The Split Delivery Capacitated Team Orienteering Problem

C. Archetti[(1)] *N. Bianchessi*[(1)] *A. Hertz*[(2)]
M.G. Speranza[(1)]

[(1)] *Department of Quantitative Methods*
University of Brescia, Brescia, Italy
{archetti, bianche, speranza}@eco.unibs.it
[(2)] *École Polytechnique and GERAD, Montréal, Canada*
alain.hertz@gerad.ca

July 10, 2012

Abstract

In this paper we study the capacitated team orienteering problem where split deliveries are allowed. A set of potential customers is given, each associated with a demand and a profit. The set of customers to be served by a fleet of capacitated vehicles has to be identified in such a way that the profit collected is maximized, while satisfying constraints on the maximum time duration of each route and the vehicle capacity constraints. When split deliveries are allowed each customer may be served by more than one vehicle. We show that the profit collected by allowing split deliveries may be as large as twice the profit collected under the constraint that each customer has to be served by one vehicle at most. We then present a branch-and-price exact algorithm, and a hybrid heuristic. We show the effectiveness of the proposed approaches on benchmark instances and on a new set of instances that allow us to computationally evaluate the impact of split deliveries.

Keywords: Capacitated Team Orienteering Problem, split deliveries, worst-case analysis, branch-and-price, heuristic.

1 Introduction

The Capacitated Team Orienteering Problem (CTOP) is the problem where a set of potential customers, each of them associated with a demand and a profit, is given. The subset of customers to be served by a fleet of capacitated vehicles has to be identified in such a way that the profit collected is maximized, while satisfying constraints on the

maximum time duration of each route and the vehicle capacity constraints. In the CTOP each customer can be served by one vehicle at most. In this paper we study the Split Delivery Capacitated Team Orienteering Problem (SDCTOP), where a customer may be served by more than one vehicle.

The CTOP is the capacitated version of the Team Orienteering Problem (TOP), a well known problem that belongs to the class of routing problems with profits. A survey on the problems of this class when only one vehicle is available is due to Feillet, Dejax, Gendreau [15]. The TOP was introduced by Butt, Cavalier [9] under the name Multiple Tour Maximum Collection Problem, while the definition of TOP was introduced by Chao, Golden, Wasil [10]. Heuristics for the TOP were presented by Tsiligirides [21] Tang, Miller-Hooks [20] and by Archetti, Hertz, Speranza [5], whereas an exact algorithm was presented by Boussier, Feillet, Gendreau [8]. A survey on the Orienteering Problem, i.e. the single vehicle version of the TOP, the TOP and its applications can be found in [22].

The CTOP has been introduced in Archetti et al. [3] where heuristics and a branch-and-price algorithm were proposed. The CTOP is motivated by several applications where a carrier can select actual customers from a set of potential customers on the basis of their profit. Transactions between shippers and carriers more and more take place through the web. Databases of transportation demands from shippers are posted on the web in different forms and the carriers may access those databases and evaluate these demands. Typically carriers have to bid on the most profitable customers. It is crucial for the carriers to identify the customers that are most profitable, given their capacitated fleet of vehicles.

When split deliveries are allowed each customer may be served by more than one vehicle. In the Split Delivery Vehicle Routing Problem (SDVRP), that is the traditional Vehicle Routing Problem where split deliveries are allowed, it is well known that split deliveries may reduce the cost of routes and the number of vehicles used (see Archetti, Savelsbergh, Speranza [6], where it is shown that the cost may be halved). In this paper we show that the profit collected in the SDCTOP is, in the best case, as large as twice the profit collected in the CTOP.

We then present an exact branch-and-price algorithm and two heuristics for the solution of the SDCTOP. A first heuristic makes use of the columns generated during the execution of the branch-and-price to obtain heuristic solutions. A mixed integer linear programming (MILP) problem is solved on subsets of promising columns. The other heuristic is a hybrid algorithm that combines a tabu search scheme with an improvement phase where an ad hoc MILP problem is solved to intensify the search. We show the effectiveness of the proposed approaches on benchmark instances and on a new set of instances that also allow us to evaluate the impact of split deliveries.

The paper is organized as follows. In Section 2 we introduce the SDCTOP and some properties are derived in Section 3. In Section 4 we show that the profit collected can be doubled when allowing split deliveries. A mathematical programming formulation for the SDCTOP is presented in Section 5. An exact branch-and-price and two heuristics are

presented in Sections 6 and 7, respectively. Computational results are shown in Section 8.

2 Problem definition

We consider a directed graph $G = (V, A)$, where $V = \{1, \ldots, n\}$ is the set of vertices and A is the set of arcs. Vertex 1 represents the depot, whereas each vertex $i = 2, \ldots, n$ represents a potential customer. We denote by $V' = V \setminus \{1\}$ the set of potential customers. An arc $(i, j) \in A$ represents the possibility to travel from vertex i to vertex j. A non-negative integer demand d_i and a non-negative profit p_i are associated with each potential customer $i \in V'$. A non-negative travel time t_{ij} is associated with each arc $(i, j) \in A$. We assume that travel times satisfy the triangle inequality. A set of vehicles is available to provide the service, each with limited integer capacity Q. Let F denote the index set for the vehicles, with $|F| = m$. The route associated with each vehicle $f \in F$ starts and ends at the depot and must not exceed a given time limit T_{max}. Each customer $i \in V'$ may be served by more than one vehicle and the profit associated with each served customer can be collected at most once. The objective of the Split Delivery Capacitated Team Orienteering Problem (SDCTOP) is to maximize the total collected profit while satisfying the constraints on the time duration of each route and the vehicle capacity constraints. When the restriction that each customer can be served by one vehicle at most is added we have the Capacitated Team Orienteering Problem (CTOP). Obviously, in the CTOP, it is necessary to assume that $d_i \leq Q$, $i \in V'$.

