The Split Delivery Vehicle Routing Problem: A Survey

C. Archetti and M.G. Speranza
Department of Quantitative Methods
University of Brescia
C.da S. Chiara 50, I-25122 Brescia, Italy
archetti@eco.unibs.it, speranza@eco.unibs.it

Abstract

In the classical Vehicle Routing Problem (VRP) a fleet of capacitated vehicles is available to serve a set of customers with known demand. Each customer is required to be visited by exactly one vehicle and the objective is to minimize the total distance traveled. In the Split Delivery Vehicle Routing Problem (SDVRP) the restriction that each customer has to be visited exactly once is removed, i.e., split deliveries are allowed. In this paper we present a survey of the state-of-the-art on this important problem.

Keywords: Split Delivery Vehicle Routing Problem, Survey, Computational complexity, Algorithms.

1 Introduction

We consider the *Split Delivery Vehicle Routing Problem* (SDVRP) where a fleet of capacitated homogeneous vehicles has to serve a set of customers. Each customer can be visited more than once, contrary to what is usually assumed in the classical *Vehicle Routing Problem* (VRP), and the demand of each customer may be greater than the capacity of the vehicles. There is a single depot for the vehicles and each vehicle has to start and end its tour at the depot. The problem consists in finding a set of vehicle routes that serve all the customers such that the sum of the quantities delivered in each tour does not exceed the capacity of a vehicle and the total distance traveled is minimized.

The SDVRP was introduced in the literature only a few years ago by Dror and Trudeau ([13] and [14]) who motivate the study of the SDVRP by showing that there can be savings generated by allowing split deliveries. Archetti, Savelsbergh and Speranza [3] study the maximum possible savings obtained by allowing split deliveries, while in [4] the same authors present a computational study to show how the savings depend on the characteristics of the instance. Valid inequalities for the SDVRP are described in [12]. In [8] a lower bound is proposed for the SDVRP where the demand of each customer is lower than the capacity of the vehicles and the quantity delivered by a vehicle when visiting a customer is an integer number. In [2] the authors analyze the computational complexity of the SDVRP and the case of small capacity of the vehicles.

Heuristic algorithms for the SDVRP can be found in [13] and [14], where a local search algorithm is proposed, in [1] for a tabu search and in [5] for an optimization-based heuristic. In [15] the authors present a mathematical formulation and a heuristic algorithm for the SDVRP with grid network distances and time windows constraints.

Real applications of the problem can be found in [20] where the authors consider the problem of managing a fleet of trucks for distributing feed in a large livestock ranch which is formulated as a split delivery capacitated rural postman problem with time windows. Several heuristics are proposed to solve the problem which compare favorably with the working practices on the ranch. Sierksma and Tijssen [21] consider the problem of determining the flight schedule for helicopters to off-shore platforms for exchanging crew people employed on these platforms. The problem is formulated as an SDVRP and several heuristics are proposed. In [6] Archetti and Speranza consider a waste collection problem where vehicles have a small capacity and customers can have demands larger than the capacity. A number of constraints are considered like time windows, different types of wastes, priorities among customers and different types of vehicles. They propose a heuristic algorithm that beats the solution implemented by the company which carries out the service. A similar problem is analyzed in [7], where it is called the Rollon-Rolloff Vehicle Routing Problem (RRVRP), and in [10].

The paper is organized as follows. In Section 2 we provide the problem formulation. In Section 3 we present computational complexity results and some properties of the problem. In Section 4 we describe a lower bound and valid inequalities proposed for the SDVRP. In Section 5 we analyze the savings with respect to the VRP. In Section 6 we present the heuristic algorithms proposed for the SDVRP and compare them.

2 Problem formulation

The SDVRP can be defined over a graph $G = (V, E)$ with vertex set $V = \{0, 1, ..., n\}$ where 0 denotes the depot while the other vertices are the customers, and E is the edge set. The traversal cost (also called length) c_{ij} of an edge $(i, j) \in E$ is supposed to be non-negative and to satisfy the triangle inequality. An integer demand d_i is associated with each customer $i \in V - \{0\}$. An unlimited number of vehicles is available, each with a capacity $Q \in Z^+$. We will however consider an upper bound m on the number of vehicles needed to serve the customers. For example, one can use $m = \sum_{i=1}^{n} d_i$. Each vehicle must start and end its route at the depot. The demands of the customers must be satisfied, and the quantity delivered in each tour cannot exceed Q. The objective is to minimize the total distance traveled by the vehicles. We give below a mixed integer programming formulation (P) for the SDVRP (see [1]). We use the following notations:

x_{ij}^v is a boolean variable which is equal to 1 if vehicle v travels directly from i to j, and to 0 otherwise,

y_{iv} is the quantity of the demand of i delivered by the v-th vehicle.

