Received August 4, 2020, accepted August 13, 2020, date of publication August 17, 2020, date of current version August 26, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.3017132

Split Demand One-to-One Pickup and Delivery Problems With the Shortest-Path Transport Along Real-Life Paths

JIAN XIONG[1,2], XIN QI[2], ZHUO FU[1], AND WEIXIONG ZHA[2]
[1]School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China
[2]School of Transportation and Logistics, East China Jiao Tong University, Nanchang 330013, China
Corresponding author: Xin Qi (qixin@ecjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71271220, and in part by the Research Project of Humanities and Social Sciences in the Universities of Jiangxi Province under Grant JC18213 and Grant JC19207.

ABSTRACT A variation of the One-to-one Pickup and Delivery Problem (OPDP) in connected graphs, the Split Demand One-to-one Pickup and Delivery Problem with the Shortest-path Transport along Real-life Paths (SDOPDPSTRP) is abstracted from passenger train operation plans based on networks. Unlike the classical OPDP, in the SDOPDPSTRP: the demands can be split and must be transported along the shortest path according to passengers requirements and vehicles should travel along a real-life path. A new kind of integer programming model is formulated for the SDOPDPSTRP based on the connection relationship between pickup-delivery demands (pd-pairs). Two different categories of splitting strategies are proposed to solve the SDOPDPSTRP: split the demands before the calculation and split the demands during the calculation. Two Multi-Start Variable Neighborhood Descent (a MS_VND originating from the other literature and a new MS_VND' IN developed in this article) and seven neighborhood operators are proposed for these two splitting strategies to solve the SDOPDPSTRP. The results show that Approach III outperforms Approach I and Approach II in terms of average solutions with the same algorithm termination conditions and in terms of time efficiency, which has great practical significance for real-life transport organizations.

INDEX TERMS One-to-one pickup and delivery problem, split demand, shortest-path transport, real-life connected graph, integer programming, multi-start, variable neighborhood descent, Gurobi solver.

I. INTRODUCTION

Travelling along the shortest path, an important requirement of passengers has no always been fully satisfied. As travel modes diversify, it is increasingly important to meet the needs of the passengers to increase the competitiveness of transport enterprises when formulating transportation schemes. Take Passenger Train Operation Plans (PTOP), which are based on lines, for instance, trains travel through real-life paths, and passengers' demands between every two stations are transported along the shortest path by one or more trains.

Currently, a Chinese high-speed rail network has been formed, it has become an urgent problem to design the PTOP based on networks, which is different from the general PTOP based on lines. Therefore, a new One-to-one Pickup and Delivery Problem (OPDP) is abstracted to solve this new PTOP, which can be formed as follows: There are several pickup-delivery demands (pd-pairs) and trains in a real-life connected graph. The pd-pairs that are chosen must be transported along the shortest path according to passengers' requirements. Each pd-pair can be split into different trains. Trains cannot visit (stop at or pass through) any station more than once, namely, each train should travel along a real-life path. Constraints, such as train capacity, train travel distances, and train stops, need to be considered.

This new problem can be addressed by introducing a set of maximum-income routes to be traversed by a fleet of vehicles to serve a group of known pd-pairs, which is referred to as the Split Demand One-to-one Pickup and Delivery Problems with the Shortest-path Transport along Real-life Paths (SDOPDPSTRP) in this article. Since each pd-pair must be transported along the shortest path and vehicle stops

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

need to be considered, the SDOPDPSTRP will be studied based on connected graphs, which should not be abstracted into complete graphs.

The SDOPDPSTRP, which has rarely been studied in the literature, is studied in this article. The key contributions of this work are as follows:

(i) A new OPDP, the SDOPDPSTRP, is refined from the Passenger Train Operation Plans based on real-life connected graph, and a new model for the SDOPDPSTRP is studied.

(ii) A new Multi-Start Variable Neighborhood Descent (MS_VND' in this article) with seven neighborhood operators are developed to solve the SDOPDPSTRP based on the second splitting methods in this article: splitting demands during the calculation.

(iii) New instances for the SDOPDPSTRP.

The remainder of this article is organized as follows. Section 2 presents related studies. Section 3 presents the model for the SDOPDPSTRP. Section 4 presents the solution approach. Section 5 presents the computation results. Finally, conclusions and future work are presented in Section 6.

II. RELATED LITERATURES

A. GPDP, OPDP, AND OPDPSTRP

Many scholars have carried out research on the PDP over the past few years. References [1], [2] reviewed current GPDP research and divided studies into two categories. The first category comprises the transportation of goods from a depot to line-haul customers and from back-haul customers to the depot, and this is denoted as the Vehicle Routing Problem with Back-hauls (VRPB). Research on the VRPB was reviewed by [3]. The second category considers all problems that occur when goods are transported between pickup and delivery locations, which is denoted as the General Vehicle Routing Problem with Pickups and Deliveries (GVRPPD). References [4], [5] divided the GPDP into three categories: the One-to-Many-to-One PDP (OMOPDP; [6], [7]), the Many-to-many PDP (MMPDP; [8]–[13]) and the One-to-one PDP (OPDP; [14]–[16]).

Most classical OPDPs are studied using complete graphs, and pickup points must be visited prior to delivery points (e.g. [17]–[23]). The classical OPDP can be easily formulated as a Mixed-Integer Program (MIP), such as those reported in [14]–[16]. Reference [24] classified the solution methods for the Dial-A-Ride Problem (DARP, an important category of the OPDP). Reference [25] proposed a combination of cutting planes to find feasible solutions for the OPDP with incompatibility constraints. Reference [26] studied a new kind of OPDP, the One-to-one Pickup and Delivery Problem with the Shortest-path Transport along Real-life Paths (OPDPSTRP), in which each pd-pair must be transported along the shortest path and each vehicle should travel along real-life paths in connected graphs. A new kind of modeling method was proposed for the OPDPSTRP according to its new route constructions.

Unlike the classical OPDP in a complete graph, the OPDPSTRP studied in this article is described based on connected graphs since the number of vehicle stops needs to be considered. Some research conducted on the route structure of the classical OPDP can provide references for the OPDPSTRP. Reference [27] listed four kinds of ride-sharing patterns for the Ride-sharing Problem. Reference [21] noted out that the cheapest path is not always the quickest path, and a comparison of multiple paths between every two points was necessary. Reference [28] proposed a method for relocating a pd-pair by considering four cases, and the shortest path was chosen as the optimal routing scheme in each local search move. References [29], [30] studied the Ride-sharing Problem (a kind of OPDP) in real-life networks.

Additionally, as in the OPDPSTRP, each vehicle starts at its location (regarded as a depot) and ends at the final delivery point of the contents transported by the vehicle; therefore, it can be considered to be a multi-depot (vehicles) problem. Most OPDP research is based on a single depot, such as that reviewed by [8], [9], [21], [22], [28], [29], [31]–[34]. Some OPDP research is based on multiple depots (vehicles), which is mainly concerned with the Taxi-sharing Problem and Ride-sharing Problem. For example, there is a starting point and an ending point for each vehicle in [35]–[38] while only the starting point is considered for each vehicle in [39].

B. SDVRP AND SDPDP

Since [40] introduced the split delivery vehicle routing problem (SDVRP), which is well known in the literature, a growing number of academics have worked in the field of split demand. Reference [41] provided a survey on the SDVRP that overviews its variants and, in general, all routing problems that consider split deliveries.

Splitting demands into different vehicles may result in better schedules, so another feature of the SDOPDPSTRP is studied in this article, one kind of Split Demand Pickup and Delivery Problem (SDPDP). There are many categories of the SDPDPs, which can provide some reference for the SDOPDPSTRP. References [42], [43] first proposed the vehicle routing problem with split deliveries and pickups (VRPSPDP). The one-commodity SDPDTSP is discussed by [44], and the OMOPDP with split demands has been discussed by [45]–[47], and [48]. The many-to-many SDPDP is studied by [15], [49], [50]. References [51], [52] proposed a kind of multi-vehicle One-to-one SDPDP.

C. NEIGHBORHOOD AND ALGORITHM

Reference [50] note that when demands are from 51% to 60% of the capacity of the vehicle, up to 30% of the transportation costs can be saved. They find that the PDP with split demand can perform better when the vehicle capacity is approximately twice that of the average demand. Therefore, the keys to solving the SDOPDPSTRP are "splitting or not?" and "how to split?". Some studies can provide reference to solve this problem.

As for splitting strategies, demands are split when routes are overloaded in [53]. Reference [48] split the demands for a Vehicle Routing Problem into discrete Split Deliveries and

TABLE 1. Differences between the OPDPSTRP in [26] and the SDOPDPSTRP in this article.

Studies	Reference [26]	This paper
Problems	OPDPSTRP	SDOPDPSTRP: Split Demand OPDPSTRP
Models	Integer Linear Programming (ILP)	Integer Program (IP), linearized by the methods mentioned in reference [67]
Variables	$x_{i,j}^k$, y_e^k, u_i^k and sn_n^k	$x_{i,j}^k$, qs_i^k, y_e^k, u_i^k and sn_n^k
Algorithms	Gurobi, VND, VNS, MS_VND and MS_VNS	Gurobi, MS_VND and MS_VND'
Neighborhoods	VND, VNS, MS_VND and MS_VNS Insert, Spread, Point-delete, Rout-delete, and Perturbation	(1) MS_VND Insert, Spread, Point-delete, Rout-delete and Perturbation (2) MS_VND' Split, Insert, Swap, Spread, Point-delete, Rout-delete, and Perturbation
Splitting methods	-	(1) Split demands before the calculation 20/10/5/1/x+MS_VND (Approach I) 25/10/5/1/x+MS_VND (Approach II) (2) Split demands during the calculation MS_VND' (Approach III)
Instances	84 benchmark instances of the OPDPSTRP	63 benchmark instances of the SDOPDPSTRP

Pickups (VRPSPDP) according to the ratios of 25/10/5/1/x and 20/10/5/1/x before the calculation, which are adjusted from [54].

As for neighborhoods, [55] presented eight kinds of local search moves for the OPDP: couple-exchange, block-exchange, relocate-couple, relocate-block, multi-relocate, 2-opt-L, double-bridge and shake. References [56], [57] modified three large neighborhood removal heuristics and two large neighborhood insertion heuristics from [58]–[60] for the OPDP. Additionally, the studies of [24], [61] show that the solution feasibility of the OPDP is an important issue to the neighborhood efficiency of the algorithm. Reference [53] used four operators (relocation, exchange, 2-opt and split-point reposition) for a Simultaneous Delivery and Pick up Vehicle Routing Problem with Split Loads (SDPVRPSL). Reference [48] proposed five operators for the VRPSPDP: intra-swap, intra-reverse, inter-reassignment, inter-swap and tail swap. Reference [52] propose 6 intra-route neighborhoods and 4 inter-route neighborhoods in randomized variable neighborhood descent for a SDOPDP.

As for the algorithms, [62] solved the OPDPTSP via the GRASP and the VND. Reference [15] proposed an efficient heuristic that combines the strengths of tabu search and simulated annealing for the OPDPSD. An Iterated Local Search (ILS) was proposed by [63]. Reference [52] classified the solution methods for the DARP. Reference [25] proposed a combination of cutting planes to find the feasible solutions for a Pickup and Delivery Problem with Incompatibility constraints (OPDPI). Additionally, some Local Search (LS) meta-heuristics studied for the PDP can also be used as references. The Adaptive Large Neighborhood Search (ALNS) was proposed for the PDP by [57], [64]–[66]. References [4], [5] reviewed the algorithms for the static and the dynamic PDP. Some exact methods have also been proposed for the OPDP. For example, [14] solve two mixed integer linear programming models of the OPDPTSP using the Cplex solver. Reference [16] proposed a mixed integer programming model for the green OPDP, and solved it using the Cplex solver. As for the SDOPDP, [15] proposed an efficient heuristic that combines the strengths of tabu search and aimulated annealing. Tabu search is used in [48], [53]. A VNS is used by [51]. A branch-and-cut algorithm is used by [44], [45]. A branch-and-price approach is proposed by [46], [47]. Reference [52] introduced a hybrid meta-heuristic based on the Iterated Local Search (ILS) and split loads with a new larger dynamic programming-based neighborhoods.