In the following we will denote by $z(CTOP)$ and $z(SDCTOP)$ the value of an optimal solution to problems CTOP and SDCTOP, respectively.

3 Properties

Dror and Trudeau [14] showed an important property of the SDVRP based on the following concept.

Definition 1 *Given k customers $i_1, i_2, ..., i_k$ and k routes. Route 1 visits customers i_1 and i_2, route 2 visits customers i_2 and i_3, ..., route $k-1$ visits customers i_{k-1} and i_k, and route k visits customers i_k and $i_{k+1} = i_1$. The subset of customers $i_1, i_2, ..., i_k$ is called a k-split cycle.*

Dror and Trudeau [14] showed that an optimal solution to the SDVRP exists without k-split cycles. We extend this property to the SDCTOP.

Theorem 1 *If the time matrix satisfies the triangle inequality, then there exists an optimal solution to the SDCTOP where there is no k-split cycle (for any k).*

Proof: Let us consider an optimal solution to the SDCTOP and suppose it contains k-split cycles. Let us consider one of the k-split cycles. Let d be the minimum quantity delivered by any of the k routes to any of the customers of the k-split cycle. Let r be the route that delivered the quantity d in the k-split cycle and suppose, without loss of generality, that the customer to whom quantity d was delivered by route r is i_r. We can transform the optimal solution in a non worse solution without the k-split cycle as follows. Let us delete the customer i_r from the route r and let us modify the route by connecting the preceding and the subsequent customers. The duration of route r does not increase thanks to the triangle inequality and route r has an additional residual capacity d. Such residual capacity can be used to serve d units of the demand of the customer i_{r+1} previously served by the route $r+1$ (or 1) of the k-split cycle. Thus, d units of customer i_{r+1} are moved from route $r+1$ (or 1) to route r. This operation does not modify the duration of any route. Now route $r+1$ (or 1) has an additional residual capacity of d units. The procedure is repeated until the route that preceded route r in the k-split cycle is modified to serve the quantity d of customer i_r originally served by route r. The new solution is non worse than the previous one and then still optimal. The procedure is then repeated on any other k-split cycle, if any.

□

The following property is an immediate consequence of Theorem 1.

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

From this property, another property can be derived for the SDCTOP that was shown to hold for the SDVRP by Desaulniers [12]. A pair of reverse arcs is a pair $(i, j) \in A$ and $(j, i) \in A$.

Corollary 2 *If the time matrix satisfies the triangle inequality, then there exists an optimal solution such that, for each pair of reverse arcs that join pairs of customers, at most one of them is traversed.*

Archetti, Savelsbergh and Speranza [6] showed a property of the SDVRP that relates the number of splits to the number of routes in an optimal solution. Such property holds for the SDCTOP.

Let n_i be the number of routes that visit customer i. We say that customer i is a customer with split deliveries 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 $\sum_{i \in V'}(n_i - 1)$.

Theorem 2 *If the time matrix satisfies the triangle inequality, then there exists an optimal solution to the SDCTOP where the total number of splits is less than the number of routes.*

Proof: The proof of this property for the SDVRP in the paper by Archetti, Savelsbergh and Speranza [6] (called Property 2 in [6]) is based on the existence of an optimal solution without k-split cycles. Such latter property holds for the SDCTOP (see Theorem 1) and the same proof used in [6] can be applied here too. □

4 Worst-case analysis

In this section we analyze the savings that can be achieved by allowing split deliveries in the CTOP. As this section deals with the comparison between CTOP and SDCTOP, we assume that $d_i \leq Q$, $i \in V'$.

Lemma 1 *From an optimal solution to the SDCTOP without k-split cycles we can build a solution where each customer is visited at most once such that for each route of the SDCTOP solution at most another, not longer, route is created. All routes satisfy the time and capacity constraints. The number of routes in the new solution is not greater than $2m$.*

Proof: The proof follows the lines of part of the proof of Theorem 1 in [6].

We convert the solution to the SDCTOP working on the customers with split deliveries one at a time. When considering a customer with a split delivery, we perform one of two operations on the routes visiting the customer:

- *Operation 1.* Delete the customer from the route by connecting its predecessor with its successor on the route.
- *Operation 2.* Delete the customer from the route by connecting its predecessor with its successor on the route, mark the route, and create an out-and-back tour to the customer.

It is key that we will never mark a route more than once during the construction. It is important to observe that

- the deletion of a customer can only reduce the length of a route (because of the triangle inequality),
- the constructed out-and-back tours will never be longer than the (marked) route that prompted their creation (because of the triangle inequality), and
- a route in the original optimal SDCTOP solution results in at most one out-and-back tour being created (because routes are never marked more than once).

All routes in the SDCTOP solution are initially unmarked. Now, we proceed as follows. For each customer i with a split delivery we delete the customer from all the routes which visit it and we create a single out-and-back tour to i (delivering d_i). We arbitrarily mark one of the unmarked routes from which customer i has been removed. Such an unmarked route always exists because the original optimal SDCTOP solution does not contain any k-split cycle, due to Property 1. We continue by processing, one by one, all the customers with split deliveries on marked routes. If there remain customers with split deliveries on unmarked routes that have not been considered, we repeat starting from an arbitrarily chosen customer.

Note that all routes in the created solution to the CTOP satisfy the time and capacity constraints. □

Theorem 3

$$\frac{z(CTOP)}{z(SDCTOP)} \geq \frac{1}{2}$$

and this bound is tight.

Proof: From Lemma 1, we know that from an optimal solution to the SDCTOP we can construct a set of feasible routes where each customer is visited at most once. The number of such routes is not greater than twice the number of routes in the solution to the SDCTOP. The solution is possibly infeasible for the CTOP because the number of routes may be greater than m. The profit collected in the constructed solution would be the same if we had enough vehicles. Now we order the routes of the new solution by non-increasing profit and take the first m routes. The profit of these routes is clearly greater than or equal to half the profit of the optimal solution to the SDCTOP.

