# A Matheuristic Approach for the Split Delivery Vehicle Routing Problem: An Efficient Set Covering-Based Model with Guided Route Generation Schemes

Nurul Huda Mohamed[a], Said Salhi[b], Gábor Nagy[b] and Nurul Akmal Mohamed[a]

[a]Department of Mathematics, Faculty of Science & Mathematics,
Universiti Pendidikan Sultan Idris,
35900 Tanjong Malim, Perak, Malaysia.
Email: {nurul.huda, akmal.mohamed}@fsmt.upsi.edu.my

[b]The Centre for Logistics & Heuristic Optimisation, Kent Business School,
University of Kent, Canterbury CT2 7PE, UK
Email: {S.Salhi, G.Nagy}@kent.ac.uk

## Abstract

The Split Delivery Vehicle Routing Problem (SDVRP) is a relaxed version of the classical VRP where customers can be visited more than once. The SDVRP is also applicable for problems where one or more of the customers require a demand larger than the vehicle capacity. Constructive heuristics adapted from the parallel savings and the sweep methods are first proposed to generate a set of solutions which is then used in the new and more efficient set covering-based formulation which we put forward. An effective repair mechanism to remedy any infeasibility due to the set covering problem is presented. A reduced set of promising routes is used in our model, instead of the original set of routes, proposing and using well defined reduction schemes. This set covering-based approach is tested on large data sets from the literature with encouraging results. In brief, 7 best solutions including ties are found among the 137 SDVRP instances.

Keywords: split deliveries, vehicle routing, set covering, hybrid method, matheuristic.

## 1. Introduction

The Split Delivery Vehicle Routing Problem (SDVRP) is a relaxation of the classical VRP in which a customer can be served by more than one vehicle if it reduces the overall total cost. This relaxation is very important especially for cases where the sizes of the customer orders are nearly as big as the capacity of a vehicle. Although the SDVRP is a relaxation of the VRP, it is also an NP-hard problem, as shown by Archetti and Speranza (2008). This routing problem was first introduced by Dror and Trudeau (1989) where it was found that the total travel distance and the number of vehicles required can be reduced by allowing more than one vehicle to deliver to a customer. It is worth noting that in some situations, it may not be worth to split as this could increase the travelling cost without a reduction in the number of vehicles.

Let $C = \{1,2,..,n\}$ be the set of customers, each customer $i$ has a positive integer demand, $d_i$. The SDVRP can be defined over a graph $G = (V, E)$, where $V = \{0\} \cup C$ is the set of nodes and $E = \{\{i,j\}: i,j \in V, i \neq j\}$ is the set of edges. Node 0 is the depot (with no demand), where a fleet of homogeneous vehicle with capacity $Q$ is located. A travelling cost from $i$ to $j$, $c_{ij}$ is associated with each edge $\{i,j\} \in E$. Each vehicle must start and end at the depot. The vehicle load cannot exceed the vehicle capacity, $Q$. The demand $d_i$,$(i = 1,2, \ldots, n)$ can be delivered by more than one vehicle. The objective is to find a set of routes that minimizes the total travelling cost without violating all these constraints. It is also applicable to problems with customers' demands larger than the vehicle capacity. These types of split routing problems can be applied in many real-world logistical problems.

This problem remained dormant for several years till 2006 when Archetti et al. (2006) revisited it and proposed an efficient and novel tabu search metaheuristic for its resolution. Most of the approaches are heuristic-based methods which include a scatter search method by Mota et al. (2007), a memetic algorithm by Boudia et al. (2007), a ring-based diversification method by Aleman et al. (2009), a variable neighbourhood descent by Aleman et al. (2010), a tabu search with vocabulary building approach (TSVBA) by Aleman and Hill (2010), a local search-based method by Derigs et al. (2010), a randomized granular tabu search by Berbotto et al. (2014), an iterated local search heuristic by Silva et al. (2015) and a priori splitting strategy by Chen et al. (2017). There are however a few exact methods such as the cutting plane method by Belenguer et al. (2000), the branch-and-cut algorithms by Archetti et al. (2011a; 2014) and the set partitioning approach by Archetti et al. (2011b). There are also a few hybrid methods developed for this problem, see Chen et al. (2007) and Archetti et al. (2008). For more details, the reader will find the recent review by Archetti and Speranza (2012) to be interesting, easy to read and very informative.

The contributions of this study include:

(i) The development of an effective and efficient matheuristic, a hybridisation of some constructive heuristics, a repair mechanism and a set covering approach.

(ii) A new and more powerful set covering model, which gives better solutions when there is a computation time limit imposed.

(iii) The design of interesting selection rules for identifying potential routes so to reduce the size of the problem without affecting solution quality.

(iv) The gain of competitive results.

This paper is organised as follows. In Section 2, we give a brief overview of the overall algorithm followed by Section 3 that describes the constructive heuristics which we adopt to generate a set of initial solutions. Section 4 provides the proposed set covering-based formulation and its implementation followed by a section on how to identify promising routes so to reduce the number of routes. Our computational results are presented in Section 6. Our conclusion and highlights of research avenues that we believe to be worth examining in the future are given in the last section.

## 2. An overview of the overall algorithm

The constructive heuristics which are implemented to generate a set of routes are adapted from the saving and the sweep methods which are originally based on the classical VRP and modified slightly to cater for the possibility of splitting.

The modified set covering model that considers the decision variables denoting the proportion of a customer demand on a given route is proposed. This will be compared against the existing classical formulation. The set of generated routes is reduced by identifying good routes only so to accommodate the feasibility of using an ILP solver such as CPLEX. This selection is based on the quality of the solutions where these routes belong to, the route dual information and the frequency of occurrences of the routes. This hybrid heuristic is denoted by (MSN), short for Mohamed, Salhi and Nagy. In brief, the overall algorithm of MSN can be described as follows:

**The MSN Algorithm**

Step 1 Generate a large set of routes using some constructive heuristics (VRP-based and modified ones to cater for split deliveries).

Step 2 Reduce the set of routes using well defined selection criteria.

Step 3 Apply an ILP solver using the new set covering-based formulation with the original set of routes found in Step 1 as well as the set of routes generated in Step 2.

The next three sections will describe the three steps of the MSN algorithm.

## 3. Constructive Heuristics for the SDVRP

Two approaches based on the parallel saving and the sweep method are adapted to construct a large number of initial solutions whose routes, after the removal of duplications, will be used in the modified set covering-based model which we present in the next section.

The first approach consists of two stages namely the construction of the initial VRP solutions in the first stage and then followed by an implementation of a splitting method to relax the problem in stage two. Whereas in the second approach, the solutions are obtained in only one stage with splitting integrated into the search.

A composite heuristic made up of commonly used refinement procedures which include the 2-opt, the swap move and the insertion (intra route and inter routes) is then used as the local search engine to improve upon the initial solutions. These are applied in sequence. Details of these two scheme approaches and the composite heuristic are given below.

### 3.1 Scheme 1 – A Two-stage Splitting Approach

The saving concept is first introduced by Clarke and Wright (1964) and then explored by many studies to solve the VRP and its related problems using heuristics and meta-heuristics approaches. Yellow (1970) modified the classical saving formulae by incorporating a route shape parameter $\lambda$ as follows:

$$s_{ij} = c_{i0} + c_{0j} - \lambda c_{ij}$$

where

$s_{ij}$ refers to the saving by merging customers $i$ and $j$;

$c_{i0}$ is the distance between customer $i$ and the depot which is denoted by 0;

$c_{ij}$ is the distance between customer $i$ and customer $j$.

As split deliveries are allowed in this problem, we solve the problem in two stages in this scheme.

Stage 1 (VRP Solution):

- Construct an initial solution for the VRP without any splitting using the standard parallel saving method.
- Apply the composite heuristic to improve upon each of the solutions.

Stage 2 (Including the Splitting):

- Modify the obtained VRP solution to include split deliveries by using the endpoints procedure (see The End Point Splitting Method).
- Apply the composite heuristic to improve the solution.

These two stages are implemented with various values of the route shape parameter to generate a set of feasible routes.

We generate several solutions with $\lambda \in [0,5]$ starting with $\lambda = 0$ with an increment of 0.2. This implementation was successfully used in the past by Salhi and Rand (1987) for the VRP. We opt for the parallel saving heuristic implementation instead of the sequential saving as the latter produced, in most cases, better results. See Mohamed (2012) for more details.

In Stage 2, this splitting method is implemented right after the VRP solutions are obtained to generate routes where splitting occurred. The idea is to merge two routes which are not fully loaded through their end point customers allowing concurrently splitting. This splitting is performed at one of the other 2 end points used in the combination. We refer to it as the end point splitting method which we call for short EPSM.

**The End Point Splitting Method (EPSM)**

Step 1 Start from a given route which is not fully loaded and compute the best merging of one of its endpoints with another endpoint (say customer $j$) from another route by delivering some of the demand at customer $j$ without exceeding the vehicle capacity constraint. This could lead to customer $j$ being split and served by two routes.

Step 2 Execute this merging.

Step 3 Search for another best merging until the current route is full.

Step 4 Repeat Steps 1 - 3 for the next route until all routes are explored.

**3.2 Scheme 2 – An Integrated Splitting Approach**

The aim here is to obtain a one stage feasible solution, using the following two steps:

- Construct an initial solution for the SDVRP by adapting some constructive methods. Here, we considered the modified parallel saving and the modified sweep methods, both with splitting included.
- Improve the obtained solution using the composite heuristic.

Parallel Savings with Split Deliveries (PSSD)

This method is similar to the classical parallel saving method for the VRP except that:

(i) a customer is allowed to be split when selected by the savings and

(ii) two routes can also be combined even when the total load exceeds the vehicle capacity as long as it does not violate by more than the demand of the closest customer of these routes to the depot.

This choice will allow easily a splitting to be applied on this particular customer. Note that (ii) is similar to using one application of EPSM when the two routes are fixed. The affected customer with its remaining demand will act as a new unassigned customer that will be allocated to a route according to the saving method. This is referred to as PSSD and its main steps are given next.

**The PSSD Algorithm**

Step 1 Create $n$ vehicle routes $(0, i, 0)$ for each $i = 1,2, \dots \dots, n$.

