R. Mcmullen, "Optimal product design using a colony of virtual ants," *Eur. J. Oper. Res.*, vol. 176, no. 1, pp. 498–520, Jan. 2007.

[11] X. Sun, K. Zhang, M. Ma, and H. Su, "Multi-population ant colony algorithm for virtual machine deployment," *IEEE Access*, vol. 5, pp. 27014–27022, 2017.

[12] D. Lin, L. He, X. Feng, and W. Luo, "Niching Pareto ant colony optimization algorithm for bi-objective pathfinding problem," *IEEE Access*, vol. 6, pp. 21184–21194, 2018.

[13] E. J. Beltrami and L. D. Bodin, "Networks and vehicle routing for municipal waste collection," *Networks*, vol. 4, no. 1, pp. 65–94, 1974.

[14] H. N. Psaraftis, "A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem," *Transp. Sci.*, vol. 14, no. 2, pp. 130–154, May 1980.

[15] 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.

[16] C. Malandraki and M. S. Daskin, "Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms," *Transp. Sci.*, vol. 26, no. 3, pp. 185–200, Aug. 1992.

[17] M. Gendreau, G. Laporte, and R. Séguin, "Stochastic vehicle routing," *Eur. J. Oper. Res.*, vol. 88, no. 1, pp. 3–12, 1996.

[18] J. Renaud, G. Laporte, and F. F. Boctor, "A tabu search heuristic for the multi-depot vehicle routing problem," *Comput. Oper. Res.*, vol. 23, no. 3, pp. 229–235, Mar. 1996.

[19] D. Sariklis and S. Powell, "A heuristic method for the open vehicle routing problem," *J. Oper. Res. Soc.*, vol. 51, no. 5, p. 564, May 2000.

[20] R. H. Xie, Z. Q. Qiu, and Y. Y. Zhang, "A new heuristics for VRP with simultaneous delivery and pick-up," in *Proc. Int. Conf. Transp. Eng.*, Jul. 2007, pp. 2175–2180.

[21] G. Perboli, R. Tadei, and D. Vigo, "The two-echelon capacitated vehicle routing problem: Models and math-based heuristics," *Transp. Sci.*, vol. 45, no. 3, pp. 364–380, Aug. 2011.

[22] S. Erdogan and E. Miller-Hooks, "A green vehicle routing problem," *Transp. Res. Part E, Logistics Transp. Rev.*, vol. 48, no. 1, pp. 100–114, 2012.

[23] 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.

[24] 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.

[25] S. Ray, A. Soeanu, J. Berger, and M. Debbabi, "The multi-depot split-delivery vehicle routing problem: Model and solution algorithm," *Knowl.-Based Syst.*, vol. 71, pp. 238–265, Nov. 2014.

[26] C. Archetti, A. M. Campbell, and M. G. Speranza, "Multicommodity vs. Single-commodity routing," *Transp. Sci.*, vol. 50, no. 2, pp. 461–472, May 2016.

[27] A. F.-W. Han and Y.-C. Chu, "A multi-start heuristic approach for the split-delivery vehicle routing problem with minimum delivery amounts," *Transp. Res. Part E, Logistics Transp. Rev.*, vol. 88, pp. 11–31, Apr. 2016.

[28] X. Li, M. Yuan, D. Chen, J. Yao, and J. Zeng, "A data-driven three-layer algorithm for split delivery vehicle routing problem with 3d container loading constraint," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 528–536.

[29] Y. Li and S. H. Chung, "Disaster relief routing under uncertainty: A robust optimization approach," *IISE Trans.*, vol. 51, no. 8, pp. 869–886, Aug. 2019.

[30] M. Dror, G. Laporte, and P. Trudeau, "Vehicle routing with split deliveries," *Discrete Appl. Math.*, vol. 50, no. 3, pp. 239–254, May 1994.

[31] M. Jin, K. Liu, and R. O. Bowden, "A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem," *Int. J. Prod. Econ.*, vol. 105, no. 1, pp. 228–242, Jan. 2007.

[32] C. Archetti, N. Bianchessi, and M. G. Speranza, "Branch-and-cut algorithms for the split delivery vehicle routing problem," *Eur. J. Oper. Res.*, vol. 238, no. 3, pp. 685–698, Nov. 2014.

[33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 80–671, 1983.

[34] D. Wang, X. Qian, C. Quek, A.-H. Tan, C. Miao, X. Zhang, G. S. Ng, and Y. Zhou, "An interpretable neural fuzzy inference system for predictions of underpricing in initial public offerings," *Neurocomputing*, vol. 319, pp. 102–117, Nov. 2018.

