Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A two-stage heuristic method for vehicle routing problem with split deliveries and pickups*

Yong WANG[†1], Xiao-lei MA[†2], Yun-teng LAO[3], Hai-yan YU[1], Yong LIU[1]

([1]*School of Management, Chongqing Jiaotong University, Chongqing 400074, China*)

([2]*School of Transportation Science and Engineering, Beihang University, Beijing 100191, China*)

([3]*Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195-2700, USA*)

[†]E-mail: yongwx6@gmail.com; xiaolei@buaa.edu.cn

Received July 3, 2013; Revision accepted Dec. 19, 2013; Crosschecked Feb. 19, 2014

Abstract: The vehicle routing problem (VRP) is a well-known combinatorial optimization issue in transportation and logistics network systems. There exist several limitations associated with the traditional VRP. Releasing the restricted conditions of traditional VRP has become a research focus in the past few decades. The vehicle routing problem with split deliveries and pickups (VRPSPDP) is particularly proposed to release the constraints on the visiting times per customer and vehicle capacity, that is, to allow the deliveries and pickups for each customer to be simultaneously split more than once. Few studies have focused on the VRPSPDP problem. In this paper we propose a two-stage heuristic method integrating the initial heuristic algorithm and hybrid heuristic algorithm to study the VRPSPDP problem. To validate the proposed algorithm, Solomon benchmark datasets and extended Solomon benchmark datasets were modified to compare with three other popular algorithms. A total of 18 datasets were used to evaluate the effectiveness of the proposed method. The computational results indicated that the proposed algorithm is superior to these three algorithms for VRPSPDP in terms of total travel cost and average loading rate.

Key words: Vehicle routing problem with split deliveries and pickups (VRPSPDP), Two-stage heuristic method, Hybrid heuristic algorithm, Solomon benchmark datasets

doi:10.1631/jzus.C1300177 **Document code:** A **CLC number:** TP3; TU121

1 Introduction

With the rapid development of urban economy, logistics companies are seeking an effective and efficient solution to manage and optimize the flow of resources in order to satisfy customers' demands (Wang *et al.*, 2012). Vehicle routing optimization is considered one critical countermeasure to reduce cost and improve service quality for logistics operators (Chepuri and Homem-de-Mello, 2005; Mitra, 2005). The classic vehicle routing problem (VRP) is a combinatorial optimization procedure (Baldacci *et al.*, 2010; Wang *et al.*, 2013), and can be described as: Several vehicles depart from the depot, serve a series of customers, and return to the same depot. The objective of VRP is to determine the optimal set of routes with minimized various costs (e.g., total travel distance and the number of vehicles) (Chen and Wu, 2006; Chen *et al.*, 2006; Ai and Kachitvichyanukul, 2009; de Oliveira and Vasconcelos, 2010).

Traditional VRP requires that the vehicle routes should start and end at the same depot, each customer should be visited only once, and the demand of each customer should not exceed the vehicle capacity. These requirements restrict the widespread use of VRP in reality. Therefore, relaxing VRP's restrictions is desired and has drawn more attention from practitioners as well as researchers. Several variants of traditional VRP have emerged in the past decades.

* Project supported by the National Natural Science Foundation of China (No. 51138003) and the National Social Science Foundation of Chongqing of China (No. 2013YBJJ035)

The split delivery vehicle routing problem (SDVRP) was proposed by Dror and Trudeau (1989) to remove the second and third restrictions by allowing multiple visits for each customer. Under this circumstance, the strategy to split deliveries should be adopted. Recently, SDVRP has gradually attracted many researchers' attention. Lee *et al*. (2006) established a deterministic dynamic program formulation with finite state and action spaces, and presented a new exact algorithm based on the shortest path searching procedure for SDVRP. Jin *et al*. (2007) presented a two-stage algorithm with valid inequalities. This algorithm can be used to solve the SDVRP. Archetti *et al*. (2008) found that splitting the deliveries is the most beneficial when customers' average demands are more than half of the vehicle capacity and little demand variance exists. Moreno *et al*. (2010) proposed an algorithm to obtain the lower bounds for the SDVRP. Gulczynski *et al*. (2010) developed a heuristic approach to solve the SDVRP and reported computational results on a wide range of problem sets. Li *et al*. (2011) formulated a non-linear programming model to coordinate split deliveries in multi-retailer distribution systems. Their algorithm aimed to reduce the inventory costs of total retailers without increasing travel costs.

With the advent of reverse logistics, logistics companies began to shift their attention to collect the unsold goods or returned items from the point of consumption (e.g., customers or supermarkets) to the point of origin (e.g., manufacturers or distributors) in a cost-efficient manner (Sheu, 2008; Lambert *et al*., 2011). The vehicle routing problem with simultaneous delivery and pickup (VRPSDP) was raised to tackle such scenarios. In this case, customers request delivery and pick-up service concurrently since they may prefer to be served fewer times. Logistics companies can also receive cost reduction due to the simultaneous delivery and pick-up operations. Many researchers have put their efforts into seeking the optimal solution to VRPSDP in these years. Montané and Galvão (2006) developed a tabu algorithm to solve VRPSDP. They used a 2-opt procedure to gain alternative within-route solutions and three types of movement to obtain a between-route adjacent solution. Ai and Kachitvichyanukul (2009) presented a generalized formulation of VRPSDP and proposed a solution method based on the particle swarm optimization algorithm to address the VRPSDP. Gajpal and Abad (2009) solved the VRPSDP through an ant colony system (ACS) algorithm with a construction rule as well as two multi-route local search schemes. Hoff *et al*. (2009) developed a tabu search heuristic algorithm to produce lasso solutions to the VRPSDP. Çatay (2010) first proposed an ant colony algorithm (ACO) approach with a new saving-based visibility function and pheromone update procedure for the VRPSDP. Subramanian *et al*. (2011) developed a branch-and-cut algorithm with lazy separation. They tested the algorithm in a large logistics network with up to 200 customers, and the results demonstrated that their approach is able to improve most of the previously known lower bounds.