Step 2 Calculate the savings $s_{ij} = c_{0i} + c_{0j} - c_{ij}$ for $i = 1,2, \dots, n$ and $i \neq j$.

Step 3 Order the calculated savings in decreasing order.

Step 4 Starting with the highest savings, $s_{ij}$ check whether there exist two routes that can feasibly be merged.

Step 5 Choose the route containing $i$, either as the first or the last customer in the route. Choose another route containing $j$ as the first or the last customer in the route.

Step 6 Merge these two routes to form a new larger route with $i$ and $j$ acting as the first or the last customer of each route.

Step 7 If these two routes cannot be merged together due to the vehicle capacity constraint. However, if both routes are still not fully loaded, we check the splitting point for each route so that one of their loads is equal to the vehicle capacity. Select the nearest splitting point to the depot as the point to be split. Merge $i$ and $j$ to get one full route, using the farthest splitting point from the depot, which ends or starts at the selected splitting point, while the other route, which also starts or ends at the same selected splitting point, will become smaller.

Step 8 Repeat Step 4 using the next savings until there is no more possible combination left.

The Sweep-based Approach with Split Deliveries (SASD)

The sweep method initially proposed by Gillett and Miller (1974) is also investigated here to generate additional sets of possible routes. The aim is to create a cluster of customers that are geographically close together from an angular viewpoint. We have extended this algorithm by generating all possible routes while allowing splitting. In this implementation, we start from each customer location and use both clockwise and counter-clockwise directions. The sweep-based splitting approach, which we refer to SASD for short, is given next.

**The Sweep-based Approach with Split Deliveries (SASD)**

Step 1 Set the depot coordinate as the starting point. Calculate the angle, $\theta_i$ of each customer $i$, as the relative angle between the depot and the customer location and arrange the angle, $\theta_i$ in ascending order.

Step 2 Starting from the first empty route, assign customers to the route according to counterclockwise (or clockwise) direction until the vehicle capacity is full.

(i) If the last customer on the route is not fully served, split its demand and start the next route with the customer as the first customer in the second route.

(ii) If the last customer is fully served, start the next route with the next customer in the list.

Step 3 Stop when all customers are served.

Step 4 Repeat Steps 2 and 3, starting from the next customer in the list creating $n$ sets of solutions.

Step 5 Repeat Steps 2 to 4 using the other direction.

## 3.3 A Composite Heuristic

As mentioned before, a composite heuristic is used as the local search engine to refine the obtained initial solutions and these refinement procedures are applied in sequence.

The 2-Opt

This procedure starts from a given route, then compute the best edge exchange of two non adjacent edges with other two new edges while maintaining the route structure that improves the original route. Update the exchange and the direction of the arcs connecting these two edges. This process is repeated until no further improvement is possible.

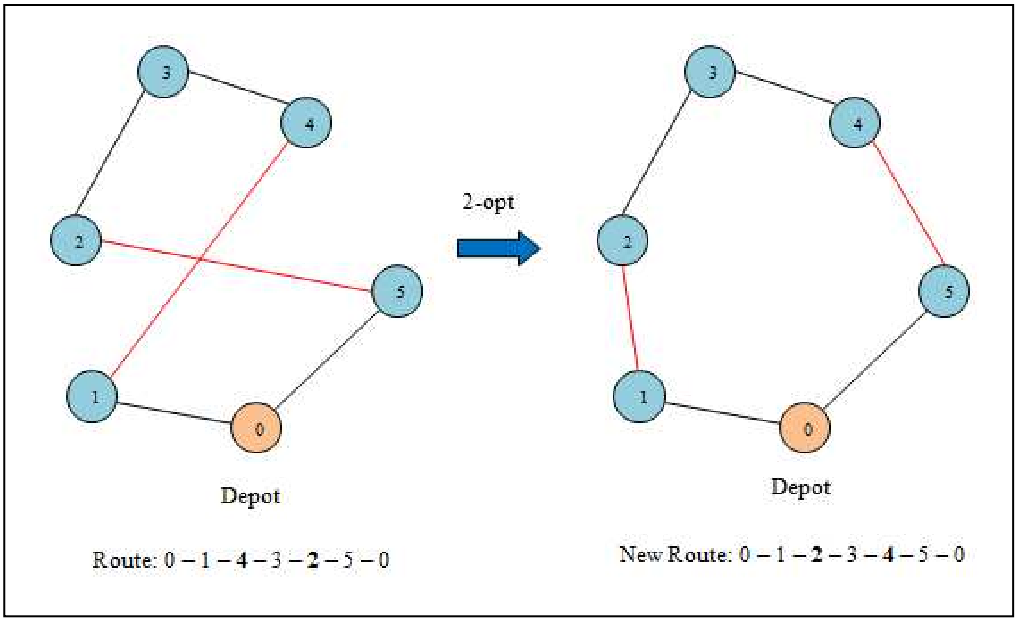


Figure 1: The 2-Opt routine within a route

Figure 1 illustrates an example of a 2-Opt routine within a route by exchanging the positions of two nodes. In the example, the location of customer 4 is exchanged with the position of customer 2. By executing this exchange, the arc that connects these two nodes is diverted, where 4 – 3 – 2 becomes 2 – 3 – 4. The profit from the exchange can be calculated as: $Gain\ =\ c_{12} + c_{45} - c_{14} - c_{25}$. There is a well-known property such that a route should never cross given that the triangular inequality holds and there are no constraints such as time windows.

The Swap Move

This routine involves two routes, where a node $i$ from a given route, say $R_1$ is exchanged with a node $j$ from another route, say $R_2$ excluding the given route ($R_1 \neq R_2$) but not necessary at the same positions. The process starts with removing node $i$ and node $j$ from their original routes, searching for the best possible position to insert $j$ into route $R_1$ and the best feasible position to insert node $i$ into route $R_2$. We implement the best improvement strategy where each pair of nodes for each pair of routes are explored to find the best swap move. Once found, the mode is executed and the process is repeated until no further improvement is possible. An illustrative example is shown in Figure 2.

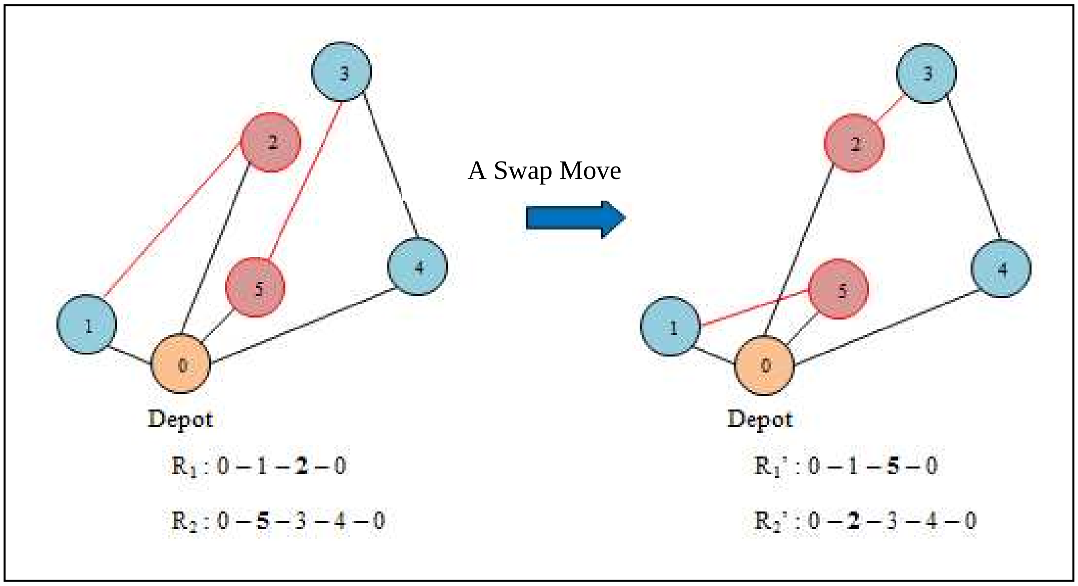


Figure 2: A Swap move inter routes

Figure 2 shows nodes 2 and 5 are removed from their original routes and then inserted into each other's route, node 2 into route $R_2$ and node 5 into route $R_1$.

Insertion (intra route and inter routes)

This routine involves one route (intra route) or two routes (inter routes) at a time, where a node $i$ from a given route, say $R_1$ is removed from the route to be inserted back into the same route at a different position or into another route, say $R_2$. The process starts by removing node $i$ from its original route, searching for the best possible position based on the insertion cost to insert $i$ into any possible route including $R_1$. The insertion move is implemented based on the best improvement strategy where the insertion is only executed after all customer $i$ is explored. The process is repeated until no further improvement is possible.

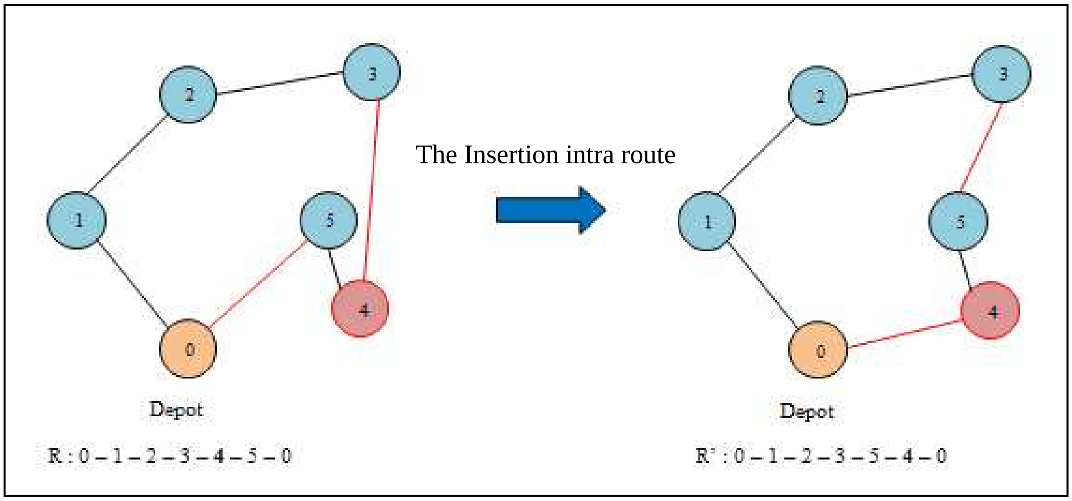


Figure 3: An example of the Insertion move within a route

Figure 3 illustrates an example of this insertion procedure within a route, where node 4 which was in between nodes 3 and 5 is removed from the route before being inserted back into the route in between node 5 and the depot.