The SDVRP can now be formulated as follows:

$$Min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{v=1}^{m} c_{ij} x_{ij}^v \tag{1}$$

subject to:

$$\sum_{i=0}^{n}\sum_{v=1}^{m} x_{ij}^{v} \geq 1 \quad j = 0, ..., n \tag{2}$$

$$\sum_{i=0}^{n} x_{ip}^{v} - \sum_{j=0}^{n} x_{pj}^{v} = 0 \quad p = 0, ..., n;\ v = 1, ..., m \tag{3}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij}^{v} \leq |S| - 1 \quad v = 1, ..., m;\ S \subseteq V - \{0\} \tag{4}$$

$$y_{iv} \leq d_i \sum_{j=0}^{n} x_{ij}^{v} \quad i = 1, ..., n;\ v = 1, ..., m \tag{5}$$

$$\sum_{v=1}^{m} y_{iv} = d_i \quad i = 1, ..., n \tag{6}$$

$$\sum_{i=1}^{n} y_{iv} \leq Q \quad v = 1, ..., m \tag{7}$$

$$x_{ij}^{v} \in \{0, 1\} \quad i = 0, ..., n;\ j = 0, ..., n;\ v = 1, ..., m \tag{8}$$

$$y_{iv} \geq 0 \quad i = 1, ..., n;\ v = 1, ..., m. \tag{9}$$

Constraints (2)-(4) are the classical routing constraints: constraints (2) impose that each vertex is visited at least once, (3) are the flow conservation constraints while (4) are the subtours elimination constraints. Constraints (5)-(7) concern the allocation of the demands of the customers among the vehicles: constraints (5) impose that customer i can be served by vehicle v only if v passes through i, constraints (6) ensure that the entire demand of each vertex is satisfied, while constraints (7) impose that the quantity delivered by each vehicle does not exceed the capacity.

It has been shown [1] that there always exists an optimal integer solution to (P).

Theorem 1. *[1] If (P) has feasible solutions, then there always exists an optimal solution in which variables $y_{iv} \in Z^{+}$.*

3 Complexity and properties

In this section we present the results concerning the computational complexity of the SDVRP. We also show some properties of the SDVRP which can be really useful to reduce the solution space when solving the problem.

Theorem 2. *[2] The SDVRP with $Q = 2$ can be solved in polynomial time.*

Theorem 3. *[2] The SDVRP where each customer has unitary demand is NP-hard for $Q \geq 3$.*

Definition 1. *A SDVRP instance is reducible if an optimal solution exists such that each vertex is served by as many direct trips as possible from the depot to the vertex, with full load in each trip, until the demand of each vertex is lower than the vehicle capacity Q.*

When an instance of the problem is reducible, we call *reduced* the instance which is obtained by changing the demand d_i of customer i with $(d_i \ mod \ Q)$ and deleting the vertices, and related arcs, when $(d_i \ mod \ Q) = 0$. The reduction of the original instance requires a linear time in the number of the vertices.

Theorem 4. *[2] The SDVRP with $Q = 2$ is reducible.*

In [2] it is also shown that, even in the case of Euclidean distances, the SDVRP is not reducible for $Q \geq 3$.

Dror and Trudeau [13] have shown an interesting property of optimal solutions to the SDVRP. To understand their result we first need the following definition.

Definition 2. *Consider a set $C = \{i_1, i_2, ..., i_k\}$ of customers and suppose that there exist k routes $r_1, ..., r_k$, $k \geq 2$, such that r_w contains customers i_w and i_{w+1}, $w = 1, ..., k-1$, and r_k contains customers i_1 and i_k. Such a configuration is called a k-split cycle.*

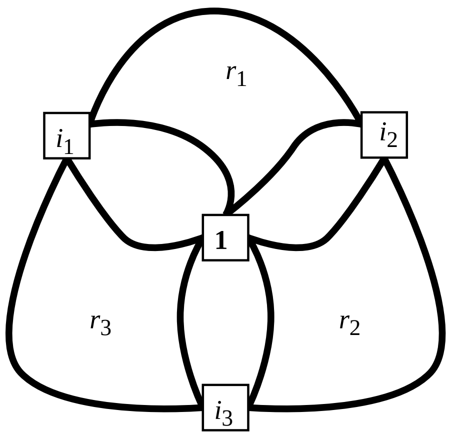

Figure 1: A 3-split cycle

An example of a 3-split cycle can be found in Figure 1. Dror and Trudeau have shown that, if the distances satisfy the triangle inequality, then there always exists an optimal solution to the SDVRP which does not contain k-split cycles, $k \geq 2$.