Combining SDVRP and VRPSDP triggers a new research topic, the vehicle routing problem with split deliveries and pickups (VRPSPDP). The VRPSPDP problem releases the capacity constraint of the traditional VRP problem that customers' total demands cannot exceed the vehicle capacity. Split deliveries can be beneficial even in cases where customer demand does not exceed vehicle capacity. Nowak *et al*. (2009) made an empirical study on the benefit of split loads with the delivery and pickup problem, where each load is associated with a pickup and a delivery location, and found that load sizes just over one half of vehicle capacity can be split. However, they investigated only the impact of different variables on the benefit of split loads with the pick and delivery problem, and did not provide any systematic approaches to solving the VRPSPDP problem. In addition, Hennig *et al*. (2012) described the split delivery and pickup problem of the maritime oil tanker based on realistic transport operations, and introduced a path flow model to solve this problem.

To the best of our knowledge, in the literature there is no suitable solution to the VRPSPDP in the logistics network. To fill this gap, a two-stage method is proposed to solve the VRPSPDP for minimizing the total travel cost. The first step is to seek the initial solution using clustering operations, and then the initial solution is further improved by a hybrid heuristic algorithm. To evaluate the effectiveness of the proposed method, Solomon datasets and extended Solomon datasets (Solomon, 1987; Saberi and Verbas, 2012; Wang and Chen, 2012) are modified and used to compare with several other solution approaches.

2 Related notations for VRPSPDP

Notations and parameters are defined in this section. We consider n+1 locations, 0, 1, …, n, where 1, 2, …, n represent n customers and 0 represents the depot. The minimum number of vehicles is used as an input to the formulation, which is calculated as follows:

$$\left\lceil \frac{\text{max(total delivery demand, total pickup demand)}}{\text{vehicle capacity}} \right\rceil,$$

where total delivery demand and pickup demand refer to the cumulative delivery demand and cumulative pickup demand at the customers respectively, and $\lceil x \rceil$ denotes the smallest integer that is equal to or greater than x. The total required number of vehicles throughout the entire algorithm is assumed to be the minimum number of vehicles. Although the actual number of vehicles may exceed the minimum number of vehicles in reality, to reduce the complexity of the algorithm, the actual number of vehicles is constant in this study.

The notations used in the following two-stage heuristic method are listed as follows:

J is the set of all customer locations, J={1, 2, ..., n}, $\forall i, j \in J$ and $i \neq j$; N_0 is the set of all locations, $N_0=J\cup\{0\}$, in which location 0 is the depot; m is the total number of vehicles; V is the set of vehicles, $V=\{k|k=1, 2, ..., m\}$, $k\in V$; Q is the vehicle capacity; D_j is the delivery demand at customer j; R_j is the pickup demand at customer j; d_{ij} is the travel cost (travel distance) from customer i to customer j. To simplify the VRPSPDP problem, $d_{ij}=d_{ji}$, $d_{00}=0$.

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels directly from} \\ & \text{customer } i \text{ to customer } j, i \neq j, \\ 0, & \text{otherwise,} \end{cases}$$

y_{ijk} is the quantity of customer j's goods delivered from vehicle k using the route from customer i to customer j, and z_{ijk} is the quantity of customer j's goods picked up from vehicle k using the route from customer i to customer j.

In the computation, the travel cost is minimized as the objective function, which is equivalent to minimizing $\sum_{k\in V}\sum_{i\in N_0}\sum_{j\in N_0} d_{ij}x_{ijk}$.

3 Heuristic algorithms

Our approach is based on a two-step process. First, we compute an initial feasible solution based on the clustering procedure and initial route construction procedure. Next, a hybrid heuristic algorithm is proposed to further improve the initial solution.

3.1 Heuristic algorithm for an initial solution

The clustering procedure and initial route construction procedure are illustrated in this subsection. Several notations are explained as follows: C is the set of clusters, $C=\{C_k|k=1, 2, …, m\}$; j' is the spatially closest unvisited customer from customer i; N' is the set of unvisited customers; DQ_k is the delivery demand for vehicle k (k=1, 2, …, m); PQ_k is the pickup demand for vehicle k (k=1, 2, …, m); DL is the quantity of delivery goods carried by each vehicle; PL is quantity of pickup goods carried by each vehicle.

In the heuristic algorithm, C is used during the clustering procedure. The essence of the algorithm is described as follows: in the initial route construction procedure, when vehicle k cannot accommodate the quantity of delivery/pickup goods of customer j' and the remaining quantity of delivery/pickup goods from vehicle k can partially satisfy the requirement of customer j', the remaining demand for customer j' can be further fulfilled either by vehicle k during its returning route or by the other vehicles on the next route. Customer j' can be chosen in N'. DL and PL are constantly changing, and we can assume that vehicle k leaves the depot with delivery demand DL and no pickup demand. DL is reduced and PL is increased after vehicle k traverses a series of customers in the set of customers. When vehicle k finally returns to the depot, it should carry a certain amount of PL and no delivery demand. In this case, total delivery and pickup demands of all customers are dynamic, and are updated as the requirement of each customer is satisfied or partially satisfied.

The heuristic algorithm for an initial solution is composed of two major steps, clustering procedure and initial route construction procedure, which are further expanded below.