To show that the bound is tight, we take an instance similar to the instance presented by Archetti, Savelsbergh and Speranza [6]. There are m vehicles with capacity Q, with $Q \geq 2m$. The time limit is taken to be large. We consider $2m$ customers. Each customer has demand $\frac{Q}{2} + 1$ and profit p. An optimal solution to the CTOP will have to visit all customers with out-and-back tours. As only m vehicles are available only m customers can be visited with a total collected profit equal to mp. On the other hand, a feasible solution to the SDCTOP uses $m-1$ vehicles to visit two customers together, delivering $\frac{Q}{2} + 1$ to a customer and $\frac{Q}{2} - 1$ to another customer. The last vehicle is used to deliver the 2 missing units to the $m-1$ partially served customers. The total collected profit is equal to $2(m-1)p$. Therefore, $\frac{z(CTOP)}{z(SDCTOP)}$ tends to $\frac{1}{2}$ when m tends to infinity. □

Remark 1 *The above result and proof hold also if the time limit is removed in the problem definition.*

5 Mathematical programming formulation

In order to define the mathematical programming formulation for the problem let us consider the following additional notation. Define $V^+(i) = \{j \in V | (i,j) \in A\}$ and $V^-(i) = \{j \in V | (j,i) \in A\}$ as the set of successors and predecessors of $i \in V$, respectively. Moreover, let $\overline{d}_i = min\{d_i, Q\}$ denote the maximum quantity that can be delivered to customer i by a single vehicle.

Using this notation, the arc flow formulation of the SDCTOP is the following:

$$\max \sum_{i \in V'} p_i z_i \tag{1}$$

$$\sum_{f \in F} \delta_i^f = d_i z_i \qquad i \in V' \tag{2}$$

$$\sum_{f \in F} (x_{ij}^f + x_{ji}^f) \leq 1 \qquad i,j \in V', j > i \tag{3}$$

$$\sum_{f \in F} \sum_{j \in V^+(1)} x_{1j}^f \leq m \tag{4}$$

$$\sum_{j \in V^+(1)} x_{1j}^f = 1 \qquad f \in F \tag{5}$$

$$\sum_{j \in V^+(i)} x_{ij}^f - \sum_{j \in V^-(i)} x_{ji}^f = 0 \qquad f \in F, i \in V' \tag{6}$$

$$\sum_{j \in V^-(1)} x_{j,1}^f = 1 \qquad f \in F \tag{7}$$

$$\sum_{i \in U} \sum_{j \in U \setminus \{i\}} x_{ij}^f \leq |U| - 1 \qquad f \in F, U \subseteq V', |U| \geq 2 \tag{8}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij}^f \leq T_{max} \qquad f \in F \tag{9}$$

$$\sum_{i \in V'} \delta_i^f \leq Q \qquad f \in F \tag{10}$$

$$\sum_{j \in V^+(i)} x_{ij}^f \leq \delta_i^f \leq \overline{d}_i \sum_{j \in V^+(i)} x_{ij}^f \qquad f \in F, i \in V' \tag{11}$$

$$\delta_i^f \geq 0, \text{integer} \qquad f \in F, i \in V' \tag{12}$$

$$z_i \in \{0,1\} \qquad i \in V' \tag{13}$$

$$x_{ij}^f \in \{0,1\} \qquad f \in F, (i,j) \in A \tag{14}$$

where x_{ij}^f is a binary variable equal to 1 if vehicle $f \in F$ traverses arc $(i,j) \in A$, 0 otherwise; δ_i^f is a non-negative variable that gives the quantity delivered by vehicle $f \in F$ to customer $i \in V'$; and z_i is a binary variable equal to 1 if customer $i \in V'$ is selected to be served, 0 otherwise.

In (1) the total profit collected by the vehicles is maximized. Constraints (2) impose that the total quantity delivered to any customer i may be 0 or the total demand d_i. Constraints (3) follow from Corollary 2. Constraint (4) limits the number of routes to m. Constraints (5), (6) and (7) are the network flow constraints for each route. Constraints (8) are the subtour elimination constraints. The limitation on the time duration of each route is imposed in (9), whereas constraints (10) are the vehicle capacity constraints. Consistency between variables δ_i^f and x_{ij}^f is imposed in constraints (11). Finally, (12)–(14) are the non-negativity and integrality constraints for the problem variables.

Constraints (3) and the lower bound on the δ variables in (11) are not necessary for the correctness of the formulation and are added to strengthen it. Similarly, only one of the sets (5) and (7) is necessary. Moreover, the integrality condition in constraints (12) may be relaxed when the customers demands and the vehicle capacity are integer, as it can be proved that an optimal solution exists where the variables that define the quantities to be delivered to the customers are integer (see Archetti, Hertz and Speranza [4] for a similar proof).

6 The branch-and-price approach

The branch-and-price approach is based on the extended model obtained applying the Dantzig-Wolfe decomposition to the problem formulation (1)-(14). Actually, constraints from (5) to (14) are separable by vehicle. For each $f \in F$, this latter group of constraints defines a finite set D^f including all feasible routes the vehicle f may perform, where each route is also characterized by the quantities to be delivered to the visited customers. In particular, as observed by Jin, Liu and Eksioglu [16], Desaulniers [12] and Archetti, Bianchessi and Speranza [2], the extreme points of each set D^f, $f \in F$, are routes associated with extreme delivery patterns, that is delivery patterns where customers receive either a full delivery (that is, a quantity equal to $\overline{d}_i$), or a delivery of 1 unit and at most one customer receives a split delivery greater than 1 and lower than $\overline{d}_i$ (this definition of extreme delivery pattern was introduced by Archetti, Bianchessi and Speranza [2], whereas Desaulniers [12] proposed a slightly different definition of extreme delivery pattern). This implies that integrality requirements cannot be imposed in the extended model on the variables associated with the extreme points of the subproblem feasible regions. Optimal solutions where in a route several customers have to be served with a split delivery greater than 1 could not be found otherwise. Integrality requirements have instead to be imposed on additional variables as done by Desaulniers [12], to which the reader is referred for

details on the decomposition. The decomposition gives rise to the master problem and subproblem formulations described in the following two subsections.

