INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 24 (2017) 27–41
DOI: 10.1111/itor.12250

A novel approach to solve the split delivery vehicle routing problem

Ping Chen[a], Bruce Golden[b], Xingyin Wang[c] and Edward Wasil[d]

[a] *Business School, Nankai University, Tianjin, China*
[b] *Robert H. Smith School of Business, University of Maryland, College Park, MD, USA*
[c] *Department of Mathematics, University of Maryland, College Park, MD, USA*
[d] *Kogod School of Business, American University, Washington DC, USA*
E-mail: chenpingteda@nankai.edu.cn [Chen]; bgolden@rhsmith.umd.edu [Golden]; wangxy@umd.edu [Wang]; ewasil@american.edu [Wasil]

Received 8 April 2015; received in revised form 23 September 2015; accepted 11 November 2015

Abstract

The split delivery vehicle routing problem (SDVRP) is a relaxed version of the classic capacitated vehicle routing problem (CVRP). Customer demands are allowed to split among vehicles. This problem is computationally challenging and the state-of-the-art heuristics are often complicated to describe and difficult to implement, and usually have long computing times. All these hinder their application by practitioners to solve real-world problems. We propose a novel, efficient, and easily implemented approach to solve the SDVRP using an *a priori* split strategy, that is, each customer demand is split into small pieces in advance. Our computational experiments on 82 benchmark instances show that our algorithm is overall much more efficient and produces results that are comparable to those from the state-of-the-art approaches.

Keywords: vehicle routing problem; split delivery

1. Introduction

The split delivery vehicle routing problem (SDVRP) was introduced by Dror and Trudeau (1989, 1990). It is derived from the traditional capacitated vehicle routing problem (CVRP) in which customers can be visited only once by exactly one vehicle. In the SDVRP, this restriction is removed. It has been shown that there is a potential to reduce the total cost (by at most 50%; Archetti et al., 2006a) by splitting customer demands among vehicles (Dror and Trudeau, 1989, 1990) for problems in which all demands are less than the capacity of a vehicle. The survey by Archetti and Speranza (2012) provides additional details about the SDVRP and its variants.

In this paper, we provide a simple, effective, and efficient method for the SDVRP using an *a priori* splitting strategy, where each demand is split into several small demands. We transform the SDVRP into a corresponding CVRP using our splitting strategies. We combine our splitting strategies with the open source code VRPH (Groër, 2011) in order to solve benchmark instances. Our motivation is to provide a method for the SDVRP that is easy to understand, simple to implement, and provides high-quality solutions quickly. Most significantly, our method can be used with any CVRP solver, thereby allowing practitioners to generate solutions to the SDVRP efficiently and effectively.

In Section 2, we describe the SDVRP and related problems. In Section 3, we present our heuristic for the SDVRP. In Section 4, we report the results of computational experiments with our heuristic. In Section 5, we provide our conclusions.

2. SDVRP and related problems

The SDVRP is specified by a graph $G = (N, A)$, where $N = \{N_0, N_1, \ldots, N_n\}$ is the set of nodes, and $A = \{(N_i, N_j)|N_i, N_j \in N, i \neq j\}$ is the set of arcs. Node N_0 denotes the depot, where a fleet of homogeneous vehicles is located. $N_i (i = 1, 2, \ldots, n)$ denotes customer i with demand of D_i. c_{ij} is associated with each arc (N_i, N_j) and gives the travel cost from N_i to N_j. Each vehicle must leave from and return to the depot, and the load cannot exceed the vehicle capacity Q. The demand of each customer can be split among several vehicles. The objective is to find a set of routes that minimizes the total travel cost while satisfying all constraints.

With split deliveries, problems can be solved when customer demand is greater than vehicle capacity. Even when all demands are less than the vehicle capacity, it may be beneficial both in terms of the number of routes and total travel distance to split deliveries (Archetti et al., 2014). The SDVRP has been studied using a limited fleet of vehicles. The number of routes is limited to the minimum number of vehicles (K) needed to serve all customers, that is, $K = \lceil \sum_{i=1}^{n} \frac{D_i}{Q} \rceil$. However, as Chen et al. (2007) have pointed out, in general, for both CVRP and SDVRP, using fewer vehicles does not necessarily reduce the total distance traveled by the fleet (even if the triangle inequality holds). In this paper, we minimize the total distance traveled assuming an unlimited number of vehicles.

The SDVRP studies have focused on theoretical analysis (complexity [Archetti et al., 2011b] and worst-case analysis [Archetti et al., 2006a]) and computational experiments. Archetti et al. (2011b) studied the computational complexity of the SDVRP and CVRP with a limited number and an unlimited number of vehicles. They showed that only the SDVRP on a straight line or on a circle is solvable in polynomial time. Archetti et al. (2006a) proved that the savings from split deliveries could be as large as 50% in the best case analysis. Archetti et al. (2008) showed that the savings depended on the magnitude of the mean and variance of customer demands relative to the vehicle capacity, but did not depend on the customer locations.

