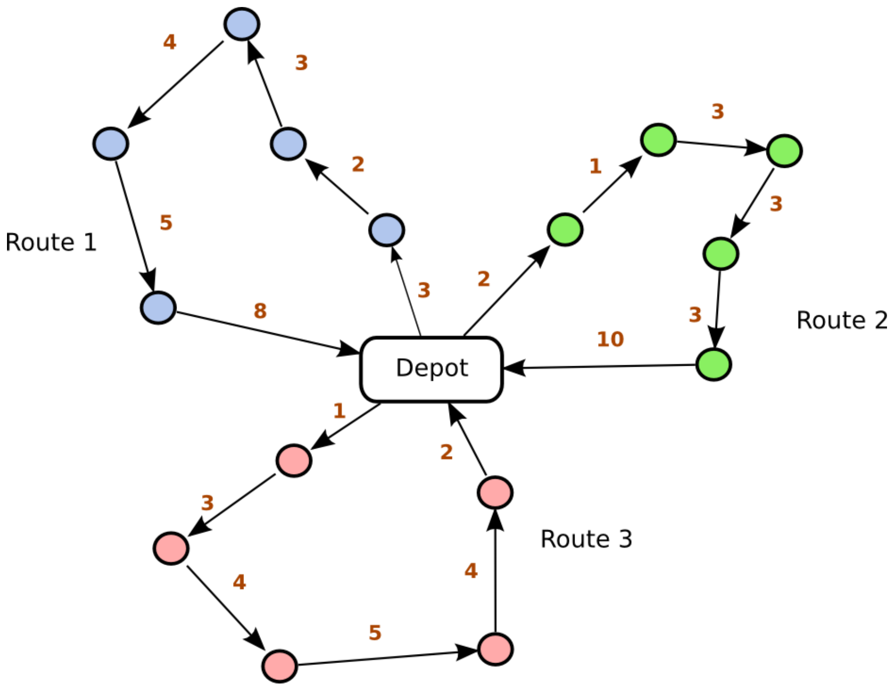

The Vehicle Routing Problem with Discrete Split Delivery and Time Windows

Ilaria Vacca
Matteo Salani

STRC 2009 **September 2009**

TRANSP-OR
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

STRC 2009

The Vehicle Routing Problem with Discrete Split Delivery and Time Windows

Ilaria Vacca
Transp-OR
EPFL
1015 Lausanne
phone: +41 21 693 24 32
fax: +41 21 693 80 60
ilaria.vacca@epfl.ch

Matteo Salani
Transp-OR
EPFL
1015 Lausanne
phone: +41 21 693 24 32
fax: +41 21 693 80 60
matteo.salani@epfl.ch

September 2009

Abstract

The Discrete Split Delivery Vehicle Routing Problem with Time Windows (DSDVRPTW) consists of designing the optimal set of routes to serve, at least cost, a given set of customers while respecting constraints on vehicles' capacity and customer time windows. The delivery request of a customer consists of several discrete items which cannot be split further. The problem belongs to the class of split delivery problems since each customer's demand can be split in orders, i.e. feasible combinations of items, and each customer can be visited by more than one vehicle. In this work, we model the DSDVRPTW as a mixed integer linear program, assuming that all feasible orders are known in advance and that each vehicle can serve at most one order per customer. Remarkably, service time at customer's location depends on the serviced combination of items, which is a modeling feature rarely found in literature. We present a branch-and-price algorithm, analyzing the implications of the classical Dantzig-Wolfe reformulation. Preliminary computational results on instances based on Solomon's data set are discussed.

Keywords

vehicle routing, discrete split delivery, Dantzig-Wolfe decomposition, column generation, branch-and-price

1 Introduction

The capacitated Vehicle Routing Problem (VRP) consists of designing the optimal routes for a set of vehicles with given capacity in order to serve a set of customers. Customer's demand must be delivered by exactly one vehicle, while respecting vehicles' capacity.

The Split Delivery Vehicle Routing Problem (SDVRP) is a relaxed version of the classical capacitated VRP in which the number of visits to customer locations is no longer constrained to be at most one. In the SDVRP each customer can be visited by more than one vehicle which serves a fraction of its demand. It has been shown that this relaxation could yield to substantial savings on the total traveled distance, up to 50% in some instances (Archetti *et al.*, 2006a, 2008a).