Property 1. *[14] If the cost matrix satisfies the triangle inequality, then there exists an optimal solution to the SDVRP where there is no k-split cycle (for any k).*

This property is of great importance since it reduces remarkably the number of interesting solutions to the SDVRP, as shown in the following corollary.

Corollary 1. *[14] If the cost matrix satisfies the triangle inequality, then there exists an optimal solution to the SDVRP where no two routes have more than one customer with a split delivery in common.*

We derive another structural property of optimal solutions to the SDVRP in which we relate the number of splits to the number of routes. Let n_i be the number of deliveries received by customer i, i.e., the number of routes that visit customer i. We say that customer i is a customer with a split delivery if $n_i > 1$ and that the number of splits at customer i is $n_i - 1$. Therefore, the total number of splits is equal to $\sum_{i=1}^{n}(n_i - 1)$.

Property 2. *[3] If the cost matrix satisfies the triangle inequality, then there exists an optimal solution to the SDVRP where the number of splits is less than the number of routes.*

4 A lower bound and valid inequalities

In this section we describe a set of valid inequalities proposed in [12] and a lower bound for the SDVRP proposed in [8] which is, to the best of our knowledge, the only lower bound proposed in the literature for the SDVRP.

In [12] Dror, Laporte and Trudeau first analyze the classical VRP valid inequalities in order to check whether they are valid also for the SDVRP. First of all, they analyze the subtour elimination constraints for the VRP. Defining as $N(S)$ the number of vehicles required to serve all the vertices in S, they show that, while the subtour elimination constraints written in the following form:

$$\sum_{v=1}^{m}\sum_{i\in S}\sum_{j\in S} x_{ij}^{v} \leq |S| - N(S) \quad S \subseteq V - \{0\}\,; |S| \geq 2$$

are not valid for the SDVRP, in the equivalent form

$$\sum_{v=1}^{m}\sum_{i\in S}\sum_{j\in V-S} x_{ij}^{v} \geq N(S) \quad S \subseteq V - \{0\}\,; |S| \geq 2 \tag{10}$$

they represent valid inequalities for the SDVRP. From (10) they derive the following class of valid inequalities:

$$\sum_{v=1}^{m}\sum_{i\in S}\sum_{j\in S} x_{ij}^{v} \leq \sum_{i\in S} d_i - N(S) \quad S \subseteq V - \{0\}\,; |S| \geq 2. \tag{11}$$

They also present a class of constraints, called *fractional cycle elimination constraints*, which represent valid inequalities for the SDVRP:

$$\sum_{i\in S}\sum_{j\in V-S} x_{ij}^{v} \geq \frac{\sum_{i\in S}\sum_{j\in S} x_{ij}^{v}}{|S| - 1} \quad S \subseteq V - \{0\}\,; |S| \geq 2; v = 1, ..., m \tag{12}$$

$$x_{ij}^{v} \leq \sum_{k\neq i} x_{jk}^{v} \quad i = 1, .., n; j = 1, .., n; v = 1, ..., m. \tag{13}$$

The authors tested the effectiveness of inequalities (10)-(13) by comparing the value of the LP relaxation of the problem before and after the introduction of the inequalities with an upper bound. The upper bound used is the value of the solution of the algorithm described in [13]. The computational results show that the gap between the value of the LP relaxation and the upper bound is always below 9% when the valid inequalities are added to the model, while it can be over 50% if these inequalities are not considered.

In [8] Belenguer, Martinez and Mota study the polyhedron of the SDVRP finding valid inequalities that define facets of the polyhedron. These facets are used in a cutting plane algorithm in order to find a lower bound for the SDVRP. The authors made computational experiments to test the effectiveness of their lower bound. They test it on instances from the TSPLIB and randomly generated instances. To measure the performance of the lower bound, they measure the gap between the lower bound and an upper bound obtained by solving the instances through a heuristic algorithm for the VRP. Computational results show that the average gap with respect to the upper bound is 3.05% for the TSPLIB instances and 7.81% for the randomly generated instances.

Finally, in [16] an exact algorithm is proposed for the SDVRP with time windows. It uses a set covering formulation of the problem and a column generation approach to solve it. The column generation scheme is included in a branch and bound tree obtaining a branch and price exact algorithm. The algorithm is introduced in [18] where instances with up to 25 customers are solved. In [16] the algorithm is improved and solves almost all instances with up to 50 customers and a subset of instances with 100 customers.