Except for some special classes of instances, the SDVRP is NP-hard (Archetti et al., 2011b; Dror and Trudeau, 1990) in general. Many exact (Archetti et al., 2011a, 2014; Belenguer et al., 2000; Jin et al., 2008; Moreno et al., 2010) and heuristic (Aleman and Hill, 2010; Aleman et al., 2010; Archetti et al., 2008; Berbotto et al., 2014; Chen et al., 2007) approaches have been proposed in the last decade. The branch-and-cut algorithm proposed by Archetti et al. (2014) was capable of solving most instances with as many as 50 customers and two instances with 75 and 100 customers. For large problems, heuristics are preferred to obtain high-quality solutions in a reasonable amount of computing time, which is desirable in real-world applications. However, many sophisticated

Table 1
Heuristic approaches for the SDVRP

Authors	Year	Methodology	Fleet constraint	Machine
Dror and Trudeau	1989, 1990	Local search	UF	
Archetti et al.	2006	Tabu search (SPLITABU)	UF	PC P4 2.4 GHz
Chen et al.	2007	Mixed integer program and record-to-record travel algorithm (EMIP + VRTR)	UF	PC P4 1.7 GHz
Archetti et al.	2008	Tabu search integrated with an integer program (OpBH)	UF	
Aleman and Hill	2010	Tabu search with vocabulary building (TSVBA)	UF	PC P4 2.8 GHz
Aleman et al.	2010	Variable neighborhood descent and adaptive memory algorithm	LF	PC P4 2.8 GHz
Derigs et al.	2010	Metaheuristic-controlled local search algorithms (ABHC)	UF	PC 3.0 GHz
Archetti et al.	2011	Branch-and-price-and-cut method (BPC)	UF	PC Dual Core P4 2.4 GHz
			LF	
Berbotto et al.	2013	Randomized granular tabu search and record-to-record travel algorithm	LF	PC 2.10 GHz
Silva et al.	2015	Iterated local search (SplitILS)	UF LF	PC Core i7 2.93 GHz

LF, limited fleet size; UF, unlimited fleet size.

heuristics (namely, metaheuristics) are difficult to implement, and typically have long computing times. This may hinder their application in real-world settings. An earlier paper by Burrows (1988) mentioned the idea of splitting demands as follows:

1. Apply a modified Clarke–Wright savings heuristic.
2. When merging two tours based on the largest savings, split a customer demand in order to exactly fill a vehicle.
3. Continue until all customer demands are fully serviced.

Burrows's (1988) idea is different from our *a priori* split strategy. We split the demands in advance; he does not. In addition, one can apply any standard CVRP solver to our resulting problem; Burrows (1988) modifies an early CVRP heuristic.

In this paper, we focus on heuristics for solving the SDVRP. Many heuristics have been applied to the SDVRP with success including local search based heuristics. In Table 1, we show heuristic approaches for the SDVRP that have appeared in the literature since 1989.

3. New heuristic for the SDVRP

Our heuristic approach for the SDVRP combines an *a priori* split strategy and the VRPH (Groër, 2011) solver. The SDVRP is transformed into a corresponding CVRP at the expense of an increased number of customers. Next, the VRPH is applied to solve this CVRP. Finally, the solution to the CVRP is transformed into a solution to the original SDVRP.

3.1. A priori split strategy

We propose an *a priori* split strategy to split deliveries in advance, and not during the algorithm procedure. We aim to split each demand into several small demands, so that it is possible to make full use of a vehicle's capacity. We propose a 20/10/5/1 rule that splits each demand, D_i, into m_{20} pieces of $0.20Q$, m_{10} pieces of $0.10Q$, m_5 pieces of $0.05Q$, m_1 pieces of $0.01Q$, and at most one remaining piece less than $0.01Q$, where Q is the vehicle capacity, and

- $m_{20} = \max\{m \in \mathbb{Z}^+ \cup \{0\} | 0.20Qm \leq D_i\}$,
- $m_{10} = \max\{m \in \mathbb{Z}^+ \cup \{0\} | 0.10Qm \leq D_i - 0.20Qm_{20}\}$,
- $m_5 = \max\{m \in \mathbb{Z}^+ \cup \{0\} | 0.05Qm \leq D_i - 0.20Qm_{20} - 0.10Qm_{10}\}$,
- $m_1 = \max\{m \in \mathbb{Z}^+ \cup \{0\} | 0.01Qm \leq D_i - 0.20Qm_{20} - 0.10Qm_{10} - 0.05Qm_5\}$.

We also propose a 25/10/5/1 rule. For example, if $Q = 100$ and $D_i = 76$, the demand is split into 20, 20, 20, 10, 5, and 1 by the 20/10/5/1 rule, and into 25, 25, 25, and 1 by the 25/10/5/1 rule. We use both rules in our computational experiments.

3.2. Vehicle routing problem heuristic

The VRPH solver (Groër, 2011; Groër et al., 2010) is an effective and efficient implementation of state-of-the-art heuristic methods for the CVRP, and its codes are freely accessible to the public. Therefore, we apply the VRPH solver without any modification of codes for the CVRP. All parameters are set to default values. We use VRPH as a stand-alone application, which takes a CVRP instance written in a file and outputs the solution to a file. The input file format resembles a file from the Traveling Salesman Problem Library (TSPLIB). We point out that a practitioner can use any free or commercial CVRP software to implement our *a priori* split idea. We chose VRPH because it is publicly available and produces high-quality solutions.

It is true that an *a priori* split rule (20/10/5/1 or 25/10/5/1) may eliminate the optimal SDVRP solution. If both vehicle capacity and customer demands are integers, splitting demands into unit demands seems to be the only way that will not eliminate the optimal solution. However, splitting into unit demands will increase the size of the CVRP dramatically, increase the running time, and may result in an inferior objective function value of VRPH in the second stage. If one uses an exact CVRP solver in the second stage, splitting customer demands into unit demands is expected to improve the objective function value at the expense of a much longer running time. But if a heuristic is implemented in the second stage, despite the increase in running time due to an increased problem size, the solution quality may actually deteriorate.