Figure 4 on the other hand demonstrates an example of this insertion procedure between two routes, where node 2 from route $R_1$ is removed from the route, and then inserted into route $R_2$.

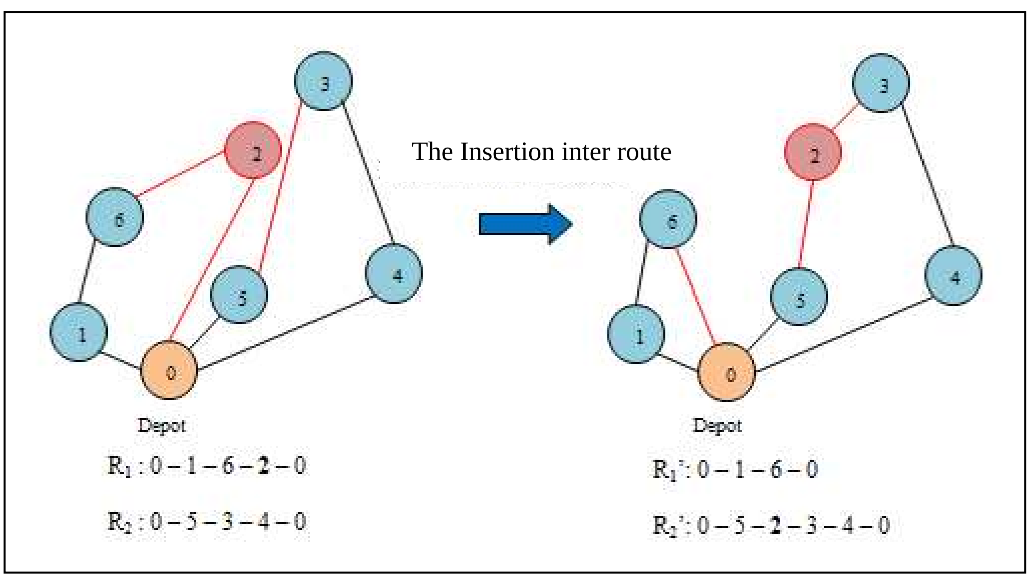


Figure 4: An example of the Insertion between routes

## 4. A Set Covering-based Matheuristic

There are two types of mathematical formulations for the SDVRP namely the classical mixed integer programming and the set covering-based model (SCM). Archetti and Speranza (2008) produce an overview on the studies in the SDVRP where comparisons have been conducted to highlight the benefits and the drawbacks of each model. Note that if the problem is highly constrained (capacity, time windows), the set of routes becomes smaller and hence the SCM becomes more attractive and relatively easier to solve. In this study we will concentrate on the latter formulation.

The SCM is based on a collection of possible feasible routes from which the best feasible solution could then be obtained. In this study, the routes found by the heuristics, as described in the earlier section, will be used as a basis to construct the set of routes. As the set covering model may generate routes with some customers being served more than their required demand due to the constraints (8) and (10), a repair mechanism will be given. In addition, as many routes may be duplicated, a scheme to avoid such duplications will also be introduced. The hybridisation of heuristics and exact method is a novel and powerful approach known as matheuristics. For an overview on heuristic search including matheuristics, see Salhi (2017).

### 4.1 The Original Set Covering-Based Formulation for the SDVRP

The model objective is to design a solution with a set of selected routes from a large set of feasible routes $R$. This is an extension of the Set Partition Problem (SPP) given by Alvarenga et al. (2007) to cater for split deliveries. The model presented by Archetti et al. (2008) and Archetti and Speranza (2008) also uses the following notation and assumptions.

$n$ = the number of customers ($i = 1,2, \dots, n$);

C = the set of customers ($i \in C = \{1, \dots, n\}$, $|C| = n$);

$V$ = the set of nodes, $V = \{0,1, \dots, n\}$ (node 0 denotes the depot), $\{0\} \cup C$;

$d_i$ = the demand of customer $i \in C$;

$c_{ij}$ = the travel cost between customer $i$ and $j$, $\forall\, i, j \in V - \{0\}$ ($c_{ji} = c_{ij}$);

$m$ = the number of vehicles ($l = 1,2, \dots, m$);

$Q$ = the vehicle capacity for each vehicle $l$ ($l = 1,2, \dots, m$);

$y_r^i$ = the quantity of the demand of customer $i$ delivered in route $r$.

$R$ = the set of all possible routes ($r \in R$);

$c_r$ = the travel distance on the route $r$ ($r \in R$) ;

$x_r$ = decision variable, 1 if the route $r$ is considered in the solution and 0 otherwise;

The objective is to choose the subset of routes from $R$ with the least total cost while ensuring that each customer is served at least by one route.

Let ($P_0$) be the original model:

$$
(P_0) \begin{cases}
\min \sum_{r \in R} c_r x_r & (1) \\
\text{s.t:} \quad \sum_{i \in r} y_r^i \leq Q x_r \quad r \in R & (2) \\
\sum_{r \in R : i \in r} y_r^i \geq d_i \quad i \in C & (3) \\
x_r \in \{0,1\} \quad r \in R & (4) \\
y_r^i \geq 0 \quad r \in R; i \in C & (5)
\end{cases}
$$

The objective function (1) is to minimise the total cost of the selected routes. Constraints (2) enforce that a delivery to a customer $i$ on route $r$ can only take place if route $r$ is selected and that the maximum total quantity delivered on a selected route must not exceed the vehicle capacity. Constraints (3) make sure that the demand $d_i$ of customer $i$ is fully satisfied.

Note that if $R$ contains all the possible feasible routes and if it is possible to solve ($P_0$) to optimality then the optimal solution will obviously be guaranteed.

### 4.2 The New Set Covering-based Formulation ($P_1$)

The model formulation used in this study is modified from the original ($P_0$) of Archetti et al. (2008). Several modified models have been studied (see Mohamed, 2012) but we only provide the best one in this paper. Similar to ($P_0$), there are two decision variables namely

$x_r$ and $y_r^i$. However, in this model, $y_r^i$ is restricted to be a fractional variable rather than just non-negative. In the original model, the optimiser decides the quantity to be delivered to each customer $i$ on route $r$ and Archetti et al. (2008) made a useful observation where they were having difficulties in solving this integer problem even with some cuts strengthening introduced. As the quantity delivered to a customer on a route was relaxed, this creates a large search space for the optimiser.

Their observation inspired us to make use of this information so to consider the maximum amount delivered to customer $i$ on route $r$, namely $d_i$. The constraints (2) and (3) have also been modified to reflect for this change.

This modified model which we refer to as ($P_1$) uses the same notations and assumptions as ($P_0$) except for the following: $y_r^i$ represents the proportion of the $i^{th}$customer demand delivered to customer $i$ on route $r$ (i.e., $0 \leq y_r^i \leq 1$) and (2) & (3) are replaced by (7) & (8) respectively.

$$
(P_1)\quad \left\{
\begin{array}{lll}
 & \min \sum_{r\in R} c_r x_r & (6) \\
\text{s.t:} & \sum_{i\in r} d_i y_r^i \leq Q x_r \qquad r \in R & (7) \\
 & \sum_{r\in R: i\in r} y_r^i \geq 1 \qquad i \in C & (8) \\
 & x_r \in \{0,1\} \qquad r \in R & (9) \\
 & 0 \leq y_r^i \leq 1 \quad r \in R; i \in C & (10)
\end{array}
\right.
$$

This set of possible routes is then used to solve the set covering problem (SCP) by calling the optimiser ILOG CPLEX Callable Library.

We have tested this idea on several problem instances and it is proved empirically that this information is very useful. It makes the SCP easier to be solved while producing better quality solutions whenever optimality was not guaranteed within the same amount of CPU time.

### 4.3 Repair Mechanism

As the above models are based on set covering formulations, the solutions obtained may select routes where some customers could be served with more than their required demand. To overcome this shortcoming, a simple but effective repair mechanism is introduced to ensure that every customer receives exactly its demand. This routine besides ensuring feasibility could also reduce, in some cases, the total routing cost. Mathematically, this can obviously be avoided by replacing (8) with equality constraints instead, as in the SPP, but this would require an excessive amount of computational effort.

In brief, for a customer receiving more than its demand, this can lead to this customer being:

(i) either served from one route only or
(ii) this customer remains to be served by the existing number of routes.

In (i) this will systematically lead to some reduction in routing cost whereas in (ii) the corresponding customer request will be adjusted accordingly without any saving in routing cost. Note that these two routes were not part of the set $R$, otherwise they would have been selected. For instance the VRP solutions always fit into (i). Figures 5 and 6 illustrate these two cases. More details including mathematical expressions are available in Mohamed (2012).