6.1 The master problem

Let us consider the following additional notation:

- R: set of all feasible routes, that is circuits in graph G starting and ending at the depot and with total time not exceeding the given limit T_{max}. The empty route is denoted with 0.
- $R^s \subseteq R$: set of routes visiting a single customer.
- $R^t \subseteq R$: set of routes visiting more than one customer;
- b_{ijr}: binary parameter equal to 1 if arc $(i, j) \in A$ is traversed by route $r \in R$, 0 otherwise;
- W_r: set of all feasible extreme delivery patterns compatible with route $r \in R$, that is, patterns assigning delivery quantities to the customer visited in route $r \in R$ such that at most one customer receives a split delivery greater than 1 and lower than $\overline{d}_i$ and the total quantity delivered is at most Q;
- δ_{irw}: quantity delivered to customer $i \in V'$ in delivery pattern $w \in W_r$, $r \in R$;
- $p_{rw} = \sum_{i \in V'} p_i \frac{\delta_{irw}}{d_i}$: profit collected by delivery pattern $w \in W_r$, $r \in R$.

The master problem can be then formulated as follows:

$$\max \sum_{r\in R}\sum_{w\in W_r} p_{rw}\theta_{rw} \tag{15}$$

$$\sum_{r\in R}\sum_{w\in W_r} \delta_{irw}\theta_{rw} = d_i z_i \qquad i \in V' \tag{16}$$

$$\sum_{r\in R}\sum_{w\in W_r} (b_{ijr} + b_{jir})\theta_{rw} \leq 1 \qquad i,j \in V', j > i \tag{17}$$

$$\sum_{r\in R\setminus\{0\}}\sum_{w\in W_r} \theta_{rw} \leq m \tag{18}$$

$$\theta_{rw} \geq 0 \qquad r \in R, w \in W_r \tag{19}$$

$$\theta_r = \sum_{w\in W_r} \theta_{rw} \qquad r \in R \tag{20}$$

$$\theta_r \in \{0,1\} \qquad r \in R^t \tag{21}$$

$$\theta_r \text{ integer} \qquad r \in R^s \tag{22}$$

$$z_i \in \{0,1\} \qquad i \in V', \tag{23}$$

where θ_{rw} is a non-negative variable indicating the number of vehicles assigned to route $r \in R$ and delivery pattern $w \in W_r$, and θ_r is a non-negative integer (resp. binary) variable indicating the number of vehicles assigned to route $r \in R^s$ (resp. R^t). Note that variables θ_{rw} are associated with the extreme points of the subproblem feasible region, that is with extreme delivery patterns, whereas variables θ_r are the additional problem variables on which integrality requirements are imposed.

The objective function (15) and constraints (16)-(18) are the reformulation in terms of the θ_{rw} variables of the objective (1) and the linking constraints (2)-(4) appearing in the arc-flow formulation of the problem. Note that constraints (17) are not needed. They are inserted in order to strengthen the linear relaxation of the master problem. Constraints (19) enforce non-negativity on the θ_{rw} variables. In constraints (20), for each route $r \in R$, the corresponding additional problem variable θ_r is defined as the convex combination of the variables θ_{rw} associated with route r. Given this definition, imposing binary and integrality requirements (21) and (22) on the θ_r variables and letting the θ_{rw} variables to be continuous, allows us to guarantee that routes can contain more than one split customer (see Desaulniers [12]).

Our solution approach is based on this formulation which we call from now on Master Problem (MP). In particular, to cope with the exponential number of non-negative variables of the MP we develop a branch-and-price algorithm where, at each node of the tree, the continuous relaxation of the MP augmented by the generated branching constraints, referred to as LMP, is solved by means of column generation. This means that, to solve

the LMP, we restrict the set of variables θ_{rw} to a subset, obtaining the Restricted Linear Master Problem (RLMP). Then, a pricing problem is iteratively solved to find new variables with positive reduced cost to be added to the RLMP. When no such variable (column) exists, an optimal solution to the RLMP corresponds to an optimal solution to the LMP.

6.2 The subproblem

The subproblem, which aims at finding variables θ_{rw} with positive reduced cost with respect to the LMP, is also called the pricing problem. At the root node the LMP is obtained by removing constraints (20)–(22) and relaxing integrality constraints (23).

Let us define the following dual variables for the LMP at the root node:

- π_i: non-negative dual variable of constraint (16) for customer $i \in V'$;
- ρ_{ij}: dual variable of constraint (17) for arcs (i, j) and (j, i), with $i, j \in V'$. In order to simplify the formulation of the subproblem, we impose $\rho_{ji} = \rho_{ij}$, $j > i$;
- β: dual variable of constraint (18);

The subproblem formulation makes use of the variables x and δ as defined in the arc-flow formulation of the SDCTOP without the index of the vehicle f.