Furthermore, our *a priori* split rules (20/10/5/1 and 25/10/5/1) may not eliminate the optimal solution if the trucks that serve customers partially are not fully loaded. If the two vehicles serving a single customer have remaining capacities in an optimal solution to the SDVRP, there are many different ways to split the demand in an optimal solution. To make a reasonable trade-off between running time and solution quality, we split each demand into some moderate-sized pieces and some small pieces.

4. Computational experiments and results

We apply our VRPH algorithm with *a priori* split strategies (denoted as VRPHAS) to 82 SDVRP benchmark instances and report the results in this section. We compare our results to the best-known solutions and the results produced by other approaches found in the literature, both in terms of solution quality and running time. All experiments were performed on a personal computer with an Intel i7-4700MQ CPU 2.40 GHz and 12 GB RAM.

4.1. Benchmark instances

The 82 instances are divided into four sets. Sets 1 and 2 were proposed by Belenguer et al. (2000). Set 1 has 11 instances from TSPLIB. Set 2 was randomly generated using the coordinates of eil51, eil76, and eil101 from TSPLIB. The capacity of a vehicle is 160. The demands are generated using six different scenarios: $[\lceil 0.01Q\rceil, \lfloor 0.1Q\rfloor]$; $[\lceil 0.1Q\rceil, \lfloor 0.3Q\rfloor]$; $[\lceil 0.1Q\rceil, \lfloor 0.5Q\rfloor]$; $[\lceil 0.1Q\rceil, \lfloor 0.9Q\rfloor]$; $[\lceil 0.3Q\rceil, \lfloor 0.7Q\rfloor]$; $[\lceil 0.7Q\rceil, \lfloor 0.9Q\rfloor]$. There are 14 instances in Set 2.

Set 3 uses the same six demand scenarios found in Set 2 and has 36 instances. These instances were generated by Archetti et al. (2008). The set has 42 instances, but 6 of them are repeated, so we include only 36 instances. A similar set was introduced by Campos et al. (2008). The two sets have the same customer locations, but different demand levels. According to Silva et al. (2015), the papers by Boudia et al. (2007), Aleman and Hill (2010), and Aleman et al. (2010) used the data set by Campos et al. (2008), so we do not include these three algorithms in our comparison in Table 4.

Set 4 has 21 instances from Chen et al. (2007). In these instances, customers are placed uniformly on concentric circles centered at the depot. Customer demands are either 60 or 90 while the vehicle capacity is 100. The instances vary in the number of concentric circles and the number of customers on a circle (from 8 to 288).

4.2. Performances of a priori split strategies

In order to evaluate the performances of our *a priori* split strategy, we conduct experiments on the 36 benchmark instances in Set 3, and compare our solutions to the nonsplit solutions. The nonsplit solutions are produced by the VRPH solver. Table 2 shows the savings over the nonsplit results using VRPHAS with *a priori* split rules 20/10/5/1 and 25/10/5/1.

In Table 2, the first and second columns are the identifier and size of the problems, respectively. The numbers after the hyphen in an identifier name indicate the demand range. For example, P01-110 has demand in the range $[\lceil 0.01Q\rceil, \lfloor 0.1Q\rfloor]$. The third and fourth columns are the percent savings produced by the two *a priori* split strategies, computed by $[(f(s_{nonsplit}) - f(s))/f(s_{nonsplit})] \times 100\%$, where $f(s)$ is the distance of the solution s produced by the *a priori* split strategy, and $f(s_{nonsplit})$ denotes the distance of the solution to the SDVRP without splits. From Table 2, we observe that

Table 2
Percent savings produced by two *a priori* strategies over the nonsplit strategy on 36 instances in Set 3 (n is the number of customers)

Instance	n	20/10/5/1	25/10/5/1
P01-110	50	0.00	0.00
P02-110	75	0.87	0.95
P03-110	100	1.74	1.05
P04-110	150	9.30	8.16
P05-110	199	9.80	11.05
P11-110	120	7.79	7.47
P01-1030	50	−0.81	−0.82
P02-1030	75	0.38	−0.04
P03-1030	100	1.22	1.75
P04-1030	150	6.39	5.87
P05-1030	199	8.73	8.78
P11-1030	120	9.00	7.98
P01-1050	50	2.60	2.60
P02-1050	75	1.13	1.84
P03-1050	100	3.01	2.52
P04-1050	150	9.30	9.36
P05-1050	199	12.79	13.14
P11-1050	120	10.30	10.16
P01-1090	50	0.87	0.87
P02-1090	75	1.15	1.63
P03-1090	100	3.74	4.42
P04-1090	150	11.50	11.51
P05-1090	199	14.56	15.23
P11-1090	120	11.42	11.61
P01-3070	50	1.86	1.86
P02-3070	75	1.80	2.02
P03-3070	100	2.71	2.79
P04-3070	150	10.05	10.02
P05-3070	199	12.83	12.62
P11-3070	120	13.29	13.24
P01-7090	50	1.28	1.28
P02-7090	75	2.12	1.82
P03-7090	100	1.67	1.34
P04-7090	150	10.24	9.73
P05-7090	199	12.24	12.74
P11-7090	120	15.32	13.95
Average		6.17	6.12

the *a priori* split strategies improve the nonsplit solutions in nearly all problems. On average, over all 36 instances, the savings are 6.17% for the 20/10/5/1 strategy and 6.12% for the 25/10/5/1 strategy. The two split rules usually produce different solutions and perform almost equally well. The 20/10/5/1 rule produces larger savings in 16 instances, while the 25/10/5/1 rule produces larger savings in 15 instances. They tie in five instances. Given these performances, we decided to use both rules with VRPHAS in our experiments.