5 SDVRP vs VRP

The interest in the SDVRP comes from the fact that costs can be reduced with respect to the costs of the VRP by allowing split deliveries. In this section, we discuss the amount of the saving. This is an important information in practice, because of the additional organizational difficulties deriving from the multiple visits to the same customer. In the following we will consider both the case where the demand of each customer is lower than or equal to the capacity Q and the case where the demand of a customer can be greater than Q. For this latter case there is the need to define a variant of the classical VRP since, when the demand of a customer is greater than the vehicle capacity, it has to be split and the customer has to be visited more than once. Thus, in order to distinguish the cases, in this section we define as *extended VRP* the problem where each customer is visited the minimum number of times and *extended SDVRP* the problem where this restriction is relaxed.

We will indicate as:

- VRP the problem of finding the optimal solution when the demand of each customer is lower than or equal to Q and each customer is visited exactly once. $z(VRP)$ is the value of the corresponding optimal solution;

- $SDVRP$ the problem of finding the optimal solution when the demand of each customer is lower than or equal to Q and each customer can be visited any number of times. $z(SDVRP)$ is the value of the corresponding optimal solution;

- VRP^+ (*extended VRP*) the problem of finding the optimal solution when the demand of each customer can be greater than Q and each customer is visited exactly the minimum number of times, i.e., $t_i = \lceil \frac{d_i}{Q} \rceil$, where t_i is the number of visits to customer i. $z(VRP^+)$ is the value of the corresponding optimal solution;

- $SDVRP^+$ (*extended SDVRP*) the problem of finding the optimal solution when the demand of each customer can be greater than Q and each customer is visited any number of times. $z(SDVRP^+)$ is the value of the corresponding optimal solution.

For both extended problems we present the performance of the following heuristic algorithm: make full truckload deliveries using out-and-back tours to customers with demand greater than the vehicle capacity until their remaining demand is less than or equal to the vehicle capacity. Then, solve a VRP (obtaining a heuristic for the VRP^+) or an SDVRP (obtaining a heuristic for the $SDVRP^+$) to find a minimum cost set of routes serving the remaining demands of all customers. We will indicate as:

- H^{VRP^+} the heuristic for problem VRP^+ and $z(H^{VRP^+})$ the value of the corresponding solution;

- H^{SDVRP^+} the heuristic for problem $SDVRP^+$ and $z(H^{SDVRP^+})$ the value of the corresponding solution.

5.1 Worst-case analysis

In [3] it is shown that

$$\frac{z(VRP)}{z(SDVRP)} \leq 2$$

and that this bound is tight, i.e., there exists an instance in which the optimal VRP solution has a value that is twice as large as the value of the optimal SDVRP solution.

For the case where the demand of each customer can be greater than Q the ratio is still the same, i.e.,

$$\frac{z(VRP^+)}{z(SDVRP^+)} \leq 2$$

and this bound is tight.

In [3] the authors have also studied the performance ratio of heuristics H^{VRP^+} and H^{SDVRP^+}. They found that

$$\frac{z(H^{VRP^+})}{z(VRP^+)} \leq 2,$$

$$\frac{z(H^{SDVRP^+})}{z(SDVRP^+)} \leq 2$$

and that both bounds are tight.

The instances used to demonstrate the tightness of the bounds all have a large vehicle capacity and discrete demands. In [3] there are also results for the case where the capacity is small and the demands are discrete. In particular, when the capacity is $Q = 3$, then $\frac{z(VRP)}{z(SDVRP)} \leq \frac{3}{2}$ and $\frac{z(VRP^{+})}{z(SDVRP^{+})} \leq \frac{3}{2}$ and these bounds are tight.

It has long been recognized that a major benefit, if not the major benefit, of allowing split deliveries is a reduction in the number of delivery routes required to satisfy all demand. Therefore, it is worthwhile to study the ratio $\frac{r(VRP)}{r(SDVRP)}$, where $r(VRP)$ and $r(SDVRP)$ denote the minimum number of delivery routes required to satisfy customer demand in a solution to the VRP and the SDVRP, respectively. In [4] it is shown that

$$\frac{r(VRP)}{r(SDVRP)} \leq 2$$

and that the bound is tight.

5.2 Computational analysis

While the worst-case results discussed above are of great theoretical and also practical relevance, important additional information can be obtained from an empirical study of the ratios $\frac{r(VRP)}{r(SDVRP)}$ and $\frac{z(VRP)}{z(SDVRP)}$.

In Figure 2 (taken from [4]) the ratio $\frac{r(VRP)}{r(SDVRP)}$ is reported as a function of the demand of the customers, d, for an instance where $n = Q = 149$ and all customers share the same location and the same demand d. It is interesting to note that the ratio reaches its maximum value of 1.987 for demand size 75, i.e., when $d = \lceil \frac{Q}{2} \rceil$. For demand size 75, an optimal VRP solution has to serve each customer with an out-and-back tour, i.e., a total of 149 routes are needed, whereas in the SDVRP solution the demands of two customers can be combined in a single route leaving only one unit of demand to be picked up by another route, i.e., a total of 75 routes are needed, resulting in a ratio of 1.987. The other peaks are reached for values of d equal to $\lceil \frac{Q}{k} \rceil$, $k \in N$, $k > 2$.