In summation, there is far more research on the classical OPDP and SDPDP than on the SDOPDPSTRP, but there is no research focusing directly on the SDOPDPSTRP proposed in this article. A new kind of OPDP studied by [26], the OPDPSTRP, can provide a reference. This study extends the work of [26] by introducing "split demand" to the OPDPSTRP. The main differences between these two works are listed in TABLE 1.

III. PROBLEM DEFINITION AND MATHEMATICAL MODEL

A. PROBLEM DEFINITION

To define the proposed SDOPDPSTRP in mathematical terms, we specify a connected graph, $G=(N, E, P, K)$, where $N =\{1,\ldots, n_0\}$ for vertexes, $E =\{1,\ldots, e_0\}$ for edges, $P =\{1,\ldots, P\}$ for pd-pairs, and $K =\{1,\ldots, m\}$ for vehicles. Each pd-pair i with demand q_i yields income $\pi_i \times q_i$. Each vehicle $K \epsilon K$ has a maximum capacity Q_k and a fixed cost vc^k. The transportation cost per unit length of vehicle k is tc^k. Each vehicle k has a stop cost sc_n^k at node n.

The system also obeys the following assumptions.

(i) Each pd-pair can be split (different from OPDPSTRP in [1]), and must be transported through the shortest path according to passengers' requirements, with the pickup point being visited prior to the delivery point.

(ii) Each vehicle must travel along a real-life path beginning with the first pickup point and ending at the last delivery point, namely each point cannot be accessed multiple times by one vehicle, a common practice in the Passenger Train Operation Plans and other similar plans.

(iii) For each vehicle, the travel distance limit (from the first pickup point to the last delivery point) is D, and the maximum number of stops is M_0. No vehicle can be overloaded.

(iv) The total cost of each vehicle consists of the constant cost, travel cost, and stop cost. To maximize income, not all pd-pairs need to be transported.

(v) There is only one shortest path between any two nodes in the graph.

By defining the aforementioned problem, we hope to identify a suitable scheme to help optimize the benefits.

B. PARAMETERS AND VARIABLES

(i) Parameters

q_i: Demand of pd-pair i.

π_i: Revenue of pd-pair i.

Q^k: Capacity of vehicle k.

vc^k: Fixed cost of vehicle k.

tc^k: Transportation cost per unit length of vehicle k.

sc_n^k : Stop cost of vehicle k at node n.

le_e: Length of edge e.

$ld_{i,e}$: Judgment parameter for whether pd-pair i moves via edge e or not.

$lc_{i,j}$: Length of the connecting section for pd-pair j to connect to vehicle/pd-pair i, where $i \in P$ for pd-pairs and $i = \{p+1\}$ for vehicles.

$ct_{i,j}$: Judgment parameter for whether pd-pair j can (or cannot) connect to vehicle/pd-pair i, where $i \in P$ for pd-pairs and $i = \{p+1\}$ for vehicles.

$ca_{i,j}$: Judgment parameter for whether pd-pair j can (or cannot) connect after vehicle/pd-pair i, where $i \in P$ for pd-pairs and $i = \{p+1\}$ for vehicles.

$sod_{i,n}$: Judgment parameter for whether pd-pair i can (or cannot) be picked up/delivered at node n.

All the above parameters can be set as in [26].

(ii) Variables

$x_{i,j}^k$: pd-pair j connects to vehicle or pd-pair i in vehicle k or not, where $i \in P$ for pd-pairs and $i = \{p+1\}$ for vehicles.

y_e^k : Vehicle k travels by way of edge e with pd-pairs or not.

qs_i^k : Load of each pd-pair i transported by vehicle k.

u_i^k : Sequence number of pd-pair i transported by vehicle k, namely $u_i^k < u_j^k$ when $x_{i,j}^k = 1$.

sn_n^k : Vehicle k stops at node n or not.

C. MATHEMATICAL MODEL

The route structure of the SDOPDPSTRP is actually similar to that of the OPDPSTRP described in [26]. The SDOPDPSTRP can be formulated as an integer programming (IP) model. It should be noted that in this IP model, the decision variables are set based on the relationships between pd-pairs, which is totally different from the classical OPDP. In the OPDP mathematical model, the values of the decision variables are based on the relationships between nodes. Take the variable $x_{i,j}^k$ for instance, it means that node j come after node i in the classic OPDP, while it means that PD-pair j come after PD-pair i in the OPDPSTRP and the SDOPDPSTRP. For details of the new model methods and the new route construction rules, referring to [26].

1) OBJECTIVE FUNCTIONS

The objective function of the proposed IP model mainly consists of four components, i.e., the total income, the total fixed cost of using vehicle, the total transportation cost and the total stop cost.

(i) Total income

$$\sum_{k\in K}\sum_{i\in P}\pi_i \cdot qs_i^k \tag{1}$$

(ii) Total fixed cost of using a vehicle

$$\sum_{k\in K}\sum_{j\in P} vc^k \cdot x_{p+1,j}^k \tag{2}$$

(iii) Total transportation cost

$$\sum_{k\in K} tc^k \cdot (\sum_{e\in E} le_e \cdot y_e^k + \sum_{i\in P\cup\{p+1\}}\sum_{j\in P} lc_{i,j}\cdot x_{i,j}^k) \tag{3}$$

(iv) Total stop cost

$$\sum_{k\in K}\sum_{n\in N} sc_n^k \cdot sn_n^k \tag{4}$$

It is hoped that we can identify a suitable vehicle routing scheme to maximize the benefits:

$$\begin{aligned}&\sum_{k\in K}\sum_{i\in P}\pi_i \cdot qs_i^k - [\sum_{k\in K}\sum_{j\in P} vc^k \cdot x_{p+1,j}^k\\ &\quad + \sum_{k\in K} tc^k \cdot (\sum_{e\in E} le_e \cdot y_e^k + \sum_{i\in P\cup\{p+1\}}\sum_{j\in P} lc_{i,j}\cdot x_{i,j}^k)\\ &\quad + \sum_{k\in K}\sum_{n\in N} sc_n^k \cdot sn_n^k].\end{aligned}$$

2) CONSTRAINTS

(i) The constraints of determining the order between pd-pairs/vehicle are

$$x_{i,j}^k \le ct_{i,j} \quad \forall k\in K,\ i\in P\cup\{p+1\},\ j\in P \tag{5}$$

$$x_{i,j}^k \le \sum_{i_0\in P\cup\{p+1\}} ca_{i_0,i}\cdot x_{i_0,i}^k \quad \forall k\in K,\ i\in P,\ j\in P \tag{6}$$

$$\sum_{j\in P} ca_{i,j}\cdot x_{i,j}^k \le 1 \quad \forall k\in K,\ i\in P\cup\{p+1\} \tag{7}$$

$$x_{i,i}^k = 0 \quad \forall k\in K,\ i\in P \tag{8}$$

$$u_i^k - u_j^k + p\cdot x_{i,j}^k \le p-1 \quad \forall k\in K, i,\ j\in P \tag{9}$$

$$\sum_{i\in P\cup\{p+1\}} x_{i,j}^k \le 1 \quad \forall k\in K, i,\ j\in P \tag{10}$$

(ii) The constraints of the splitting demands are

$$\sum_{k\in K} qs_i^k \le q_i \quad \forall i\in P \tag{11}$$

$$\sum_{j\in P\cup\{p+1\}} x_{j,i}^k \le qs_i^k \quad \forall k\in K, i\in P \tag{12}$$

(iii) The capacity constraints are

$$\sum_{i\in P}(ld_{i,e}\cdot qs_i^k\cdot \sum_{j\in P\cup\{p+1\}} x_{j,i}^k)\le Q^k \quad \forall k\in K,\ e\in E \tag{13}$$

(iv) The stop constraints are

$$sn_n^k \ge sod_{i,n}\cdot \sum_{j\in P\cup\{p+1\}} x_{j,i}^k \quad \forall k\in K,\ i\in P,\ n\in N \tag{14}$$

$$\sum_{n\in N} sn_n^k \le M_0 \quad \forall k\in K \tag{15}$$

(v) The constraints of whether a vehicle is traveling along edge e or not are

$$ld_{i,e}\cdot \sum_{j\in P\cup\{p+1\}} x_{j,i}^k \le y_e^k \quad \forall k\in K,\ i\in P,\ e\in E \tag{16}$$

(vi) The constraints of whether a vehicle is assigned vehicle or not are

$$y_e^k \le \sum_{j\in P} x_{p+1,j}^k \quad \forall k\in K, e\in E \tag{17}$$

(vii) The route length constraints are

$$\sum_{e\in E} le_e\cdot y_e^k + \sum_{i\in P}\sum_{j\in P} lc_{i,j}\cdot x_{i,j}^k \le D \quad \forall k\in K \tag{18}$$

(viii) The domains of the variables are

$$x_{i,j}^k \in \{0,1\} \quad \forall k\in K,\ i\in P\cup\{p+1\},\ j\in P \tag{19}$$
$$qs_i^k \in \{0,1,2,\ldots\} \quad \forall k\in K,\ i\in P \tag{20}$$
$$y_e^k \in \{0,1\} \quad \forall k\in K,\ e\in E \tag{21}$$
$$u_i^k \in \{1,2,3,\ldots\} \quad \forall k\in K,\ i\in P \tag{22}$$
$$sn_n^k \in \{0,1\} \quad \forall k\in K,\ n\in N \tag{23}$$

3) LINEARIZATION OF THE IP MODEL

The above model is an IP model because the constraint (13) is nonlinear. Reference [67] proposed methods to convert nonlinear formulas into linear formulas. For example, the nonlinear formula $r = zy$ can be replaced by the linear formulas (24) and (25); where z is an 0-1 variable and M is a positive constant with a sufficiently large value.

$$y-(1-z)\cdot M \le r \le y+(1-z)\cdot M \tag{24}$$
$$-z\cdot M \le r \le z\cdot M \tag{25}$$

Formula (13) is nonlinear, and $\sum_{j\in P\cup\{p+1\}} x_{j,i}^k$ is 0/1 variables obviously.

Let:

$$qsx_i^k = qs_i^k\cdot \sum_{j\in P\cup\{p+1\}} x_{j,i}^k \quad \forall k\in K,\ i\in P \tag{26}$$

After replacing the nonlinear constrain (13) by formula (27), (28) and (29) according to the methods mentioned in [67], the integer programming (IP) model for the SDOPDPSTRP is converted into a new integer linear programming (ILP) model that can be solved by the Gurobi solver. M is a constant with a sufficiently large value.

$$qs_i^k - (1-\sum_{j\in P\cup\{p+1\}} x_{j,i}^k)\cdot M \le qsx_i^k \le qs_i^k + (1-\sum_{j\in P\cup\{p+1\}} x_{j,i}^k)\cdot M \quad \forall k\in K,\ i\in P \tag{27}$$

$$-(1-\sum_{j\in P\cup\{p+1\}} x_{j,i}^k)\cdot M \le qsx_i^k \le (1-\sum_{j\in P\cup\{p+1\}} x_{j,i}^k)\cdot M \quad k\in K,\ i\in P \tag{28}$$

$$\sum_{i\in P}(ld_{i,e}\cdot qsx_i^k) \le Q^k \quad \forall k\in K,\ e\in E \tag{29}$$

$$qsx_i^k \in \{0,1,2,\ldots\} \quad \forall k\in K,\ i\in P \tag{30}$$

D. A FEASIBLE SOLUTION FOR A SMALL INSTANCE

For a better introduction to the SDOPDPSTRP model, a small instance is given as follows. FIGURE 1 is a connected graph, and the edge lengths are shown in the figure. In a feasible schedule, 5 pd-pairs (demands: 2, 2, 6, 2 and 2) are transported by two vehicles (capacity: 5, maximum distance: 30, and maximum stops: 6) along two routes (paths). Points 1 and 14 are the vehicles locations, and points 2, 3, 4, 15, 14, 12, 6, 8 and 10 are the stop nodes.

