A TSP tabu search heuristic algorithm for the split delivery vehicle routing problem

Hao Xiong¹, Huili Yan²

Abstract The split delivery vehicle routing problem (SDVRP) is a relaxed problem of the classical capacitated vehicle routing problem (CVRP), which allows that each customer can be visited for multiple times while the CVRP allows only once. And it is also an NP-hard problem. In this paper, a tabu search heuristic algorithm with TSP cutting and splitting based on a bi-level program model of SDVRP is proposed. The bi-level programming model redefined the objective function of the SDVRP. The whole cost of the solution can be seemed as the cost of a big TSP path and the increased cost of cutting and splitting. So, the algorithm includes three phases. First, a big TSP path including the distribution depot and all the customers is constructed. Then, the big TSP is cut and split. And in the third phase, the algorithm solves the traveling salesman problem within each route to improve the solution. Finally, a computational study shows that the proposed method can really solve the SDVRP effectively on the benchmark instances in the SDVRP literature.

Keywords Vehicle routing problem; Split deliveries; Tabu search; Random cutting

1 Introduction

The split delivery vehicle routing problem (SDVRP) is a variant of classical Capacity Vehicle Routing Problem (CVRP): routes with minimum total cost must be determined for a fleet of equally capacitated vehicles based at one depot serving the demand of geographically dispersed customers. It differs from the CVRP in that each customer can be visited more than once. This relaxation of the classical CVRP makes the model more realistic and can lead to important savings on the total solution cost as well as a reduction in the total number of vehicles. Moreover, individual demands larger than the vehicle capacity are allowed and there is always a feasible solution using the minimum number of vehicles, which does not happen in the CVRP.

The SDVRP was introduced by Dror and Trudeau (1989, 1990). A complete survey of the SDVRP can be found in Archetti and Speranza (2012). As the classical CVRP, the SDVRP is also NP-hard (Dror and Trudeau 1990). Therefore, the exact algorithms cannot solve the problems with large size.

Because of the already mentioned complexity of the problem, heuristic procedures

Foundation: Natural Science Foundation of China (Grant No. 71461006, No.71461007 and No.71761009), the Natural Science Foundation of Hainan Province (No. 20167239), the project of Hainan province philosophy and social sciences (No. HNSK(QN)15-4), and the Education fund item of Hainan province (Hnky2015ZD-7, Hnky2015ZD-2).

E-mail: xh_xionghao@126.com

¹ Hao Xiong: Professor, School of Economics and Management, Hainan University, Haikou, 570228, China

² Huili Yan: Associate Professor, School of Tourism, Hainan University, Haikou, 570228, China
are more often found in the literature. In Dror and Trudeau (1989) and Dror and Trudeau (1990), a first local search approach is presented. Then, tabu search heuristics are used in Ho and Haugland (2004), Archetti et al. (2006b) and Aleman and Hill (2010). In Ho and Haugland (2004) the authors compare the saving between solutions with and without split delivery in a model with time windows. A tabu search with only two procedures named Order Routes and Best Neighbor is implemented in Archetti et al. (2006b). They also add an improving phase using the GENIUS algorithm developed in Gendreau et al. (1992) and a k-split cycles elimination procedure. They compare their results with Dror and Trudeau (1989) and improve almost all the considered instances. Finally, in Aleman and Hill (2010) a tabu search and a learning procedure are developed. The method is based on a set of initial solutions that are used to build a new solution with higher quality. The generated solutions are improved with a variable neighborhood descendent procedure previously introduced in Aleman et al. (2010).

Other meta-heuristics based techniques can be found in Campos et al. (2008), where the first algorithm based on a scatter search methodology for this problem is proposed, and in Boudia et al. (2007), where a memetic algorithm with population management is implemented by combining a genetic algorithm with local search for intensification and diversification. In Archetti et al. (2008), the tabu search procedure of Archetti et al. (2006b) is used to identify which part of the solution space has higher probability of containing a good solution. After the identification, an integer program is run to obtain improved feasible solutions. Another hybrid technique can be found in Chen et al. (2007), where a mixed integer program is combined with a record-to-record travel algorithm to produce high quality solutions. Finally, a review on different techniques for solving the SDVRP can be found in Aleman et al. (2009). This last paper also shows a new diversification methodology.