In [4] the authors study the ratio between costs, i.e., $\frac{z(VRP)}{z(SDVRP)}$, also to see if is strictly related to the ratio $\frac{r(VRP)}{r(SDVRP)}$. They analyze what are the main characteristics of the instances that influence both ratios, focusing on three aspects:

- location of the customers;
- mean demand of the customers;
- variance of the demands of the customers.

Studying the ratio $\frac{z(VRP)}{z(SDVRP)}$ for reasonable size instances can only be done using heuristics, since both the VRP and the SDVRP are NP-hard. The use of heuristics is further justified by the fact that practitioners will also use heuristics for the solution of a VRP or an SDVRP, and comparing the values obtained through heuristics therefore gives a realistic measure of the benefit practitioners may obtain from allowing split deliveries. For their computational study, Archetti, Savelsbergh and

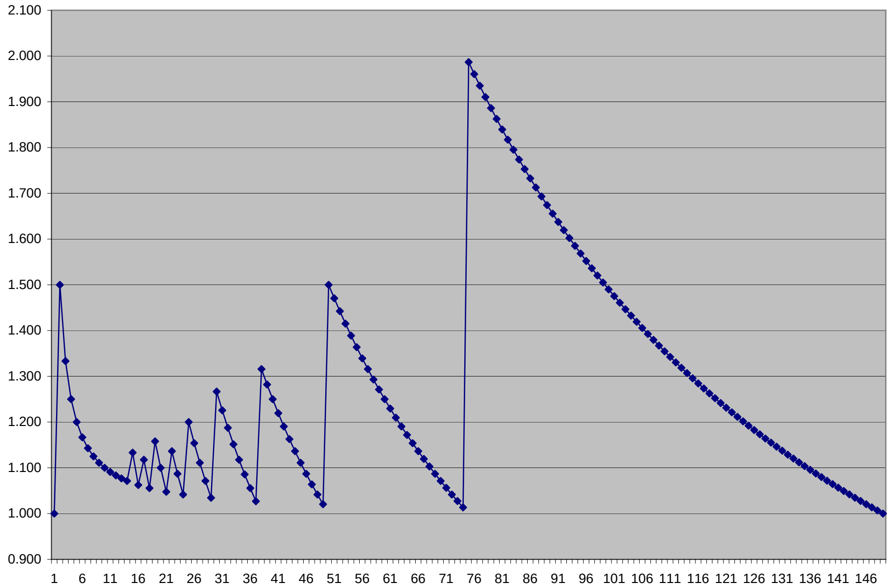

Figure 2: Ratio $\frac{r(VRP)}{r(SDVRP)}$ as a function of d for an instance with 149 customers and vehicles with capacity 149.

Speranza [4] use state-of-the-art meta-heuristics. The VRP solution is obtained using the granular tabu search heuristic of Toth and Vigo [22]. The SDVRP solution is obtained using the tabu search heuristic developed by Archetti, Hertz and Speranza [1]. Both algorithms are considered to be highly effective and therefore should provide reasonably accurate and acceptable ratios.

The computational study confirms that allowing split deliveries can result in substantial benefits, but also shows that these substantial benefits only occur for customers demands with fairly specific characteristics. The following insights have been obtained:

1. the benefits from allowing split deliveries mainly depend on the relation between mean demand and vehicle capacity and on demand variance; there does not appear to be a dependence on customer locations;

2. the major benefit of allowing split deliveries appears to be the ability to reduce the number of delivery routes and, as a consequence, the cost of the routes (this also explains the fact that there does not appear to be a dependence on customer locations);

3. the largest benefits are obtained when the mean demand is greater than half the vehicle capacity but less than three quarters of the vehicle capacity and the demand variance is relatively small.

6 Heuristics for the SDVRP

In this section we present the heuristic algorithms proposed in the literature to solve the SDVRP. To the best of our knowledge, three algorithms have been proposed: a local search heuristic by Dror and Trudeau [13], a tabu search heuristic by Archetti, Hertz and Speranza [1] and an optimization-based heuristic by Archetti, Savelsbergh and Speranza [5]. We are also aware of a heuristic by Chen, Golden and Wasil [9], but since the paper did not yet appear we are unable to provide any additional information.

6.1 Dror and Trudeau heuristic

We give in this section a short description of the algorithm proposed by Dror and Trudeau for the SDVRP [13]. The algorithm is designed only for the case where the demand of each customer is lower than the capacity of the vehicles. The heuristic is a local search algorithm and is composed of the following two main procedures.