3.1.1 Clustering procedure

A clustering procedure should be undertaken prior to the initial route construction procedure. The number of clusters is equal to the minimum number of vehicles given in Section 2, and each cluster can be served by one vehicle. The maximum total demand can be obtained by comparing total delivery demand and total pickup demand, and then the minimum number of vehicles can be calculated by using the maximum total demand divided by the capacity of each vehicle. Therefore, the minimum number of vehicles is a fixed value. A clustering procedure based on previous research (Sheu, 2006; Mitra, 2008; Özdamar and Demir, 2012; Wang *et al*., 2014) is designed as follows:

Step 1: For each cluster, we can set the first and last elements as the depots in each cluster.

Step 2: The second element is chosen in the first cluster C_1 as the customer who is away from the depot in the farthest distance. Then we can find the customer with the maximum distance from the second element in the first cluster. This new customer will be designated as the second element in the second cluster. Similarly, for the remaining clusters, the second element in each cluster C_t (t>2) can be chosen as the customer with the maximum sum of distances from all the second elements in the previous clusters.

Step 3: For the third element in each cluster, we can obtain the customer with the minimum sum of the distances from the existing assigned elements within this cluster.

Step 4: The remaining elements in each cluster can be extracted by following a process similar to step 3. The clustering procedure will continue until the cumulative delivery or pickup demands assigned to the cluster reach the vehicle capacity.

With the above steps, all customers can be assigned to the corresponding clusters. The customer clustering sets up a base for the initial route construction procedure.

3.1.2 Initial route construction procedure

The initial route construction procedure is used to adjust the order of customers within each cluster obtained from the above clustering procedure. The core of this procedure is to update each vehicle's remaining capacity, and construct a sequence of stops for each vehicle to fulfill each customer's delivery and pickup requirements. This procedure is detailed as follows:

Step 1: Set the number of iterations as k=1.

Step 2: Set j'=0, DL=DQ$_k$, PL=0, vehicle k starts from the depot, and the vehicles' remaining capacity (Q−DL−PL) is equal to or greater than zero.

Step 3: Set i=j', and select the unvisited customer j' that is the spatially closest to customer i (i.e., the minimum travel cost).

Step 4: Set DL=DL−$D_{j'}$, and $D_{j'}$=0.

If the pickup demand of customer j' is less than the vehicle's remaining capacity ($R_{j'}$<Q−DL−PL), then set PL=PL+$R_{j'}$, $R_{j'}$=0.

If the pickup demand of customer j' is more than the vehicle's remaining capacity ($R_{j'}$>Q−DL−PL), then set $R_{j'}$=$R_{j'}$−(Q−DL−PL), PL=Q−DL.

If the pickup demand of customer j' is equal to the vehicle's remaining capacity ($R_{j'}$=Q−DL−PL), then set PL=PL+$R_{j'}$, $R_{j'}$=0, Q−DL−PL=0.

Step 5: Insert customer j' into the current route k, compute the spare capacity (Q−DL−PL) of vehicle k, and return to step 3 until no more insertions are valid.

Step 6: Set k=k+1 and return to step 2, until all customers are served in clusters.

With the above heuristic algorithm, the initial routes can be obtained.

3.2 Hybrid heuristic algorithm for improving the initial solution

3.2.1 Local search operators

Local search methods have been proved effective in solving the VRPSDP (Subramanian *et al*., 2010; Zhang *et al*., 2012). Several operators from the local search method are 2-opt, 2-opt* exchange, or-opt, swap move, shift move (Chen and Wu, 2006; Subramanian *et al*., 2010), and two other operators are relocate and relocate split (Ho and Haugland, 2004). These operators are applied to find the solution to the VRPSPDP. Among these operators, the relocate operator, relocate split operator, 2-opt* exchange operator, and swap move operator are commonly used to solve the SDVRP, so we adopt these operators and further illustrate the details of each operator as follows:

1. Relocate operator (Fig. 1): Suppose that customer i is in route R_k, and customer j in route R_l . With the relocate operation, customer i can be removed from route R_k and inserted after customer j in route R_l. Then we can obtain the new routes denoted as R_k'=(0, ..., i−1, i+1, ..., 0) and R_l'=(0, ..., j, i, j+1, ..., 0).

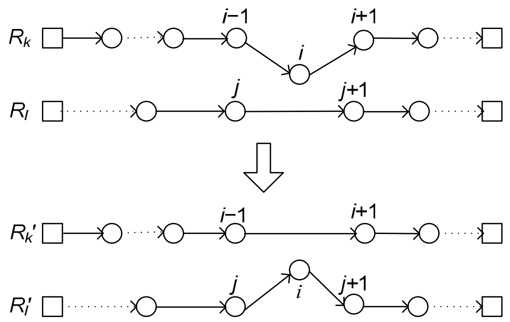

Fig. 1 A relocation procedure

2. Relocate split operator (Fig. 2): Suppose that customer i is split by routes R_k and R_l, i.e., customer $i \in R_k \cap R_l$. In addition, customer i belongs to route R_k and customer j belongs to route R_k. We can remove customer i from route R_k, denoted as R_k=(0, ..., i−1, i+1, ..., 0), and the demand of customer j can be split by routes R_k and R_l. To keep the demand unchanged, we can increase the amount of customer i's delivery and pickup demand in R_l by the same amount of customer i's delivery and pickup demand in R_k, and transfer the same amount of deliveries and pickups at customer j from R_l to R_k.