The problem and some properties have been introduced by Dror and Trudeau (1989) with a local search heuristic. Next, Dror *et al.* (1994) introduce a mathematical formulation based on integer programming and solved through a cutting plane approach. Lower bounds have been studied by Belenguer *et al.* (2000). Exact methods (Gueguen, 1999; Jin *et al.*, 2007) as well as heuristic algorithms (Archetti *et al.*, 2006b; Chen *et al.*, 2007; Jin *et al.*, 2008; Archetti *et al.*, 2008b) have been proposed to solve the SDVRP. Gendreau *et al.* (2006) and Desaulniers (2008) address the problem with time windows and present exact approaches based on column generation and branch-and-bound techniques. Lower bounds have been studied by Ceselli *et al.* (2009b) and a tabu search algorithm has been proposed by (Ho and Haugland, 2004).

In the Discrete Split Delivery Vehicle Routing Problem (DSDVRP) the demand of a customer is discretized and consists of several items which cannot be split further. The problem belongs to the class of split delivery problems since each customer's demand can be fractionated and each customer can be visited by more than one vehicle.

Nakao and Nagamochi (2007) present the problem and propose a dynamic programming based heuristic. The algorithm is compared to other existing heuristics for the VRP and computational results on real-world instances are provided. Ceselli *et al.* (2009a) present an exact approach to a real-world VRP in which customers' orders can be split among several vehicles in a discrete fashion. The authors propose a three level order aggregation which ends up, at the last level, in considering any possible combination of items. The VRP with splittable and discrete demand arises in some practical applications, such as the routing of helicopters for crew exchanges on off-shore locations (Sierksma and Tijssen, 1998) and the so-called Field Technician Scheduling Problem (Xu and Chiu, 2001); however, authors do not specifically relate their problems to the DSDVRP.

In the reminder of the paper we study the Discrete Split Delivery Vehicle Routing Problem with Time Windows (DSDVRPTW). We assume that demand can be split in orders, i.e. feasible combinations of items, that each vehicle can serve at most one order per customer and that service time at customer's location depends on the delivered combination of items. Remarkably, this is a modeling feature rarely found in literature, where service times are usually assumed to be independent of the delivered quantities. Section 2 provides an arc-flow formulation for the DSDVRPTW. In Section 3 we reformulate the problem using Dantzig-Wolfe decomposition and we illustrate the column generation scheme. Implementation issues are discussed in Section 4 and preliminary computational results are presented in Section 5. Finally, conclusions and future research perspectives are discussed in Section 6.

2 The model

In this section we present a mixed integer linear program for the DSDVRPTW based on an arc-flow formulation.

Let $G(V, E)$ be a complete graph with $V = \{0\} \cup N$, where vertex $\{0\}$ represents the depot and $N = \{1, ..., n\}$ is the set of customers to be served. Each arc $(i, j) \in E$ has a cost c_{ij} and a travel time t_{ij}. K denotes the set of available vehicles with identical capacity Q. The set of items R is defined as $R = \bigcup_{i \in N} R_i$, where R_i represents the set of items to be delivered to customer $i \in N$. Furthermore, $R_i \cap R_j = \emptyset \ \forall i \neq j, \ \ i, j \in N$, meaning that any item $r \in R$ is univocally associated to a customer $i \in N$. Each item $r \in R$ has a size q^r and a service time t^r. Items are delivered in orders, i.e. combinations of items. The set of orders C is defined as $C = \bigcup_{i \in N} C_i$, where C_i represents the set of feasible orders for customer $i \in N$. Furthermore, $C_i \cap C_j = \emptyset \ \forall i \neq j, \ \ i, j \in N$, meaning that any order $c \in C$ is univocally associated to a customer $i \in N$. Each combination $c \in C$ has a size $q_c = \sum_{r \in R} e_c^r q^r$ and a service time t_c such that:

$$\max_{r \in R} e_c^r t^r \leq t_c \leq \sum_{r \in R} e_c^r t^r \tag{1}$$

where e_c^r is a binary parameter equal 1 if item $r \in R$ is delivered in order $c \in C$ and 0 otherwise. Interval $[a_i, b_i]$ denotes the time window for customer $i \in N$.

We define the following decision variables:

x_{ij}^k	binary, equal to 1 if arc $(i, j) \in E$ is used by vehicle $k \in K$;
y_c^k	binary, equal to 1 if vehicle $k \in K$ delivers order $c \in C$;
T_i^k	≥ 0, represents the arrival time of vehicle $k \in K$ at location of customer $i \in N$.