4.3. Comparisons with the state-of-the-art approaches

Next, we compare the results produced by VRPHAS with both *a priori* split strategies to the results produced by seven procedures for the SDVRP with an unlimited number of vehicles. The seven procedures are tabu search (SPLITABU) (Archetti et al., 2006b), hybrid heuristic (EMIP + VRTR) (Chen et al., 2007), optimization-based heuristic (OpBH) (Archetti et al., 2008), tabu search with vocabulary building (TSVBA) (Aleman and Hill, 2010), metaheuristic-controlled local search (ABHC) (Derigs et al., 2010), branch-and-price-and-cut (BPC) (Archetti et al., 2011a), and iterated local search (SplitILS) (Silva et al., 2015).

Tables 3–5 show the results produced by the procedures. In these three tables, the first and second columns give the instance name and problem size. The third column is the best-known solution to each instance, while the fourth column is the reference that produced the best-known solution. If VRPHAS updates an existing best-known solution, the reference cell is left blank and the new best solution is shown in bold. Other columns show the results by each approach in terms of deviation from the best-known solution (gap) and computing time (time) in seconds. The gap is computed by $[(f(s) - f(s_{best\text{-}known}))/f(s_{best\text{-}known})] \times 100\%$. We also report the total distance (cost) and the number of vehicles used (k) in the best solution produced by VRPHAS. The computer used in each approach is given in Table 1. The results that we report from VRPHAS the better ones from running both 20/10/5/1 and 25/10/5/1. The reported computing time is the amount of time to run both strategies.

In Table 3, we compare VRPHAS to TSVBA, BPC, and SplitILS on Sets 1 and 2. On the instances in Set 1, SplitILS produces the smallest average gap (0.45%) from the best-known solutions followed by VRPHAS (0.74%). Both TSVBA and BPC produce solutions with average gaps greater than 2%. The four best-known solutions shown in bold are new solutions found by VRPHAS. On Set 2, SplitILS produces all the best-known solutions. TSVBA, BPC, and VRPHAS produce solutions with average gaps of 3.06%, 2.04%, and 2.24%, respectively. It is not straightforward to compare the running times because the algorithms do not run on the same machine. Based on the average running times per instance, VRPHAS and SplitILS are more efficient than TSVBA and BPC on the first set of instances, and VRPHAS is the most efficient on the second set.

In Table 4, we compare VRPHAS to SPLITABU, OpBH, ABHC, BPC, and SplitILS on Set 3. (The SPLITABU results were provided by the authors in Archetti et al., 2006b via private communication.) SplitILS performs the best in terms of objective function value, with an average gap of 0.09% from the best-known solutions, followed by ABHC (1.43%) and VRPHAS (1.50%). SplitILS produces the best-known solutions in 31 of the 36 instances. VRPHAS finds four new best-known solutions. In terms of running time, VRPHAS is the fastest, taking 45.51 seconds, on average, to solve an instance, followed by BPC (99.71 seconds). The other four procedures take more than 1000 seconds on average to solve an instance.

In Table 5, we compare VRPHAS to EMIP + VRTR, TSVBA, ABHC, BPC, and SplitILS on Set 4. SplitILS produces the smallest average gap (0.05%). TSVBA, ABHC, BPC, and VRPHAS produce solutions with average gaps that are less than 1%. TSVBA and VRPHAS are both efficient on this set, with average running times of 8.91 and 27.29 seconds. In Fig. 1, we plot the running times of TSVBA, SplitILS, and VRPHAS against the instances in increasing order of their sizes (number of customers). The running times are plotted on a logarithmic scale. We observe that the running times of VRPHAS increase more slowly than the running times of TSVBA and SplitILS.

Table 3
Performance of TSVBA, BPC, SplitILS, and VRPHAS on 25 benchmark instances in Sets 1 and 2

				TSVBA (Aleman and Hill, 2010)		BPC (Archetti et al., 2011a)		SplitILS (Silva et al., 2015)		VRPHAS			
Instance	*n*	Best-known Solution	Reference	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	*k*	Cost	Gap (%)	Time (seconds)
Set 1													
eil22	21	375.28	Aleman and Hill (2010)	0.00	2.58	–	–	0.00	0.15	4	375.28	0.00	1.65
eil23	22	568.56	Silva et al. (2015)	0.21	1.59	–	–	0.00	0.13	3	568.56	0.00	1.58
eil30	29	**497.53**		1.50	7.45	–	–	1.50	0.24	3	**497.53**	0.00	2.18
eil33	32	**826.41**		2.08	8.38	–	–	1.29	0.51	4	**826.41**	0.00	3.51
eil51	50	524.61	Silva et al. (2015)	0.58	49.84	0.64	–	0.00	1.79	5	524.61	0.00	9.37
eilA76	75	823.89	Silva et al. (2015)	3.56	145.78	1.07	–	0.00	30.76	11	849.60	3.12	17.76
eilB76	75	1009.04	Silva et al. (2015)	2.49	91.36	3.78	–	0.00	51.83	15	1024.44	1.53	18.81
eilC76	75	738.67	Silva et al. (2015)	3.10	151.13	3.61	–	0.00	16.96	8	748.51	1.33	15.36
eilD76	75	**684.53**		1.67	122.52	3.03	–	0.45	11.16	7	**684.53**	0.00	14.72
eilA101	100	**812.51**		3.90	295.22	5.11	–	1.68	38.90	8	**812.51**	0.00	14.73
eilB101	100	1076.26	Silva et al. (2015)	3.33	173.13	3.99	–	0.00	110.61	14	1099.21	2.13	21.61
Average				2.24	95.36	3.03	99.71	0.45	23.91			0.74	11.03