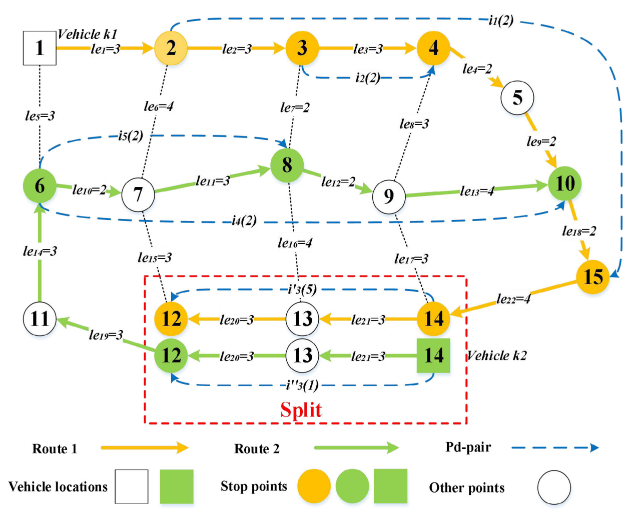


FIGURE 1. A connected graph with two routes.

$ct_{i,j}$, $ca_{i,j}$ and $lc_{i,j}$ are given in TABLE 2.

Let $\pi_i = 15$, $vc^k = 1$, $tc^k = 1$, and $sc_n^k = 1$. pd-pair $q_3 = 6$ is split into $qs_3^{k_1} = 5$ and $qs_3^{k_2} = 1$ because the vehicle capacity is 5. The values of the decision variables for the schedule are listed in TABLE 3 (only non-zero variable are listed).

Therefore, $\sum_{k\in K}\sum_{i\in P}\pi_i\cdot qs_i^k = 15\times(2+2+5+2+2+1) = 210$, $\sum_{k\in K}\sum_{j\in P} vc^k\cdot x_{p+1,j}^k = 1\times 2 = 2$, $\sum_{k\in K} tc^k\cdot \sum_{e\in E}$

TABLE 2. Values of $ct_{i,j}$ and $ca_{i,j}$.

$ct_{i,j}/ca_{i,j}/lc_{i,j}$	j=1	j=2	j=3	j=4	j=5
i=1	-	1/0/0	$1/1/le_{22}$	$0/0/\infty$	$1/1/le_{22}+le_{21}+le_{20}+le_{19}+le_{14}$
i=2	$0/0/\infty$	-	$1/1/le_4+le_9+le_{18}+le_{22}$	$0/0/\infty$	$1/1/le_4+le_9+le_{18}+le_{22}+le_{21}+le_{20}+le_{19}+le_{14}$
i=3	$1/1/le_{19}+le_{14}+le_5+le_1$	$1/1/le_{19}+le_{14}+le_5+le_1+le_2$	-	$1/1/le_{19}+le_{14}$	$1/1/le_{19}+le_{14}$
i=4	$0/0/\infty$	$0/0/\infty$	$1/1/le_{18}+le_{22}$	-	1/0/0
i=5	$0/0/\infty$	$0/0/\infty$	$1/1/le_{12}+le_{13}+le_{18}+le_{22}$	1/1/0	-
k_1	$1/1/le_1$	$1/1/le_1+le_2$	$1/1/le_5+le_{14}+le_{19}+le_{20}+le_{21}$	$1/1/le_5$	$1/1/le_5$
k_2	$1/1/le_{21}+le_{20}+le_{19}+le_{14}+le_5+le_1$	$1/1/e_{21}+e_{20}+e_{19}+e_{14}+e_5+e_1+e_2$	1/1/0	$1/1/e_{21}+e_{20}+e_{19}+e_{14}$	$1/1/le_{21}+le_{20}+le_{19}+le_{14}$

TABLE 3. Decision variables for the schedule.

Variables	*Route 1*	*Route 2*
$x_{i,j}^k$	$x_{k_1,i_1}^{k_1}=1,\ x_{i_1,i_2}^{k_1}=1,\ x_{i_1,i_3}^{k_1}=1$	$x_{k_2,i_3}^{k_2}=1,\ x_{i_3,i_4}^{k_2}=1, x_{i_4,i_5}^{k_2}=1$ or $x_{k_2,i_3}^{k_2}=1,\ x_{i_3,i_5}^{k_2}=1, x_{i_5,i_4}^{k_2}=1$
y_e^k	$y_{e_2}^{k_1}=1,\ y_{e_3}^{k_1}=1,\ y_{e_4}^{k_1}=1,\ y_{e_9}^{k_1}=1,\ y_{e_{18}}^{k_1}=1,\ y_{e_{21}}^{k_1}=1,\ y_{e_{20}}^{k_1}=1$	$y_{e_{21}}^{k_2}=1,\ y_{e_{20}}^{k_2}=1,\ y_{e_{10}}^{k_2}=1,\ y_{e_{11}}^{k_2}=1,\ y_{e_{12}}^{k_2}=1,\ y_{e_{13}}^{k_2}=1$
sn_n^k	$sn_2^{k_1}=1,\ sn_3^{k_1}=1,\ sn_4^{k_1}=1,\ sn_{15}^{k_1}=1,\ sn_{14}^{k_1}=1,\ sn_{12}^{k_1}=1$	$sn_{14}^{k_2}=1,\ sn_{12}^{k_2}=1,\ sn_6^{k_2}=1,\ sn_8^{k_2}=1,\ sn_{10}^{k_2}=1$
qs_i^k	$qs_1^{k_1}=2,\ qs_2^{k_1}=2,\ qs_3^{k_1}=5$	$qs_4^{k_2}=2,\ qs_5^{k_2}=2,\ qs_3^{k_2}=1$
u_i^k	$u_{i_1}^{k_1}<u_{i_2}^{k_1},\ u_{i_1}^{k_1}<u_{i_3}^{k_1}$	$u_{i_3}^{k_2}<u_{i_4}^{k_2},\ u_{i_4}^{k_2}<u_{i_5}^{k_2}$ or $u_{i_3}^{k_2}<u_{i_5}^{k_2},\ u_{i_5}^{k_2}<u_{i_4}^{k_2}$

$le_e \cdot y_e^k = 1\times(3+3+2+2+2+3+3+3+3+2+3+2+4) = 35$, $\sum_{k\in K} tc^k \cdot \sum_{i\in P\cup\{p+1\}} \sum_{j\in P} lc_{i,j} \cdot x_{i,j}^k = 1\times(3+4+3+3) = 13$, and $\sum_{k\in K}\sum_{n\in N} sc_n^k \cdot sn_n^k = 1\times 11 = 11$. Total profit is $210-(2+35+13+11) = 149$.

IV. SOLUTION APPROACH

Two Multi-Start Variable Neighborhood Descent (MS_VND and MS_VND') and seven neighborhood operators are proposed to solve the SDOPDPSTRP based on two categories of strategies: splitting demands before the calculation and splitting demands during the calculation.

A. NEIGHBORHOODS

Reference [53] used four operators (relocation, exchange, 2-opt and split-point re-positioning) for a Simultaneous Delivery and Pick up Vehicle Routing Problem with Split Loads (SDPVRPSL). Reference [48] proposed five operators for the VRPSPDP: intra-swap, intra-reverse, inter-reassignment, inter-swap and tail swap for the VRPSPDP. Reference [52] proposed 6 intra-route neighborhoods and 4 inter-route neighborhoods in randomized variable neighborhood descent for a SDOPDP.

The solution for a small instance was provided in the last part of Section III, FIGURE 1 shows the route structure of the solution, TABLE 3 shows the variables. Since the route structure of the SDOPDPSTRP is quite different from that of the classical OPDP, there are not as many pd-pairs that can be reinserted into a new route in the SDOPDPSTRP as in the classical OPDP. Therefore seven neighborhoods, which are modified from [26], are presented for the SDOPDPSTRP.

1) SPLIT

As in FIGURE 2, pd-pair i is selected randomly and inserted into a new route j_2 chosen according to the route structure feasibility strategy studied in [26]. If route j_2 is overloaded after being inserted, split pd-pair i and leave the overload part in route j_1.

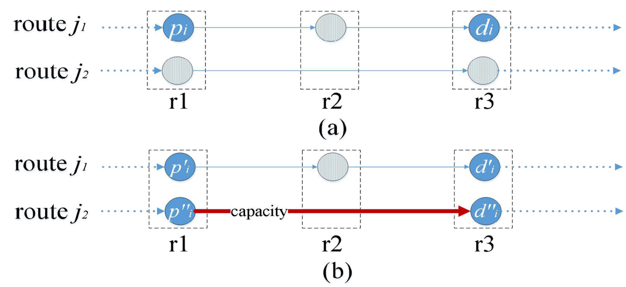


FIGURE 2. Construction methods for Split.

2) INSERT

As in FIGURE 3, pd-pair i is selected randomly and inserted into a new route j_2 chosen according to the route structure feasibility strategy.

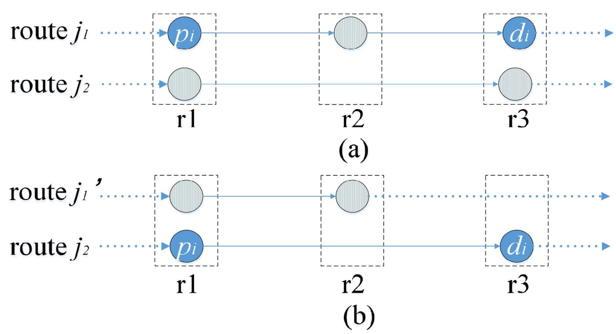


FIGURE 3. Construction methods for Insert.

3) SWAP

In the *Swap*, route i is chosen randomly and pd-pairs are inserted into the other routes from route i until the number of "insert" are more than the number of iterations K or the solution is improved or there is no pd-pair on route i. Then, we choose pd-pairs according to the route structure feasibility strategy, and insert them into route i until the number of "insert" are more than the numbers of iterations K or the solution is improved.

4) SPREAD

As in FIGURE 4, a pd-pair is selected and inserted into a new route j_2 as an *Insert* operation. If the vehicle is overloaded, the success rate can be improved by choosing a new pd-pair i from the route j_2 and transferring i into a new route j_3 selected according to the route structure feasibility strategy. This cycle will continue until the vehicle is no longer overloaded, or if the number cyclic k exceeds the preset number of iterations K. The task of the preset value K is to control the computing time of this operation.

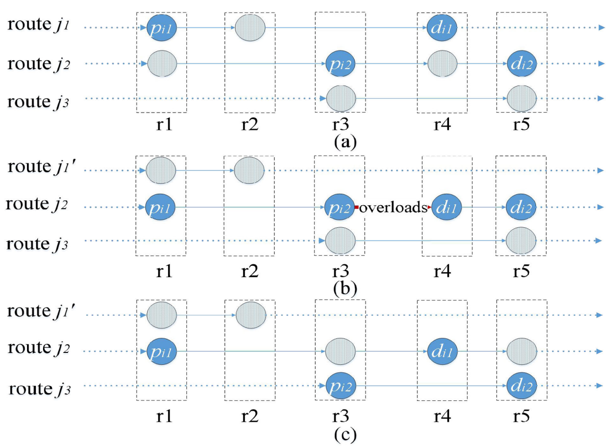


FIGURE 4. Construction methods for Spread.