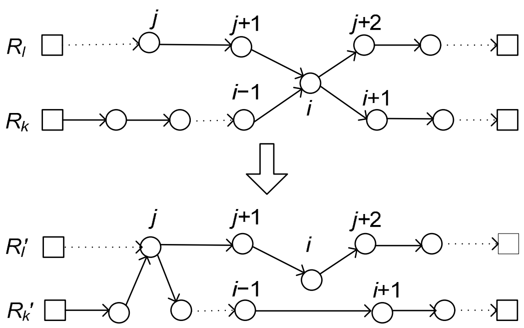

Fig. 2 A relocate split procedure

3. 2-opt* exchange operator (Fig. 3): The 2-opt* exchange operator is also known as the 2-opt* procedure (Mullaseril *et al.*, 1997; Chen and Wu, 2006; Liu *et al.*, 2013). If we denote customer $i \in R_k$ and customer $j \in R_l$, then routes R_k and R_l can be expressed as R_k=(0, ..., i, i+1, ..., 0) and R_l=(0, ..., j, j+1, ..., 0). The new routes become R_k'=(0, ..., i, j+1, j+2, ..., 0) and R_l'=(0, ..., j, i+1, i+2, ..., 0); with the 2-opt* exchange operator, a split delivery and pickup between R_k and R_l may be eliminated.

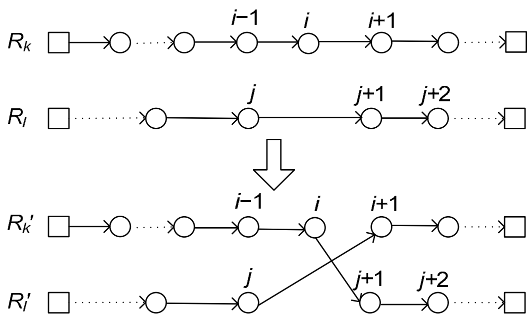

Fig. 3 2-opt* exchange procedure

4. Swap move operator (Fig. 4): The swap move operator is described as an interchange of customers between two different routes. In the procedure, the selected sequence is moved from one route to another or swapped between two routes with both split delivery and pickup. For example, we interchange customers $i \in R_k$, $h \in R_k$, and $j \in R_l$ between routes R_k and R_l. Customer i is inserted into route R_l and customer j is inserted to a certain position of route R_k. To balance the delivery and pickup quantity, the demand of customer h may be split by routes R_k and R_l. The (m, n) swap move operator refers to interchanging m customers and n customers between two different routes.

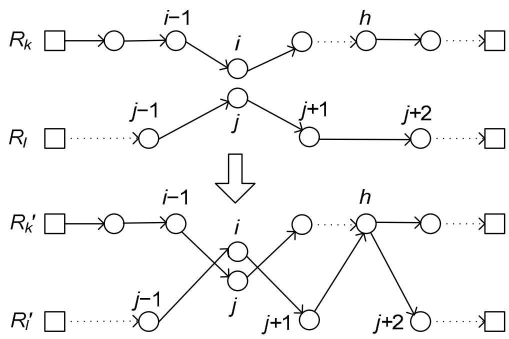

Fig. 4 Swap move procedure

3.2.2 Hybrid heuristic algorithm

Based on the above description of local search methods and two operators relevant to the split procedure, the hybrid heuristic algorithm can be detailed as follows:

Step 1: Calculate the initial solution from the initial routes, record the total travel cost, and set the iteration number as l'=1.

Step 2: Rearrange the initial routes into a random order from the initial solution in the initial routes. Each permutation contains a distinct sequence of routes, and the number of permutations is p.

Step 3: Select two consecutive routes in pairs starting from the first route for each permutation, and repeat the relocate procedure until no more improvement can be found over the best known solution within β consecutive iterations.

Step 4: Similar to step 3, repeat the relocate split procedure until no more improvement can be found over the best known solution within β consecutive iterations.

Step 5: Apply the 2-opt procedure within each route until no more improvement can be found over the best known solution within μ consecutive iterations.

Step 6: Apply the 2-opt* exchange procedure until no more improvement can be found over the best known solution within β consecutive iterations.

Step 7: Apply the or-opt procedure within each route until no more improvement can be found over the best known solution within μ consecutive iterations.

Step 8: Apply the (2, 0) shift move procedure until no more improvement can be found over the best known solution within β consecutive iterations.

Step 9: same as step 4.

Step 10: same as step 7.

Step 11: Apply the (1, 1), (2, 1), and (2, 2) swap move procedures recursively until no more improvement can be found over the best known solution within β consecutive iterations.

Step 12: same as step 4.

Step 13: same as step 6.

Step 14: Apply the 2-opt and or-opt procedures respectively until no more improvement can be found over the best known solution within μ consecutive iterations.

Step 15: Record the best total travel cost as F_1, set $l'=l'+1$, and determine whether l' reaches the maximum iteration number γ. If not, select the best routes set, and return to step 2; otherwise, the hybrid heuristic algorithm ends.

In the above hybrid heuristic algorithm, rearranging the initial routes into a particular order can diversify the candidate solutions, and then between-route operations (relocate, relocate split, 2-opt* exchange, shift move, and swap move) can be used to adjust the visited customer sequence between different routes, and these operations can be considered a coarse-turning process. Within-route operation (2-opt and or-opt) can be used to adjust the order of visited customers within each route, and these operations can be considered a fine-turning process. Both the coarse-turning process (between-route) and fine-turning process (within-route) are recursive until the best solution is found.

In addition, each of the relocate split operator, 2-opt* exchange operator, and swap move operator computes the newly updated travel cost after each iteration in the hybrid heuristic algorithm. If the updated travel cost is lower than the one in the previous iteration, the current route adjustment will be accepted; otherwise, the previous route will be kept. This procedure will continue until no 'better' travel cost can be reached. In this case, the feasibility of the routing solution can be preserved.

4 Computational results

The proposed two-stage method was applied to the revised Solomon benchmark and extended Solomon benchmark data (Montané and Galvão, 2006; Wang and Chen, 2012). We also implemented three recently developed algorithms: tabu search algorithm (TSA) (Montané and Galvão, 2006), particle swarm optimization (PSO) (Ai and Kachitvichyanukul, 2009), and parallel heuristic approach (PHA) (Subramanian *et al*., 2010). These three algorithms were modified with relocate split operators for fair comparison.