Continued

Table 3
Continued

Instance	n	Best-known Solution	Reference	TSVBA (Aleman and Hill, 2010) Gap (%)	TSVBA Time (seconds)	BPC (Archetti et al., 2011a) Gap (%)	BPC Time (seconds)	SplitILS (Silva et al., 2015) Gap (%)	SplitILS Time (seconds)	VRPHAS k	VRPHAS Cost	VRPHAS Gap (%)	VRPHAS Time (seconds)
Set 2													
S51D1	50	459.50	Archetti et al. (2011a)	2.02	13.56	0.00	–	0.00	1.24	3	459.50	0.00	7.23
S51D2	50	709.29	Silva et al. (2015)	1.33	31.66	1.11	–	0.00	11.20	10	716.82	1.06	11.46
S51D3	50	948.06	Silva et al. (2015)	2.29	18.75	1.30	–	0.00	15.74	16	964.83	1.77	15.88
S51D4	50	1562.01	Silva et al. (2015)	4.24	19.77	0.51	–	0.00	56.28	28	1592.23	1.93	18.90
S51D5	50	1333.67	Silva et al. (2015)	2.14	15.39	0.43	–	0.00	36.69	24	1371.41	2.83	19.66
S51D6	50	2169.10	Silva et al. (2015)	3.09	14.38	0.60	–	0.00	62.55	43	2240.46	3.29	28.04
S76D1	75	598.94	Silva et al. (2015)	2.46	252.28	5.83	–	0.00	4.86	4	614.31	2.57	15.07
S76D2	75	1087.40	Silva et al. (2015)	3.75	60.44	1.58	–	0.00	69.36	16	1120.71	3.06	21.54
S76D3	75	1427.86	Silva et al. (2015)	3.16	51.13	0.51	–	0.00	96.50	24	1445.23	1.22	34.28
S76D4	75	2079.76	Silva et al. (2015)	4.83	53.56	1.29	–	0.00	188.38	37	2138.64	2.83	29.74
S101D1	100	726.59	Silva et al. (2015)	3.21	860.31	8.88	–	0.00	15.93	6	746.08	2.68	22.02
S101D2	100	1378.43	Silva et al. (2015)	2.22	219.52	3.46	–	0.00	151.66	21	1412.98	2.51	37.16
S101D3	100	1874.81	Silva et al. (2015)	3.88	132.19	1.95	–	0.00	317.29	31	1924.39	2.64	39.49
S101D5	100	2791.22	Silva et al. (2015)	4.28	131.16	1.18	–	0.00	572.13	50	2874.86	3.00	45.44
Average				3.06	133.86	2.04	99.66	0.00	114.27			2.24	24.71

Bold indicates new best-known solution.

Table 4
Performance of six methods on 36 benchmark instances in Set 3

				SPLITABU (Archetti et al., 2006b)		OpBH (Archetti et al., 2008)		ABHC (Derigs et al., 2010)		BPC (Archetti et al., 2011a)		SplitILS (Silva et al., 2015)		VRPHAS			
Instance	n	Best-known solution	Reference	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	k	Cost	Gap (%)	Time (seconds)
P01-110	50	459.50	Archetti et al. (2011a), Silva et al. (2015)	0.74	9	–	–	–	–	0.00	–	0.00	1.16	3	459.50	0.00	6.55
P01-1030	50	757.15	Silva et al. (2015)	1.08	27	0.70	256	2.54	3600	1.72	–	0.00	13.24	11	776.06	2.50	15.54
P01-1050	50	1005.75	Silva et al. (2015)	3.32	56	3.01	866	0.68	3600	1.14	–	0.00	20.70	16	1027.92	2.21	15.87
P01-1090	50	1488.58	Silva et al. (2015)	1.57	34	1.49	2939	0.07	3600	0.05	–	0.00	43.04	27	1525.09	2.45	22.07
P01-3070	50	1481.71	Silva et al. (2015)	1.50	52	1.37	1684	0.44	3600	1.19	–	0.00	44.11	27	1511.66	2.02	22.47
P01-7090	50	2156.14	Silva et al. (2015)	0.81	160	0.75	834	0.85	3600	0.47	–	0.00	78.49	43	2215.09	2.73	26.32
P02-110	75	617.85	Silva et al. (2015)	0.99	42	–	–	–	–	5.68	–	0.00	5.70	5	628.86	1.79	10.83
P02-1030	75	1109.62	Silva et al. (2015)	2.20	78	1.19	161	1.29	3600	1.10	–	0.00	53.26	17	1137.43	2.50	23.08
P02-1050	75	1502.05	Silva et al. (2015)	3.64	71	3.09	646	0.44	3600	0.82	–	0.00	108.37	24	1520.83	1.25	28.19
P02-1090	75	2298.58	Silva et al. (2015)	1.74	311	1.70	361	1.81	3600	0.86	–	0.00	207.22	42	2356.70	2.53	32.42
P02-3070	75	2219.97	Silva et al. (2015)	3.31	184	2.20	2551	1.08	3600	0.78	–	0.00	265.00	41	2286.06	2.97	43.51
P02-7090	75	3223.40	Silva et al. (2015)	1.92	437	1.58	1872	1.35	3600	1.08	–	0.00	378.95	63	3303.98	2.50	44.78
P03-110	100	**752.62**		2.50	59	–	–	–	–	4.73	–	0.98	21.65	6	**752.62**	0.00	13.76
P03-1030	100	1458.46	Silva et al. (2015)	3.89	122	3.21	159	1.38	3600	1.30	–	0.00	179.69	22	1469.84	0.78	22.70
P03-1050	100	1996.76	Silva et al. (2015)	2.87	206	1.37	201	1.96	3600	2.21	–	0.00	362.02	33	2017.94	1.06	29.37
P03-1090	100	3085.69	Silva et al. (2015)	2.25	412	1.64	620	1.93	3600	1.34	–	0.00	736.52	58	3143.09	1.86	40.35
P03-3070	100	2989.30	Silva et al. (2015)	2.73	454	2.22	1605	0.83	3600	1.38	–	0.00	742.88	54	3044.73	1.85	40.98
P03-7090	100	4387.32	Silva et al. (2015)	1.90	1891	1.48	2433	1.37	3600	1.83	–	0.00	675.01	83	4475.32	2.01	52.49