[35] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[36] Y. Saeed, K. Ahmed, M. Zareei, A. Zeb, C. Vargas-Rosales, and K. M. Awan, "In-vehicle cognitive route decision using fuzzy modeling and artificial neural network," *IEEE Access*, vol. 7, pp. 20262–20272, 2019.

[37] J. Zhang, F. Yang, and X. Weng, "An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows," *IEEE Access*, vol. 6, pp. 63468–63485, 2018.

[38] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[39] T. Stutzle and H. Hoos, "MAX-MIN ant system and local search for the traveling salesman problem," in *Proc. IEEE Int. Conf. Evol. Comput. (ICEC)*, Apr. 1997, pp. 309–314.

[40] T. Stützle and H. H. Hoos, "MAX–MIN ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, 2000.

[41] C. Archetti, R. Mansini, and M. G. Speranza, "Complexity and reducibility of the skip delivery problem," *Transp. Sci.*, vol. 39, no. 2, pp. 182–187, May 2005.

[42] C. Archetti, D. Feillet, M. Gendreau, and M. Grazia Speranza, "Complexity of the VRP and SDVRP," *Transp. Res. Part C, Emerg. Technol.*, vol. 19, no. 5, pp. 741–750, Aug. 2011.

[43] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, "Modification of the ant colony optimization for solving the multiple traveling salesman problem," *Romanian J. Inf. Sci. Technol.*, vol. 16, no. 1, pp. 65–80, 2013.

[44] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," in *Proc. Ethnographic Praxis Ind. Conf.*, 1988, vol. 9, no. 2, pp. 95–99.

[45] C. M. Chen, W. C. Xie, and S. S. Fan, "The research on VRP based on max-min ant colony algorithm," *Adv. Mater. Res.*, vols. 219–220, pp. 1285–1288, Mar. 2011.

[46] G.-S. Wang and Y.-X. Yu, "An improved ant colony algorithm for VRP problem," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat.*, Apr. 2010, pp. 129–133.

[47] X. Liu, C. Yang, and X. Liu, "Optimization of vehicle routing problem based on max-min ant system with parameter adaptation," in *Proc. 7th Int. Conf. Comput. Intell. Secur.*, Dec. 2011, pp. 129–133.

[48] J. M. Belenguer, M. C. Martinez, and E. Mota, "A lower bound for the split delivery vehicle routing problem," *Oper. Res.*, vol. 48, no. 5, pp. 801–810, Oct. 2000.

WENZHE YANG received the B.S. degree in computer science and technology from Jilin University, in 2017, where she is currently pursuing the M.S. degree with the computer science and technology with the Machine Learning Engineering Laboratory. Her current interests include ant colony optimization, image processing, and deep learning.

DI WANG (Senior Member, IEEE) received the B.Eng. degree in computer engineering and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2003 and 2014, respectively. He is currently working as a Senior Research Fellow and the Research Manager of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University. His research interests include computational intelligence, decision support systems, computational neuroscience, autonomous agents, affective computing, and ubiquitous computing.

WEI PANG received the Ph.D. degree in computing science from the University of Aberdeen, U.K., in 2009. He is currently an Associate Professor with Heriot-Watt University, Edinburgh, U.K. He is also an Honorary Senior Lecturer with the University of Aberdeen and a part of the Edinburgh Centre for Robotics. He has authored more than 100 articles, including 50 journal articles. His research interests include bio-inspired computing, data mining, machine learning, and qualitative reasoning. He was a recipient of the Best Paper Award in the 19th Annual U.K., Workshop on Computational Intelligence (UKCI 2019) and the Best Paper Runner Up Award in the 12th International Conference on Advanced Data Mining and Applications (ADMA 2016). His research has been funded by EPSRC, ESRC, Royal Society, Royal Society of Edinburgh, and most recently, Cancer Research U.K.

AH-HWEE TAN (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in computer and information science from the National University of Singapore and the Ph.D. degree in cognitive and neural systems from Boston University, Boston, MA, USA. He was a Research Manager with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore, spearheading the Text Mining and Intelligent Agents research programs. He is currently a Professor of computer science and the Associate Chair (a Research) of the School of Computer Science and Engineering, Nanyang Technological University (NTU). His current research interests include cognitive and neural systems, brain inspired intelligent agents, machine learning, knowledge discovery, and text mining.

YOU ZHOU received the bachelor's and Ph.D. degrees from Jilin University, in 2002 and 2008, respectively. He is currently a Professor with the College of Computer Science and Technology, Jilin University, Changchun, China. His research interests include machine learning, pattern recognition, and bioinformatics.