The discrete split delivery vehicle routing problem with time windows can be formulated as follows:

$$z^*_{IP} = \min \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x^k_{ij} \tag{2}$$

$$\sum_{j \in V} x^k_{0j} = 1 \qquad \forall k \in K, \tag{3}$$

$$\sum_{j \in V} x^k_{ij} - \sum_{j \in V} x^k_{ji} = 0 \qquad \forall k \in K, \forall i \in V, \tag{4}$$

$$\sum_{j \in V} x^k_{ij} = \sum_{c \in C_i} y^k_c \qquad \forall k \in K, \forall i \in N, \tag{5}$$

$$\sum_{k \in K} \sum_{c \in C} e^r_c y^k_c = 1 \qquad \forall r \in R, \tag{6}$$

$$\sum_{c \in C_i} y^k_c \leq 1 \qquad \forall k \in K, \forall i \in N, \tag{7}$$

$$T^k_i + \sum_{c \in C_i} t_c y^k_c + t_{ij} - T^k_j \leq (1 - x^k_{ij}) M \qquad \forall k \in K, \forall i \in N, \forall j \in V, \tag{8}$$

$$T^k_i - t_{0i} \geq (1 - x^k_{0i}) M \qquad \forall k \in K, \forall i \in N, \tag{9}$$

$$T^k_i \geq a_i \sum_{j \in V} x^k_{ij} \qquad \forall k \in K, \forall i \in N, \tag{10}$$

$$T^k_i + \sum_{c \in C_i} t_c y^k_c \leq b_i \sum_{j \in V} x^k_{ij} \qquad \forall k \in K, \forall i \in N, \tag{11}$$

$$\sum_{c \in C} q_c y^k_c \leq Q \qquad \forall k \in K, \tag{12}$$

$$x^k_{ij} \in \{0, 1\} \qquad \forall k \in K, \forall (i,j) \in E, \tag{13}$$

$$y^k_c \in \{0, 1\} \qquad \forall k \in K, \forall c \in C, \tag{14}$$

$$T^k_i \geq 0 \qquad \forall k \in K, \forall i \in N. \tag{15}$$

where M is a sufficiently large constant. The objective function (2) minimizes the total traveling costs. Flow conservation is ensured by constraints (3)–(4), while constraints (5) link variables x and y. Demand satisfaction is ensured by constraints (6): all items must be delivered (but not all combinations). Constraints (7) ensure that every vehicle delivers at most one order per customer. Precedence, time windows and capacity constraints are ensured by constraints (8)–(9), (10)–(11) and (12). Finally, the domain of variables is defined by (13), (14) and (15).

Remarkably, the service time at customer location depends on the selected order. This feature is modeled by the term $\sum_{c \in C_i} t_c y^k_c$ in constraints (8): it increases the complexity of the model, with respect to the same type of precedence constraints in classical VRP formulations with time windows. In particular, Gendreau *et al.* (2006) and Desaulniers (2008) assume that service times are independent on the quantities delivered.

3 Column generation

In this section we reformulate the DSDVRPTW model (2)–(15) via Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) and provide the formulations of the master problem and pricing subproblem. The master problem is solved by means of column generation.

3.1 Master problem

Let (3)-(5) and (7)-(15) be the constraints that define the subproblem and let $D^k = conv\{(x^k, y^k, T^k) \,|\, (x^k, y^k, T^k)$ satisfies $(3) - (5); (7) - (15)$ for $k\}$ be the feasible bounded domain of the subproblem associated to vehicle $k \in K$. Let P^k be the set of extreme points of D^k. Each extreme point $d_p = (x_p^k, y_p^k, T_p^k)$, $p \in P^k$ represents a feasible route for vehicle k with respect to vehicle's capacity and customers' time windows, delivering a unique order to every customer visited by the tour.

Since vehicles present identical restrictions (i.e. same capacity), all subproblems can be aggregated into a single subproblem. We denote as $D = conv\{(x, y, T) \,|\, (x, y, T)$ satisfies $(3) - (5); (7) - (15)\}$ the feasible domain of the subproblem and P the set of extreme points of D. Each extreme point $d_p = (x_p, y_p, T_p)$, $p \in P$ represents now a feasible route that can be covered by any vehicle among the $|K|$ available.

The definition of the master problem requires the following additional notation:

c_p	cost of path $p \in P$: $c_p := \sum_{(i,j)\in p} c_{ij}$;
α_p^i	binary parameter equal to 1 if path $p \in P$ visits $i \in N$;
β_{pc}^r	binary parameter equal to 1 if path $p \in P$ delivers item $r \in R$ in order $c \in C$;
γ_p^r	binary parameter equal to 1 if path $p \in P$ delivers item $r \in R$: $\gamma_p^r := \sum_{c\in C} \beta_{pc}^r$.