The subproblem can be formulated as follows:

$$\tilde{c}^* = \max \sum_{i \in V'} (\frac{p_i}{d_i} - \pi_i)\delta_i + \sum_{(i,j) \in A} \bar{c}_{ij} x_{ij} \tag{24}$$

$$\sum_{j \in V^+(1)} x_{1j} = 1 \tag{25}$$

$$\sum_{j \in V^+(i)} x_{ij} - \sum_{j \in V^-(i)} x_{ji} = 0 \qquad i \in V' \tag{26}$$

$$\sum_{j \in V^-(1)} x_{j,1} = 1 \tag{27}$$

$$\sum_{i \in U} \sum_{j \in U} x_{ij} \leq |U| - 1 \qquad U \subseteq V', |U| \geq 2 \tag{28}$$

$$\sum_{i \in V'} \delta_i \leq Q \tag{29}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T_{max} \tag{30}$$

$$\sum_{j \in V^+(i)} x_{ij} \leq \delta_i \leq \overline{d}_i \sum_{j \in V^+(i)} x_{ij} \qquad i \in V' \tag{31}$$

$$\delta_i \geq 0, \text{integer} \qquad i \in V' \tag{32}$$

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

where $\bar{c}_{ij}$ is the reduced cost of arc $(i,j) \in A$. It is $\bar{c}_{1j} = 0$, $\bar{c}_{i,j} = -\rho_{ij}$ and $\bar{c}_{i1} = -\beta$, $i, j \in V'$.

6.3 The branch-and-price algorithm

The solution approach we propose is inspired by the branch-and-price-and-cut presented by Archetti, Bianchessi and Speranza [2] to address the SDVRP. The algorithm for the solution of the pricing problem is described in Section 6.3.1. A mixed integer linear programming (MILP) based primal heuristic that takes advantage of the columns generated in the dual bounding procedure is discussed in Section 6.3.2. The set of branching rules described in [2] has to be completed with an additonal branching rule since here not all the customers have to be served. This branching rule applies to fractional values of variables z_i and is introduced at the second level of the hierarchical scheme presented in [2]. For all the remaining details not covered here, we refer to [2].

6.3.1 Solving the subproblem

Instead of solving directly formulation (24)-(33), the subproblem is modeled and solved as an *Elementary Shortest Path Problem with Resource Constraints* (ESPPRC) defined over an expanded graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where the vertex set $\mathcal{V}$ and the arc set $\mathcal{A}$ are defined as follows. The vertex set includes two different vertices, s and t, associated with the depot, and a set of $\overline{d}_i$ vertices for each customer $i \in V'$, each of them representing the customer served with a feasible quantity. Let us denote by v_h^i the vertex that represents customer $i \in V'$ with delivered quantity h, $h = 1, \ldots, \overline{d}_i$. Each arc $(1, j) \in A$ is replaced in $\mathcal{A}$ by two sets of arcs: arcs (s, v_h^j), $h = 1, \ldots, \overline{d}_j$, of cost $c'_{sv_h^j} = (\frac{p_j}{d_j} - \pi_j)h + \bar{c}_{1j}$, associated with load and time consumption equal to h and $t_{s,v_h^j} = t_{1j}$, respectively (see Figure 1 where $\bar{p}_j = \frac{p_j}{d_j} - \pi_j$), and arcs $(v_h^j, v_{\overline{d}_j}^j)$, $h = 1, \ldots, \overline{d}_j - 1$, associated with null cost and null resource consumption (dashed arcs in Figure 1). Arcs $(i, j) \in A$ are similarly replaced as shown in Figure 2. Finally, arcs $(i, 1) \in A$ are replaced by arcs $(v_{\overline{d}_i}^i, t) \in \mathcal{A}$ of value $c'_{v_{\overline{d}_i}^i t} = \bar{c}_{i1}$ and associated with null load consumption and time consumption $t_{v_{\overline{d}_i}^i, t} = t_{i1}$ (see Figure 3).

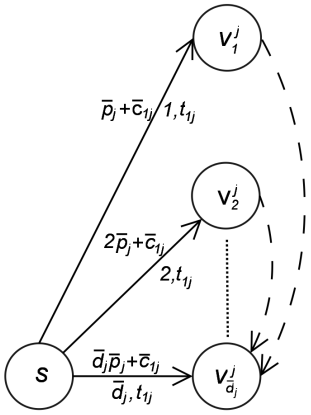

Figure 1: Expansion of arc $(1, j) \in A$.

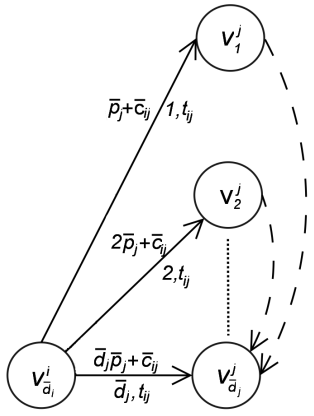

Figure 2: Expansion of arc $(i, j) \in A$.

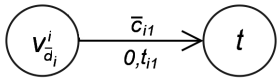

Figure 3: Expansion of arc $(i, 1) \in A$.

As finding elementary paths is very cumbersome, we allow paths that contain cycles. This does not affect the correctness of the overall approach and generates a remarkable

decrease of the computational time. Thus, we solve a *Shortest Path Problem with Resource Constraints* (SPPRC) over graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$.

The problem is solved by means of a label setting dynamic programming algorithm where each partial path ending in node $i \in \mathcal{V}$ is associated with a *state* represented by a *label* $(\lambda, q, \tau, \varsigma, C, i)$, where C is the value (reduced cost) of the path and i is the last vertex visited in the path, while the first four terms represent resources. In particular

- λ is the number of customers visited along the path,
- q is the quantity loaded on the vehicle when leaving vertex i,
- τ is the duration of the path when leaving vertex i,
- ς is the number of split customers visited along the path.