In this paper, we present a tabu search heuristic that we call TSP Cutting Tabu Search (TSPTS). A big TSP route planning including all the customers and the depot will be transformed to a SDVRP solution by the cutting and splitting on the TSP, which based on the idea of an endpoint mixed integer program in SI Chen and Bruce Golden(2007). The 3-opt local search method is used to search the big TSP, which can reduce the necessary computational time to explore the neighborhoods of a clusters solution. An important idea is that this algorithm uses constant cutting rule to cut the big TSP to many sub-routing clusters whose whole demand is satisfied the limited of the vehicle capacity. We define a split threshold for the last customer in each route by introducing information of the remaining vehicle capacity of the route and the distant between the customer and the depot based on known properties of the problem and on the triangle inequality. This split threshold focus on making “good” splits of the demands, whenever splitting is necessary. According to our knowledge, this paper is the first that uses these idea to solve the Split Delivery Vehicle Routing Problem. After the 3-opt local search, some satisfied TSP is found, the TSP will be cut and split to several routes, and each routes will be improved by TSP search. Finally, we compare the performance of our method with the best known results found in the literature. The results show that TSPTS is very competitive with the existing methods.
The rest of the paper is structured as follows: Section 2 introduces the mathematical formulation for the SDVRP. Section 3 describes the TSPTS. The computational results are exposed in Section 4. Finally, some conclusions and further research directions are given in Section 5.

2 Mathematical formulation

The solution of the SDVRP will be constructed from a big tour include every customer, which is start from the depot and back to it. Each arc of the big tour is allowed to be cut. And the big tour will be transformed to trips, whose total demands should not exceed the capacity of the vehicle. After cutting, the cost of the route planning will possibly increase as the endpoints of the cutting arc will connect to the depot. The increase cost equal to the saving cost of the endpoints as in the Clarke and Wright algorithm.

In this section, We formulate a bi-level program of TSP cutting model. Let \( C=\{1,2,\ldots,n\} \) be the set of customers, each customer \( i \) with a positive integer demand \( d_i \). The SDVRP is defined on a directed graph \( G=(V,A) \), where \( A \) is the set of arcs \( \{(i,j)/i,j \in V, i \neq j\} \) and \( V=\{0\} \cup C \) is the set of nodes, with 0 representing the depot (with no demand). Each arc \( (i,j)/i,j \in V \) has a non-negative travel cost \( c_{ij} \) (usually distances). These costs are assumed to satisfy the triangle inequality. A fleet of \( K \) identical vehicles, each with capacity \( Q \in \mathbb{Z}^+ \), is available at the depot. We assume that \( K \) is equal to \( K_{\text{min}} \), the minimum number of vehicles needed to serve all the customers, which is determined as

\[
K_{\text{min}} = \left\lceil \frac{d(C)}{Q} \right\rceil
\]

where \( d(C) \) is the sum of all the demands. When the big TSP is get and cut, we will get some other parameters: Let \( R=\{1,2,\ldots,r\} \) be the set of the trips after TSP cutting. \( w_{ij} \) is the saving cost of the customers \( i \) and \( j \). \( p_i \) is the number of the customers in the trip \( i \). \( q_{ij} \) is demand of the customers \( j \) in the trip \( i \).