5) POINT-DELETE

As in FIGURE 5, *Point-delete* starts by choosing a route at random. Then, the point with the minimal number of picking stops and delivery stops on the route is isolated, and these pd-pairs $i \in P$ are subsequently inserted into different routes selected according to the route structure feasibility strategy, thus making it possible to delete the point from the first route.

6) ROUTE-DELETE

In *Route-delete*, the net income ne_i of each route i is computed. Route k with the max negative income is selected. All pd-pairs transported by route k are changed to the state of being non-carried.

7) PERTURBATION

The key of the *Perturbation* is *Reassign-vehicle*, which is an Assignment Problem (AP). In *Reassign-vehicle*, vehicles are reassigned to routes to achieve the best scheme by the Gurobi solver in Matlab.

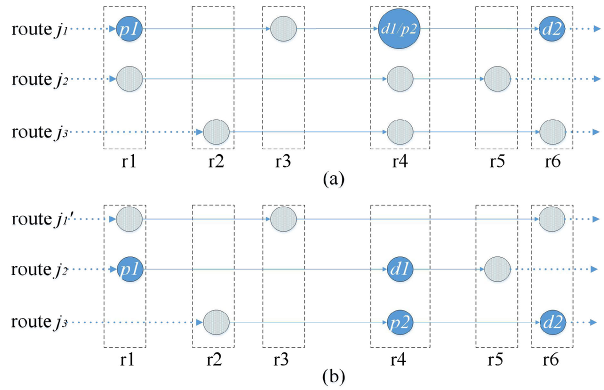


FIGURE 5. Construction methods for Point-delete.

Since *Reassign-vehicle* may make the most significant change to the solution but cannot always improve the solution and requires more CPU time, it is better to choose to perturb the local best solution according to a low probability. Therefore, a kind of *Perturbation* is proposed to shock the local best solution instead of conducting *Reassign-vehicle*. In *Perturbation*, *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete* and *Reassign-vehicle* are chosen according to the ***operator choosing probabilities*** p_1, p_2, p_3, p_4, p_5 and p_6 respectively.

B. APPROACHES WITH DIFFERENT DEMAND SPLITTING STRATEGIES

As in Table 4, 3 kinds of approaches with two different categories of splitting strategies are proposed to solve the SDOPDPSTRP in this article: (i) demands are split before the calculation, and then, the new problems is solved as the OPDPSTRP; (ii) demands are split during the calculation.

TABLE 4. Approaches with different demand splitting strategies.

Appraoches	Splitting strategies	Pre-splitting methods	Algorithms
Appraoch I	Split demands before the calculation	20/10/5/1/x	MS_VND in [26]
Appraoch II		25/10/5/1/x	
Appraoch III	Split demands during the calculation	-	MS_VND' in this paper

1) SPLIT DEMANDS BEFORE THE CALCULATION

(i) Pre-treatment of the demands

In the strategy that splits demands before the calculation, the demands are split according to the ratios of 20/10/5/1/x and 25/10/5/1/x before the calculation as in [48], [54].

20/10/5/1/x Splitting Strategy: Each demand is split into to 5 separate groups, each of which has a different quantity. The first four demands are set as $0.2Q$, $0.1Q$, $0.05Q$ and $0.01Q$. The fifth demand is set as the load of the quantity less than $0.01Q$. For example, if $Q = 300$ and $q = 205$, then we split q into q_1 =60, q_2 =60, q_3 =60, q_4 =15, q_5 =3, q_6 =3, q_7 =3 and q_8 =1.

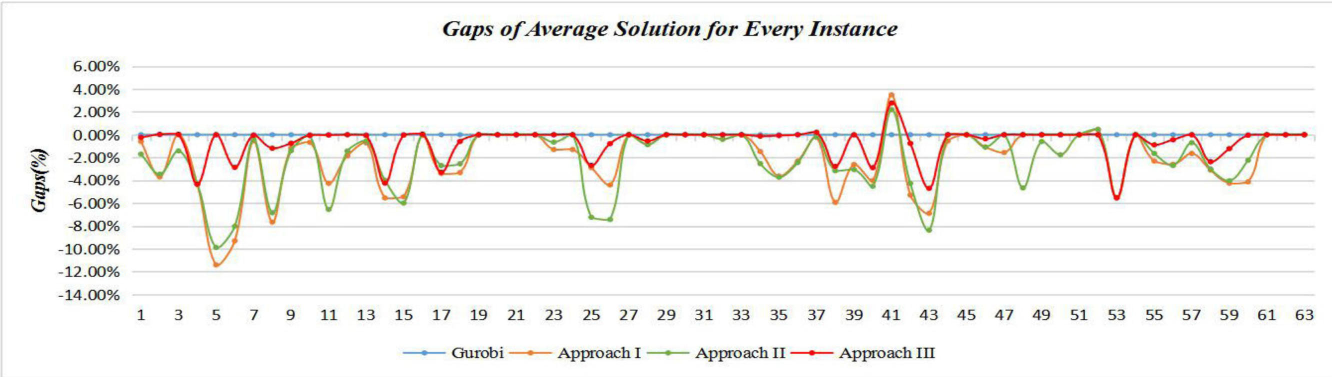


FIGURE 6. Gaps of the Average Solution for Every Instance.

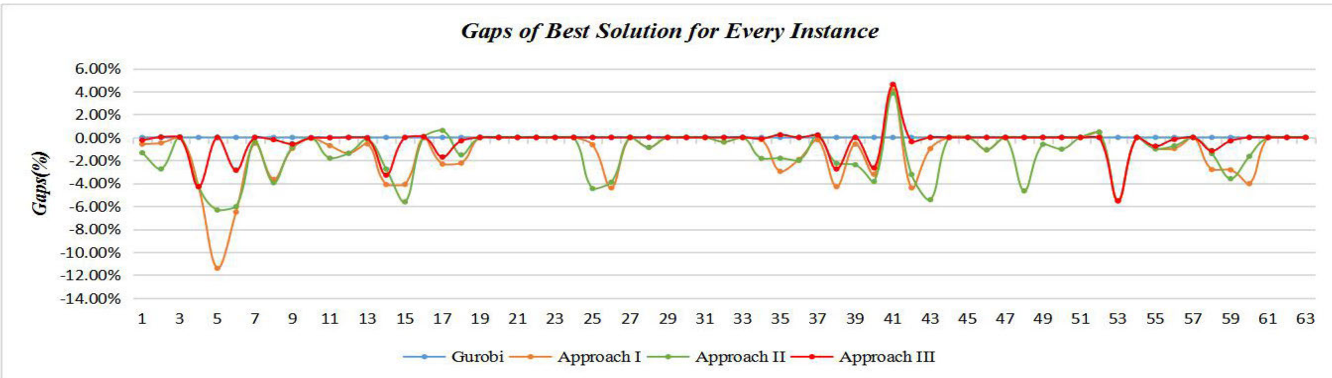


FIGURE 7. Gaps of the Best Solution for Every Instance.

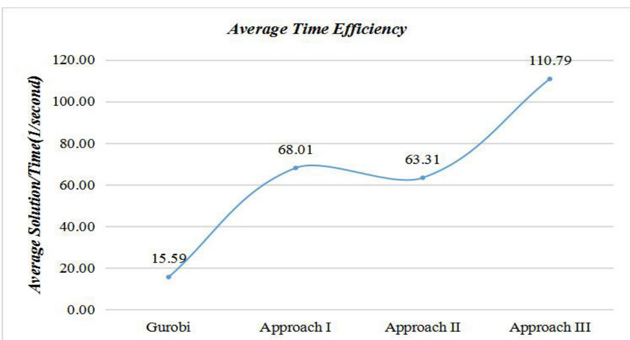


FIGURE 8. Average Time Efficiency.

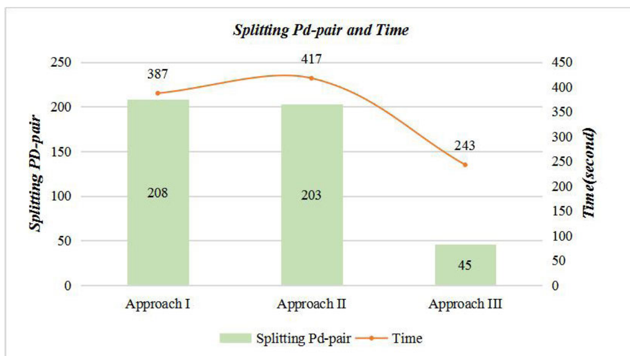


FIGURE 9. Splitting Pd-pairs and Time.

25/10/5/1/x Splitting Strategy: Each demand is split into to 5 separate groups, each of which has a different quantity. The first four demands are set as $0.25Q$, $0.1Q$, $0.05Q$ and $0.01Q$. The fifth demand is set as the load of the quantity less than $0.01Q$. For example, if $Q = 300$ and $q = 205$, then we split q into q_1 =75, q_2 =75, q_3 =30, q_4 =15, q_5 =3, q_6 =3, q_7 =3 and q_8 =1.

Then, the SDOPDPSTRP is converted to the OPDPSTRP and can be solved by a Multi-Start Variable Neighborhood Descent (MS_VND), which was proposed in [26] and shown as in Algorithm 2.

(ii) Generation steps of the initial solution

Generation steps of the initial solution are presented in Algorithm 1.

(iii) Algorithm

The steps of the MS_VND are presented in Algorithm 2, which were mentioned in [26].

To improve the search, the evaluation value of the algorithm is set as:

$$s = (z - z_1 \times M) \times (M/1000) + (z_0 - z_1 \times M) \quad (31)$$

Algorithm 1 Pseudo-Code of Generation Steps of Initial Solution

Input: let $P = \{1, \ldots, p\}$ is set of pd-pairs set; let routes set $R = \{R_i | R_i = pd\_pair\ i\}(i \in P)$ without vehicles; Input Multi-start candidate solution set size n; let i=0;
Reassign vehicles to routes by neighborhood *Reassign-vehicle* and solution s^0 is obtained;
let $S = \{s_i = s^0, i = 1, \ldots, n\}$ be Multi-Start candidate solution set;
for $i = 1 : n$
  select pd_pair j and route R_k randomly in solution s^0; try to insert pd-pair j into route R_k;
  if succeed
    $s_i \leftarrow s^0$; Update Multi-Start candidate solution set S;
  end if
end for

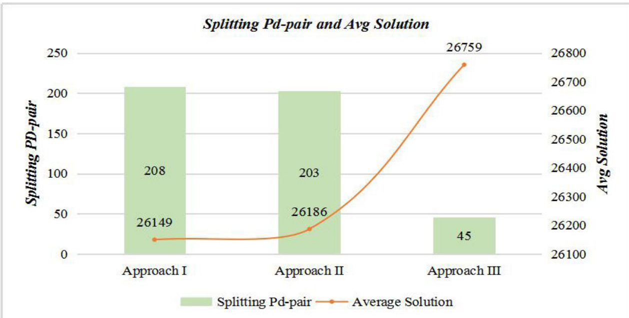


FIGURE 10. Splitting Pd-pair and Avg Solution.

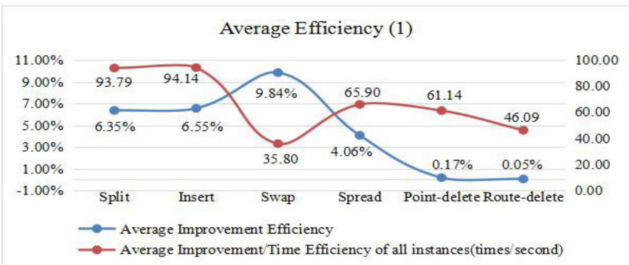


FIGURE 11. Average efficiency of the 6 operators for parameter p_k.

where

s: evaluation value of the algorithm,

z: objection value of the model,

z_1: The product of overloads and distances,

z_0: Total benefit of the schedule without assigning vehicles and

M: A large penalty.