The initial state at vertex s is represented by the label $(0, 0, 0, 0, 0, s)$. At a given vertex i, a state $(\lambda, q, \tau, \varsigma, C, i)$ is feasible if $\lambda \leq |V'|$, $q \leq Q$, $\tau \leq T_{max}$ and $\varsigma \leq 1$. When arc $(w, z) \in \mathcal{A}$ is traversed, a label $(\lambda, q, \tau, \varsigma, C, w)$ is extended to the label $(\lambda', q', \tau', \varsigma', C', z)$ as follows.
Along arcs (s, v^j_h) and $(v^i_{\overline{d}_i}, v^j_h)$ the extension rules for the resource consumptions are

$$\begin{aligned}
\lambda' &= \lambda + 1 \\
q' &= q + h \\
\tau' &= \tau + t_{w,v^j_h} \text{ (with } w = s \text{ or } w = v^i_{\overline{d}_i}, \text{respectively)} \\
\varsigma' &= \begin{cases} \varsigma + 1 & \text{if } 1 < h < \overline{d}_j \\ \varsigma & \text{otherwise,} \end{cases}
\end{aligned}$$

with C' equal to $C + c'_{s,v^j_h}$ and to $C + c'_{v^i_{\overline{d}_i},v^j_h}$, respectively.
Along arcs $(v^i_h, v^i_{\overline{d}_i})$, $h = 1, \ldots, \overline{d}_i - 1$, the resource consumptions do not vary and their values are simply reported in the new label, while along arcs $(v^i_{\overline{d}_i}, t)$ the only resource which varies is $\tau' = \tau + t_{v^i_{\overline{d}_i} t}$. The value C remains unchanged if arcs $(v^i_h, v^i_{\overline{d}_i})$ are traversed, with $h = 1, \ldots, \overline{d}_i - 1$, whereas when arcs $(v^i_{\overline{d}_i}, t)$ are considered $C' = C + c'_{v^i_{\overline{d}_i},t}$.

The efficiency of the dynamic programming algorithm depends on the ability to discard paths that are dominated by other paths. Let $L' \equiv (\lambda', q', \tau', \varsigma', C', i)$ and $L'' \equiv (\lambda'', q'', \tau'', \varsigma'', C'', i)$ be two different labels representing states associated with the same vertex i. Then, L' dominates L'' if $\lambda' \leq \lambda''$, $q' \leq q''$, $\tau' \leq \tau''$, $\varsigma' \leq \varsigma''$ and $C' \geq C''$.

The label setting algorithm performs Q iterations and, at each iteration $q \in \{0, \ldots, Q-1\}$, all partial paths associated with labels characterized by a cumulated load equal to q

are extended. The *2-cycle elimination* technique presented by Desrochers, Desrosiers and Solomon [13] and the *bounded bidirectional search* technique proposed by Righini and Salani [19] are incorporated in the algorithm in order to improve its efficiency. In particular, concerning the latter technique, the amount q that can be loaded on a vehicle is considered as *critical resource*.

In order to accelerate the solution of LMP, at each column generation iteration, heuristic methods are applied before solving the pricing problem to optimality. Each heuristic consists in applying the implemented dynamic programming algorithm to a subgraph of $\mathcal{G}$ (see Archetti, Bianchessi and Speranza [2]).

6.3.2 Restricted master heuristic

The solution of the MP restricted to any subset of the columns provides a heuristic solution. The restricted set of columns can include either columns generated heuristically or columns generated during the solution of the RLMPs, or a combination of both. The resulting heuristic belongs to the class of the so called *restricted master heuristics* [17].

The restricted master heuristic we designed plays the role of a primal bound heuristic within the branch-and-price algorithm. The heuristic takes advantage of the information provided by the dual bounding procedure, that is of the columns provided by the column generation algorithm. Let Ω be a subset of the columns considered while solving a given LMP. Columns in Ω represent routes associated with quantities to deliver to the visited customers. Then, let R^{Ω} represent the set of the routes associated with columns in Ω. A pre-processing is first applied in order to eventually make routes in R^{Ω} cycle-free. To this aim, each route is scanned and all, with the exception of the first, visits to the same customer are removed. Then, on the basis of the routes included in R^{Ω}, the following MILP problem is defined and solved

$$\max \sum_{i \in V'} p_i z_i \tag{34}$$

$$\text{s.t.:} \sum_{r \in R^{\Omega}} \alpha_i^r = d_i z_i \qquad \forall i \in V' \tag{35}$$

$$\sum_{i \in V'} \alpha_i^r \leq Q y^r \qquad \forall r \in R^{\Omega} \tag{36}$$

$$\sum_{r \in R^{\Omega}} y^r \leq m \tag{37}$$

$$z_i \in \{0,1\} \qquad \forall i \in V' \tag{38}$$

$$y^r \in \{0,1\} \qquad \forall r \in R^{\Omega} \tag{39}$$

$$\alpha_i^r \geq 0 \text{ and integer} \qquad \forall i \in V', \forall r \in R^{\Omega}, \tag{40}$$

where α_i^r is a variable representing the quantity delivered to customer i in route r, y^r is a binary variable equal to 1 if route r is selected and z_i is a binary variable equal to 1 if customer i is served.

The objective (34) aims at the maximization of the collected profit. Inequalities (35) ensure that exactly the demand will be delivered to each customer selected to be served. Constraints (36) state the vehicle capacity constraints and, finally, we have the constraint on the maximum number of vehicles (37).

The set of columns Ω should include a large number of columns to allow the heuristic to find high quality solutions, but, at the same time, the solution of (34)-(40) should be obtained in a reasonable time. Starting from the columns used to solve a given LMP, Ω is initialized with one of the columns that have null reduced cost in the LMP. Then, at most $\bar{n}_O$ subsets of columns are included in Ω. Each subset is defined as follows. The columns not in Ω are considered in non-increasing order of reduced cost. A column is inserted in the subset if it has at least one vertex that is not covered by the columns previously inserted in the subset. The process stops when all vertices are covered by the columns in the subset or there are no more columns available. The maximum number of subsets that can be included in Ω is modified after each solution of (34)-(40). If the problem is infeasible or the solution found is close to the corresponding dual bound, $\bar{n}_O$ is increased. Otherwise, $\bar{n}_O$ is decreased. Increasing and decreasing values are allowed provided that $\bar{n}_O \in [\bar{n}_{min}, \bar{n}_{max}]$.