After some standard adjustments and aggregation, the master problem can be formulated as follows:

$$\min \sum_{p\in P} c_p \lambda_p \tag{16}$$

$$\sum_{p\in P} \gamma_p^r \lambda_p = 1 \quad \forall r \in R \quad (\pi_r) \tag{17}$$

$$0 \leq \sum_{p\in P} \lambda_p \leq |K| \qquad (\pi_0) \tag{18}$$

$$0 \leq \sum_{p\in P} \alpha_p^i \lambda_p \leq |K| \quad \forall i \in N \quad (\mu_i) \tag{19}$$

$$\lambda_p \geq 0 \quad \forall p \in P. \tag{20}$$

where λ_p are the decision variables associated to paths $p \in P$, π_r are the dual variables associated to constraints (17), π_0 is the dual variable associated to constraint (18) and μ_i are the dual variables associated to constraints (19).

The objective function (16) minimizes the total traveling costs. Constraints (17) ensure that all items are delivered to customers, while constraint (18) ensures that the number of chosen routes does not exceed the number of available vehicles. Constraints (19) are redundant, since they ensure that the number of chosen routes visiting a given customer does not exceed, again, the number of available vehicles. However, they are needed to implement the branching scheme in the solution algorithm (see Section 4).

We remark that constraints (17) need to be modeled as partitioning constraints in the DSD-VRPTW, unlike common reformulations for routing problems that generally make use of covering constraints. This is due to the fact that, for every customer $i \in N$, the set of orders C_i does not necessarily contain all subsets of items $r \in R_i$, but only the subsets that are considered feasible with respect to the problem definition (incompatibilities between specific items, restrictions on the order size, etc.).

3.2 Pricing subproblem

The reduced cost of a route $p \in P$ is defined as:

$$\tilde{c}_p := c_p - \sum_{r \in R} \pi_r \gamma_p^r - \pi_0 - \sum_{i \in N} \mu_i \alpha_p^i \tag{21}$$

In a column generation scheme, given a dual solution of the (restricted) master problem, the pricing subproblem identifies the route (column) p^* with the minimum reduced cost:

$$p^* = \arg\min_{p \in P} \{\tilde{c}_p\} = \arg\min_{p \in P} \{c_p - \sum_{r \in R} \pi_r \gamma_p^r - \pi_0 - \sum_{i \in N} \mu_i \alpha_p^i\} \tag{22}$$

If $\tilde{c}_{p^*} < 0$, the column is added to the (restricted) master problem and the procedure is iterated; otherwise, the current primal solution is proven to be optimal for the master problem and the procedure terminates.

The subproblem formulation relies on variables x, y and T defined in Section 2 (without index k, since we have aggregated the subproblems) and can be written as follows:

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} - \sum_{r\in R} \pi_r (\sum_{c\in C} y_c e_c^r) - \pi_0 - \sum_{i\in N} \mu_i (\sum_{j\in N\cup\{0\}} x_{ij}) \tag{23}$$

$$\sum_{j\in V} x_{0j} = 1 \tag{24}$$

$$\sum_{j\in V} x_{ij} - \sum_{j\in V} x_{ji} = 0 \qquad \forall i \in V, \tag{25}$$

$$\sum_{j\in V} x_{ij} = \sum_{c\in C_i} y_c \qquad \forall i \in N, \tag{26}$$

$$\sum_{c\in C_i} y_c \leq 1 \qquad \forall i \in N, \tag{27}$$

$$T_i + \sum_{c\in C_i} t_c y_c + t_{ij} - T_j \leq (1 - x_{ij})M \qquad \forall i \in N, \forall j \in V, \tag{28}$$

$$T_i - t_{0i} \geq (1 - x_{0i})M \qquad \forall i \in N, \tag{29}$$

$$T_i \geq a_i \sum_{j\in V} x_{ij} \qquad \forall i \in N, \tag{30}$$

$$T_i + \sum_{c\in C_i} t_c y_c \leq b_i \sum_{j\in V} x_{ij} \qquad \forall i \in N, \tag{31}$$

$$\sum_{c\in C} q_c y_c \leq Q \tag{32}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E, \tag{33}$$

$$y_c \in \{0,1\} \qquad \forall c \in C, \tag{34}$$

$$T_i \geq 0 \qquad \forall i \in N. \tag{35}$$

Analyzing the objective function, we can observe that two major decisions are made in the subproblem:

a) the sequence of customers $i \in N$ visited in the route (cost component c_{ij});

b) for each customer in the route, the order $c \in C$ to be delivered, and therefore the subset of items $r \in R$ delivered by the route (cost component e_c^r).