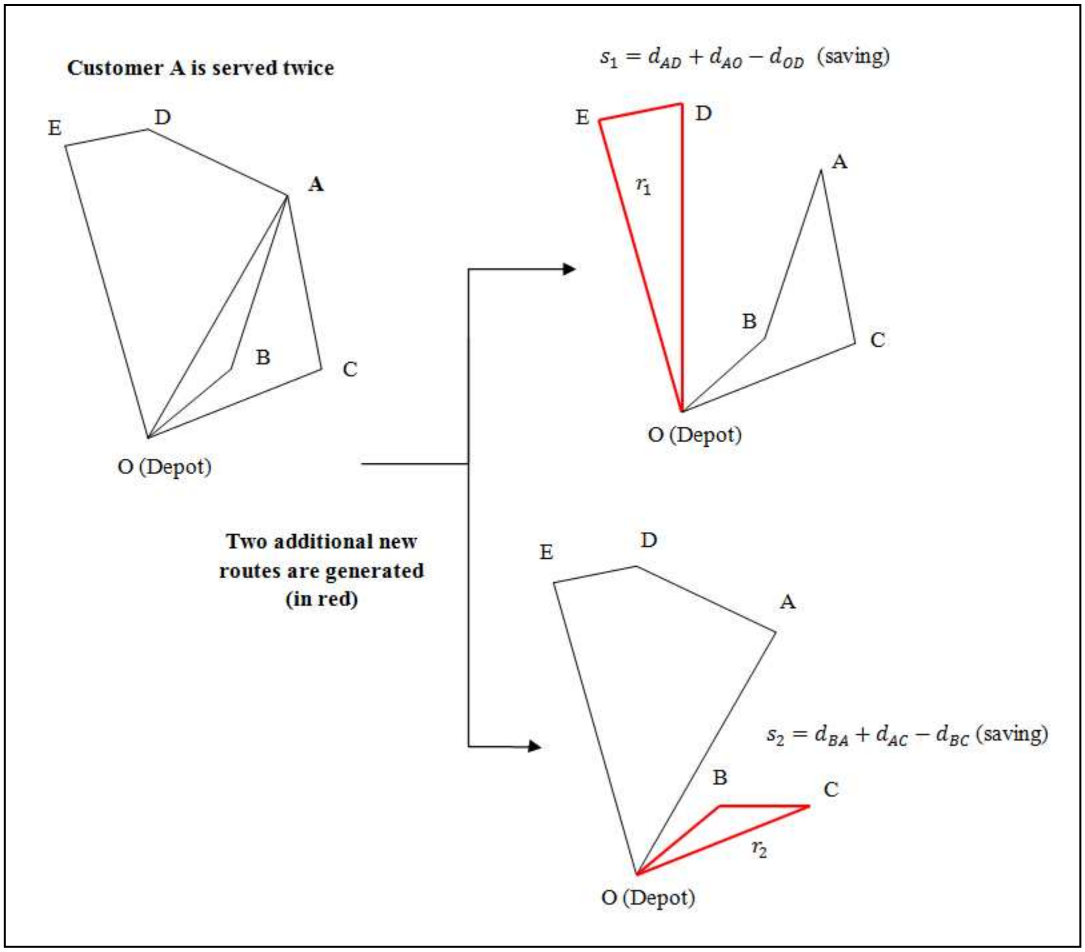


Figure 5: An example of the route generation heuristic for the VRP case (i)

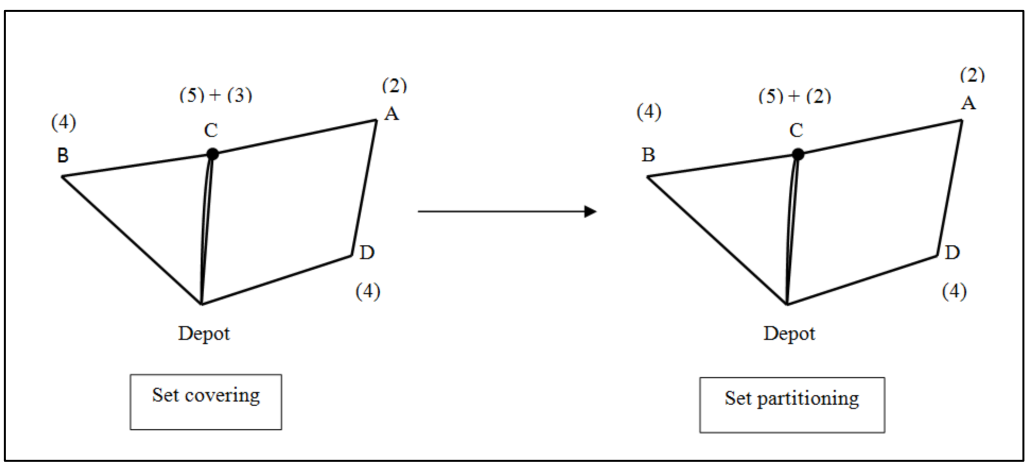


Figure 6: An example of the route generation heuristic for SDVRP case (ii)

### 4.4 A Route Duplication Removal Scheme

Once the set $R$ is obtained, it is then cleaned by eliminating any duplicate route using the following procedure. We achieve this by checking for each route $r$ $(r \in R)$ its total route distance, its route load, its number of customers served and the customers served on the route. Also, if these four attributes happen to be the same, then the route that has its split customer with the highest quantity delivered to it will be stored only. As this scheme is route-based, the pitfall caused by having similar solutions is avoided from the outset as the non-duplicate routes are stored only.

## 5. The Identification of Promising Routes

This obtained set of routes (R) could become too big to be handled by commercial LP/ILP solvers such as ILOG CPLEX. Besides, this large set may also contain many 'not so good' routes. The idea would be to identify a set of 'promising' routes to be solved in MIP using our modified model ($P_1$). The question is how to identify these promising routes? Obviously we could not guarantee that the optimal results are part of the new subset as optimality will only be guaranteed if the set of routes contains all possible routes and the optimiser is run till the end. By restricting the computational time to a maximum of 2 hours, the search area becomes relatively smaller, covering good solutions and hence the solver may be able to find a better solution faster (a good upper bound). We consider the promising routes to be those that

(i) belong to the top best solutions obtained from the heuristics,

(ii) have dual values obtained from the relaxation of the set covering-based formulation to be larger than a certain threshold,

(iii) appear more than twice in the solutions generated by the selected heuristics.

These three selection schemes are briefly outlined next, followed by a scheme that combines them all.

### 5.1 Solution quality-based route selection

Let $Z_0$ be the cost of the best solution found so far from the heuristics and $Z_k$ be the cost of the $k^{th}$ best solution. Any routes contained in the solutions with $Z_k \leq (1+\beta)Z_0$ are included in the new subset. We define this subset as $R_H^{'} = \{r \in R \text{ such that } Z_k \leq (1+\beta)Z_0\}$ where $\beta$ is a threshold parameter (a small positive value close to zero). A pilot test using values of $\beta$, set to 1%, 5%, 10% and 15% is conducted under the time limit of 2 hours. Better solutions are observed with $\beta = 10\%$. It is also observed that a larger $|R_H'|$ does not necessarily guarantee a better solution when a time limit is imposed when using CPLEX. Further detailed can be found in Mohamed (2012).

### 5.2 Dual Values-based route selection

The second way of identifying good routes is based on the routes' dual values related to constraint set (7), in the LP relaxation of ($P_1$). Let $\mu_r$ be the dual price related to route $r$. The idea is then to choose routes with $\mu_r \geq \varepsilon$ ($\varepsilon$ >0). In other words, the new subset is defined as $R_M^{'} = \{r \in R / \mu_r \geq \varepsilon\}$. The question is how to choose the most suitable value of $\varepsilon$? A simple experiment on a sample using several values of $\varepsilon$ is conducted. We tested the cases for $\varepsilon = \bar{\mu}$ , $\varepsilon = \bar{\mu} - \sigma$ and $\varepsilon = \bar{\mu} + \sigma$ with $\bar{\mu}$ and $\sigma$ referring to the average and the standard deviation of the $\mu_r$ $(r \in R)$ respectively.

It is found that in most problem cases, CPLEX running time has been reduced for the case of $\varepsilon = \bar{\mu} + \sigma$ but at the expense of solution quality. When $\varepsilon = \bar{\mu} - \sigma$, the results were found to be rather inferior while reaching the time limit in most instances. The best results were obtained when $\varepsilon = \bar{\mu}$ so $R_M^{'} = \{r \in R / \mu_r \geq \bar{\mu}\}$.

### 5.3 Frequency-based route selection

The third and last scheme of our set reduction is to include those routes which appear more than twice in the set $R$. This is because poor quality heuristic solutions might contain good routes and also routes which appear only once or twice in the set may have happened just by luck. Here, we select the subset as $R_F^{'} = \{r \in R / F_r > 2\}$ where $F_r$ being the frequency of occurrence of route r $(r \in R)$. We also tested the subset $\{r \in R / F_r > 1\}$ to see its effect but without any success. This could be due to the larger feasible region for

CPLEX to explore given the same limited amount of CPU time is imposed. The subset with $F_r > 3$ was also found to be not promising as it is rather small and hence the solution quality was sacrificed with the benefit of a relatively smaller amount of CPU time.

### 5.4 The Combined Scheme

We combined the three selection schemes described earlier to form our set of promising routes as $R'^{*} = R'_H \cup (R'_M \cap R'_F)$. We limit the size of $R'^{*}$ to $\frac{|R|}{2}$ with the following restrictions:

(i) $|R'_H| = \min(0.8|R|, |\{r \in R / Z_k \leq (1+\beta)Z_0\}|)$

(ii) The rest of the routes are then selected if they are found in the two subsets $R'_M$ and $R'_F$ and count for at least $0.2|R|$.

In (i) we opt for $\beta = 10\%$ as the results obtained using this subset alone were found to be better than the other selections. We proceed to fill $R'^{*}$ by using all the routes from $R'_H$ followed by the routes which are in both $R'_F$ and $R'_M$. Note that no route duplication is permitted. In other words, once a route is in $R'^{*}$, it cannot be chosen again from any of the other subsets.

In brief, $R'^{*}$ is then used instead of $R$ in the CPLEX Callable Library to solve ($P_1$). We have tested some combination of the selection schemes on several problem instances and it is proved empirically that the above combination is the best for this SDVRP.

## 6. Computational Results

The constructive heuristics are coded in C++ whereas the Set Covering-based approaches are solved using ILOG CPLEX 12.3 solver with Microsoft Visual C++ interface and the CPLEX Callable Library. Both approaches are executed on a PC with an Intel® CoreTM i7-620M, 2.66GHz processor with 8.0GB of RAM. For simplicity and convenience, a maximum CPU time of 2 hours is capped for each problem instance. If time is not a main concern, better results would be found if the problem is solved optimally using our new

Set Covering formulation. Our methods are tested on the four data sets from the literature namely Archetti et al. (2006), Mota et al. (2007), Chen et al. (2007) and Belenguer et al. (2000).