At the root node the restricted master heuristic is run before each exact solution of the pricing problem, whereas at non-root nodes it is run after the LMP solution. The solution found has no impact on the other components of the branch-and-price algorithm. The solution value, however, helps in pruning the nodes of the tree. From this point of view, the time limit for the heuristic has to be set according to the quality of the solutions found in the allotted time. Details concerning the use of the restricted master heuristic are given in Section 8.

7 A hybrid heuristic for the SDCTOP

In this section we describe a heuristic algorithm for the SDCTOP. Let s be a solution to the SDCTOP. Thus, s is a collection $\mathcal{R}$ of m routes. Let $\Lambda \subseteq V$ be the set of vertices visited by $\mathcal{R}$ and $P(\Lambda) = \sum_{i \in \Lambda} p_i$ be the total profit of the customers in Λ. Moreover, for each route $r \in \mathcal{R}$, we define as α_i^r the quantity delivered to customer i by route r, $T(r)$ as the duration of route r and $D(r) = \sum_{i \in \Lambda} \alpha_i^r$ as the total quantity delivered by r. A solution s is feasible if, for each route $r \in \mathcal{R}$, $T(r) \leq T_{max}$ and $D(r) \leq Q$. When a route r is infeasible, we define $I(r) = max\{D(r) - Q, 0\}^2 + \varrho * max\{T(r) - T_{max}, 0\}^2$, where $\varrho = \frac{Q}{T_{max}}$. Hence, $I(r) = 0$ if and only if r is feasible. The infeasibility of a solution s is defined as $\sum_{r \in \mathcal{R}} I(r)$.

The algorithm is composed by the following four procedures:

1. *Initial solution*;
2. *Tabu search*(s);
3. *Improve*(s);
4. *Optimize*(s).

The general scheme of the approach is presented in Algorithm 1 where s_{best} is the best solution found so far. In the following subsections we describe each procedure.

Algorithm 1 A hybrid heuristic for the SDCTOP

Apply the *Initial solution* procedure to generate an initial solution s. Set $s_{best} \leftarrow s$.
while no stopping criterion is met **do**
  Apply the *Tabu search*(s) procedure to find a new solution s'.
  if $I(s') > 0$ **then**
    Apply the *Improve*(s') procedure to possibly remove the infeasibility in s'.
  end if
  if s' is better than s_{best} **then**
    Apply the *Optimize*(s') procedure to possibly improve s' and set $s_{best} \leftarrow s'$.
  end if
  Set $s \leftarrow s'$.
end while

7.1 Initial solution

Since any feasible solution to the CTOP is a feasible solution also to the SDCTOP, we decided to use a feasible solution of CTOP as the initial solution for the SDCTOP heuristic algorithm. In particular, we use the solution given by the Variable Neighborhood Search (VNS) algorithm described in Archetti et al. [3], run for ten minutes.

7.2 Tabu search (s)

The tabu search we have implemented uses two kinds of moves:

- *Insertion*: Consider a customer $i \in V' \setminus \Lambda$. Insert i in $\mathcal{R}$ in the following way:
  1. Order the routes in $\mathcal{R}$ on the basis of a non-decreasing value of $I(r)$. Let $r_1, ..., r_m$ be the list of the ordered routes in $\mathcal{R}$.

2. Starting from $j = 1$, insert i in route r_j using the cheapest insertion method. The quantity inserted is $\alpha_i^{r_j} = max\{0, min\{Q - D(r_j), d_i - \sum_{t=1}^{j-1} \alpha_i^{r_t}\}\}$, unless $j = m$ or $D(r_j) \geq Q$. When $j = m$ we set $\alpha_i^{r_1} = d_i - \sum_{t=1}^{m} \alpha_i^{r_t}$, while when $D(r_j) \geq Q$ we set $\alpha_i^{r_j} = d_i - \sum_{t=1}^{j-1} \alpha_i^{r_t}$.

- *Removal*: Let i be a customer in Λ. For each route $r \in \mathcal{R}$, remove i from r by joining its predecessor with its successor.

A temporary tabu status forbids customers to be inserted in $\mathcal{R}$ once they have been removed and viceversa for a number of iterations equal to:

$$L + random(\kappa\sqrt{n}), \tag{41}$$

where L and κ are constants and $random(x)$ is an integer uniformly selected in the set $\{0, \cdots, x\}$.

The solution s' resulting from an insertion move applied to s possibly contains infeasible routes. Thus, the value of a solution is evaluated through the following modified objective function:

$$f'(s) = P(\Lambda) - \gamma \sum_{r \in \mathcal{R}} max\{T(r) - T_{max}, 0\}^2 - \varphi \sum_{r \in \mathcal{R}} max\{D(r) - Q, 0\}^2, \tag{42}$$

where γ and φ are two parameters that penalize the duration infeasibility and the capacity infeasibility, respectively. They are both set to 1 at the beginning of the algorithm and are then updated as follows: if the last 10 iterations were feasible with respect to duration (capacity), then γ (φ) is halved. If instead the last 10 iterations were infeasible with respect to duration (capacity), then γ (φ) is doubled. In the definition of $f'(s)$, both the duration and capacity infeasibility are defined as the square of the exceeding time and exceeding load, respectively, since we prefer to have a solution containing different routes with a small infeasibility (and thus easy to correct) rather than a solution having few routes with a large infeasibility.