The pricing problem (23)–(35) is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) defined on a network which has one node for every order $c \in C$ and whose arcs have transit time equals to $(t_{ij} + t_c)$. In particular, the choice on the orders to be delivered by the route has impact on the complexity of the subproblem.

4 Implementation

For solving the DSDVRPTW we have implemented a branch-and-price algorithm (Barnhart *et al.*, 1998; Lübbecke and Desrosiers, 2005) which relies on the master problem and the pricing subproblem introduced in Section 3.

The implementation is standard. The pricing problem is solved using bounded bi-directional dynamic programming (Righini and Salani, 2006) with decremental state space relaxation (Righini and Salani, 2008). The algorithm is initialized by a preprocessing phase, used to identify and remove trivially dominated combinations, and by a simple greedy algorithm used to find a feasible solution to the problem. Such solution allows to compute an upper bound to the cost of the solution and to the number of vehicles.

4.1 Branching scheme

In the search tree, branching is required when the optimal solution of the master problem obtained via column generation is not integer. We have implemented a branching scheme which consists of three hierarchical levels:

1. if the total number of vehicles $\tilde{K} = \sum_{p \in P} \lambda_p$ is fractional, branching is performed on constraint (18) by enforcing $\sum_{p \in P} \lambda_p \leq \lfloor \tilde{K} \rfloor$ on the first child node and $\lceil \tilde{K} \rceil \leq \sum_{p \in P} \lambda_p$ on the second child node;
2. if the number of vehicles visiting a customer $\tilde{K}_i = \sum_{p \in P} \alpha_p^i \lambda_p, i \in N$ is fractional, branching is performed on constraint (19) by enforcing $\sum_{p \in P} \alpha_p^i \lambda_p \leq \lfloor \tilde{K}_i \rfloor$ on the first child node and $\lceil \tilde{K}_i \rceil \leq \sum_{p \in P} \alpha_p^i \lambda_p$ on the second child node;
3. finally, if there is an arc $(i, j) \in E$ visited a fractional number of times, branching is performed by enforcing $x_{ij} = 1$ on the first child node and $x_{ij} = 0$ on the second child node. This additional constraint is handled by modifying the network of the pricing subproblem.

5 Computational experiments

Algorithms are coded in ANSI C, compiled with gcc 4.1.2 and computational experience is run under a linux operating system on a 2Ghz Intel processor equipped with 2GB of RAM. All restricted master problems are solved using GLPK version 4.39.

5.1 Instances

To the best of our knowledge there is no standard dataset used in the literature for the DSD-VRPTW. The most related contribution is that of Nakao and Nagamochi (2007) for which the instances are not available.

We generated our test bed from the well-known Solomon's data set (Solomon, 1983). Solomon's instances are divided in classes. In class R1 customer locations are randomly generated by a random uniform distribution, while in class C1 customers are placed in clusters. Class RC1 contains a mix of random and clustered locations.

For every instance of classes R1 (12 instances), C1 (9 instances) and RC1 (8 instances) we considered the first $n = 25, 50$ customers and we discretized the demand of each customer in 12 items ($|R_i| = 12 \ \ \forall i \in N$). For each customer, we generated 7 orders as follows:

- 1 full order (containing 12 items);
- 2 complementary orders 50%-50% (containing 6 items each, partitioned);
- 2 complementary orders 75%-25% (containing 9 and 3 items respectively, partitioned);
- 2 complementary 90%-10% orders (containing 11 and 1 items respectively, partitioned);

and we considered 3 possible scenarios:

A: full order + 50-50% orders ($|C_i| = 3$);
B: full order + 50-50% orders + 75-25% orders ($|C_i| = 5$);
C: full order + 50-50% orders + 75-25% orders + 90-10% orders ($|C_i| = 7$).

The full order has been always included in order to allow the comparison of the DSDVRPTW with the classical VRP with Time Windows (VRPTW). The unsplittable case, which is trivially composed of the full order only ($|C_i| = 1$), is denoted as scenario O.

In order to enhance splitting, we considered more restrictive capacities than Solomon's, as already suggested by Gendreau *et al.* (2006). Instances have been tested with $Q = 30, 50$ and 100.