2) SPLIT DEMANDS DURING THE CALCULATION

A new MS_VND, which is named MS_VND', is also proposed for the SDOPDPSTRP. In MS_VND', the demands are split during the calculation. To analyze the efficiency of these two splitting methods mentioned in this article, the generation steps of the initial solution are set as the same as Algorithm 1. The steps of MS_VND' for the SDOPDPSTRP are presented in Algorithm 3, which are revised from MS_VND, and two new operator *Split* and *Swap* are added in it. The evaluation value in Algorithm 3 is set as the same as in Algorithm 2.

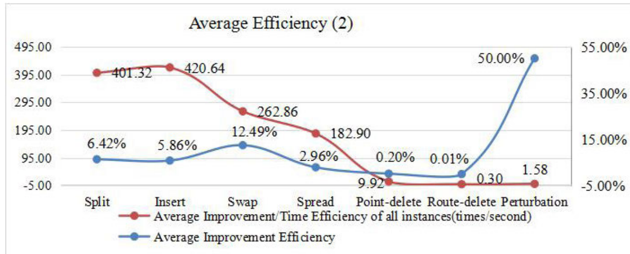


FIGURE 12. Average efficiency of the 7 operators for parameter p_k.

V. INSTANCES AND COMPUTATIONAL RESULTS

A. GENERATION OF INSTANCES

Sixty three instances are provided to test the methods for the SDOPDPSTRP, which are generated as follows.

Each instance name has the format *m0×m1-m2-m3-m4-m5-m6-m7*, where *m0×m1* is the size of a connected graph, the distance between any two node is randomly set as [0.5, 1.5], *1/m2* is the probability that each edge in this graph is deleted, *1/m3* is the pd-pair generation probability between every two nodes, *1/m4* is the vehicle generation percentage for each node, *m5* is ratio of the income/cost, the capacity is set as *m6* times the average demands, and the vehicles are randomly chosen as 1/*m7* times the demands. Additionally, [50] noted that when demands are from 51% to 60% of the capacity of the vehicle, up to 30% of the transportation costs can be saved. Therefore, *m6* is set as 2∼3, namely, the capacity is set as 2∼3 times of the average demands in this article.

Consider the instance 3 × 4.10-*10-2-1-2-2* as an example. The size of the incomplete digraph is 3 × 4 (12 nodes and 144 node-pairs), the distance between any two node is set as [0.5 1.5], the deletion probability of each edge is 1/10, the pd-pair generation probability between every two nodes is 1/10, the vehicle generation percentage for each node is 1/2, the ratio of income/cost is set as 1, the capacity is set as twice the average demands, and the number of vehicles is 1/2 the demand. It has been checked that there is only one shortest path between any two nodes in each graph. For each vehicle, the maximum travel distance limit is set as $D = (m0+1)\times 2$, and the maximum number of stops is $M = (m0+1)\times 2$.

Algorithm 2 Pseudo-Code of the ***MS_VND*** Meta-Heuristic

Input: let *bestsofar_s*=max$\{s_i\}$, let *Insert*, *Spread*, *Point-delete*, *Rout-delete* and *Perturbation* be operators *opt(k),* ($k = 1, 2, 3, 4$ and 5); input K for *Spread*, Selection controlling value T_0 and replacing proportion m, input p_1, p_2, p_3, p_4 and p_5 for *Perturbation*, Multi-start candidate solution set size n, algorithm termination iterations *constant_T*, *total_iteration*; Input Multi-Start candidate solution set S, let *constant*=0, *constant0*=0, *iteration*=0.
while *constant*<*constant_T* or *iteration*<*total_iteration*
  for every $S_i \in S$
    if $constant0 < T_0$
      $k = 1$;
    else if $T_0 \leq constant0 < T_0 * 2$
      $k = 2$;
    else if $T_0 * 2 \leq constant0 < T_0 * 3$
      $k = 3$;
    else if $T_0 * 3 \leq constant0 < T_0 * 4$
      $k = 4$;
    else if $T_0 * 4 \leq constant0 < T_0 * 5$
      $k = 5$;
    else
      $k = 6$;(*Perturbation*)
      if *Reassign-vehicle* is chosen in *Perturbation*
        constant 0=0;
      end if
    end if
    $s'_i \leftarrow opt(k, s_i)$;
    if s'_i is not inferior than s_i
      let $s_i = s'_i$; update S;
    else
      if *constant*>*constant_T/2* and $f(s'_i)$-$f(s_i) \geq$-$f(s_i)/2$
        let $s_i = s'_i$ according to the probability 50%; update S;
      end if
    end if
  end for
  find the local best solution *localbest_s* in S; *iteration*=*iteration*+1;
  if *localbest_s* is better than *bestsofar_s*
    let *bestsofar_s*=*localbest_s*; *constant*=*0*;
  else
    constant=*constant*+1;*constant 0*=*constant 0*+1;
  end if
  if *constant*≤*constant_T/2*
    replace the worst m solutions in S with *bestsofar_s*;
  end if
end while

The demand of each pd-pair is randomly set as [10, 20], namely, the average demand q is set as 15.

All the data of the 63 instances can been found in APPENDIX A.

B. COMPUTING ENVIRONMENT

All experiments were conducted on a desktop equipped with an Intel(R) Core(TM) i7-4510U 2.00 GHz processor and 8 GB of RAM. The operating system of this PC was 64-bit Windows 8. The new integer linear programming (ILP) model was solved using the Gurobi solver 7.5.2, which was embedded into Matlab R2015a by the Yalmip toolbox. All the algorithms in this article were also programmed in Matlab.

C. PARAMETER SETTING

The ILP model is solved by the Gurobi solver with the termination conditions set for a computing time over 1000 seconds or the gap is less than 5%. The long preset time aims to ensure that the Gurobi solver can obtain at least one feasible solution served as a comparison indicator with the proposed approaches, although in some cases it failed to

Algorithm 3 Pseudo-Code of the MS_VND' Meta-Heuristic

Input: let *bestsofar_s*=max$\{s_i\}$, let *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete* and *Perturbation* be operators *opt(k)*, ($k = 1, 2, 3, 4, 5, 6$ and 7); input K for *Spread*, Selection controlling value T_0 and replacing proportion m, input $p_1, p_2, p_3, p_4, p_5, p_7$ and p_8 for *Perturbation*, Multi-start candidate solution set size n, algorithm termination iterations *constant_T*, *total_iteration*; Input Multi-Start candidate solution set S, let *constant*=0, *constant0*=0, *iteration*=0.
while *constant*<*constant_T* or *iteration*<*total_iteration*
  for every $s_i \in S$
    if $constant0 < T_0$
      $k = 1$;
    else if $T_0 \leq constant0 < T_0 * 2$
      $k = 2$;
    else if $T_0 * 2 \leq constant0 < T_0 * 3$
      $k = 3$;
    else if $T_0 * 3 \leq constant0 < T_0 * 4$
      $k = 4$;
    else if $T_0 * 4 \leq constant0 < T_0 * 5$
      $k = 5$;
    else if $T_0 * 5 \leq constant0 < T_0 * 6$
      $k = 6$;
    else
      $k = 7$;(*Perturbation*)
      if *Reassign-vehicle* is chosen in *Perturbation*
        constant0=0;
      end if
    end if
    $s_i' \leftarrow opt(k, s_i)$;
    if s_i' is not inferior than s_i
      let $s_i = s_i'$; update S;
    else
      if *constant*>*constant_T*/2 and $f(s_i') - f(s_i) \geq -f(s_i)/2$
        let $s_i = s_i'$ according to the probability 50%; update S;
      end if
    end if
  end for
  find the local best solution *localbest_s* in S; *iteration*=*iteration*+1;
  if *localbest_s* is better than *bestsofar_s*
    let *bestsofar_s*=*localbest_s*; *constant=0*;
  else
    constant=*constant*+1;*constant0*=*constant0*+1;
  end if
  if *constant*≤*constant_T*/2
    replace the worst m solutions in S with *bestsofar_s*;
  end if
end while

achieve this goal. We provided the upper bounds found by the Gurobi solver as well a more in-depth reference to evaluate the performances of the proposed approaches: Approach I, Approach II and Approach III.

Almost all of numbers of the pd-pairs of the above 63 instances are not more than 230 after being pretreated. Therefore, the parameters values (n, *constant_T*, *total_iteration*, *T0*, K and m) of MS_VND can be set the same as in [26], where the numbers of pd-pairs are not more than 236. Then, the parameters values (n, *constant_T*, *total_iteration*, *T0*, K and m) of MS_VND' are set the same as for MS_VND to better compare the efficiency of these two splitting strategies. Furthermore, the operator sequence *opt(k)* and the operator choosing probability p_k in MS_VND' are reanalyzed in APPENDIX B because two new operators, *Split* and *Swap*, are added to MS_VND'.

TABLE 5. Parameter setting for MS_VND and MS_VND'.

Symbol	Definition	Value
n	Size of the Multi-Start candidate solution set	90
constant_T	Algorithm termination condition 1: limit on the number of iterations where the solution does not change	*constant_T=exp(-20/(2+num_pd-pairs))*700*, where *num_pd-pairs* is the number of pd-pairs
total_iteration	Algorithm termination condition 2: limit on the total number of iterations	*num_pd-pairs*100*, where *num_pd-pairs* is the number of pd-pairs
T_0	Selected controlling value	20
K	Number of iterations for ***Spread***	3
opt(k)	Operators sequences	MS_VND: *Insert*, *Spread*, *Point-delete*, *Rout-delete*, and *Perturbation* MS_VND': *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete*, and *Perturbation*
p_k	Operator choosing probabilities in ***Perturbation***	MS_VND: 9/24, 7/24, 1/24, 1/24, 6/24 for *Insert*, *Spread*, *Point-delete*, *Rout-delete*, and *Reassign-vehicle*, respectively MS_VND': 6/35, 7/35, 10/35, 4/35, 1/35, 1/35, 6/35 for *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete*, and *Reassign-vehicle*, respectively
m	Replacement proportion for Multi-Start solution set	1/8

TABLE 6. Abbreviation of the experiment indicators and definitions.

Abbreviation	Definition
UB	The upper bound of the ILP model obtained by the Gurobi solver in a preset running time.
LB	The best feasible objective value found by the Gurobi solver in a preset running time.
Gap	The gap between ***UB*** and ***LB***: (***UB-LB***)/***UB***.
LB_Avg	The average feasible objective value obtained by Approach I, Approach II and Approach III after a preset number of iterations.
LB_Best	The best feasible objective value obtained by Approach I, Approach II and Approach III after a preset number of iterations.
Gap_Avg	The gap between ***LB_Avg*** and ***LB***: (***LB_Avg-LB***)/***LB***.
Gap_Best	The gap between ***LB_Best*** and ***LB***: (***LB_Best-LB***)/***LB***.
Time	Average CPU time for solving the ILP model by Approach I, Approach II and Approach III (second).
Initial pd-pairs	Number of initial pd-pairs of each instance
Splitting pd-pairs	Number of pd-pairs of each instance in the solution

TABLE 7. Computational results for small size graphs.