Defining variables \( x_{ij} \) as binary variables that take value one if vehicle travels directly from \( i \) to \( j \) (and zero otherwise) and \( y_{ij} \) as binary variables that take value one if arcs \( (i,j) \) is cut (and zero otherwise), the vehicle flow formulation for the SDVRP is:

\[
U: \text{Minimize } \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} + \sum_{i=0}^{n} \sum_{j=0}^{n} w_{ij} y_{ij}
\]  

(1)

\[
\sum_{i=1}^{n} x_{ij} \geq 1; \forall j \in V
\]  

(2)

\[
\sum_{i=1}^{n} x_{ji} = 1;
\]  

(3)

\[
\sum_{j=0}^{n} x_{j0} = 1;
\]  

(4)

\[
\sum_{i=1}^{n} x_{ih} - \sum_{j=1}^{n} x_{jh} = 0; \forall h \in C
\]  

(5)
\[ x_{ij} \in \{0,1\}; \forall i, j \in C \]  \hspace{1cm} (6)

\[ D: \text{Minimize} \sum_{i=0}^{n} \sum_{j=0}^{n} w_{ij} y_{ij} \]  \hspace{1cm} (7)

\[ \sum_{j=0}^{n} q_{ij} \leq Q; \forall i \in R \]  \hspace{1cm} (8)

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} y_{ij} > K-1 \]  \hspace{1cm} (9)

\[ y_{ij} \in \{0,1\}; \forall i, j \in C \]  \hspace{1cm} (10)

The first-stage model \( U \) is a clustering sub-problem to assign the demand to the vehicles without considering the capacitated limitation. The objective function (1) is the total transportation cost of big tour and the saving cost of the cutting arcs. Constraints (2) state that every customer must be visited by at least one vehicle while constraints (3) and constraints (4) impose that exactly the big tour is start and end at the depot. Constraints (5) indicate that, if vehicle visits node \( h \), then it must leave it (flow conservation constraints). The second-stage model \( D \) guarantees that a minimum cutting solution is done on the big TSP tour. Constraints (8) limit to \( Q \) the maximum load of each vehicle while constraints (9) ensure that the big tour should be cut to \( K \) trips.

3 Solution method

3.1 General Principles of TSPTS

In this section we present a tabu search heuristic with TSP cutting and splitting that we called TSP Cutting and Splitting Tabu Search (TSPTS) designed for the SDVRP. The Big Tour, or the big TSP tour called here, is a tour with the \( n \) clients and the depot. Then, a procedure called cutting and splitting is required to cut and split it into trips to get a feasible SDVRP solution.

Given \( c_{ij} \) is the distance between point \( i \) and point \( j \). If the \( \{c_{ij}\} \) matrix satisfies the triangular inequality, then no two routes in the optimal solution of SDVRP can have more than one split demand point in common. So, for each two adjacent trips transformed from the big TSP route, if they have a common split point, there are two possible constructions: (1) The split point is on the common arc of the adjacent two trips; (2) the split point is not on the common arc. Both structures are illustrated in Figure 1, which present examples of the two adjacent trips having a common split point. In Figure 1, there are 9 customers with certain demand in the bracket. Vehicle capacity is 10. The split point is customer 4. Observing the Figure 1(a), route R1 (0-1-2-3-5-4-0) and R2 (0-4-7-8-9-6-0) can be easily get from the big route (0-1-2-3-5-4-7-8-9-6-0) by increasing a round-trip from the split point and depot because the split is on the common arc of the two trips. However, for the second case, the route R3 and R4 can not be directly get from the big TSP route. Fortunately, the route R3 and R4 can be transformed by the route R1 and R2. So, in this case, we can first get the route R1 and R2 from the big TSP tour, then transformed them to route R3 and R4.
Symposium on Service Innovation in Big Data Area

Figure 1 Example illustrating the structure of the trips with a common split point

(a) split point is on the common arc of the two trips
(b) split point is not on the common arc of the two trips

So, Tabu search with the TSP Cutting and Splitting is a diversification strategy within the tabu search framework that explores the big TSP tour connected from trips of the SDVRP solution. And the trips knowledge representing solution attributes is stored in big TSP structures. Cutting and splitting is a solution formation concept to decompose the big tour in a feasible and optimized way. After cutting and splitting, the tabu algorithm also must make an improvement to avoid local optimality when the split point is not on the common arc of the two adjacent trips of the solution as showed in figure 1(b).