4.1 Modified Solomon dataset

The Solomon datasets and extended Solomon datasets (Solomon, 1987; Saberi and Verbas, 2012; Wang and Chen, 2012) have been widely used as a benchmark to compare a series of VRPs (Nanry and Barnes, 2000; Montané and Galvão, 2006; de Oliveira and Vasconcelos, 2010; Subramanian *et al*., 2010; Wang and Chen, 2012). The Solomon datasets and extended Solomon datasets are composed of six different problem types (C1, C2, R1, R2, RC1, RC2). These datasets contain different numbers of customers (Solomon datasets contain 100 customers, and extended Solomon datasets contain more than 100 customers). We chose 100-customer datasets in Solomon datasets, and 200- and 400-customer datasets in extended Solomon datasets to test the proposed algorithm.

Both Solomon datasets and extended Solomon datasets provide delivery demand in each sub-dataset. However, Solomon datasets and extended Solomon datasets cannot be directly used for the VRPSPDP, since they do not include pickup demand data. Necessary revisions are needed to test our algorithms.

To incorporate pickup demand, we can extract delivery demand from one dataset and merge it into another dataset to make a complete dataset for analysis. Table 1 shows how we combine two datasets into one dataset. For example, when delivery demand R101 is extracted as the pickup demand, and then merged into dataset C101, the new combined dataset CR101 is constructed.

Table 1 New datasets used for analysis

Number of customers	New dataset	Pick up demand	Delivery demand
100	CR101	R101	C101
	CR201	R201	C201
	R_C101	C101	R101
	R_C201	C201	R201
	RCR101	R101	RC101
	RCR201	R201	RC201
200	CR1_2_1	R1_2_1	C1_2_1
	CR2_2_1	R2_2_1	C2_2_1
	R_C1_2_1	C1_2_1	R1_2_1
	R_C2_2_1	C2_2_1	R2_2_1
	RCR1_2_1	R1_2_1	RC1_2_1
	RCR2_2_1	R2_2_1	RC2_2_1
400	CR1_4_1	R1_4_1	C1_4_1
	CR2_4_1	R2_4_1	C2_4_1
	R_C1_4_1	C1_4_1	R1_4_1
	R_C2_4_1	C2_4_1	R2_4_1
	RCR1_4_1	R1_4_1	RC1_4_1
	RCR2_4_1	R2_4_1	RC2_4_1

4.2 Parameter settings for heuristic algorithms

Several parameters need to be defined before the heuristic algorithms are executed. Parameters were chosen based on previous studies on the VRPSPDP (Nanry and Barnes, 2000; Chen and Wu, 2006; Rieck and Zimmermann, 2010; Subramanian *et al.*, 2010; Wang and Chen, 2012). The parameters are given as follows: p=100, 150, and 200 are the numbers of permutations used to increase the diversity of initial routes in 100-, 200-, and 400-customer problems, respectively; β=15 is the maximum number of iterations to obtain the best known solution without any improvement between two routes; μ=10 is the maximum number of iterations to obtain the best known solution without any improvement within routes; γ=1500 is the maximum number of iterations to terminate the hybrid heuristic algorithm.

In addition, we set y_{ijk}' as the remaining amount of the delivery goods from customer i to customer j for vehicle k, and the average loading rate (ALR) is calculated for comparison:

$$\mathrm{ALR}=\frac{1}{mQ\sum_{i\in N_0}\sum_{j\in N_0}x_{ijk}}\sum_{k=1}^{m}\sum_{i\in N_0}\sum_{j\in N_0}(y_{ijk}'+\sum_{i=0,i\neq j}^{n}z_{ijk}).$$

4.3 Results analysis and comparison

For comparison, we implemented and tested TSA, PSO, and PHA using Solomon benchmark datasets with 100 customers, extended Solomon benchmark datasets with 200 customers, and extended Solomon benchmark datasets with 400 customers, respectively. TSA, PSO, and PHA can be modified by adding the relocate split operator during their between-route movements. Each algorithm is further illustrated as follows:

TSA: The revised neighborhoods for the tabu search contain five types of movement, among which relocation, relocate split, interchange, and crossover are the between-route movements, and 2-opt exchange is the intra-route movement. We generate an initial solution using the independent grouping and routing (IGR) procedure, and then the intensification phase, diversification phase, standard phase, interactive phase, and re-initialization are implemented successively until the stopping rules are reached.

PSO: The initialized particles as a swarm can be generated with a random position, and relocate split is designed in the encoding procedure to optimize the initial solution. In the solution representation procedure, the dimensions in each particle represent the priorities of customers. Each customer can be represented by two or three dimensions, and the values in these dimensions can be converted in the decoding step.

PHA: The initial solution can be generated using the greedy procedure in the parallel heuristic approach, and the relocate split procedure can be included in between-route movements of local search to improve the initial solution. Then both perturbation mechanisms and parallel calculation processes in the diversification phase are performed until the optimal solution is achieved.

Adding the relocate split operator to each algorithm increases the diversity of solutions, thereby improving the performance of each algorithm.