Continued

Table 4
Continued

				SPLITABU (Archetti et al., 2006b)		OpBH (Archetti et al., 2008)		ABHC (Derigs et al., 2010)		BPC (Archetti et al., 2011a)		SplitILS (Silva et al., 2015)		VRPHAS			
Instance	n	Best-known solution	Reference	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	k	Cost	Gap (%)	Time (seconds)
P04-110	150	**919.17**		3.04	258	–	–	–	–	7.13	–	0.30	161.90	9	**919.17**	0.00	30.30
P04-1030	150	2016.97	Silva et al. (2015)	4.21	545	3.79	1152	1.89	3600	2.45	–	0.00	1109.88	32	2039.21	1.10	40.70
P04-1050	150	2849.66	Silva et al. (2015)	4.98	564	4.47	517	2.19	3600	2.39	–	0.00	1518.82	49	2876.70	0.95	58.02
P04-1090	150	4545.46	Silva et al. (2015)	2.83	1822	2.51	592	2.05	3600	2.93	–	0.00	2410.48	84	4616.62	1.56	70.00
P04-3070	150	4334.71	Silva et al. (2015)	3.74	1512	3.01	251	2.34	3600	2.40	–	0.00	2357.52	81	4420.73	1.98	74.35
P04-7090	150	6395.41	Silva et al. (2015)	1.36	8783	1.05	2460	1.12	3600	2.00	–	0.00	1926.20	124	6506.25	1.73	87.03
P05-110	199	1074.18	Silva et al. (2015)	6.90	754	–	–	–	–	18.12	–	0.00	626.12	11	1074.58	0.04	33.66
P05-1030	199	2478.40	Silva et al. (2015)	4.34	1224	4.20	567	2.49	3600	4.78	–	0.00	2661.25	41	2500.49	0.89	59.47
P05-1050	199	3471.41	Silva et al. (2015)	4.40	3811	3.52	1138	3.18	3600	2.79	–	0.00	3014.82	63	3517.12	1.32	75.93
P05-1090	199	5521.57	Silva et al. (2015)	3.52	2598	3.42	806	2.67	3600	2.75	–	0.00	4349.61	106	5618.96	1.76	87.18
P05-3070	199	5409.76	Silva et al. (2015)	2.98	2279	2.59	1702	2.43	3600	2.78	–	0.00	4524.33	103	5496.88	1.61	119.96
P05-7090	199	8192.03	Silva et al. (2015)	2.44	11,347	1.99	656	1.29	3600	2.67	–	0.00	3258.29	162	8331.44	1.71	133.69
P11-110	120	**1031.11**		2.34	61	–	–	–	–	3.93	–	1.17	109.76	7	**1031.11**	0.00	16.19
P11-1030	120	**2881.80**		6.20	516	4.71	585	1.09	3600	3.54	–	0.58	895.11	26	**2881.80**	0.00	30.06
P11-1050	120	4219.01	Silva et al. (2015)	6.72	259	6.09	365	1.22	3600	0.97	–	0.00	1957.37	43	4265.64	1.10	39.86
P11-1090	120	6854.09	Silva et al. (2015)	7.24	1037	3.73	4882	0.53	3600	2.07	–	0.00	3442.31	68	7004.71	2.20	64.02
P11-3070	120	6671.04	Derigs et al. (2010)	7.45	477	6.83	7174	0.00	3600	2.27	–	0.04	2354.02	64	6776.88	1.58	54.73
P11-7090	120	10,204.81	Silva et al. (2015)	4.59	2033	2.14	3948	0.28	3600	1.69	–	0.00	2279.57	98	10,364.33	1.56	72.05
Average				3.29	1235.69	2.70	1466.10	1.43	3600	2.65	99.71	0.09	1191.51			1.50	45.51