3.2 Tabu search for the big TSP tour

Tabu search for the big TSP tour is a global search procedure that solves problems by certain evolutions of general TSP tabu search algorithm. As the final solution of the SDVRP is the cutting and the splitting of a big TSP tour, the fitness function for the big TSP tour is set as the cost of the big TSP route and the cost of the cutting and splitting. So, the local search for the big TSP tour is focus on the whole construct of the TSP route while for the general TSP is care about every position of the TSP route.

A good local search process for the SDVRP should ensure to explore more new TSP tour much different with the previous one. That is, an efficient local search can not only change the internal customers’ orders of each trip but also exchange customers between trips. On the contrary, it is not a good local search method if it always only change the internal orders of the customers in a trip, such as: Relocate, Exchange and 2-opt. then customers’ orders maybe recover again after an improvement process of TSPTS algorithm. We have tested the quality of the feasible solutions obtained by these four procedures (Relocate, Exchange, 2-opt and 3-opt) on the set of test instances used in this paper. The results indicate that the values of feasible solutions obtained by the former three methods are always worst than the values obtained by the 3-opt procedure. So, we use the 3-opt as the local search operator for the tabu search algorithm of the big TSP tour.

3-Opt is one of the most famous local search algorithms. This move deletes three edges, thus breaking the tour into three paths, and then reconnects those paths in the other four possible ways, and then evaluating each reconnection method to find the optimum one. This process is then repeated for a different set of 3 connections. See Figure 2.
3.2 Cutting

From a big TSP tour it is always possible to obtain \( k \) routes through the cutting and splitting procedures.

Let us first renumber the clients so that the big TSP tour is 0-1-2-…-(n-1)-n-0. The first vehicle leaves the depot to serve the customer 1 and services, successively, clients 2 and 3 up to client \( i \) for which the total demand serviced of this route will equal or over the vehicle’s capacity. Client \( i \) is either completely serviced or its demand is split between routes 1 and 2. In the first case, edges \((i,0)\) and \((0,i+1)\) are added, corresponding to the last edge in route 1 and the first one in route 2. In the second case, edge \((i,0)\) is added twice and route 2 continues with edge \((i, i+1)\). The feasible solution finally obtained satisfies the two structural properties of the optimal SDVRP solution, i.e., the total number of splits is less than the number of routes and any two routes have, at most, one client in common.

3.3 Splitting

Considering the splitting procedure, not every last customer of the route is split even if the demand can’t be served fully by vehicle. We observe that customer with high demand have a high probability of being split demand of high quality solutions. In order to define what a “split” demand is, a split threshold value \( g \) is defined as:

\[
g^i_k = \beta \frac{C_k}{Q} + \lambda \frac{d_i - C_k}{d_i}
\]

Where \( \beta \) and \( \lambda \) is a positive parameter, \( C_k \) is the spare capacity of the trip \( k \) before customer \( i \) is assigned, \( d_i \) is the demand of customer \( i \) decided to assign to the trip \( k \). When \( g^i_k \geq \Delta \), then the customer \( i \) in trip \( k \) should be split; otherwise, it should be allocate to the trip \( k+1 \). Note that the split threshold is defined for each trip because it depends on the remaining capacity of each route \( k \) and it is calculated at each cutting of the big TSP tour.

3.4 Improving

After cutting and splitting procedures, the customers’ order in the trip unchanged as in the TSP tour. As the split demand is not always at the end of the route, some improvement search in the route could be executed in each route, such as exchange algorithm, 2-opt algorithm. 3-opt neighborhood search algorithm can also be used.
when the number of customers of the sub-route is large.