K-split interchange

Consider a vertex i and its total demand d_i :

1. remove vertex i from all the routes where it is visited;

2. consider all subsets R of routes such that the total residual capacity is greater than or equal to d_i. For each such subset R compute the total insertion cost of i into all routes of R. Choose the subset R that leads to the least insertion cost and insert i into all routes of R.

Route addition

Consider a customer i which appears in at least two routes r_1 and r_2. Eliminate the split of i on these two routes and create a new route in the following way:

1. preserve the four principle route segments on r_1 and r_2 (from the depot to the vertex preceding i and from the vertex succeeding i to the depot);

2. create three routes considering all the possible combinations between the principle route segments and i (which must not be split) and choose the best one.

There are 9 possible combinations (for details see [13]). The same procedure is considered when customer i is split among 3 different routes. In this case there are 19 possible combinations to be considered. If a vertex is visited by more than 3 routes, the algorithm considers all the possible combinations of 2 and 3 routes.

Moreover, Dror and Trudeau use the following classical improvement procedures which have been developed for the capacitated VRP.

Node interchanges This procedure is based on one-node moves and two-nodes swaps between routes and is described in detail in [11].

2-opt This is the classical 2-opt procedure for the TSP [19].

Defining boolean variables *split_impr* and *add_impr*, the main algorithm works as follows.

Dror and Trudeau algorithm

1. Construct a feasible VRP solution.
2. *Node interchanges*: execute all node interchange improvements.
3. *2-opt*: execute all 2-opt route improvements.
4. Set *split_impr* = "false" and *add_impr* = "false".
5. *K-split interchange*: execute all k-split interchange improvements. If there is at least one improvement then set *split_impr* = "true".
6. *Route addition*: execute all route addition improvements. If there is at least one improvement then set *add_impr* = "true".
7. If *add_impr* = "true" then go to step 5. Otherwise, if *split_impr* = "true" go to step 2 else STOP.

6.2 Tabu search heuristic

In [1] a tabu search algorithm for the SDVRP, called SPLITABU, is presented and tested. It is a very simple algorithm, easy to implement, where there are only two parameters to be set: the length of the tabu list and the maximum number of iterations the algorithm can run without improvement of the best solution found. The algorithm is composed of the three following phases:

- *Phase 1: construction of an initial feasible solution.* Initially, the instance is reduced by making as many full load out-and-back tours as possible from the depot to each vertex. Then, a traveling salesman problem is solved on the reduced instance by means of the GENIUS algorithm proposed by Gendreau, Hertz and Laporte [17]. Finally, the giant tour is cut into pieces so that the capacity constraint is satisfied.
- *Phase 2: tabu search phase.* In the tabu phase, a move from a solution s to a neighbour solution s' is performed by inserting a customer i into a route r and by removing i from a subset U of routes visiting i. The subset U is determined on the basis of the savings caused by removing i. The insertion of a customer i into a route r is made with the classical cheapest insertion method. Customer i can be totally or partially removed from each route in U. The route r, where customer i is inserted, can be a route which already visits i or not, and can be also a new route. When a customer i is added to a route r it can not be removed for a number of iterations. Similarly, when a customer i is removed from a route u it can not be added for a number of iterations.
- *Phase 3: final improvement of the solution found by the tabu search phase.* The final solution of Phase 2 is improved by deleting all k-split cycles and by applying the GENIUS algorithm to each individual route.

6.3 Optimization-based heuristic

The heuristic proposed in [4] makes use of information provided by the tabu search described in the previous section in order to construct a set of good routes. These routes are then passed to a MIP program which determines the best ones. One of the key ideas underlying this solution approach is that the tabu search can identify parts of the solution space that are likely to contain high quality solutions.

The simplest use of this idea is the identification of a set C' of customers which are likely to be served by a single vehicle in high-quality SDVRP solutions. If a customer is never, or rarely, split in the solutions encountered during the tabu search, this information is interpreted as an indication that it is likely that the customer will be served by a single vehicle in high quality SDVRP solutions (and therefore should be in the set C'). This idea is implemented as follows. Let S denote the set of all SDVRP solutions encountered during the tabu search. For each customer i, calculate the *node counter* n_i, the number of times customer i is split in the solutions in S, where we say that a customer is split $k-1$ times if the customer is served by k routes in a solution $s \in S$. Let $n_{max} = \max_i n_i$. A customer i is included in C' if $n_i < 0.1 \times n_{max}$ and if i is not split in the final solution of the tabu search.