The proposed algorithms along with the three algorithms were implemented on a laptop with Intel Core i5 Quad CPU and 8 GB RAM using the MATLAB programming language. The results are demonstrated in Tables 2–4 and Fig. 5. The measures of effectiveness include the total travel cost and average loading rate. We can find that:

1. The travel costs were significantly reduced using the proposed algorithms in all 18 scenarios, compared to the three algorithms. For example, in the scenario of R_C1_4_1, the travel cost of the proposed

Table 2 Comparison between the proposed algorithms and TSA, PSO, PHA on 100 customers

Dataset	Travel cost (mile)				Average loading rate			
	Proposed algorithms	TSA	PSO	PHA	Proposed algorithms	TSA	PSO	PHA
CR101	1152.43	1293.71	1342.30	1267.21	89%	81%	79%	82%
CR201	625.21	721.13	755.26	702.93	93%	83%	80%	85%
R_C101	987.75	1126.52	1191.33	1076.07	91%	75%	72%	78%
R_C201	659.36	733.27	759.02	702.40	90%	79%	76%	82%
RCR101	1013.12	1127.85	1185.81	1086.35	89%	80%	78%	83%
RCR201	636.51	716.43	749.39	701.16	92%	82%	81%	84%
Average	845.73	953.15	997.19	922.69	91%	80%	78%	82%

Table 3 Comparison between the proposed algorithms and TSA, PSO, PHA on 200 customers

Dataset	Travel cost (mile)				Average loading rate			
	Proposed algorithms	TSA	PSO	PHA	Proposed algorithms	TSA	PSO	PHA
CR1_2_1	3412.17	3843.22	3993.56	3819.25	93%	87%	85%	88%
CR2_2_1	1593.41	1771.03	1785.80	1734.03	94%	80%	79%	82%
R_C1_2_1	3127.06	3498.51	3543.72	3450.62	91%	78%	75%	80%
R_C2_2_1	1581.53	1702.30	1736.12	1672.01	90%	83%	81%	85%
RCR1_2_1	3129.19	3485.16	3546.38	3438.20	89%	81%	77%	83%
RCR2_2_1	1466.03	1573.87	1621.17	1551.35	93%	83%	79%	85%
Average	2384.90	2645.68	2704.46	2610.91	92%	82%	79%	84%

Table 4 Comparison between the proposed algorithms and TSA, PSO, PHA on 400 customers

Dataset	Travel cost (mile)				Average loading rate			
	Proposed algorithms	TSA	PSO	PHA	Proposed algorithms	TSA	PSO	PHA
CR1_4_1	10 527.82	12 450.07	13 028.49	11 930.25	92%	80%	76%	83%
CR2_4_1	3276.91	3739.22	3825.15	3683.67	88%	75%	72%	77%
R_C1_4_1	8545.37	10 912.06	11 043.33	10 786.05	91%	79%	77%	82%
R_C2_4_1	3162.43	3565.31	3699.76	3527.88	93%	80%	78%	81%
RCR1_4_1	8597.20	9583.19	9896.01	9332.61	91%	91%	75%	79%
RCR2_4_1	3104.51	3564.41	3671.89	3502.17	92%	79%	78%	80%
Average	6202.37	7302.38	7527.44	7127.11	91%	78%	76%	80%

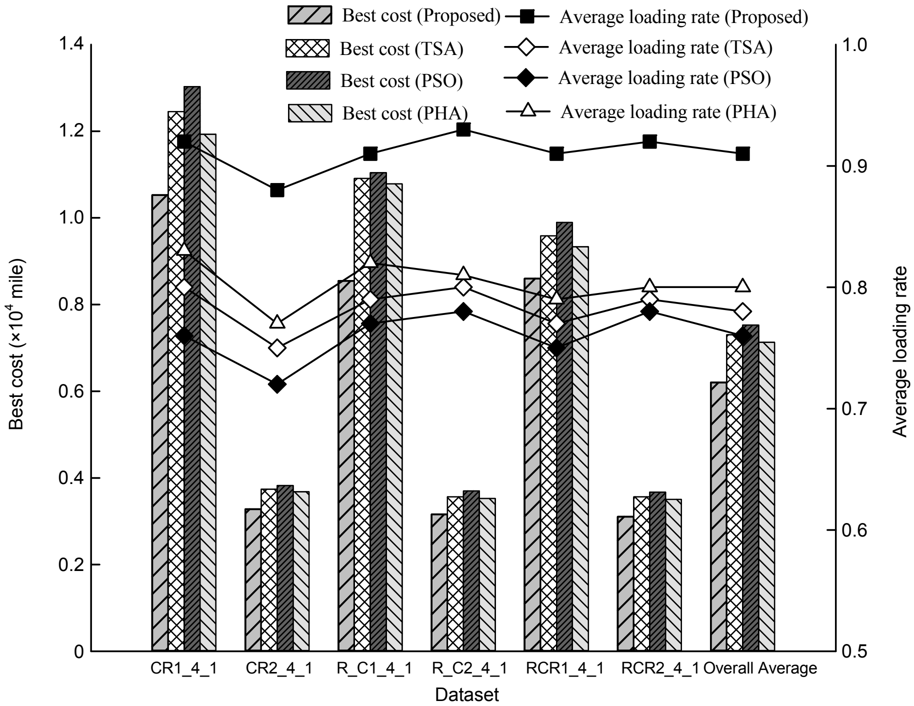

Fig. 5 Comparison between the proposed algorithms and TSA, PSO, PHA on 400 customers

algorithms decreased from 10786.05 to 8545.37 miles compared with the PHA method.

2. The average loading rate of the proposed algorithms was greater than those of the three algorithms in all 18 scenarios. For example, the average loading rate on 200-customer problems ranged from 89% to 94% for the proposed algorithms, and 75% to 85% for PSO.

3. The average travel costs were effectively reduced using the proposed algorithms. For example, the average travel cost of the proposed algorithms decreased from 7302.38 to 6202.37 miles compared with the TSA method (Table 4).