The summary results including the total cost, the average deviation and the best solution are given in Tables 1- 4. The detailed average deviations and the route configurations of the best solutions can be found in Mohamed (2012) or requested from the authors. The deviation (in %) for each instance is computed as in Equation 11 below:

$$\text{Deviation}(\%) = \left(\frac{COST_M(p) - Z_{BEST}(p)}{Z_{BEST}(p)} \times 100\right) \qquad (11)$$

where $Z_{BEST}(p)$ and $COST_M(p)$ refer, for the p$^{th}$ instance, to the overall best cost and the cost found by a given method $(\mathrm{M})$ respectively.

**The Archetti et al. (2006) Data Set**

Table 1 shows the summary results on Archetti et al. (2006) data set. The best solution for each problem is reported in bold. Based on the average deviations on 30 instances, it is considered that MSN using the set $R'*$ is the third best performer after SplitILS by Silva et al. (2015) and Local Search Method by Derigs et al. (2011).

For comparison purpose and to be consistent with Archetti et al. (2008), we also include the solutions obtained using the original set covering model when using the set of routes, $R$. By using our modified model, we obtained our solutions faster besides being of a better quality (or at least the same) on most of the instances tested except for p120_7090 and p150_0110.

**The Mota et al. (2007) Data Set**

Table 2 shows the summary results on Mota et al. (2007) data set where the best solution is shown in bold. Among the 49 instances, MSN yields 1 best solution. In brief, MSN is the third best performer after SplitILS and the memetic algorithm with population management (MA|PM) by Boudia et al. (2007).

In addition, when comparing the solutions obtained from the original model ($P_0$) against those from our modified model ($P_1$) when using the same set of routes $\mathrm{R}$. It is found that

($P_0$) produced slightly inferior solutions with an average of 2752.02 compared to 2631.20, found by MSN (using ($P_1$)).

**The Chen et al. (2007) Data Set**

Table 3 illustrates the summary results on Chen et al. (2007). Among the methods which are tested on this data set, MSN using the reduced set $R'*$ is considered as the third best performer producing an average cost of 9048.36 after SplitILS and TSVBA by Aleman and Hill (2010) with an average cost of 9006.20 and 9043.31 respectively. Among the 21 instances, SplitILS produces the best result with a 0.05% average deviation. Branch-and-price-and-cut (BC) by Archetti et al. (2011a) is the second best performer with a 0.30% average deviation but using 6 hours of execution time in their branch and price cut algorithm. Our modified set covering-based approach yields an average deviation of 0.42%, the third best performer when using the set $R$.

**The Belenguer et al. (2000) Data Set**

In Table 4 we compare our MSN to TSVBA, VRPHAS (Chen et al., 2017) and SplitILS on Sets 1 and 2 of Belenguer et al. (2000) data set. SplitILS is the best performer on the instances in Set 1 by giving the smallest average deviation of 0.45%, followed by VRPHAS with the average deviation of 0.74%. While MSN is the third best performer with an average deviation of 1.59% using the set $R$. In Set 2, MSN produces the second best solutions with an average deviation of 1.30% using the reduced set $R'*$ after SplitILS.

## 7. Conclusions and Suggestions

The Split Delivery Vehicle Routing Problem (SDVRP) is examined using a new formulation and an efficient implementation within a set covering-based methodology. The saving-based and the sweep-based heuristics are adopted to generate the set of routes. A modified set covering-based formulation which outperforms an existing one is proposed to solve this problem. An effective repair mechanism is also proposed to remedy any infeasibility due to a customer receiving more than its original demand when solving the set covering problem. Reduction schemes to identify the set of promising routes are also carefully explored using dual routes information, the quality of the solution obtained from the heuristics and the frequency of occurrence of the generated routes. This hybrid method, which can also be called a matheuristic, produced 7 best solutions including ties when tested on the 137 instances taken from the literature.

A possible future study is to extend this methodology by solving a series of smaller subsets for the SDVRP and incorporating a learning scheme from one run to the next. Another approach is to integrate evolutionary algorithms such as GA with our set covering-based model. Other related SDVRP that incorporate vehicle fleet mix, presence of time windows, backhauling and multi depots could also be worth exploring in the near future.

## Acknowledgements

The authors would like to acknowledge Universiti Pendidikan Sultan Idris and Ministry of Higher Education, Malaysia for their financial support in respect of this study (RAGS: 2013-0005-102-72). The authors are also grateful to the referees comments for the useful comments that improved the presentation and the content of the paper.

## References

Aleman, R. E., Hill, R. R. (2010) 'A tabu search with vocabulary building approach for the vehicle routing problem with split demands', International Journal of Metaheuristics, 1, 55-80.

Aleman, R. E., Zhang, X. & Hill, R. R. (2010) 'An adaptive memory algorithm for the split delivery vehicle routing problem', Journal of Heuristics, 16, 441-473.

Aleman, R. E., Zhang, X. & Hill, R. R. (2009) 'A ring-based diversification scheme for routing problems', International Journal of Mathematics in Operational Research, 1, 163–190.

Alvarenga, G.B., Mateus, G.R. & Tomi, G.D. (2007) 'A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows', Journal Computers and Operations Research, 34, 1561-1584.

Archetti, C. & Speranza, M. G. (2008) 'The split delivery vehicle routing problem: A survey', in Golden, B. et al (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Springer, USA, pp. 103-122.

Archetti, C. & Speranza, M. G. (2012) 'Vehicle routing problems with split deliveries', Int. Transactions in Operational Research, 19, 3-22.

Archetti, C., Bianchessi, N. & Speranza, M. G. (2011a) 'A column generation approach for the split delivery vehicle routing problem', Networks, 58, 241-254.

Archetti, C., Bianchessi, N. & Speranza, M. (2014) 'Branch-and-cut algorithms for the split delivery vehicle routing problem', European Journal of Operational Research, 238, 685-698.

Archetti, C., Feiller, D., Gendreau, M. and Speranza, M. G. (2011b) 'Complexity of the VRP and SDVRP', Transportation Res. Part C: Emerging Technologies, 19, 741-750.

Archetti, C., Hertz, A. and Speranza, M. G. (2006) 'A tabu search algorithm for the split delivery vehicle routing problem', Transportation Science, 40, 64-73.

Archetti, C., Savelsbergh, A. and Speranza, M. G. (2008) 'An-optimization based heuristic for the split delivery vehicle routing problem', Transportation Science, 42, 22-31.

Belenguer, J., Martinez, M. C. and Mota, E. (2000) 'A lower bound for the split delivery vehicle routing problem', Operations Research, 48, 801-810.

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

Boudia, M., Prins, C. and Reghioui, M. (2007) 'An effective memetic algorithm with population management for the split delivery vehicle routing problem', in Bartz-Beielstein, T. et al (Eds.), Hybrid Metaheuristics, Lecture Notes in Computer Science 4771, Springer, Berlin/Heidelberg, pp. 16-30.

Chen, S., Golden, B., Wang, X. and Wasil, E. (2017) 'A novel approach to solve the split delivery vehicle routing problem', International Transactions in Operational Research, doi: 10.1111/itor.12250

Chen, S., Golden, B. and Wasil, E. (2007) 'The split delivery vehicle routing problem: Applications, algorithms, test problems and computational results', Networks, 49, 318-329.

Clarke, G. andWright, J. W. (1964) 'Scheduling of vehicles from a central depot to a number of delivery points', Operations Research, 12, 568-581.

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

Dror, M. and Trudeau, P. (1989) 'Savings by split delivery routing', Transportation Science, 23, 141-145.

Gillett, B.E. and Miller, L.R. (1974) 'A heuristic algorithm for the vehicle dispatch problem', Operations Research, 22, 340-349.

IBM, ILOG CPLEX V12.1: User’s Manual for CPLEX. IBM Corp, 2009.

Mohamed, N.H.B. (2012) Hybridisation of Heuristics and Exact Methods for the Split Delivery Vehicle Routing Problem, Unpublished PhD thesis, University of Kent, Canterbury, Kent, United Kingdom.

Mota, E., Campos, V. and Corberan, A. (2007) 'A new metaheuristic for the vehicle routing problem with split demands', in Cotta, C. and Van Hemert, J. (Eds.), Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science 4446, Springer, Berlin/Heidelberg, pp. 121-129.

Salhi, S. (2017) Heuristic Search: The Emerging Science of Problem Solving, Palgrave MacMillan, Springer International Publishing, AG, Cham, Switzerland.

Salhi, S. and Rand, G.K. (1987) 'Improvements to Vehicle Routing Heuristics', Journal of the Operational Research Society, 38, 293-295.

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

Yellow, P. (1970) 'A computational modification to the savings method of vehicle scheduling', Operational Research Quarterly, 21, 281-283.