From the 29 original Solomon's instances (12 for class R1, 9 for class C1 and 8 for class RC1), we derived 174 instances: 29×2 (customers) $\times 3$ (capacities) . Each instance has been tested under the three DSDVRPTW scenarios A, B, C and compared to the VRPTW scenario O.

				A		B		C	
n	class	nb_inst	Q	nb_solved	t	nb_solved	t	nb_solved	t
25	R1	12	30	12	87	10	694	6	1554
			50	11	342	6	463	5	522
			100	9	16	10	129	9	551
25	C1	9	50	9	273	0	x	0	x
			100	3	947	0	x	0	x
25	RC1	8	30	8	317	0	x	0	x
			100	8	222	2	1542	0	x
50	R1	12	30	1	3011	0	x	0	x
			50	1	1527	0	x	0	x
			100	2	120	2	509	1	93
50	RC1	8	50	7	723	0	x	0	x
			100	1	1953	0	x	0	x

Table 1: Summary of the branch-and-price results.

5.2 Preliminary results

Table 1 presents a summary of the instances solved by the branch-and-price within 1 hour of computational time. Instances are grouped by the number of customers (n) and the capacity (Q). The number of instances of each class is also provided (*nb_inst*). For each group, the table provides the number of instances solved at optimality (*nb_solved*) and the average computational time in seconds (*t*) for each DSDVRPTW scenario.

We were able to solve 72, 30 and 21 out of 174 instances for scenarios A, B and C, respectively. The difficulty of solving the instances increases with the size of $|C|$: 75, 125 and 175 orders with 25 customers and 150, 250, and 350 orders with 50 customers for scenarios A, B and C, respectively. This difficulty also increases with the number of customers: we were able to solve 69% (A), 32% (B) and 23% (C) of instances with $n = 25$, whereas only 14% (A), 2% (B) and 1% (C) of instances with $n = 50$ were solved at optimality. The average computational time is also affected by the size of $|C|$ and the number of customers.

Instances of class C1 are the most difficult to solve; on the contrary, instances of class R1 are the easiest to solve. For 25 customers, there are 32 (A), 26 (B) and 20 (C) solved instances out of 36 for class R1; 12 solved instances out of 27 for scenario A in class C1; 16 (A) and 2 (B) solved instances out of 24 for class RC1. On average, 72% of instances were solved in class R1, 25% in class RC1 and only 15% in class C1. For 50 customers, class RC1 seems slightly easier to solve than class R1 (on average, 11% versus 6% of solved instances), while

			O			**A**			**B**			**C**		
n	Q	id	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
25	30	R101	795.6	13	0	**795.1**	13	2	x			x		
		R102	789.1	13	0	**772.3**	13	29	x			**761.2**	**12**	797
		R103	759.6	12	0	759.6	12	73	**751.7**	12	882	**745.3**	12	1644
		R104	759.6	12	0	759.6	12	218	**747.0**	12	990	**745.3**	12	2593
		R105	775.7	12	0	**775.3**	13	4	**773.2**	12	82	**773.2**	12	1581
		R106	772.6	13	0	**763.7**	**12**	29	**756.6**	**12**	294	**753.4**	**12**	1019
		R107	748.5	12	0	748.5	12	80	**744.1**	12	1056	x		
		R108	748.5	12	0	748.5	12	326	**744.1**	12	1388	x		
		R109	754.6	12	0	754.6	12	12	**750.2**	12	109	**750.2**	12	1690
		R110	748.5	12	0	748.5	12	43	**744.1**	12	240	x		
		R111	754.6	12	0	754.6	12	63	**750.2**	12	786	x		
		R112	748.5	12	0	748.5	12	171	**744.1**	12	1117	x		
25	50	R101	635.0	9	0	**631.5**	**8**	0	**631.5**	**8**	0	**631.5**	**8**	1
		R102	580.7	8	0	580.7	8	8	580.7	8	1973	580.7	8	644
		R103	534.3	7	0	534.3	7	11	x			534.3	7	1852
		R104	527.3	7	0	527.3	7	17	x			x		
		R105	596.1	8	0	**588.9**	8	1	**585.4**	8	5	**585.4**	8	23
		R106	543.3	7	0	**542.5**	7	9	**542.3**	7	273	x		
		R107	527.7	7	3	527.7	7	2348	x			x		
		R109	524.6	7	0	524.6	7	3	524.6	7	60	524.6	7	91
		R110	536.7	7	0	**529.1**	7	446	x			x		
		R111	521.6	7	2	521.6	7	889	x			x		
		R112	515.8	7	0	515.8	7	28	515.8	7	470	x		
25	100	R101	617.1	8	0	617.1	8	0	617.1	8	0	617.1	8	1
		R102	547.1	7	0	x			547.1	7	10	x		
		R103	454.6	5	0	454.6	5	4	454.6	5	22	454.6	5	82
		R104	416.9	4	0	416.9	4	24	416.9	4	135	416.9	4	430
		R105	530.5	6	0	530.5	6	1	530.5	6	4	530.5	6	12
		R106	465.4	5	1	465.4	5	9	465.4	5	184	465.4	5	1394
		R107	428.4	4	1	428.4	4	36	428.4	4	306	428.4	4	1058
		R109	441.3	5	0	441.3	5	5	441.3	5	26	441.3	5	92
		R110	444.1	5	1	444.1	5	45	444.1	5	394	444.1	5	1344
		R111	428.8	4	1	428.8	4	23	428.8	4	215	428.8	4	547