To demonstrate the algorithm efficiency, six datasets with 400 customers were used to test the execution time of each algorithm. The average running time for the proposed algorithms was 2255.992 s, whereas those for TSA, PSO, and PHA were 1796.616, 1995.817, and 1897.192 s, respectively. The proposed algorithms resulted in 26%, 13%, and 19% more CPU time compared with TSA, PSO, and PHA, respectively. This indicates that the proposed algorithms are less efficient than these three algorithms.

In summary, by using the proposed two-stage method, the total travel cost and average loading rate were significantly improved in different datasets with 100, 200, and 400 customers. The travel cost was reduced by more than 17.6%, and the average loading rate was increased by 15% for the dataset with 400 customers. This demonstrates that our proposed algorithm outperforms the other three algorithms in terms of travel cost and average loading rate. The computational complexity of our proposed algorithm, however, is higher than those of the three algorithms. With the emergence of clouding computing technology, the running time of the proposed algorithm can be foreseen not a critical issue in the near future.

The improvement of the proposed algorithms is owing to the following three major reasons:

1. More realistic factors are introduced in the problem formulation; for example, by splitting both delivery and pickup demands, each customer is allowed to be served more than once. In this situation, the vehicle is still able to satisfy customers' partial requirements by delivering or picking up the remaining loads even if the customer's demand

exceeds the total or remaining vehicle capacity. Thereby, this strategy improves the accessibility of service for customers, and gains more benefits for transportation and logistics operators.

2. The initial solution from the clustering procedures and heuristic algorithm helps increase the possibility that the final solution converges to the known best solution.

3. The hybrid heuristic algorithm is able to improve the initial solution. For example, the relocate operator and relocate split operator in the hybrid heuristic algorithm can split and relocate the customers between different routes. In addition, between-route (coarse-turning) and within-route (fine-turning) operations are applied interchangeably until the known best solution is achieved. The known best solution can thus reduce the travel cost and increase the average loading rate.

5 Conclusions

This study aims to tackle the vehicle routing problem with split deliveries and pickups (VRPSPDP) in an effective manner. A heuristic algorithm with clustering operation and initial heuristic procedures are proposed to obtain an initial solution. A hybrid heuristic algorithm with several operators is used to further improve the initial solution. The merits of this study lie in the following aspects: (1) The clustering procedure ensures that vehicle routes can start and end at the same depot. (2) The proposed two-stage method with clustering capability has inherent advantages on large-scale logistics network optimization problems. (3) The hybrid heuristic algorithm with between-route (coarse-turning) and within-route (fine-turning) operations is able to polish the initial solution. The proposed method was applied to the revised Solomon datasets and extended Solomon datasets. The results revealed that the proposed two-stage method has significantly reduced the total travel cost and increased the average loading rate. Specifically, for the 400-customer problem datasets, travel cost was reduced by more than 17.6% and the average loading rate was increased by 15%. The different dataset experiments also showed that splitting deliveries and pickups under a certain condition is beneficial for transportation and logistics enterprises, reducing the total travel cost and increasing the average loading rate.

For future research, the following directions can be considered: (1) The total number of vehicle varies according to the actual customer demands; (2) The customer demands are time dependent and stochastic, rather than constant; (3) Improve the algorithm efficiency using distributed and parallel computing techniques or other advanced optimization algorithms to find the 'best' solution more quickly.

Acknowledgements

Special thanks are given to Matt Dunlap from University of Washington, USA for his valuable suggestions and revisions.

References

Ai, T.J., Kachitvichyanukul, V., 2009. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.*, **36**(5):1693-1702. [doi:10.1016/j.cor.2008.04.003]

Archetti, C., Savelsbergh, M.W.P., Speranza, M.G., 2008. To split or not to split: that is the question. *Transp. Res. PT E-Logist. Transp.*, **44**(1):114-123. [doi:10.1016/j.tre.2006.04.003]

Baldacci, R., Toth, P., Vigo, D., 2010. Exact algorithms for routing problems under vehicle capacity constraints. *Ann. Oper. Res.*, **175**(1):213-245. [doi:10.1007/s10479-009-0650-0]

Çatay, B., 2010. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Syst. Appl.*, **37**(10):6809-6817. [doi:10.1016/j.eswa.2010.03.045]

Chen, A.L., Yang, G.K., Wu, Z.M., 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *J. Zhejiang Univ.-Sci. A*, **7**(4):607-614. [doi:10.1631/jzus.2006.A0607]

Chen, J.F., Wu, T.H., 2006. Vehicle routing problem with simultaneous deliveries and pickups. *J. Oper. Res. Soc.*, **57**(5):579-587. [doi:10.1057/palgrave.jors.2602028]

Chepuri, K., Homem-de-Mello, T., 2005. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Ann. Oper. Res.*, **134**(1):153-181. [doi:10.1007/s10479-005-5729-7]

de Oliveira, H.C.B., Vasconcelos, G.C., 2010. A hybrid search method for the vehicle routing problem with time windows. *Ann. Oper. Res.*, **180**(1):125-144. [doi:10.1007/s10479-008-0487-y]

Dror, M., Trudeau, P., 1989. Savings by split delivery routing. *Transp. Sci.*, **23**(2):141-145.