### 3.5 Algorithm Overview

The general structure of the TSPTS can now be summarized in algorithm 1. The initial big TSP tour can be random generalized or obtained with the Clarke and Wright heuristic. The cutting and splitting procedures are used to adjust the TSP tour and get the initial solution $S_0$.

The main tabu search loop performs a fixed number of cycles, maximum cycles and returns the best solution $S_i$ at the end. Each basic iteration, the big TSP tour undergoes $M$ times the 3-opt local search with a given tabu list. And the resulting neighbors are $M$ big TSP tours. Then the 3-opt tabu list be updated. Each big TSP tour is cut and split to construct the feasible solution. The split threshold $g_k$ is calculated at the end of each trip cut from the big TSP tour.

To sum up, the structure of the algorithm is the following:

<table>
<thead>
<tr>
<th>Table 1 The algorithm overview</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase 1: TSP tabu search with cutting and splitting</strong></td>
</tr>
<tr>
<td><strong>Step 1:</strong> Generalize an initial big TSP tour $T_0$. Cutting and splitting the $T_0$ to get an initial solution $S_0$. Calculate the cost of the $S_0$ to be $C_0$.</td>
</tr>
<tr>
<td>$Bestsolution \leftarrow S_0$</td>
</tr>
<tr>
<td>for $i = 1:n$ do</td>
</tr>
<tr>
<td>accumulated the demands of route $k$ to be $Q_i$</td>
</tr>
<tr>
<td>if $Q_i \leq Q$ &amp; $Q_i + q_i &gt; Q$ then</td>
</tr>
<tr>
<td>calculate the threshold $g_k^i$</td>
</tr>
<tr>
<td>if $g_k^i &gt; \Delta$, then</td>
</tr>
<tr>
<td>split the customer $i$</td>
</tr>
<tr>
<td>and edge $(i,0)$ is added both in route $k$ and $k+1$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>broken the edge $(i,i+1)$</td>
</tr>
<tr>
<td>and edges $(i,0)$ and $(0,i+1)$ are added in route $k$ and $k+1$</td>
</tr>
<tr>
<td>endif</td>
</tr>
<tr>
<td>endfor</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Do $M$ times the 3-opt local search operators on the best big TSP $S_i$ from last generation to get $M$ TSP tour $(T_1^i, T_2^i, \cdots, T_M^i)$; then, cutting and splitting every new big TSP tour $S_j^i$; and select the best solution and update the Tabu list.</td>
</tr>
</tbody>
</table>
for $i = 1$: max generations
for $j = 1$: M do

$T^i_j \leftarrow$ $3$-opt local search
$S^i_j \leftarrow$ cutting and splitting

choose the best solution($S_{best}^i$) and update the tabu list
endfor
if $C_{S_{new}} < C_{Best_{solution}}$ then

$Best_{solution} \leftarrow S_{best}^i$
endif
endfor

Phase 2: Improvement of the solution found by the tabu search phase

Step 3: Improve each individual route of final solution given by the phase 1.

4 Computational Experiments

This section contains computational experiments to evaluate the performance of TSPTS. The algorithm was implemented in Matlab 9.0 on a PC Core i5, 4G RAM, CPU 2.60 GHz.

In order to evaluate the performance of the proposed algorithm, we compare the solutions obtained with the state-of-the-art heuristics for solving the SDVRP. We use four different sets of instances that have been tested in the literature. The sets of instances and the papers that use them are the following:

- Set 1: Two instances of Liu Wangsheng et al. (2012). The numbers of customers are 15 and 20. The vehicle capacities are 500 and 5.

- Set 2: An instance of Fanchao et al. (2010). The vehicle capacity is $Q=1$ for the 36 instances and the demands are from 0.1 to 1.4 units. There are two customers’ demands exceeded the vehicle capacity.