Instances	*Initial pd-pairs*	*Vehicles*	*Gurobi*				*Approach I (pre-split demands according to 20/10/5/1/x)*						*Approach II (pre-split demands according to 25/10/5/1/x)*						*Approach III (split demands during calculation)*					
			UB	*LB*	*Gap*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*
3-4-10-10-1-1-2-2	12	6	2147	2146	0.05%	1	2133	2133	-0.59%	-0.59%	73	62	2109	2117	-1.70%	-1.34%	72	59	2141	2141	-0.24%	-0.24%	13	35
3-4-10-10-1-1-2-4	12	3	1464	1464	0.00%	1	1410	1457	-3.71%	-0.48%	73	92	1413	1424	-3.46%	-2.75%	72	87	1464	1464	0.03%	0.03%	13	46
3-4-10-10-1-1-3-4	12	3	1479	1479	0.00%	1	1479	1479	0.03%	0.03%	61	55	1458	1479	-1.42%	0.03%	55	52	1479	1479	0.03%	0.03%	12	32
3-4-10-10-1-4-2-2	13	7	13535	13535	0.00%	29	12947	12953	-4.34%	-4.30%	63	45	12949	12953	-4.33%	-4.30%	60	40	12947	12953	-4.34%	-4.30%	15	25
3-4-10-10-1-4-2-4	13	3	7858	7858	0.00%	1	6961	6961	-11.41%	-11.41%	63	95	7081	7361	-9.88%	-6.33%	60	96	7858	7858	0.00%	0.00%	15	36
3-4-10-10-1-4-3-4	13	3	8857	8857	0.00%	3	8033	8280	-9.30%	-6.51%	59	69	8146	8322	-8.02%	-6.04%	60	71	8604	8604	-2.86%	-2.86%	16	40
3-4-10-5-1-1-2-2	25	13	5526	5445	1.49%	474	5415	5417	-0.54%	-0.51%	115	136	5423	5424	-0.40%	-0.38%	110	132	5442	5445	-0.05%	0.00%	27	130
3-4-10-5-1-1-2-4	25	8	5016	4864	3.13%	1184	4491	4686	-7.66%	-3.65%	115	160	4531	4672	-6.84%	-3.95%	110	155	4806	4854	-1.19%	-0.20%	30	146
3-4-10-5-1-1-3-4	25	8	5205	5172	0.64%	246	5112	5122	-1.16%	-0.96%	125	128	5099	5128	-1.40%	-0.85%	130	128	5133	5142	-0.75%	-0.57%	25	94
3-4-10-5-1-4-2-2	23	12	22500	22418	0.37%	549	22268	22399	-0.67%	-0.08%	125	118	22395	22408	-0.10%	-0.04%	117	121	22407	22409	-0.05%	-0.04%	25	92
3-4-10-5-1-4-2-4	23	6	16706	16706	0.00%	150	15995	16585	-4.26%	-0.72%	125	243	15612	16403	-6.55%	-1.81%	117	253	16702	16702	-0.03%	-0.03%	27	168
3-4-10-5-1-4-3-4	23	6	18062	18062	0.00%	10	17728	17812	-1.85%	-1.38%	114	110	17803	17812	-1.43%	-1.38%	110	121	18062	18062	0.00%	0.00%	24	151
3-4-10-3-1-1-2-2	42	21	9888	9613	2.86%	1628	9543	9559	-0.72%	-0.56%	211	334	9559	9598	-0.56%	-0.15%	211	321	9606	9608	-0.07%	-0.05%	45	261
3-4-10-3-1-1-2-4	42	11	9742	8609	13.16%	1081	8132	8255	-5.54%	-4.11%	211	516	8266	8372	-3.99%	-2.75%	211	644	8244	8326	-4.24%	-3.29%	46	152
3-4-10-3-1-1-3-4	42	11	9831	9493	3.56%	1263	8977	9104	-5.44%	-4.10%	221	362	8925	8960	-5.99%	-5.62%	219	338	9490	9493	-0.03%	0.00%	45	214
3-4-10-3-1-4-2-2	42	21	51138	50703	0.86%	1140	50672	50686	-0.06%	-0.03%	220	327	50709	50730	0.01%	0.05%	220	373	50730	50736	0.05%	0.07%	49	163
3-4-10-3-1-4-2-4	42	11	48026	38918	23.40%	1278	37601	38015	-3.39%	-2.32%	220	1037	37866	39155	-2.70%	0.61%	220	1251	37638	38251	-3.29%	-1.71%	51	301
3-4-10-3-1-4-3-4	42	11	49446	46541	6.24%	1658	44995	45498	-3.32%	-2.24%	222	589	45356	45833	-2.55%	-1.52%	223	653	46267	46406	-0.59%	-0.29%	46	182
3-4-10-1-1-1-2-2	132	66	-	-	-	-	31262	31262	-	-	661	2098	31301	31301	-	-	640	2102	31476	31501	-	-	153	1523
3-4-10-1-1-1-2-4	132	33	-	-	-	-	25619	25719	-	-	661	1580	25496	25785	-	-	640	1552	26973	27145	-	-	179	1432
3-4-10-1-1-1-3-4	132	33	-	-	-		30784	31172	-	-	633	1833	30922	31073	-	-	639	1843	31764	31843	-	-	152	1645
Avg	41	14	15913	15105	3.10%	594	16741	16884	-3.55%	-2.44%	208	476	16782	16967	-3.41%	-2.14%	205	495	17106	17163	-0.98%	-0.75%	48	327

Note: The indexes in this table are introduced in TABLE VI.

The parameters values of MS_VND and MS_VND' are given in TABLE 5.

D. TEST RESULTS

The abbreviations of the experimental indicators and corresponding definitions are listed in TABLE 6.

The results found by Gurobi solver, Approach I, Approach II and Approach III are shown in TABLE 7 (small size graphs), TABLE 8 (medium size graphs) and TABLE 9 (large size graphs). Each instance has been solved 10 times by each method.

According to the results, the following can be found:

1) Approach III can always obtain better solutions than Approach I and Approach II with the same algorithm termination conditions. FIGURE 6 and FIGURE 7

TABLE 8. Computational results for medium size graphs.

Instances	Initial pd-pairs	Vehicles	Gurobi				Approach I (pre-split demands according to 20/10/5/1/x)						Approach II (pre-split demands according to 25/10/5/1/x)						Approach III (split demands during calculation)					
			UB	*LB*	*Gap*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*
6-8-10-200-1-1-2-2	14	7	5460	5460	0.00%	2	5460	5460	0.00%	0.00%	70	46	5460	5460	0.00%	0.00%	67	45	5460	5460	0.00%	0.00%	14	45
6-8-10-200-1-1-2-4	14	4	3561	3561	0.00%	1	3515	3561	-1.30%	0.00%	70	32	3538	3561	-0.65%	0.00%	67	30	3561	3561	0.00%	0.00%	14	25
6-8-10-200-1-1-3-4	14	4	3561	3561	0.00%	1	3515	3561	-1.30%	0.00%	77	30	3561	3561	0.00%	0.00%	70	31	3561	3561	0.00%	0.00%	14	23
6-8-10-200-1-4-2-2	15	8	30282	30282	0.00%	12	29407	30087	-2.89%	-0.64%	76	41	28090	28933	-7.24%	-4.46%	69	48	29469	30282	-2.69%	0.00%	17	40
6-8-10-200-1-4-2-4	15	4	18283	18283	0.00%	1	17479	17479	-4.40%	-4.40%	76	36	16925	17570	-7.43%	-3.90%	69	37	18141	18283	-0.78%	0.00%	17	37
6-8-10-200-1-4-3-4	15	4	18857	18857	0.00%	1	18857	18857	0.00%	0.00%	74	19	18857	18857	0.00%	0.00%	68	18	18857	18857	0.00%	0.00%	15	39
6-8-10-100-1-1-2-2	19	10	6672	6672	0.00%	2	6614	6614	-0.87%	-0.87%	90	24	6614	6614	-0.87%	-0.87%	85	24	6635	6672	-0.55%	0.00%	19	59
6-8-10-100-1-1-2-4	19	5	3965	3965	0.00%	1	3965	3965	0.00%	0.00%	90	31	3965	3965	0.00%	0.00%	85	33	3965	3965	0.00%	0.00%	19	47
6-8-10-100-1-1-3-4	19	5	3965	3965	0.00%	1	3965	3965	0.00%	0.00%	89	34	3965	3965	0.00%	0.00%	82	29	3965	3965	0.00%	0.00%	19	50
6-8-10-100-1-4-2-2	20	10	34875	34875	0.00%	2	34875	34875	0.00%	0.00%	102	28	34875	34875	0.00%	0.00%	101	29	34875	34875	0.00%	0.00%	21	51
6-8-10-100-1-4-2-4	20	5	20316	20316	0.00%	1	20237	20237	-0.39%	-0.39%	102	29	20237	20237	-0.39%	-0.39%	101	32	20316	20316	0.00%	0.00%	21	58
6-8-10-100-1-4-3-4	20	5	20367	20367	0.00%	1	20367	20367	0.00%	0.00%	99	31	20367	20367	0.00%	0.00%	93	32	20367	20367	0.00%	0.00%	20	57
6-8-10-50-1-1-2-2	42	21	16264	16252	0.07%	530	16013	16221	-1.47%	-0.19%	188	143	15841	15954	-2.53%	-1.83%	188	114	16232	16235	-0.13%	-0.11%	43	101
6-8-10-50-1-1-2-4	42	11	11316	10885	3.96%	641	10490	10562	-3.63%	-2.96%	188	220	10479	10688	-3.73%	-1.81%	188	230	10876	10910	-0.08%	0.23%	47	105
6-8-10-50-1-1-3-4	42	11	11171	11171	0.00%	537	10913	10956	-2.31%	-1.92%	214	225	10899	10946	-2.44%	-2.01%	199	205	11171	11171	0.00%	0.00%	48	107
6-8-10-50-1-4-2-2	44	22	75888	73049	3.89%	1131	72886	72891	-0.22%	-0.22%	213	134	72911	73189	-0.19%	0.19%	206	88	73200	73200	0.21%	0.21%	45	122
6-8-10-50-1-4-2-4	44	11	49186	46551	5.66%	1825	43791	44550	-5.93%	-4.30%	213	235	45082	45503	-3.16%	-2.25%	206	238	45261	45271	-2.77%	-2.75%	45	80
6-8-10-50-1-4-3-4	44	11	49658	49658	0.00%	171	48363	49370	-2.61%	-0.58%	222	127	48137	48476	-3.06%	-2.38%	220	125	49652	49658	-0.01%	0.00%	46	97
6-8-10-25-1-1-2-2	94	47	49846	47906	4.05%	21600	45977	46367	-4.03%	-3.21%	484	776	45742	46068	-4.52%	-3.84%	459	781	46519	46636	-2.89%	-2.65%	100	315
6-8-10-25-1-1-2-4	94	24	36694	29201	25.66%	19963	30220	30399	3.49%	4.10%	484	864	29842	30329	2.20%	3.86%	459	777	30014	30551	2.78%	4.62%	110	508
6-8-10-25-1-1-3-4	94	24	36704	35785	2.57%	19980	33893	34210	-5.29%	-4.40%	471	793	34258	34628	-4.27%	-3.23%	468	822	35513	35652	-0.76%	-0.37%	96	213
Avg	35	12	24138	23363	2.18%	3162	22895	23074	-1.58%	-0.95%	176	186	22840	23035	-1.98%	-1.21%	169	194	23220	23307	-0.40%	-0.04%	38	104

Note: The indexes in this table are introduced in TABLE VI.

TABLE 9. Computational results for large size graphs.