Table 1: Summary results on Archetti et al. (2006) data set

| Problem | ZBest | [a]SPLITABU-DT | [b]Best Opt-Based | [c]B&C | [d]Local Search | [e]SplitILS | [f]VRPHAS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model (set R) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p50_00 | **524.61** | 530.79 | 527.68 | N/A | **524.61** | **524.61** | N/A | 535.96 | 524.93 | 524.95 | 524.93 |
| p75_00 | **823.89** | 854.28 | 853.61 | N/A | 829.89 | **823.89** | N/A | 856.14 | 843.33 | 845.77 | 850.93 |
| p100_00 | **826.14** | 841.36 | 840.12 | N/A | **826.14** | **826.14** | N/A | 852.69 | 844.58 | 840.96 | 851.43 |
| p120_00 | **1037.88** | 1056.96 | 1056.96 | N/A | 1042.12 | **1037.88** | N/A | 1051.31 | 1048.39 | 1048.39 | 1048.39 |
| p150_00 | **1023.87** | 1070.86 | 1055.08 | N/A | 1028.42 | **1023.87** | N/A | 1082.01 | 1067.33 | 1062.83 | 1133.56 |
| p199_00 | **1289.89** | 1340.35 | 1338.38 | N/A | 1302.89 | **1289.89** | N/A | 1367.61 | 1359.14 | 1367.61 | 1947.05 |
| p50_0110 | **459.50** | 462.91 | N/A | **459.50** | N/A | **459.50** | **459.50** | 465.95 | 465.95 | 462.45 | 465.95 |
| p75_0110 | **617.85** | 623.94 | N/A | 652.93 | N/A | **617.85** | 628.86 | 646.70 | 642.84 | 644.05 | 642.84 |
| p100_0110 | **752.62** | 771.46 | N/A | 788.23 | N/A | 760.00 | **752.62** | 792.79 | 782.51 | 782.51 | 782.51 |
| p120_0110 | **1031.11** | 1055.28 | N/A | 1071.58 | N/A | 1043.19 | **1031.11** | 1059.97 | 1047.63 | 1046.56 | 1047.63 |
| p150_0110 | **919.17** | 947.14 | N/A | 984.69 | N/A | 921.91 | **919.17** | 969.07 | 969.07 | 969.07 | 968.44 |
| p199_0110 | **1074.18** | 1148.27 | N/A | 1268.79 | N/A | 1074.18 | **1074.58** | 1134.82 | 1134.82 | 1134.82 | 1185.47 |
| p50_1030 | **757.15** | 765.31 | 758.20 | 770.19 | 776.42 | **757.15** | 776.06 | 786.55 | 768.17 | 768.95 | 772.91 |
| p75_1030 | **1109.62** | 1134.08 | 1122.91 | 1121.82 | 1123.97 | **1109.62** | 1137.43 | 1154.66 | 1114.20 | 1112.69 | 1129.42 |
| p100_1030 | **1458.46** | 1515.17 | 1505.46 | 1477.35 | 1478.59 | **1458.46** | 1469.84 | 1523.33 | 1476.61 | 1483.59 | 1595.09 |
| p120_1030 | **2881.80** | 3060.47 | 3017.92 | 2983.82 | 2913.09 | 2898.50 | **2881.80** | 2950.79 | 2950.79 | 2929.21 | 3414.40 |
| p150_1030 | **2016.97** | 2101.80 | 2093.28 | 2066.46 | 2055.18 | **2016.97** | 2039.21 | 2122.10 | 2077.29 | 2073.18 | 2300.13 |
| p199_1030 | **2478.40** | 2585.85 | 2582.62 | 2596.94 | 2540.06 | **2478.40** | 2500.49 | 2590.47 | 2576.43 | 2561.38 | 2873.27 |
| p50_1050 | **1005.75** | 1039.11 | 1021.02 | 1017.18 | 1012.56 | **1005.75** | 1027.92 | 1058.52 | 1014.69 | 1019.15 | 1019.08 |
| p75_1050 | **1502.05** | 1556.69 | 1548.54 | 1514.39 | 1508.73 | **1502.05** | 1520.83 | 1556.46 | 1507.13 | 1506.45 | 1522.08 |
| p100_1050 | **1996.76** | 2054.13 | 2024.58 | 2040.92 | 2035.91 | **1996.76** | 2017.94 | 2107.38 | 2027.35 | 2031.11 | 2071.92 |
| p120_1050 | **4219.01** | 4502.62 | 4476.38 | 4259.94 | 4270.38 | **4219.01** | 4265.64 | 4338.41 | 4338.41 | 4253.48 | 4632.14 |
| p150_1050 | **2849.66** | 2991.64 | 2977.00 | 2917.80 | 2912.08 | **2849.66** | 2876.70 | 3002.49 | 2878.77 | 2913.94 | 3275.80 |
| p199_1050 | **3471.41** | 3624.20 | 3594.00 | 3568.25 | 3581.66 | **3471.41** | 3517.12 | 3618.82 | 3546.46 | 3530.34 | 4209.91 |

Continued on the next page

| Problem | ZBest | [a]SPLITABU-DT | [b]Best Opt-Based | [c]B&C | [d]Local Search | [e]SplitILS | [f]VRPHAS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model (set R) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p50_1090 | **1488.58** | 1511.98 | 1497.28 | 1489.37 | 1489.64 | **1488.58** | 1525.09 | 1587.50 | 1501.50 | 1510.43 | 1502.68 |
| p75_1090 | **2298.58** | 2338.67 | 2337.81 | 2318.28 | 2340.09 | **2298.58** | 2356.70 | 2369.46 | 2303.15 | 2323.70 | 2323.80 |
| p100_1090 | **3085.69** | 3155.22 | 3136.29 | 3127.06 | 3145.33 | **3085.69** | 3143.09 | 3192.52 | 3120.62 | 3130.05 | 3179.31 |
| p120_1090 | **6854.09** | 7350.11 | 7117.24 | 6995.85 | 6890.39 | **6854.09** | 7004.71 | 6950.79 | 6950.79 | 6918.01 | 7089.50 |
| p150_1090 | **4545.46** | 4674.13 | 4659.90 | 4678.52 | 4638.74 | **4545.46** | 4616.62 | 4707.61 | 4678.68 | 4657.50 | 4861.73 |
| p199_1090 | **5521.57** | 5715.85 | 5710.21 | 5673.18 | 5669.26 | **5521.57** | 5618.96 | 5693.60 | 5665.46 | 5686.08 | 5909.54 |
| p50_3070 | **1481.71** | 1503.95 | 1502.00 | 1499.29 | 1488.28 | **1481.71** | 1511.66 | 1552.73 | 1494.31 | 1491.52 | 1498.02 |
| p75_3070 | **2219.97** | 2293.55 | 2263.12 | 2237.19 | 2243.93 | **2219.97** | 2286.06 | 2320.31 | 2251.17 | 2256.50 | 2264.20 |
| p100_3070 | **2989.30** | 3070.90 | 3055.51 | 3030.66 | 3014.08 | **2989.30** | 3044.73 | 3120.44 | 3020.70 | 3033.00 | 3097.36 |
| p120_3070 | **6671.04** | 7168.26 | 7126.84 | 6822.31 | **6671.04** | 6673.95 | 6776.88 | 6778.32 | 6766.97 | 6778.32 | 6828.67 |
| p150_3070 | **4334.71** | 4496.86 | 4465.47 | 4438.76 | 4435.95 | **4334.71** | 4420.73 | 4476.23 | 4411.13 | 4425.77 | 4636.08 |
| p199_3070 | **5409.76** | 5571.13 | 5549.77 | 5560.29 | 5541.09 | **5409.76** | 5496.88 | 5581.28 | 5581.28 | 5545.76 | 6006.03 |
| p50_7090 | **2156.14** | 2173.63 | 2166.80 | 2166.30 | 2174.54 | **2156.14** | 2215.09 | 2228.22 | 2159.83 | 2180.42 | 2160.22 |
| p75_7090 | **3223.40** | 3285.37 | 3250.39 | 3258.15 | 3266.78 | **3223.40** | 3303.98 | 3325.45 | 3234.61 | 3239.30 | 3261.28 |
| p100_7090 | **4387.32** | 4470.71 | 4452.56 | 4467.59 | 4447.47 | **4387.32** | 4475.32 | 4490.52 | 4429.21 | 4416.81 | 4432.94 |
| p120_7090 | **10204.81** | 10673.31 | 10429.75 | 10376.94 | 10233.37 | **10204.81** | 10364.33 | 10399.64 | 10332.33 | 10248.73 | 10288.27 |
| p150_7090 | **6395.41** | 6482.19 | 6462.78 | 6523.22 | 6467.17 | **6395.41** | 6506.25 | 6537.03 | 6499.71 | 6476.39 | 6580.22 |
| p199_7090 | **8192.03** | 8392.11 | 8355.45 | 8410.38 | 8297.71 | **8192.03** | 8331.44 | 8365.01 | 8296.87 | 8291.50 | 8535.30 |
| Average Deviation (%) | | 3.39 | 2.57 | 1.86 | 1.43 | 0.02 | 1.74 | 3.65 | 1.52 | 1.50 | 5.80 |
| Average Deviation[+] (%) | | 3.22 | N/A | N/A | N/A | 0.07 | N/A | 3.74 | 1.98 | 1.95 | 6.42 |
| # Best | | 0 | 0 | 1 | 3 | 36 | 6 | 0 | 0 | 0 | 0 |
| # Best[+] | | 0 | N/A | N/A | N/A | 36 | N/A | 0 | 0 | 0 | 0 |

[a]Archetti et al. (2006); [b]Archetti et al. (2008); [c]Archetti et al. (2011a); [d]Derigs et al. (2011); [e]Silva et al. (2015); [f]Chen et al. (2017); [1]Our Constructive Heuristics; [2]Our Modified Set Covering-based Approach ($P_1$) using set R ; [3]Our Modified Set Covering-based Approach ($P_1$) using the reduced set R'*; [4]Original Model ($P_0$) using set R.

+based on all instances.