- Set 3: 6 instances of Archetti et al. (2006b). The set considers the problems 01 from Gendreau et al. (1994). The number of customers is 50. Five additional sets of instances are created by changing the demands of the customers with a lower bound and upper bound, $\alpha$ and $\gamma$ respectively, expressed as a fraction of the vehicle capacity $Q$ and $\alpha \leq \beta$. Thus, the demand $d_i$ of customer $i$ is:

$$d_i = \alpha Q + \delta (\gamma - \alpha)Q$$

for some random value $\delta \in (0,1)$. Therefore, the demand $d_i$ of customer $i$ is set randomly in the interval $[\alpha Q, \gamma Q]$. The following lower and upper bound combinations are used to construct the demands of these new instances ($\alpha$, $\gamma$) = (0.1,0.3), (0.1,0.5), (0.1,0.9), (0.3,0.7) and (0.7,0.9). These instances are solved in. In Belenguer et al. (2000), Archetti et al. (2006b), and Aleman et al. (2009) solutions for these instances are found.

Our algorithm has 8 parameters: (1) the tabu list size ($t$), (2) candidate list size ($m$), (3) maximum number of consecutive iterations $N$, (4) the split threshold coefficient ($\beta$,
\( y, \Delta \). Based on a balance between solution quality and computational time, we selected the following values based on calibration experiments:

\[
t = \sqrt{n(n-1)/2}, \quad m = n(n-1)/2, \quad N = n(n-1)/4, \quad \beta = 0.7, \quad \gamma = 0.3, \quad \Delta = 0.5.
\]

The results and the comparative results for Set 1 and Set 2 are shown in Table 2 and 3.

### Table 2 The results for Set 1 and Set 2

<table>
<thead>
<tr>
<th>Data set</th>
<th>Solution</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1(a)</td>
<td>0-8-5-0-5-13-0-13-6-0-15-0-2-11-3-0-9-0-7-12-0-12-1-0-1-10-4-14-0</td>
<td>1563.2</td>
</tr>
<tr>
<td>Set 1(b)</td>
<td>0-20-11-6-0-17-3-4-14-0-5-0-8-1-0-18-0-10-7-0-2-12-9-0-19-15-16-13-0</td>
<td>163.68</td>
</tr>
</tbody>
</table>

### Table 3 Comparing the TSPTS versus CA and TSA of Liu et al. (2012) for Set 1 and Set 2

<table>
<thead>
<tr>
<th>Data set</th>
<th>TSPTS</th>
<th>CA</th>
<th>TSA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k</td>
<td>z</td>
<td>t</td>
</tr>
<tr>
<td>Set 1(a)</td>
<td>9</td>
<td>1563.2</td>
<td>0.37</td>
</tr>
<tr>
<td>Set 1(b)</td>
<td>8</td>
<td>163.7</td>
<td>1.0</td>
</tr>
<tr>
<td>Set 2</td>
<td>15</td>
<td>312.7</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Table 3 summarizes the results shown in Table 2. It gives the average number of vehicles used (k), the average solution cost (z) and the average computing time in seconds (t) on each subset. Considering the small problems with up to 40 customers, the problem could be solved in a few seconds with better quality solutions.

Table 4 shows the best known solution produced by TSVBA in the problem set of Archetti et al. (2006) and a comparison with other existing approaches. The existing algorithms are the scatter search (SS) of Mota et al. (2007), A Tabu Search with Vocabulary Building Approach (TSVBA) of Aleman et al. (2007), the memetic algorithm with population management (MA|PM) of Boudia et al. (2007), the tabu searches (Splitabu-DT) of Archetti, Hertz, and Speranza (2006), and the hybrid algorithm (EMIP+VRTR) of Chen et al. (2007).

For each problem, table 4 shows the size of the set, the best solutions, by all the approaches. And table 5 shows the comparison results. Results in this table and bold fonts show a clear dominance of TSPTS over the SS of Archetti et al. (2006); TSPTS also improves TSVBA in two cases. And the distance between our solution and the best solution is bigger than 9%. It is important to highlight the fact that TSPTS produces the final solutions just in a few seconds.