Instances	Initial pd-pairs	Vehicles	Gurobi				Approach I (pre-split demands according to 20/10/5/1/x)						Approach II (pre-split demands according to 25/10/5/1/x)						Approach III (split demands during calculation)					
			UB	*LB*	*Gap*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*	*LB_Avg*	*LB_Best*	*Gap_Avg*	*Gap_Best*	*Splitting pd-pairs*	*Time (second)*
10-10-10-1000-1-1-2-2	9	5	6095	6095	0.00%	3	5675	6035	-6.89%	-0.98%	48	10	5585	5764	-8.37%	-5.43%	46	9	5808	6095	-4.70%	0.00%	10	27
10-10-10-1000-1-1-2-4	9	2	3648	3648	0.00%	1	3628	3648	-0.55%	0.00%	48	9	3648	3648	0.00%	0.00%	46	9	3648	3648	0.00%	0.00%	10	26
10-10-10-1000-1-1-3-4	9	2	3768	3768	0.00%	1	3768	3768	0.00%	0.00%	50	9	3768	3768	0.00%	0.00%	50	8	3768	3768	0.00%	0.00%	9	17
10-10-10-1000-1-4-2-2	11	6	28290	28290	0.00%	2	27985	27985	-1.08%	-1.08%	53	17	27985	27985	-1.08%	-1.08%	55	18	28188	28290	-0.36%	0.00%	11	29
10-10-10-1000-1-4-2-4	11	3	16380	16380	0.00%	2	16125	16380	-1.56%	0.00%	53	23	16380	16380	0.00%	0.00%	55	20	16380	16380	0.00%	0.00%	11	31
10-10-10-1000-1-4-3-4	11	3	16380	16380	0.00%	2	16380	16380	0.00%	0.00%	53	24	15616	15616	-4.66%	-4.66%	48	18	16380	16380	0.00%	0.00%	11	34
10-10-10-500-1-1-2-2	20	10	11204	11204	0.00%	20	11204	11204	0.00%	0.00%	108	42	11137	11137	-0.60%	-0.60%	100	42	11204	11204	0.00%	0.00%	21	61
10-10-10-500-1-1-2-4	20	5	6678	6678	0.00%	3	6678	6678	0.00%	0.00%	108	59	6560	6610	-1.76%	-1.02%	100	48	6678	6678	0.00%	0.00%	21	146
10-10-10-500-1-1-3-4	20	5	6948	6948	0.00%	1	6948	6948	0.00%	0.00%	104	73	6948	6948	0.00%	0.00%	106	63	6948	6948	0.00%	0.00%	20	132
10-10-10-500-1-4-2-2	13	7	34551	34551	0.00%	2	34708	34708	0.46%	0.46%	62	10	34708	34708	0.46%	0.46%	61	10	34551	34551	0.00%	0.00%	14	47
10-10-10-500-1-4-2-4	13	3	18266	18266	0.00%	2	17256	17256	-5.53%	-5.53%	62	15	17256	17256	-5.53%	-5.53%	61	15	17256	17256	-5.53%	-5.53%	13	33
10-10-10-500-1-4-3-4	13	3	18424	18424	0.00%	2	18424	18424	0.00%	0.00%	63	14	18424	18424	0.00%	0.00%	62	13	18424	18424	0.00%	0.00%	13	38
10-10-10-200-1-1-2-2	44	22	27930	27577	1.28%	1459	26940	27309	-2.31%	-0.97%	235	207	27124	27301	-1.64%	-1.00%	232	211	27331	27368	-0.89%	-0.76%	46	185
10-10-10-200-1-1-2-4	44	11	17615	17615	0.00%	936	17159	17443	-2.59%	-0.97%	235	333	17140	17485	-2.70%	-0.74%	232	356	17538	17585	-0.44%	-0.17%	47	244
10-10-10-200-1-1-3-4	44	11	17782	17782	0.00%	577	17491	17782	-1.64%	0.00%	218	271	17658	17782	-0.70%	0.00%	216	249	17782	17782	0.00%	0.00%	44	165
10-10-10-200-1-4-2-2	49	25	135576	133053	1.90%	1051	128923	129329	-3.10%	-2.80%	239	276	129052	131179	-3.01%	-1.41%	236	281	129898	131508	-2.37%	-1.16%	51	227
10-10-10-200-1-4-2-4	49	12	84652	79640	6.29%	3206	76257	77384	-4.25%	-2.83%	239	328	76419	76777	-4.04%	-3.60%	236	330	78671	79401	-1.22%	-0.30%	53	376
10-10-10-200-1-4-3-4	49	12	85988	84092	2.25%	3559	80616	80702	-4.13%	-4.03%	251	340	82205	82707	-2.24%	-1.65%	250	383	84046	84092	-0.05%	0.00%	50	245
10-10-10-50-1-1-2-2	188	94	-	-	-	-	127327	127865	-	-	939	2175	126702	128064	-	-	908	2135	129291	129447	-	-	191	1252
10-10-10-50-1-1-2-4	188	47	-	-	-	-	81355	81455	-	-	939	3351	82838	83892	-	-	908	2670	87290	88099	-	-	213	1856
10-10-10-50-1-1-3-4	188	47	-	-	-	-	90209	90443	-	-	931	2902	90529	90958	-	-	923	2720	97892	98486	-	-	189	1076
Avg	48	16	30010	29466	0.65%	602	38812	39006	-1.84%	-1.04%	240	499	38937	39257	-1.89%	-1.41%	235	563	39951	40161	-0.86%	-0.44%	50	297

Note: The indexes in this table are introduced in TABLE VI.

TABLE 10. Parameters setting of *OPT(K)* and P_K for the MS_VND'.

Symbol	Definition	Value
opt(k)	Operators sequences	MS_VND': *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete*, and *Perturbation*
p_k	Operator choosing probabilities in ***Perturbation***	MS_VND': 6/35, 7/35, 10/35, 4/35, 1/35, 1/35, 6/35 for *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Rout-delete*, and *Reassign-vehicle*, respectively

show the gaps between the ***LB*** and ***LB_Avg*** and ***LB_Best*** solved by Approach I, Approach III and Approach III. The ***Gap_Avg*** and ***Gap_Best*** of the Approach III are the lowest.

2) Approach III outperforms Approach I and Approach II in terms of time efficiency. As shown in FIGURE 8, the average time efficiency (Average Solution/Time) of Approach III is better than that of Approach I and Approach II.
3) There is a strong link between the number of splitting pd-pairs and the time for solving the SDOPDPSTRP. As in FIGURE 9, Approach I and Approach II take more time than that of Approach III to solve the SDOPDPSTRP with the same algorithm termination conditions, because they split more pd-pairs than Approach III.

(iv) In theory, the fewer the demands that are split, the better the solutions that can be obtained. However, this phenomenon is not immutable. As shown in FIGURE 10, the average number of splitting pd-pairs is 45 in Approach III, but Approach III outperforms Approach I and Approach II in terms of solution quality with more than 500 on average. Namely, ''how to split'' is worthy of further study in the future.

VI. CONCLUSION

A new Pickup and Delivery Problem with a new route structure and split demand, the Split Demand One-to-one

Pickup and Delivery Problems with the Shortest-path Transport along Real-life Paths (SDOPDPSTRP), which is proposed using connected graphs, is introduced and formulated in a new way. Three methods are proposed to solve the SDOPDPSTRP: two methods (Approach I and Approach II) split demands before the calculation, and the other method (Approach III) splits demands during the calculation.

The results show that Approach III outperform Approach I and Approach II in terms of the average solutions and time efficiency under the same algorithm termination conditions, which has great practical significance for real-life transport organizations. It is also be found that "how to split" is the key to solving the SDOPDPSTRP and that the time it takes to solve the SDOPDPSTRP is closely related to the number of split pd-pairs, especially for large instances. Therefore, the splitting methods for large instances are worthy of further study in the future.

APPENDIX

A. INSTANCES

The relative data for the instances can be found online at the following link:

https://www.researchgate.net/publication/343415212_Instances_of_the_SDOPDPSTRP.

B. PARAMETERS SETTING OF opt(K) AND p_K FOR THE MD_VND'

The parameter setting, such as the operator sequence *opt(k)* and the operator choosing probability p_k in *Perturbation* will be revised in this section because two new operators, *Split* and *Swap*, are proposed in the MD_VND'.

These two parameters have also been tested over 9 instances (including small size connected graphs, medium size connected graphs and large size connected graphs, chosen from the 64 instances in APPENDIX A), as in the parameter setting is tuned by determining the trade off between the solution quality and CPU time after numerous experiments.

FIGURE 11 shows the performance of the six operators (*Split*, *Insert*, *Swap*, *Spread*, *Point-delete* and *Route-delete*) which are chosen separately in the MD_VND' (the other operators are removed).

According to FIGURE 11, the operator choosing ratios between neighborhoods is set as 6:7:10:4:1:1 (according to the average improvement efficiency) in the *Perturbation* for MD_VND'.

FIGURE 12 shows the performance of the seven operators in the MD_VND' algorithm with the empirical values given in TABLE 10. The results show that *Reassign-vehicle* will take a large amount of CPU time. Therefore, the sequence of choosing the operators is determined as *Split*, *Insert*, *Swap*, *Spread*, *Point-delete*, *Route-delete* and *Perturbation* for the MS_VND'.

For a better comparison, the final parameters are shown in the TABLE 10.

REFERENCES

[1] S. Parragh, K. Doerner, and R. Hartl, "A survey on pickup and delivery problems I," *J. Für Betriebswirtschaft*, vol. 58, pp. 21–51, Oct. 2008.

[2] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations," *J. Für Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, Jun. 2008.

[3] Ç. Koç and G. Laporte, "Vehicle routing with backhauls: Review and research perspectives," *Comput. Oper. Res.*, vol. 91, pp. 79–91, Mar. 2018.

[4] G. Berbeglia, J. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems a classification scheme and survey," *Top*, vol. 15, pp. 1–31, Apr. 2007.

[5] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 8–15, Apr. 2010.

[6] I. Gribkovskaia and G. Laporte, "One-to-many-to-one single vehicle pickup and delivery problems," *Oper. Res. Comput. Sci. Interface*, vol. 43, pp. 359–377, 2008.

[7] Z. Zhu, J. Xiao, S. He, Z. Ji, and Y. Sun, "A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem," *Inf. Sci.*, vol. 329, pp. 73–89, Feb. 2016.

[8] H. N. Psaraftis, "An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows," *Transp. Sci.*, vol. 17, no. 3, pp. 351–357, Aug. 1983.

[9] H. N. Psaraftis, "Analysis of an O(N_2) heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem," *Transp. Res. B, Methodol.*, vol. 17, no. 2, pp. 133–145, Apr. 1983.

[10] T. R. Sexton and L. D. Bodin, "Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling," *Transp. Sci.*, vol. 19, no. 4, pp. 378–410, Nov. 1985.

[11] T. R. Sexton and L. D. Bodin, "Optimizing single vehicle many-to-many operations with desired delivery times: II. Routing," *Transp. Sci.*, vol. 19, no. 4, pp. 411–435, Nov. 1985.

[12] R. Camargo, G. Miranda, and A. Løkketangen, "A new formulation and an exact approach for the many-to-many hub location-routing problem," *Appl. Math. Model.*, vol. 37, pp. 7465–7480, May 2013.

[13] J. Rieck, C. Ehrenberg, and J. Zimmermann, "Many-to-many location-routing with inter-hub transport and multi-commodity pickup-and-delivery," *Eur. J. Oper. Res.*, vol. 236, no. 3, pp. 863–878, Aug. 2014.

[14] H. Hernández-Pérez and J.-J. Salazar-González, "The multi-commodity one-to-one pickup-and-delivery traveling salesman problem," *Eur. J. Oper. Res.*, vol. 196, no. 3, pp. 987–995, Aug. 2009.

[15] M. Åahin, G. Çavuälar, T. Öncan, G. Åahin, and D. T. Aksu, "An efficient heuristic for the multi-vehicle One-to-one pickup and delivery problem with split loads," *Transp. Res. C, Emerg. Technol.*, vol. 27, pp. 169–188, Feb. 2013.