Table 2: Summary results on Mota et al. (2007) data set

| Problem | ZBest | [a]TSVBA | [b]iVNDiv | [c]ICA+VND | [d]SS | [e]MA\|PM | [f]SplitILS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model |
|---|---|---|---|---|---|---|---|---|---|---|---|
| mgp50-01 | **524.61** | 527.67 | **524.61** | 540.82 | 531.02 | **524.61** | **524.61** | 535.96 | 524.93 | 524.95 | 524.93 |
| mgp75-02 | **823.89** | 853.20 | 851.24 | 880.28 | 839.75 | **823.89** | **823.89** | 856.14 | 843.33 | 845.77 | 850.93 |
| mgp100-03 | **826.14** | 844.21 | 852.74 | 854.13 | 835.82 | 829.44 | **826.14** | 852.69 | 844.58 | 840.96 | 851.43 |
| mgp120-11 | **1037.88** | 1051.24 | 1201.83 | 1223.28 | 1042.97 | 1041.20 | **1037.88** | 1051.31 | 1048.39 | 1048.39 | 1048.39 |
| mgp150-04 | **1023.66** | 1079.55 | 1074.11 | 1088.91 | 1056.92 | 1042.37 | **1023.66** | 1082.01 | 1067.33 | 1062.83 | 1133.56 |
| mgp199-05 | **1286.92** | 1339.49 | 1368.67 | 1390.55 | 1340.44 | 1311.59 | **1286.92** | 1367.61 | 1359.14 | 1367.61 | 1947.05 |
| mgp100-12 | **819.56** | 819.60 | 824.78 | 824.82 | 820.92 | **819.56** | **819.56** | 832.30 | 820.97 | 820.97 | 820.97 |
| mgp50-01-a | **460.79** | 466.74 | 471.92 | 473.22 | **460.79** | **460.79** | **460.79** | 465.95 | 465.95 | 465.95 | 465.95 |
| mgp75-02-a | **596.25** | 614.09 | 597.46 | 617.65 | 602.67 | 600.06 | **596.25** | 621.31 | 615.38 | 615.38 | 615.38 |
| mgp100-03-a | **726.81** | 741.60 | 745.35 | 789.16 | 729.67 | **726.81** | **726.81** | 764.20 | 763.40 | 750.63 | 763.40 |
| mgp120-11-a | **975.96** | 990.59 | 1087.80 | 1101.14 | 979.57 | 976.57 | **975.96** | 995.25 | 993.71 | 988.50 | 993.71 |
| mgp150-04-a | **866.31** | 891.10 | 891.98 | 893.49 | 883.05 | 875.61 | **866.31** | 919.91 | 919.91 | 896.08 | 915.96 |
| mgp199-05-a | **1017.28** | 1069.24 | 1073.55 | 1079.04 | 1039.51 | 1018.71 | **1017.28** | 1077.09 | 1077.09 | 1077.09 | 1117.73 |
| mgp100-12-a | **632.63** | 658.99 | 673.54 | 673.54 | 633.80 | 649.73 | **632.63** | 687.44 | 645.27 | 657.79 | 650.01 |
| mgp50-01-b | **741.06** | 753.98 | 766.19 | 777.75 | 769.60 | 751.41 | **741.06** | 779.77 | **741.06** | **741.06** | **741.06** |
| mgp75-02-b | **1064.49** | 1085.70 | 1099.47 | 1099.47 | 1074.01 | 1074.46 | **1064.49** | 1100.12 | 1068.90 | 1071.55 | 1093.68 |
| mgp100-03-b | **1376.22** | 1416.35 | 1425.90 | 1452.52 | 1416.48 | 1392.85 | **1376.22** | 1423.80 | 1423.80 | 1407.48 | 1504.43 |
| mgp120-11-b | **2707.52** | 2744.74 | 2806.92 | 2806.92 | 2783.10 | 2720.38 | **2707.52** | 2759.33 | 2745.17 | 2735.22 | 3210.75 |
| mgp150-04-b | **1861.63** | 1929.91 | 1978.01 | 1978.01 | 1974.70 | 1878.71 | **1861.63** | 1954.04 | 1927.36 | 1917.36 | 2169.08 |
| mgp199-05-b | **2305.70** | 2408.16 | 2464.65 | 2502.54 | 2435.08 | 2340.14 | **2305.70** | 2413.36 | 2388.01 | 2386.57 | 2736.33 |
| mgp100-12-b | **1413.85** | 1441.48 | 1428.27 | 1428.27 | 1423.49 | 1417.28 | **1413.85** | 1451.14 | 1425.70 | 1426.34 | 1449.02 |

Continued on the next page

| Problem | ZBest | [a]TSVBA | [b]iVNDiv | [c]ICA+VND | [d]SS | [e]MA\|PM | [f]SplitILS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model |
|---|---|---|---|---|---|---|---|---|---|---|---|
| mgp50-01-c | **982.79** | 1023.24 | 1039.89 | 1045.93 | 1025.91 | 988.31 | **982.79** | 1031.37 | 1000.63 | 994.74 | 1003.34 |
| mgp75-02-c | **1393.11** | 1458.59 | 1478.67 | 1503.02 | 1484.62 | 1413.80 | **1393.11** | 1462.21 | 1409.74 | 1416.19 | 1433.92 |
| mgp100-03-c | **1823.58** | 1886.70 | 1956.13 | 1957.55 | 1926.15 | 1845.30 | **1823.58** | 1921.52 | 1856.90 | 1855.95 | 1889.50 |
| mgp120-11-c | **3907.27** | 4010.80 | 4026.53 | 4085.36 | 3996.29 | 3934.39 | **3907.27** | 4000.71 | 4000.71 | 3978.19 | 4339.23 |
| mgp150-04-c | **2527.96** | 2647.17 | 2671.62 | 2685.33 | 2649.97 | 2561.65 | **2527.96** | 2651.69 | 2580.30 | 2570.27 | 2777.43 |
| mgp199-05-c | **3156.02** | 3296.69 | 3411.38 | 3450.84 | 3310.71 | 3191.25 | **3156.02** | 3318.49 | 3247.07 | 3238.21 | 3594.99 |
| mgp100-12-c | **1967.41** | 2010.00 | 2007.11 | 2046.15 | 2022.30 | 1994.59 | **1967.41** | 2011.37 | 1993.79 | 1993.36 | 2021.53 |
| mgp50-01-d | **1456.00** | 1530.81 | 1522.43 | 1547.32 | 1580.77 | 1467.06 | **1456.00** | 1540.47 | 1461.60 | 1467.46 | 1461.60 |
| mgp75-02-d | **2081.38** | 2164.74 | 2200.51 | 2212.93 | 2233.08 | 2102.58 | **2081.38** | 2161.71 | 2094.27 | 2093.33 | 2128.63 |
| mgp100-03-d | **2749.53** | 2874.86 | 2865.86 | 2925.13 | 2932.34 | 2780.95 | **2749.53** | 2879.60 | 2778.11 | 2797.97 | 2837.03 |
| mgp120-11-d | **6195.37** | 6308.76 | 6364.87 | 6483.06 | 6361.46 | 6318.37 | **6195.37** | 6315.02 | 6306.81 | 6227.63 | 6412.24 |
| mgp150-04-d | **3988.64** | 4151.90 | 4165.18 | 4192.50 | 4185.68 | 4045.87 | **3988.64** | 4143.09 | 4044.92 | 4060.71 | 4430.10 |
| mgp199-05-d | **4843.83** | 5066.24 | 5184.57 | 5192.06 | 5085.64 | 4941.22 | **4843.83** | 4999.29 | 4999.29 | 4961.02 | 5417.32 |
| mgp100-12-d | **3088.47** | 3157.48 | 3156.31 | 3178.28 | 3187.44 | 3113.72 | **3088.47** | 3154.89 | 3113.81 | 3106.93 | 3125.09 |
| mgp50-01-e | **1467.47** | 1505.38 | 1540.39 | 1557.52 | 1568.04 | 1477.01 | **1467.47** | 1554.96 | 1478.44 | 1481.83 | 1478.48 |
| mgp75-02-e | **2111.83** | 2182.33 | 2238.98 | 2241.59 | 2228.90 | 2132.16 | **2111.83** | 2198.82 | 2122.62 | 2127.75 | 2155.10 |
| mgp100-03-e | **2813.52** | 2929.29 | 2941.64 | 2945.19 | 2986.33 | 2858.87 | **2813.52** | 2937.50 | 2866.60 | 2856.90 | 2887.92 |
| mgp120-11-e | **6373.24** | 6511.08 | 6545.50 | 6591.40 | 6481.09 | 6424.71 | **6373.24** | 6445.41 | 6445.41 | 6442.47 | 7012.94 |
| mgp150-04-e | **3985.76** | 4151.90 | 4165.18 | 4192.50 | 4185.68 | 4045.87 | **3985.76** | 4143.09 | 4045.14 | 4057.77 | 4430.10 |
| mgp199-05-e | **5063.89** | 5281.55 | 5363.65 | 5366.06 | 5265.01 | 5155.36 | **5063.89** | 5222.25 | 5207.97 | 5142.47 | 5549.95 |
| mgp100-12-e | **3125.47** | 3200.62 | 3225.63 | 3318.08 | 3248.76 | 3155.69 | **3125.47** | 3205.94 | 3158.68 | 3157.44 | 3145.72 |

Continued on the next page

| Problem | ZBest | [a]TSVBA | [b]iVNDiv | [c]ICA+VND | [d]SS | [e]MA\|PM | [f]SplitILS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model |
|---|---|---|---|---|---|---|---|---|---|---|---|
| mgp50-01-f | **2150.97** | 2219.32 | 2215.34 | 2215.34 | 2312.48 | 2154.35 | **2150.97** | 2227.86 | 2154.35 | 2171.18 | 2163.73 |
| mgp75-02-f | **3178.47** | 3278.33 | 3304.24 | 3341.26 | 3387.86 | 3200.35 | **3178.47** | 3288.62 | 3199.95 | 3193.73 | 3206.06 |
| mgp100-03-f | **4294.12** | 4435.56 | 4429.21 | 4455.14 | 4580.98 | 4312.95 | **4294.12** | 4412.03 | 4339.88 | 4349.55 | 4343.49 |
| mgp120-11-f | **10003.99** | 10186.06 | 10302.16 | 10302.16 | 10158.32 | 10063.47 | **10003.99** | 10193.66 | 10113.63 | 10093.88 | 10232.85 |
| mgp150-04-f | **6232.37** | 6416.12 | 6482.11 | 6513.36 | 6479.46 | 6267.48 | **6232.37** | 6377.65 | 6363.64 | 6319.66 | 6418.33 |
| mgp199-05-f | **8037.88** | 8333.61 | 8329.55 | 8368.35 | 8323.72 | 8081.58 | **8037.88** | 8211.65 | 8190.91 | 8144.39 | 8375.54 |
| mgp100-12-f | **4903.00** | 4996.88 | 5028.78 | 5058.76 | 5065.26 | 4919.48 | **4903.00** | 5017.97 | 4935.05 | 4979.11 | 4972.99 |
| Average | | 2672.32 | 2701.48 | 2723.42 | 2692.40 | 2616.83 | 2591.68 | 2673.87 | 2637.13 | 2631.20 | 2752.02 |
| Average Deviation (%) | | 3.08 | 4.45 | 5.55 | 3.62 | 0.90 | 0.00 | 3.73 | 1.92 | 1.76 | 6.07 |
| #Best | | 0 | 1 | 0 | 1 | 5 | 49 | 0 | 1 | 1 | 1 |

[a]Aleman and Hill (2010); [b]Aleman et al. (2009); [c]Aleman et al. (2010); [d]Mota et al. (2007); [e]Boudia et al. (2007); [f]Silva et al. (2015); Others are defined as before