### Table 4 Computational results on instances of Archetti et al. (2006)

<table>
<thead>
<tr>
<th>Data set</th>
<th>TSPTS</th>
<th>SS</th>
<th>TSVBA</th>
<th>MA</th>
<th>PM</th>
<th>Splitabu-DT</th>
<th>EMIP+VRTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01-00</td>
<td>532.28</td>
<td>531.02</td>
<td>527.67</td>
<td>524.61</td>
<td>533.55</td>
<td>524.61</td>
<td></td>
</tr>
<tr>
<td>P01-1030</td>
<td>786.89</td>
<td>769.60</td>
<td>753.98</td>
<td>751.41</td>
<td>761.40</td>
<td>723.57</td>
<td></td>
</tr>
<tr>
<td>P01-1050</td>
<td>1026.3</td>
<td>1025.91</td>
<td>1023.24</td>
<td>988.31</td>
<td>1008.67</td>
<td>943.86</td>
<td></td>
</tr>
<tr>
<td>P01-1090</td>
<td>1528.5</td>
<td>1580.77</td>
<td>1530.81</td>
<td>1467.06</td>
<td>1469.92</td>
<td>1408.34</td>
<td></td>
</tr>
<tr>
<td>P01-3070</td>
<td>1508.8</td>
<td>1568.04</td>
<td>1505.38</td>
<td>1477.01</td>
<td>1496.90</td>
<td>1408.68</td>
<td></td>
</tr>
<tr>
<td>P01-7090</td>
<td>2225.8</td>
<td>2312.48</td>
<td>2219.32</td>
<td>2154.35</td>
<td>2165.21</td>
<td>2056.01</td>
<td></td>
</tr>
</tbody>
</table>
Table 5 Compare with the other approaches

<table>
<thead>
<tr>
<th>Data set</th>
<th>Compare with the best</th>
<th>Compare with the worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01-00</td>
<td>7.67 (1.4%)</td>
<td>1.27 (0.2%)</td>
</tr>
<tr>
<td>P01-1030</td>
<td>63.32 (8.7%)</td>
<td>7.29 (2.4%)</td>
</tr>
<tr>
<td>P01-1050</td>
<td>82.44 (8.7%)</td>
<td>0.39 (0.0%)</td>
</tr>
<tr>
<td>P01-1090</td>
<td>120.16 (8.5%)</td>
<td>52.27 (3.4%)</td>
</tr>
<tr>
<td>P01-3070</td>
<td>100.12 (7.1%)</td>
<td>59.24 (3.9%)</td>
</tr>
<tr>
<td>P01-7090</td>
<td>169.79 (8.3%)</td>
<td>86.68 (3.9%)</td>
</tr>
</tbody>
</table>

5 Conclusions and future research

In this paper we have presented a new heuristic method for solving the SDVRP. The primary research result of this paper is a new solution construct method which composes the solution by a big TSP tour with the cutting and splitting processes. And a splitting threshold is introduced to decide the last added customer whether or not split. Considering the spare capacity of the trip and the demands of the customer to be split. Finally, four experiments are carried out and an exhaustive comparison with the other approaches found in the literature has been provided. The results obtained with the TSPTS procedure indicate that it is able to obtain very good feasible solutions with the minimum number of vehicles within reasonable computing times. But when the numbers of customers are over 50, the values of the solutions are not so good, because our procedure was not initially designed for these situations.

For future research, we would like to focus on the split threshold and on the big TSP local search phase. As this paper introduces many new best known solutions for the SDVRP, we think that it would be interesting to develop an exact algorithm where TSPTS is an auxiliary tool to obtain good upper bounds.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 71461006, No.71461007 and No.71761009), the Natural Science Foundation of Hainan Province (No. 20167239), the project of Hainan province philosophy and social sciences (No. HNSK (QN) 15-4), and the Education fund item of Hainan province (Hnky2015ZD-7, Hnky2015ZD-2).

References