Table 2: Optimal solutions for class R1, $n = 25$ customers.

no instances in class C1 were solved.

Optimal solutions are detailed in tables 2, 3, 4, 5 and 6. For each instance, we provide the value of the optimal integer solution (z_{IP}), the number of vehicles (veh) and the computational time in seconds (t). The three DSDVRPTW scenarios A, B, C and compared to the unsplittable VRPTW scenario O: figures highlighted in bold denote savings due to split deliveries. Instances

			O			**A**			**B**			**C**		
n	Q	id	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
25	50	C101	516.9	10	0	**516.8**	10	4	x			x		
		C102	516.6	10	0	**516.5**	10	157	x			x		
		C103	516.6	10	1	**516.5**	10	725	x			x		
		C104	516.8	10	2	**516.4**	10	1223	x			x		
		C105	516.9	10	0	**516.8**	10	33	x			x		
		C106	516.9	10	0	**516.8**	10	11	x			x		
		C107	516.9	10	0	**516.8**	10	50	x			x		
		C108	516.8	10	0	**516.7**	10	102	x			x		
		C109	516.8	10	0	**515.9**	10	153	x			x		
25	100	C101	291.9	5	0	291.9	5	336	x			x		
		C105	291.9	5	1	291.9	5	1321	x			x		
		C106	291.9	5	1	291.9	5	1183	x			x		

Table 3: Optimal solutions for class C1, $n = 25$ customers.

that are not feasible for the unsplittable case because of insufficient capacity are denoted by *"Q < demand"*. Instances not solved at optimality within 1 hour of computational time are denoted by *"x"*.

We can observe that split deliveries are more frequent for instances with small Q values, although they also occur for certain instances with $Q = 100$. In a few cases, split deliveries not only decrease the total traveling costs but also allow to save one vehicle.

6 Conclusions

Analyzing the results, we can conclude that obtaining optimal solutions is difficult, even with a small number of orders per customer. Furthermore, only a limited number of instances with 50 customers could be solved.

We guess that the bottleneck is in the pricing problem. Indeed, the underlying ESPPRC network is huge, since, in the worst case scenario, for every customer $i \in N$ we have that set C_i corresponds to the set of all subsets of R_i and therefore its size grows exponentially with the number of items. Computational results show that solving the ESPPRC on such a network may be impractical. Therefore, more efficient solution techniques need to be investigated.

			O			**A**			**B**			**C**		
n	Q	id	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
25	30	RC101	Q < demand			1438.0	18	75	x			x		
		RC102	Q < demand			1438.0	18	172	x			x		
		RC103	Q < demand			1438.0	18	342	x			x		
		RC104	Q < demand			1438.0	18	525	x			x		
		RC105	Q < demand			1438.0	18	165	x			x		
		RC106	Q < demand			1438.0	18	208	x			x		
		RC107	Q < demand			1438.0	18	373	x			x		
		RC108	Q < demand			1438.0	18	674	x			x		
25	100	RC101	534.3	6	0	534.3	6	9	534.3	6	265	x		
		RC102	523.7	6	1	523.7	6	111	x			x		
		RC103	514.7	6	1	**513.7**	6	293	x			x		
		RC104	506.7	6	3	506.7	6	496	x			x		
		RC105	527.5	6	0	527.5	6	37	x			x		
		RC106	515.6	6	0	515.6	6	27	515.6	6	2819	x		
		RC107	505.7	6	1	505.7	6	255	x			x		
		RC108	505.7	6	4	505.7	6	544	x			x		

Table 4: Optimal solutions for class RC1, $n = 25$ customers.