Table 3: Summary results on Chen et al. (2007) data set

| Problem | n | ZBest | [a]R to R | [b]B&C | [c]Local Search | [d]TSVBA | [e]iVNDiv | [f]ICA+VND | [g]SplitILS | [h]VRPHAS | [1]Our Heuristics | [2]MSN (set R) | [3]MSN (set R'*) | [4]Original Model (set R) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SD1 | 8 | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** | **228.28** |
| SD2 | 16 | **708.28** | 714.40 | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** | **708.28** |
| SD3 | 16 | **430.40** | 430.61 | **430.40** | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 | 430.58 |
| SD4 | 24 | **630.62** | 631.06 | **630.62** | 631.05 | 631.05 | 635.84 | 635.84 | 631.05 | 631.05 | 690.39 | 631.05 | 631.05 | 631.05 |
| SD5 | 32 | **1389.94** | 1408.12 | **1389.94** | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 | 1390.57 |
| SD6 | 32 | **830.86** | 831.21 | **830.86** | 831.24 | 831.24 | 831.24 | 831.24 | 831.24 | 831.24 | 893.61 | 831.24 | 831.24 | 831.24 |
| SD7 | 40 | **3640.00** | 3714.40 | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** | **3640.00** |
| SD8 | 48 | **5068.28** | 5200.00 | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** | **5068.28** |
| SD9 | 48 | **2042.88** | 2059.84 | **2042.88** | 2067.81 | 2071.03 | 2071.03 | 2071.03 | 2044.20 | 2057.62 | 2104.69 | 2052.80 | 2082.76 | 2048.67 |
| SD10 | 64 | **2683.73** | 2749.11 | **2683.73** | 2784.21 | 2747.83 | 2742.84 | 2747.83 | 2684.88 | 2707.83 | 2758.36 | 2689.15 | 2699.85 | 2698.54 |
| SD11 | 80 | **13280.00** | 13612.12 | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** | **13280.00** |
| SD12 | 80 | **7213.61** | 7399.06 | 7270.87 | 7220.36 | 7213.62 | 7265.70 | 7279.97 | **7213.61** | 7259.46 | 7279.06 | 7255.60 | 7264.23 | 7269.15 |
| SD13 | 96 | **10105.86** | 10367.06 | **10105.86** | 10277.81 | 10110.58 | 10110.58 | 10110.58 | 10110.58 | 10110.58 | 10110.60 | 10110.58 | 10110.58 | 10110.60 |
| SD14 | 120 | **10717.53** | 11023.00 | 10754.70 | 10790.58 | 10802.87 | 10829.25 | 10893.50 | **10717.53** | 10771.54 | 10837.80 | 10837.80 | 10837.80 | 10927.70 |
| SD15 | 144 | **15094.48** | 15271.77 | 15154.14 | 15152.88 | 15153.45 | 15168.28 | 15168.28 | **15094.48** | 15250.13 | 15210.40 | 15210.40 | 15188.02 | 15210.30 |
| SD16 | 144 | **3379.33** | 3449.05 | **3379.33** | 3381.29 | 3446.43 | 3580.07 | 3635.27 | 3381.26 | 3553.32 | 3428.20 | 3395.29 | 3381.28 | 3381.25 |
| SD17 | 160 | **26493.56** | 26665.76 | 26547.44 | 26536.09 | **26493.56** | 26556.13 | 26559.93 | 26496.06 | 26547.06 | 26559.00 | 26559.00 | 26533.00 | 26835.20 |
| SD18 | 160 | **14202.53** | 14546.58 | 14334.03 | 14469.10 | 14323.04 | 14372.80 | 14440.59 | **14202.53** | 14320.66 | 14378.10 | 14378.10 | 14281.00 | 14782.09 |
| SD19 | 192 | **19995.69** | 20559.21 | 20210.45 | 20420.11 | 20157.10 | 20188.62 | 20191.19 | **19995.69** | 20251.89 | 20259.40 | 20259.40 | 20197.70 | 20599.60 |
| SD20 | 240 | **39635.51** | 40408.22 | 39901.22 | 40368.58 | 39722.86 | 39803.13 | 39813.49 | **39635.51** | 39678.10 | 39757.80 | 39757.80 | 39757.80 | 40614.76 |
| SD21 | 288 | **11271.06** | 11491.67 | 11491.13 | **11271.06** | 11458.76 | 11682.09 | 11799.60 | 11345.68 | 11631.67 | 11498.10 | 11486.98 | 11473.16 | 11916.79 |
| Average | | | 9179.07 | 9051.54 | 9092.77 | 9043.31 | 9075.41 | 9091.63 | 9006.20 | 9064.20 | 9071.98 | 9057.20 | 9048.36 | 9171.57 |
| Average Deviation (%) | | | 1.61 | 0.30 | 0.67 | 0.51 | 0.92 | 1.12 | 0.05 | 0.71 | 1.54 | 0.42 | 0.43 | 1.01 |
| # Best inc. ties | | | 1 | 13 | 6 | 6 | 5 | 5 | 11 | 5 | 5 | 5 | 5 | 5 |

[a]Chen et al. (2007); [b]Archetti et al. (2011a); [c]Derigs et al. (2011); [d]Aleman and Hill (2010); [e]Aleman et al. (2009); [f]Aleman et al. (2010); [g]Silva et al. (2015); [h]Chen et al. (2017); Others are defined as before.

Table 4: Summary results on Belenguer et al. (2000) data set

| Problem (Set 1) | LB | Zbest | [a]TSVBA | [b]SplitILS | [c]VRPHAS | [1]Our Heuristics | [2]MSN (Set R) | [3]MSN (Set R'*) |
|---|---|---|---|---|---|---|---|---|
| eil22 | - | **375.28** | **375.28** | **375.28** | **375.28** | **375.28** | **375.28** | **375.28** |
| eil23 | 451.80 | **568.56** | 569.75 | **568.56** | **568.56** | 571.55 | 570.36 | 570.36 |
| eil30 | 218.92 | **497.53** | 505.01 | 505.01 | **497.53** | 506.67 | 505.01 | 505.01 |
| eil33 | - | **826.41** | 843.64 | 837.06 | **826.41** | 841.65 | 840.68 | 840.68 |
| eil51 | 518.23 | **524.61** | 527.67 | 524.61 | **524.61** | 535.96 | 524.93 | 524.93 |
| eilA76 | 809.58 | **823.89** | 853.20 | **823.89** | 849.60 | 856.14 | 841.94 | 845.59 |
| eilB76 | 984.13 | **1009.04** | 1034.21 | **1009.04** | 1024.44 | 1039.92 | 1025.48 | 1026.74 |
| eilC76 | 721.39 | **738.67** | 761.55 | **738.67** | 748.51 | 757.04 | 747.47 | 747.47 |
| eilD76 | 672.34 | **684.53** | 695.96 | 687.60 | **684.53** | 706.66 | 700.39 | 701.96 |
| eilA101 | 804.27 | **812.51** | 844.21 | 826.14 | **812.51** | 843.80 | 843.80 | 840.96 |
| eilB101 | 1055.59 | **1076.26** | 1112.15 | **1076.26** | 1099.00 | 1117.36 | 1105.90 | 1119.84 |
| Average | | | 738.42 | 724.74 | 728.27 | 741.09 | 734.66 | 736.26 |
| Average Deviation (%) | | | 2.04 | 0.45 | 0.74 | 2.43 | 1.59 | 1.75 |
| # Best inc. ties | | | 1.00 | 6 | 7 | 1 | 1 | 1 |

| Problem (Set 2) | LB | Zbest | [a]TSVBA | [b]SplitILS | [c]VRPHAS | [1]Our Heuristics | [2]MSN (Set R) | [3]MSN (Set R'*) |
|---|---|---|---|---|---|---|---|---|
| s51D1 | 457.08 | **459.50** | 468.79 | **459.50** | **459.50** | 465.95 | 465.95 | 465.95 |
| s51D2 | 697.00 | **709.29** | 718.69 | **709.29** | 716.82 | 727.84 | 713.32 | 713.32 |
| s51D3 | 933.97 | **948.06** | 969.78 | **948.06** | 964.83 | 996.90 | 951.19 | 951.09 |
| s51D4 | 1545.19 | **1562.01** | 1628.20 | **1562.01** | 1592.23 | 1636.51 | 1566.36 | 1572.65 |
| s51D5 | 1316.93 | **1333.67** | 1362.19 | **1333.67** | 1371.41 | 1388.42 | 1338.75 | 1343.51 |
| s51D6 | 2149.55 | **2169.10** | 2236.16 | **2169.10** | 2240.46 | 2268.86 | 2172.33 | 2172.33 |
| s76D1 | 590.92 | **598.94** | 613.70 | **598.94** | 614.31 | 624.19 | 624.19 | 624.19 |
| s76D2 | 1066.88 | **1087.40** | 1128.15 | **1087.40** | 1120.71 | 1129.59 | 1116.29 | 1102.88 |
| s76D3 | 1406.85 | **1427.86** | 1472.92 | **1427.86** | 1445.23 | 1485.75 | 1438.97 | 1444.95 |
| s76D4 | 2053.66 | **2079.76** | 2180.13 | **2079.76** | 2138.64 | 2158.21 | 2095.47 | 2094.52 |
| s101D1 | 714.50 | **726.59** | 749.93 | **726.59** | 746.08 | 742.43 | 761.18 | 742.43 |
| s101D2 | 1356.78 | **1378.43** | 1409.03 | **1378.43** | 1412.98 | 1437.98 | 1401.44 | 1414.11 |
| s101D3 | 1845.07 | **1874.81** | 1947.62 | **1874.81** | 1924.39 | 1969.05 | 1894.57 | 1890.06 |
| s101D5 | 2758.21 | **2791.22** | 2910.71 | **2791.22** | 2874.86 | 2935.14 | 2812.65 | 2826.50 |
| Average | | | 1414.00 | 1367.62 | 1401.60 | 1426.20 | 1382.33 | 1382.75 |
| Average Deviation (%) | | | 3.06 | 0.00 | 2.24 | 3.95 | 1.41 | 1.30 |
| # Best inc. ties | | | 0 | 14 | 1 | 0 | 0 | 0 |

[a]Aleman and Hill (2010); [b]Silva et al. (2015); [c]Chen et al. (2017); Others are defined as before.