Gajpal, Y., Abad, P., 2009. An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Comput. Oper. Res.*, **36**(12):3215-3223. [doi:10.1016/j.cor.2009.02.017]

Gulczynski, D., Golden, B., Wasil, E., 2010. The split delivery vehicle routing problem with minimum delivery amounts. *Transp. Res. PT E-Logist. Transp.*, **46**(5):612-626. [doi:10.1016/j.tre.2009.12.007]

Hennig, F., Nygreen, B., Christiansen, M., *et al.*, 2012. Maritime crude oil transportation—a split pickup and split delivery problem. *Eur. J. Oper. Res.*, **218**(3):764-774. [doi:10.1016/j.ejor.2011.09.046]

Ho, S.C., Haugland, D., 2004. A tabu search heuristic for vehicle routing problem with time windows and split deliveries. *Comput. Oper. Res.*, **31**(12):1947-1964. [doi:10.1016/S0305-0548(03)00155-2]

Hoff, A., Gribkovskaia, I., Laporte, G., *et al.*, 2009. Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *Eur. J. Oper. Res.*, **192**(3):755-766. [doi:10.1016/j.ejor.2007.10.021]

Jin, M.Z., Liu, K., Bowden, R.O., 2007. A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. *Int. J. Prod. Econ.*, **105**(1):228-242. [doi:10.1016/j.ijpe.2006.04.014]

Lambert, S., Riopel, D., Abdul-Kader, W., 2011. A reverse logistics decisions conceptual framework. *Comput. Ind. Eng.*, **61**(3):561-581. [doi:10.1016/j.cie.2011.04.012]

Lee, C.G., Epelman, M.A., White, C.C.III, *et al.*, 2006. A shortest path approach to the multiple-vehicle routing problem with pick-ups. *Transp. Res. PT B-Method.*, **40**(4):265-284. [doi:10.1016/j.trb.2004.11.004]

Li, J.X., Chu, F., Chen, H.X., 2011. Coordination of split deliveries in one-warehouse multi-retailer distribution systems. *Comput. Ind. Eng.*, **60**(2):291-301. [doi:10.1016/j.cie.2010.11.012]

Liu, R., Xie, X.L., Augusto, V., *et al.*, 2013. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *Eur. J. Oper. Res.*, **230**(3):475-486. [doi:10.1016/j.ejor.2013.04.044]

Mitra, S., 2005. An algorithm for the generalized vehicle routing problem with backhauling. *Asia Pac. J. Oper. Res.*, **22**(2):153-169. [doi:10.1142/S0217595905000522]

Mitra, S., 2008. A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *J. Oper. Res. Soc.*, **59**(11):1532-1546. [doi:10.1057/palgrave.jors.2602500]

Montané, F.A.T., Galvão, R.D., 2006. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput. Oper. Res.*, **33**(3):595-619. [doi:10.1016/j.cor.2004.07.009]

Moreno, L., de Aragão, M.P., Uchoa, E., 2010. Improved lower bounds for the split delivery vehicle routing problem. *Oper. Res. Lett.*, **38**(4):302-306. [doi:10.1016/j.orl.2010.04.008]

Mullaseril, P.A., Dror, M., Leung, J., 1997. Split-delivery routing heuristics in livestock feed distribution. *J. Oper. Res. Soc.*, **48**(2):107-116.

Nanry, W.P., Barnes, J.W., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transp. Res. PT B-Method.*, **34**(2):107-121. [doi:10.1016/S0191-2615(99)00016-8]

Nowak, M., Ergun, O., White, C.C.III, 2009. An empirical study on the benefit of split loads with the pickup and delivery problem. *Eur. J. Oper. Res.*, **198**(3):734-740. [doi:10.1016/j.ejor.2008.09.041]

Özdamar, L., Demir, O., 2012. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transp. Res. PT E-Logist. Transp.*, **48**(3):591-602. [doi:10.1016/j.tre.2011.11.003]

Rieck, J., Zimmermann, J., 2010. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Ann. Oper. Res.*, **181**(1):337-358. [doi:10.1007/s10479-010-0748-4]

Saberi, M., Verbas, İ.Ö., 2012. Continuous approximation model for the vehicle routing problem for emissions minimization at the strategic level. *J. Transp. Eng.*, **138**(11):1368-1376. [doi:10.1061/(ASCE)TE.1943-5436.0000442]

Sheu, J.B., 2006. A novel dynamic resource allocation model for demand-responsive city logistics distribution operations. *Transp. Res. PT E-Logist. Transp.*, **42**(6):445-472. [doi:10.1016/j.tre.2005.05.004]

Sheu, J.B., 2008. Green supply chain management, reverse logistics and nuclear power generation. *Transp. Res. PT E-Logist. Transp.*, **44**(1):19-46. [doi:10.1016/j.tre.2006.06.001]

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.*, **35**(2):254-265.

Subramanian, A., Drummond, L.M.A., Bentes, C., *et al.*, 2010. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.*, **37**(11):1899-1911. [doi:10.1016/j.cor.2009.10.011]

Subramanian, A., Uchoa, E., Pessoa, A.A., *et al.*, 2011. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Oper. Res. Lett.*, **39**(5):338-341. [doi:10.1016/j.orl.2011.06.012]

Wang, H.F., Chen, Y.Y., 2012. A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Comput. Ind. Eng.*, **62**(1):84-95. [doi:10.1016/j.cie.2011.08.018]

Wang, Y., Ma, X.L., Wang, Y.H., *et al.*, 2012. Location optimization of multiple distribution centers under fuzzy environment. *J. Zhejiang Univ.-Sci. A (Appl. Phys. & Eng.)*, **13**(10):782-798. [doi:10.1631/jzus.A1200137]

Wang, Y., Ma, X.L., Lao, Y.T., *et al.*, 2013. Vehicle routing problem: simultaneous deliveries and pickups with split loads and time windows. *J. Transp. Res. Board*, **2378**:120-128. [doi:10.3141/2378-13]

Wang, Y., Ma, X.L., Lao, Y.T., *et al.*, 2014. A fuzzy-based customer clustering approach with hierarchical structure for logistics network optimization. *Expert Syst. Appl.*, **41**(2):521-534. [doi:10.1016/j.eswa.2013.07.078]

Zhang, T., Chaovalitwongse, W.A., Zhang, Y.J., 2012. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Comput. Oper. Res.*, **39**(10):2277-2290. [doi:10.1016/j.cor.2011.11.021]