			O			**A**			**B**			**C**		
n	Q	id	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
50	30	R101	Q $<$ demand			1664.6	26	3011	x			x		
50	50	R101	1222.0	16	1	**1211.1**	16	1527	x			x		
50	100	R101	1044.0	12	0	1044.0	12	11	**1040.6**	12	20	**1040.6**	12	93
		R102	913.2	11	1	913.2	11	230	**911.9**	11	998	x		

Table 5: Optimal solutions for class R1, $n = 50$ customers.

			O			**A**			**B**			**C**		
n	Q	id	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t	z_{IP}	veh	t
50	50	RC101	1713.2	20	1	**1708.9**	20	100	x			x		
		RC102	1706.5	20	2	**1701.5**	20	570	x			x		
		RC103	1703.4	20	2	**1696.8**	20	501	x			x		
		RC104	1702.2	20	5	**1696.7**	20	1695	x			x		
		RC105	1703.9	20	1	**1700.1**	20	330	x			x		
		RC106	1705.7	20	1	**1699.0**	20	304	x			x		
		RC108	1702.2	20	6	**1696.7**	20	1561	x			x		
50	100	RC101	994.6	10	3	**993.8**	10	1953	x			x		

Table 6: Optimal solutions for class RC1, $n = 50$ customers.

References

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

Archetti, C., M. Savelsbergh and M. Speranza (2008a) To split or not to split: That is the question, *Transportation Research Part E*, **44**, 114–123.

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

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

Barnhart, C., E. Johnson, G. Nemhauser, M. Savelsbergh and P. Vance (1998) Branch-and-price: Column generation for solving huge integer programs, *Operations Research*, **46** (3) 316–329.

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

Ceselli, A., G. Righini and M. Salani (2009a) A column generation algorithm for a vehicle routing problem with economies of scale and additional constraints, *Transportation Science*, **43** (1) 56–69.

Ceselli, A., G. Righini and M. Salani (2009b) Column generation for the split delivery vehicle routing problem, *Technical Report*, Note del Polo - Ricerca 118, Dipartimento di Tecnologie dell'Informazione, Università degli Studi di Milano.

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

Dantzig, G. and P. Wolfe (1960) Decomposition principle for linear programs, *Operations Research*, **8**, 101–111.

Desaulniers, G. (2008) Branch-and-price-and-cut for the split delivery vehicle routing with time windows, *Technical Report*, **G-2008-32**, Les Chaiers du GERAD.

Dror, M., G. Laporte and P. Trudeau (1994) Vehicle routing with split deliveries, *Discrete and Applied Mathematics*, **50**, 239–254.

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

Gendreau, M., P. Dejax, D. Feillet and C. Gueguen (2006) Vehicle routing with time windows and split deliveries, *Technical Report*, Technical report 2006-851, Laboratoire d'Informatique d'Avignon.

Gueguen, C. (1999) Méthodes de résolution exacte pour les problèmes de tournées de véhicules, Ph.D. Thesis, Ecole centrale Paris.

Ho, S. and D. Haugland (2004) A tabu search heuristic for the vehicle routing problem with time windows and split deliveries, *Computers and Operations Research*, **31**, 1947–1964.

Jin, M., K. Liu and R. Bowden (2007) A two stage algorithm with valid inequalitites for the split delivery vehicle routing problem, *International Journal of Production Economics*, **105**, 228–242.

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

Lübbecke, M. and J. Desrosiers (2005) Selected topics in column generation, *Operations Research*, **53** (6) 1007–1023.

Nakao, Y. and H. Nagamochi (2007) A dp-based heuristic algorithm for the discrete split delivery vehicle routing problem, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, **1** (2) 217–226.

Righini, G. and M. Salani (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, *Discrete Optimization*, **3** (3) 255–273, September 2006.

Righini, G. and M. Salani (2008) New dynamic programming algorithms for the resource constrained shortest path problem., *Networks*, **51** (3) 155–170.

Sierksma, G. and G. Tijssen (1998) Routing helicopters for crew exchanges on off-shore locations, *Annals of Operations Research*, **76**, 261–286.

Solomon, M. (1983) Vehicle routing and scheduling with time windows constraints: Models and algorithms, Ph.D. Thesis, University of Pennsylvania.

Xu, J. and S. Y. Chiu (2001) Effective heuristic procedures for a field technician scheduling problem, *Journal of Heuristics*, **7** (5) 495–509, ISSN 1381-1231.