Table 5
Performance of six methods on 21 benchmark instances in Set 4

				EMIP + VRTR Chen et al., 2007)		TSVBA (Aleman and Hill, 2010)		ABHC (Derigs et al., 2010)		BPC (Archetti et al., 2011a)		SplitILS (Silva et al., 2015)		VRPHAS			
Instance	n	Best-known solution	Reference	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	Gap (%)	Time (seconds)	k	Cost	Gap (%)	Time (seconds)
SD1	8	228.28	Aleman and Hill (2010), Archetti et al. (2011a), Chen et al. (2007), Derigs et al. (2010), Silva et al. (2015)	0.00	0.70	0.00	0.00	0.00	300	0.00	–	0.00	0.05	6	228.28	0.00	0.59
SD2	16	708.28	Aleman and Hill (2010), Archetti et al. (2011a), Derigs et al. (2010), Silva et al. (2015)	0.86	54.40	0.00	0.02	0.00	300	0.00	–	0.00	0.63	12	708.28	0.00	1.65
SD3	16	430.40	Archetti et al. (2011a)	0.05	67.30	0.04	0.03	0.04	300	0.00	–	0.04	0.62	12	430.58	0.04	1.62
SD4	24	630.62	Archetti et al. (2011a)	0.07	400.00	0.07	0.08	0.07	300	0.00	–	0.07	2.26	18	631.05	0.07	3.01
SD5	32	1389.94	Archetti et al. (2011a)	1.31	402.70	0.05	0.13	0.05	300	0.00	–	0.05	6.07	24	1390.57	0.05	4.59
SD6	32	830.86	Archetti et al. (2011a)	0.04	408.30	0.05	0.14	0.05	300	0.00	–	0.05	5.81	24	831.24	0.05	4.04
SD7	40	3640.00	Aleman and Hill (2010), Archetti et al. (2011a), Derigs et al. (2010), Silva et al. (2015)	2.04	403.20	0.00	0.09	0.00	300	0.00	–	0.00	14.12	30	3640.00	0.00	9.22

Continued

Table 5
Continued

Instance	n	Best-known solution	Reference	EMIP + VRTR Chen et al., 2007) Gap (%)	Time (seconds)	TSVBA (Aleman and Hill, 2010) Gap (%)	Time (seconds)	ABHC (Derigs et al., 2010) Gap (%)	Time (seconds)	BPC (Archetti et al., 2011a) Gap (%)	Time (seconds)	SplitILS (Silva et al., 2015) Gap (%)	Time (seconds)	VRPHAS k	Cost	Gap (%)	Time (seconds)
SD8	48	5068.28	Aleman and Hill (2010), Archetti et al. (2011a), Derigs et al. (2010), Silva et al. (2015)	2.60	404.10	0.00	0.14	0.00	300	0.00	–	0.00	24.93	36	5068.28	0.00	14.51
SD9	48	2042.88	Archetti et al. (2011a)	0.83	404.30	1.38	0.36	1.22	300	0.00	–	0.06	38.78	36	2057.62	0.72	7.34
SD10	64	2683.73	Archetti et al. (2011a)	2.44	400.00	2.39	0.89	3.74	300	0.00	–	0.04	101.10	49	2707.83	0.90	16.87
SD11	80	13,280.00	Aleman and Hill (2010), Archetti et al. (2011a), Derigs et al. (2010), Silva et al. (2015)	2.50	400.10	0.00	0.41	0.00	300	0.00	–	0.00	152.42	60	13,280.00	0.00	25.14
SD12	80	7213.62	Aleman and Hill (2010), Silva et al. (2015)	2.57	408.30	0.00	0.84	0.09	300	0.79	–	0.00	210.71	61	7259.46	0.64	24.08
SD13	96	10,105.86	Archetti et al. (2011a)	2.58	404.50	0.05	1.20	1.70	300	0.00	–	0.05	189.45	72	10,110.58	0.05	27.14
SD14	120	10,717.53	Silva et al. (2015)	2.85	5021.70	0.80	2.31	0.68	3600	0.35	–	0.00	479.85	92	10,771.54	0.51	33.54
SD15	144	15,094.48	Silva et al. (2015)	1.17	5042.30	0.39	3.20	0.39	3600	0.40	–	0.00	731.98	111	15,250.13	1.03	41.46
SD16	144	3379.33	Archetti et al. (2011a)	2.06	5014.70	1.99	7.59	0.06	3600	0.00	–	0.06	930.72	114	3553.32	5.15	17.95
SD17	160	26,493.56	Aleman and Hill (2010)	0.65	5023.60	0.00	7.27	0.16	3600	0.20	–	0.01	577.29	121	26,547.06	0.20	51.29
SD18	160	14,202.53	Silva et al. (2015)	2.42	5028.60	0.85	27.95	1.88	3600	0.93	–	0.00	834.60	123	14,320.66	0.83	50.08
SD19	192	19,995.69	Silva et al. (2015)	2.82	5034.20	0.81	11.95	2.12	3600	1.07	–	0.00	1524.67	149	20,251.89	1.28	52.80
SD20	240	39,635.51	Silva et al. (2015)	1.95	5053.00	0.22	11.02	1.85	3600	0.67	–	0.00	1563.38	182	39,678.10	0.11	80.94
SD21	288	11,271.06	Derigs et al. (2010)	1.96	5051.00	1.67	111.56	0.00	3600	1.95	–	0.66	5034.56	216	11,631.67	3.20	105.20
Average				1.61	2115.57	0.51	8.91	0.67	1557.14	0.30	99.79	0.05	591.62			0.71	27.29