The use of the same idea in the identification of the set R of promising routes is more involved and summarized here. For each edge (i,j), calculate n_{ij}, the number of times edge (i,j) appears in any of the routes of the solutions in S. n_{ij} is the *edge counter* of edge (i,j). As before, a large value n_{ij} is interpreted as an indication that it is likely that edge (i,j) will be included in high quality SDVRP solutions. The edge counters n_{ij} guide the construction of a set of *promising routes* $\bar{R}$. The set $\bar{R}$ is not used directly in the route optimization IP, because it is usually too large, but the route optimization IP is solved several times with subsets R of $\bar{R}$.

We now describe the route optimization IP. Let c_r denote the cost of route r. The formulation has two sets of variables. The binary variable x_r takes on value 1 if route r is selected and 0 otherwise. The continuous variable y_r^i represents the quantity delivered to customer i on route r. The integer programming model is presented below.

$$\min \sum_r c_r x_r \tag{14}$$

$$\sum_{i \in r} y_r^i \leq Q x_r \qquad r \in R \tag{15}$$

$$\sum_{r \in R: i \in r} y_r^i \geq d_i \qquad i \in V - \{0\} \tag{16}$$

$$x_r \in \{0,1\} \qquad r \in R \tag{17}$$

$$y_r^i \geq 0 \qquad r \in R;\ i \in V - \{0\}. \tag{18}$$

The objective function (14) minimizes the total cost of the selected routes. Constraints (15) impose that a delivery to a customer i on route r can only take place if route r is selected and that the total quantity delivered on a selected route cannot exceed the vehicle capacity. Constraints (16) ensure that the demand d_i of customer

i is completely satisfied. This formulation is strengthened in [4] with additional constraints.

An overview of the proposed approach is presented in Algorithm 1.

Algorithm 1 Optimization-based heuristic

Calculate the node counters n_i for all $i \in V - \{0\}$ and determine C'.
Calculate the edge counters n_{ij} for all (i, j).
Initialize the best known solution s^* with the solution produced by the tabu search.
Generate a set of promising routes $\bar{R}$ guided by the edge counters n_{ij}.
Sort the routes in $\bar{R}$ based on a desirability measure.
while a time limit has not been reached or a maximum number of IPs has not been solved **do**
  Select a subset of routes R of $\bar{R}$;
  Solve the route optimization IP over the set R;
  if the solution found by the route optimization IP improves s^* **then**
    Update s^*.
  end if
end while

6.4 Computational results

While in [1] Dror and Trudeau algorithm and the tabu search heuristic are compared and in [4] the improvements obtained by the optimization-based heuristic upon the tabu search heuristic are reported, we compare here the three algorithms on the same set of instances. This set is formed by 49 instances which are derived from seven basic instances; the same instances used to test the tabu search algorithm of Archetti et al. ([1]). These basic instances vary in terms of the number of customers (ranging from 50 to 199) and in terms of vehicle capacity (ranging from 140 to 200). Six additional set of instances are created by changing the demand of the customers in the basic instances, but keeping all other characteristics the same. Each of the new sets of instances is characterized by a lower bound on the demand of the customers, α, and by an upper bound on the demand of the customers, γ, expressed as a fraction of the vehicle capacity Q, i.e., $\alpha, \gamma \in [0, 1]$ with $\alpha \leq \gamma$. The demand d_i of customer i is set to

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

for some random δ in [0,1], i.e., the demand d_i of customer i is chosen randomly in the interval $[\alpha Q, \gamma Q]$. The six additional sets of instances are created with the following lower and upper bound combinations (α, γ): (0.01,0.1), (0.1,0.3), (0.1,0.5), (0.1,0.9), (0.3,0.7) and (0.7,0.9) (following Dror and Trudeau [13]). Thus, customers demands are always lower than the vehicle capacity. The reason is that Dror and Trudeau algorithm works only for this case thus we can only compare the three algorithms on this case.

The results are shown in Table 1. Here, when $\alpha = \gamma = 0$ it means the the demands of the original instance are taken. The performance of the tabu search heuristic is slightly different from what is reported in [1] for two main reasons: the instances are not the same and in [1] five different runs were made on each instance for the tabu search. By observing Table 1 we see that the tabu search heuristic outperforms Dror and Trudeau heuristic in 41 cases over 49 and that the optimization-based heuristic improves the tabu search solution in 41 cases over 49. As observed in [1], the tabu search heuristic produces better results when the demands are small with respect to the vehicle capacity. In Table 2 the computational times required by the three heuristics are shown and we can observe that the computational times required by the two most effective heuristics are much higher than the time required by the local search heuristic. The tabu search heuristic takes much more time to solve instances with large demands. The reason is that in this case the number of moves to be evaluated at each iteration is very large. Two observations have to be made on these results. First, in [1] the authors made computational tests where the tabu search heuristic runs for one minute only and they showed that also in this case it is much more effective than Dror and Trudeau heuristic. In this work we let the tabu search run until the end because we needed data to run the optimization-based heuristic. Second, while the results presented here concern only a single version of the optimization-based heuristic, in [4] different variants of the algorithm were tested and, by taking the best one for each instance, it was possible to improve further the results.