[16] M. Soysal, M. Çimen, and E. Demir, "On the mathematical modeling of green one-to-one pickup and delivery problem with road segmentation," *J. Cleaner Prod.*, vol. 174, pp. 1664–1678, Feb. 2018.

[17] B. Fleischmann, "A cutting plane procedure for the traveling salesman problem on a road network," *Eur. J. Oper. Res.*, vol. 21, pp. 307–317, Apr. 1985.

[18] G. Cornuéjols, J. Fonlupt, and D. Naddef, "The traveling salesman problem on a graph and some related integer polyhedra," *Math. Program.*, vol. 33, no. 1, pp. 1–27, Sep. 1985.

[19] H. Longo, M. P. de Aragáo, and E. Uchoa, "Solving capacitated arc routing problems using a transformation to the CVRP," *Comput. Oper. Res.*, vol. 33, no. 6, pp. 1823–1837, Jun. 2006.

[20] E. Demir and T. Van Woensel, "Mathematical modeling of CO2e emissions in one-to-one pickup and delivery problems," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage.*, Dec. 2013, Art. no. 14762859.

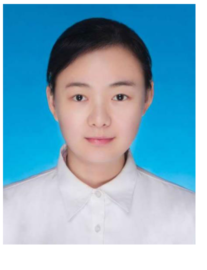
[21] A. N. Letchford, S. D. Nasiri, and A. Oukil, "Pricing routines for vehicle routing with time windows on road networks," *Comput. Oper. Res.*, vol. 51, pp. 331–337, Nov. 2014.

[22] X. Li, J. Ma, J. Cui, A. Ghiasi, and F. Zhou, "Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model," *Transp. Res. B, Methodol.*, vol. 88, pp. 21–45, Jun. 2016.

[23] M. Mahmoudi and X. Zhou, "Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations," *Transp. Res. B, Methodol.*, vol. 89, pp. 19–42, Jul. 2016.

[24] S. C. Ho, W. Y. Szeto, Y.-H. Kuo, J. M. Y. Leung, M. Petering, and T. W. H. Tou, "A survey of dial-a-ride problems: Literature review and recent developments," *Transp. Res. B, Methodol.*, vol. 111, pp. 395–421, May 2018.

[25] P. Factorovich, I. Méndez-Díaz, and P. Zabala, "Pickup and delivery problem with incompatibility constraints," *Comput. Oper. Res.*, vol. 113, Jan. 2020, Art. no. 104805.
[26] X. Qi, Z. Fu, J. Xiong, and W. Zha, "Multi-start heuristic approaches for one-to-one pickup and delivery problems with shortest-path transport along real-life paths," *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0227702.
[27] M. Furuhata, M. Dessouky, and O. Fernando, "Ride-sharing: The state-of-the-art and future directions," *Transp. Res. B, Methodol.*, vol. 57, pp. 28–46, Oct. 2013.
[28] S. Muelas, A. LaTorre, and J.-M. Peña, "A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 110–130, May 2015.
[29] Y. Lin, W. Li, F. Qiu, and H. Xu, "Research on optimization of vehicle routing problem for ride-sharing taxi," *Procedia—Social Behav. Sci.*, vol. 43, pp. 494–502, 2012.
[30] Y. Wang, B. Zheng, and E.-P. Lim, "Understanding the effects of taxi ride-sharing — A case study of singapore," *Comput., Environ. Urban Syst.*, vol. 69, pp. 124–132, May 2018.
[31] J. Desrosiers, Y. Dumas, and F. Soumis, "A dynamic programming solution of the large-scale single-vehicle Dial-A-Ride problem with time windows," *Amer. J. Math. Manage. Sci.*, vol. 6, nos. 3–4, pp. 301–325, Feb. 1986.
[32] B. Li, D. Krushinsky, H. A. Reijers, and T. Van Woensel, "The Share-a-Ride problem: People and parcels sharing taxis," *Eur. J. Oper. Res.*, vol. 238, no. 1, pp. 31–40, Oct. 2014.
[33] E. Urra, C. Cubillos, and D. Cabrera-Paniagua, "A hyperheuristic for the Dial-a-Ride problem with time windows," *Math. Problems Eng.*, vol. 2015, Oct. 2015, Art. no. 707056.
[34] X. Qiu, S. Feuerriegel, and D. Neumann, "Making the most of fleets: A profit-maximizing multi-vehicle pickup and delivery selection problem," *Eur. J. Oper. Res.*, vol. 000, pp. 1–14, Feb. 2016.
[35] H. Hosni, J. Naoum-Sawaya, and H. Artail, "The shared-taxi problem: Formulation and solution methods," *Transp. Res. B, Methodol.*, vol. 70, pp. 303–318, Dec. 2014.
[36] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1782–1795, Jul. 2015.
[37] J. N. Sawaya, R. Cogill, B. Ghaddar, S. Sajja, R. Shorten, N. Taheri, P. Tommasi, R. Verago, and W. F. 2015, "Stochastic optimization approach for the car placement problem in ride sharing systems," *Transp. Res. B, Methodol.*, vol. 80, pp. 173–184.
[38] D. O. Santos and E. C. Xavier, "Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6728–6737, Nov. 2015.
[39] P. Detti, F. Papalini, and G. Z. M. D. Lara, "A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare," *Omega*, vol. 70, pp. 1–14, Jul. 2017.
[40] M. Dror and P. Trudeau, "Savings by split delivery routing," *Transp. Sci.*, vol. 23, no. 3, pp. 141–145, 1989.
[41] 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.
[42] S. Mitra, "An algorithm for the generalized vehicle routing problem with backhauling," *Asia–Pacific J. Oper. Res.*, vol. 22, no. 02, pp. 153–169, Jun. 2005.
[43] S. Mitra, "A parallel clustering technique for the vehicle routing problem with split deliveries and pickups," *J. Oper. Res. Soc.*, vol. 59, no. 11, pp. 1532–1546, Nov. 2008.
[44] J.-J. Salazar-González and B. Santos-Hernández, "The split-demand one-commodity pickup-and-delivery travelling salesman problem," *Transp. Res. B, Methodol.*, vol. 75, pp. 58–73, May 2015.
[45] H. Hernández-Pérez, J. J. Salazar-González, and B. Santos-Hernández, "Heuristic algorithm for the split-demand one-commodity Pickup-and-Delivery travelling salesman problem," *Comput. Oper. Res.*, vol. 97, pp. 1–17, Sep. 2018.
[46] M. Casazza, A. Ceselli, and R. Wolfler Calvo, "A branch and price approach for the split pickup and split delivery VRP," *Electron. Notes Discrete Math.*, vol. 69, pp. 189–196, Aug. 2018.
[47] M. Casazza, A. Ceselli, and R. Wolfler Calvo, "A route decomposition approach for the single commodity split pickup and split delivery vehicle routing problem," *Eur. J. Oper. Res.*, vol. 5, pp. 1–7, Jul. 2019, doi: 10.1016/j.ejor.2019.07.015.
[48] M. Qiu, Z. Fu, R. Eglese, and Q. Tang, "A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups," *Comput. Oper. Res.*, vol. 100, pp. 102–116, Dec. 2018.
[49] M. Nowak, Ö. Ergun, and C. C. White, "Pickup and delivery with split loads," *Transp. Sci.*, vol. 42, no. 1, pp. 32–43, Feb. 2008.
[50] M. Nowak, O. Ergun, and C. C. White, "An empirical study on the benefit of split loads with the pickup and delivery problem," *Eur. J. Oper. Res.*, vol. 198, no. 3, pp. 734–740, Nov. 2009.
[51] Q. Chen, K. Li, and Z. Liu, "Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads," *Transp. Res. E, Logistics Transp. Rev.*, vol. 69, pp. 218–235, Sep. 2014.
[52] 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.
[53] C. Yin, L. Bu, and H. Gong, "Mathematical model and algorithm of split load vehicle routing problem with simultaneous delivery and pickup," *Int. J. Innov. Comput.*, vol. 9, no. 11, pp. 4497–4508, 2013.
[54] P. Chen, B. Golden, X. Wang, and E. Wasil, "A novel approach to solve the split delivery vehicle routing problem," *Int. Trans. Oper. Res.*, vol. 24, nos. 1–2, pp. 27–41, Jan. 2017.
[55] J. F. Cordeau, G. Laporte, and S. Ropke, "Recent models and algorithms for one-to-one pickup and delivery problems. The vehicle routing problem: Latest advances and new challenges," *Oper. Res./Comput. Sci. Interfaces*, vol. 43, pp. 327–357, Oct. 2008.
[56] G. Mattos Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," *Comput. Oper. Res.*, vol. 39, no. 3, pp. 728–735, Mar. 2012.
[57] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, Nov. 2006.
[58] P. Shaw, "A new local search algorithm providing high quality solutions to vehicle routing problems," Dept. Comput. Sci., Univ. Strathclyde, Scotland, U.K., Tech. Rep., 1997.
[59] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proc. Int. Conf. Princ. Pract.*, 1998, pp. 417–431.
[60] J. Potvin and J. Rousseau, "A parallel route building algorithm for the vehicle routing and schedulingproblem with time windows," *Eur. J. Oper. Res.*, vol. 66, p. 331, Oct. 1993.
[61] A. Grimault, N. Bostel, and F. Lehuédé, "An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization," *Comput. Oper. Res.*, vol. 88, pp. 1–14, Dec. 2017.
[62] I. Rodríguez-Martín and J. Salazar-González, "A hybrid heuristic approach for the multi-commodity one-to-one pickup-and-delivery traveling salesman problem," *J. Heuristics*, vol. 18, pp. 849–867, Feb. 2012.
[63] J. Li, P. M. Pardalos, H. Sun, J. Pei, and Y. Zhang, "Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3551–3561, May 2015.
[64] Y. Li, H. Chen, and C. Prins, "Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests," *Eur. J. Oper. Res.*, vol. 252, no. 1, pp. 27–38, Jul. 2016.
[65] F. Liu, M. Gui, C. Yi, and Y. Lan, "A fast decomposition and reconstruction framework for the pickup and delivery problem with time windows and LIFO loading," *IEEE Access*, vol. 7, pp. 71813–71826, 2019.
[66] B. Sun, Y. Yang, J. Shi, and L. Zheng, "Dynamic pick-up and delivery optimization with multiple dynamic events in real-world environment," *IEEE Access*, vol. 7, pp. 146209–146220, 2019.
[67] Y. Lin and L. Schrage, "The global solver in the LINDO API," *Optim. Methods Softw.*, vol. 24, pp. 4–5, pp. 657–668, 2009.



JIAN XIONG received the B.S. degree in transportation management and the M.S. degree in logistics engineering from Central South University, in 2004 and 2007, respectively, where she is currently pursuing the Ph.D. degree in transportation planning and management. Her research interests include logistics engineering, railway transportation, and vehicle routing problem.

XIN QI received the B.S. degree in transportation management from Central South University, in 2004, and the M.S. degree in transportation management from East China Jiao Tong University, in 2009. He is currently pursuing the Ph.D. degree in transportation planning and management with Central South University. He is also an Associate Professor with East China Jiao Tong University. His research interests include railway transportation organization and vehicle routing problem.

ZHUO FU received the B.S., M.S., and Ph.D. degrees in transportation management from Central South University, in 1982, 1985, and 2004, respectively. He is currently a Professor with Central South University. His research interests include railway transportation operation management optimization; logistics distribution optimization theory and application; and vehicle routing problem.

WEIXIONG ZHA received the B.S. degree in mathematics from Wuhan University, in 1983, the M.S. degree in mathematics from Central South University, in 1987, and the Ph.D. degree in transportation management from Beijing Jiaotong University, in 1998. He is currently a Professor with East China Jiao Tong University. His research interests include railway transportation operation management optimization and transportation economy.