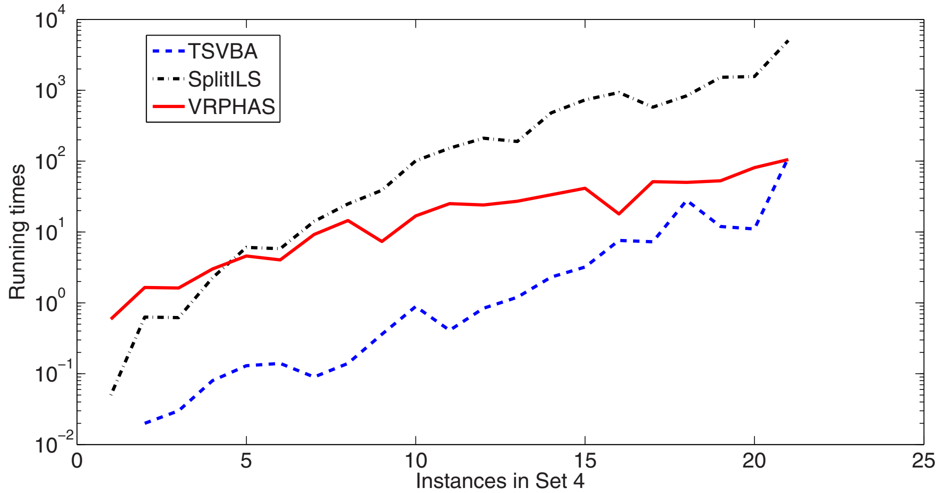

Fig. 1. Running times of TSVBA, SplitILS, and VRPHAS on instances in Set 4.

5. Conclusions

We propose a novel approach to solve the SDVRP. Each demand is split into several small pieces, and some dummy copies of the corresponding customer are generated. This idea allows us to transform the SDVRP into a CVRP and we can use any CVRP solver to solve this problem. Our computational experiments have demonstrated that VRPHAS improves the nonsplit solutions and, for the 82 benchmark instances, it is much faster than the state-of-the-art algorithms and often produces solutions with comparable quality. In addition, VRPHAS is simple to implement, so that anyone with a free or commercial CVRP solver can apply it.

Acknowledgments

This paper is supported in part by Humanity and Social Science Youth Foundation of Ministry of Education of China under grant no. 13YJC630010.

References

Aleman, R., Hill, R., 2010. A tabu search with vocabulary building approach for the vehicle routing problem with split demands. *International Journal of Metaheuristics* 1, 1, 55–80.

Aleman, R., Zhang, X., Hill, R., 2010. An adaptive memory algorithm for the split delivery vehicle routing problem. *Journal of Heuristics* 16, 3, 441–473.

Archetti, C., Bianchessi, N., Speranza, M., 2011a. A column generation approach for the split delivery vehicle routing problem. *Networks* 58, 4, 241–254.

Archetti, C., Bianchessi, N., Speranza, M., 2014. Branch-and-cut algorithms for the split delivery vehicle routing problem. *Annals of Operations Research* 238, 3, 685–698.

Archetti, C., Feillet, D., Gendreau, M., Speranza, M., 2011b. Complexity of the VRP and SDVRP. *Transportation Research Part C* 19, 5, 741–750.

Archetti, C., Savelsbergh, M., Speranza, M., 2006a. Worst-case analysis for split delivery vehicle routing problems. *Transportation Science* 40, 2, 226–234.

Archetti, C., Speranza, M., 2012. Vehicle routing problems with split deliveries. *International Transactions in Operational Research* 39, 1, 3–22.

Archetti, C., Speranza, M., Hertz, A., 2006b. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science* 40, 1, 64–73.

Archetti, C., Speranza, M., Savelsbergh, M., 2008. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science* 42, 1, 22–31.

Belenguer, J., Martinez, M., Mota, E., 2000. A lower bound for the split delivery vehicle routing problem. *Operations Research* 48, 5, 801–810.

Berbotto, L., García, S., Nogales, F., 2014. A randomized granular tabu search heuristic for the split delivery vehicle routing problem. *Annals of Operations Research* 222, 1, 153–173.

Boudia, M., Prins, C., Reghioui, M., 2007. An effective memetic algorithm with population management for the split delivery vehicle routing problem. In Bartz-Beielstein, T., Blesa Aguilera, M., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M. (eds) *Hybrid Metaheuristics*, Vol. 4771 of *Lecture Notes in Computer Science*. Springer, Berlin/Heidelberg, pp. 16–30.

Burrows, W., 1988. The vehicle routing problem with loadsplitting: a heuristic approach. Proceedings of the 24th Annual Conference of the Operational Research Society of New Zealand, Auckland, pp. 33–38.

Campos, V., Corberán, A., Mota, E., 2008. A scatter search algorithm for the split delivery vehicle routing problem. In Fink, A., Rothlauf, F. (eds) *Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management*, Vol. 144 of *Studies in Computational Intelligence*. Springer, Berlin/Heidelberg, pp. 137–152.

Chen, S., Golden, B., Wasil, E., 2007. The split delivery vehicle routing problem: applications, algorithms, test problems, and computational results. *Networks* 49, 4, 318–329.

Derigs, U., Li, B., Vogel, U., 2010. Local search-based metaheuristics for the split delivery vehicle routing problem. *Journal of the Operational Research Society* 61, 9, 1356–1364.

Dror, M., Trudeau, P., 1989. Savings by split delivery routing. *Transportaion Science* 23, 2, 141–145.

Dror, M., Trudeau, P., 1990. Split delivery routing. *Naval Research Logistics* 37, 3, 383–402.

Groër, C., 2011. VRPH. Available at http://www.coin-or.org/projects/VRPH.xml (accessed August 1, 2014).

Groër, C., Golden, B., Wasil, E., 2010. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation* 2, 2, 79–101.

Jin, M., Liu, K., Eksioglu, B., 2008. A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters* 36, 2, 265–270.

Moreno, L., de Aragão, M.P., Uchoa, E., 2010. Improved lower bounds for the split delivery vehicle routing problem. *Operations Research Letters* 38, 4, 302–306.

Silva, M., Subramanian, A., Ochi, L., 2015. An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research* 53, 234–249.