Conclusions

In this survey we have summarized what is known on the Split Delivery Vehicle Routing Problem. The problem, that is NP-hard even in simple cases, is more difficult to solve than a Vehicle Routing Problem. This is due to the fact that, while in the VRP it has to be decided whether a vehicle visits a customer or not, in the SDVRP a more complex decision has to be taken, that is how much of the demand of a customer is served by a vehicle. While some heuristics have been designed and tested, no exact algorithm has been yet proposed and this fact confirms how difficult this problem is. On the other hand, the cost savings that can be obtained by allowing split deliveries can be relevant and justify the interest that this problem has raised.

References

[1] C. Archetti, A. Hertz, M.G. Speranza, "A tabu search algorithm for the split delivery vehicle routing problem", *Transportation Science*, 40: 64-73, 2006.

[2] C. Archetti, R. Mansini, M.G. Speranza, "Complexity and reducibility of the skip delivery problem", *Transportation Science*, 39: 182-187, 2005.

[3] C. Archetti, M.W.P. Savelsbergh, M.G. Speranza, "Worst-case analysis for split delivery vehicle routing problems", *Transportation Science*, 40: 226-234, 2006.

[4] C. Archetti, M.W.P. Savelsbergh, M.G. Speranza, "To split or not to split: That is the question", to appear in *Transportation Research E.*

[5] C. Archetti, M.W.P. Savelsbergh, M.G. Speranza, "An optimization-based heuristic for the split delivery vehicle routing problem", submitted.

[6] C. Archetti, M.G. Speranza, "Vehicle routing in the 1-skip collection problem", *Journal of the Operational Research Society*, 55: 717-727, 2004.

[7] M. Ball, L. Bodin, R. Baldacci, A. Mingozzi, "The rollon-rolloff vehicle routing problem", *Transportation Science*, 34: 271-288, 2000.

[8] J.M. Belenguer, M.C. Martinez, E. Mota, "A lower bound for the split delivery vehicle routing problem", *Operations Research*, 48: 801-810, 2000.

[9] S. Chen, B. Golden, E. Wasil, "The split delivery vehicle routing problem: A new heuristic and benchmark problems", presented at ROUTE 2005.

[10] L. De Meulemeester, G. Laporte, F.V. Louveaux, F. Semet, "Optimal sequencing of skip collections and deliveries", *Journal of the Operational Research Society*, 48: 57-64, 1997.

[11] M. Dror, L. Levy, "A vehicle routing improvement algorithm comparison of a "greedy" and a matching implementation for inventory routing", *Computers and Operations Research*, 13: 33–45, 1986.

[12] M. Dror, G. Laporte, P. Trudeau, "Vehicle routing with split deliveries", *Discrete Applied Mathematics*, 50: 239-254, 1994.

[13] M. Dror, P. Trudeau, "Savings by split delivery routing", *Transportation Science*, 23: 141-145, 1989.

[14] M. Dror, P. Trudeau, "Split delivery routing", *Naval Research Logistics*, 37: 383-402, 1990.

[15] P.W. Frizzell, J.W. Giffin, "The split delivery vehicle scheduling problem with time windows and grid network distances", *Computers and Operations Research*, 22: 655-667, 1995.

[16] M. Gendreau, P. Dejax, D. Feillet, C. Gueguen, "Vehicle routing with time windows and split deliveries", working paper.

[17] M. Gendreau, A. Hertz, G. Laporte, "New insertion and postoptimization procedures for the traveling salesman problem", *Operations Research*, 40: 1086–1094, 1992.

[18] C. Gueguen, "Méthodes de résolution exacte pour les problèmes de tournées de véhicules", *PhD thesis*, École Centrale Paris, 1999.

[19] S. Lin, "Computer solutions of the traveling salesman problem", *Bell System Technical Journal*, 44: 2245–2269, 1965.

[20] P.A. Mullaseril, M. Dror, J. Leung, "Split-delivery routing in livestock feed distribution", *Journal of the Operational Research Society*, 48: 107-116, 1997.

[21] G. Sierksma, G.A. Tijssen, "Routing helicopters for crew exchanges on off-shore locations", *Annals of Operations Research*, 76: 261-286, 1998.

[22] P. Toth, D. Vigo, "The granular tabu search and its application to the vehicle-routing problem", *INFORMS Journal on Computing*, 15: 333-346, 2003.