7.3 Improve (s)

In the *Improve* (s) procedure, we first apply the 2-opt algorithm proposed by Lin [18] on each route $r \in \mathcal{R}$ of solution s. Then, we apply the following procedure SMART IMPROVE ($\mathcal{R}$) which optimizes a function F over the set of routes $\mathcal{R}$. Function F evaluates the following two terms: the total infeasibility $I(\mathcal{R}) = \sum_{r \in \mathcal{R}} I(r)$ and the total duration $T(\mathcal{R}) = \sum_{r \in \mathcal{R}} T(r)$. $F(\mathcal{R}') < F(\mathcal{R})$ if $I(\mathcal{R}') < I(\mathcal{R})$, whatever are the values of $T(\mathcal{R}')$ and $T(\mathcal{R})$. Moreover, $F(\mathcal{R}') < F(\mathcal{R})$ if $I(\mathcal{R}') = I(\mathcal{R})$ and $T(\mathcal{R}') < T(\mathcal{R})$.

SMART IMPROVE ($\mathcal{R}$)

Let us define the following moves:

- *1-move*: Consider a customer $i \in \Lambda$, a route $r \in \mathcal{R}$ visiting i and a route $r' \in \mathcal{R}$. Remove i from r by joining its predecessor with its successor and insert i in r' with the cheapest insertion method. Set $\alpha_i^{r'} = \alpha_i^{r'} + \alpha_i^r$. Note that r' can be a route already visiting i.

- *swap-move*: Let i and i' be two customers in Λ on two different routes, r and r', respectively. Remove i from r and i' from r' by joining the predecessors with the successors. Then, insert i in r' and i' in r with the cheapest insertion method. Set $\alpha_i^{r'} = \alpha_i^{r'} + \alpha_i^r$ and $\alpha_{i'}^r = \alpha_{i'}^r + \alpha_{i'}^{r'}$.

The SMART IMPROVE ($\mathcal{R}$) procedure is a local search that follows the general scheme of Figure 4.

Procedure SMART IMPROVE ($\mathcal{R}$)
Input: A set $\mathcal{R}$ of routes.
Output: A set $\mathcal{R}'$ of routes with $P(\mathcal{R}') = P(\mathcal{R})$, $\mid \mathcal{R}' \mid \leq \mid \mathcal{R} \mid$, and $F(\mathcal{R}') \leq F(\mathcal{R})$
1. Determine $\mathcal{R}'$ applying the 1-move which minimizes function F on $\mathcal{R}$.
2. If $F(\mathcal{R}') < F(\mathcal{R})$ then go to 1, else set $\mathcal{R}' \leftarrow \mathcal{R}$ and go to 3.
3. Determine $\mathcal{R}'$ applying the swap-move which minimizes function F on $\mathcal{R}$.
4. If $F(\mathcal{R}') < F(\mathcal{R})$ then go to 1, else STOP.

Figure 4. Procedure SMART IMPROVE ($\mathcal{R}$)

7.4 Optimize (s)

Every time a new best feasible solution s_{best} is found by the tabu search, we first apply the *Improve*(s_{best}) procedure described in the previous subsection obtaining a new solution s' with routes $\mathcal{R}'$. Then, a MILP model, called SMART OPT, which tries to improve s', is solved to optimality. To describe SMART OPT, we need to introduce the following additional notation:

- $\sigma_i^r = 1$ if i is visited by route r in s', 0 otherwise.

- Δ_i^r = estimated time increase for inserting i in route r according to the cheapest insertion method.

- Γ_i^r = estimated time decrease caused by removing i from route r and joining its predecessor with its successor.

SMART OPT makes use of the variables z_i and α_i^r defined in Section 6.3.2 plus the following additional variables:

- $w_i^r = 1$ if i is inserted in r, 0 otherwise.
- $v_i^r = 1$ if i is removed from r, 0 otherwise.

Then, SMART OPT is the following:

$$\max \sum_{i \in V'} p_i z_i \tag{43}$$

$$\sum_{r \in \mathcal{R}'} \alpha_i^r = d_i z_i \qquad i \in V' \tag{44}$$

$$\sum_{i \in V'} \alpha_i^r \leq Q \qquad r \in \mathcal{R}' \tag{45}$$

$$T(r) + \sum_{i \in V'} (\Delta_i^r w_i^r - \Gamma_i^r v_i^r) \leq T_{max} \qquad r \in \mathcal{R}' \tag{46}$$

$$w_i^r \leq 1 - \sigma_i^r \qquad i \in V' \quad r \in \mathcal{R}' \tag{47}$$

$$v_i^r \leq \sigma_i^r \qquad i \in V' \quad r \in \mathcal{R}' \tag{48}$$

$$\alpha_i^r \leq \bar{d}_i(\sigma_i^r - v_i^r + w_i^r) \qquad i \in V', r \in \mathcal{R}' \tag{49}$$

$$\sum_{i \in V'} v_i^r \leq 1 \qquad r \in \mathcal{R}' \tag{50}$$

$$w_i^r \in \{0,1\} \qquad i \in V', r \in \mathcal{R}' \tag{51}$$

$$v_i^r \in \{0,1\} \qquad i \in V', r \in \mathcal{R}' \tag{52}$$

$$z_i \in \{0,1\} \qquad i \in V' \tag{53}$$

$$\alpha_i^r \geq 0 \qquad i \in V', r \in \mathcal{R}' \tag{54}$$

The objective function (43) aims at maximizing the profit collected. Constraints (44) and (45) are the demand and capacity constraints, respectively. Constraint (46) is the duration constraint. Constraints (47)-(49) are coherence constraints. Constraint (50) is imposed in order to avoid big estimation errors when evaluating the new cost of a route. Finally, (51)-(54) are variable definitions.

After applying SMART OPT, we obtain a new solution $\tilde{s}$. Three different situations can occur:

1. $\tilde{s}$ is feasible and is better than s'; in this case we apply the *Improve* $(\tilde{s})$ procedure. We set s' equal to the new solution we obtained and we apply again SMART OPT.

2. $\tilde{s}$ is feasible but it is not better than s'; in this case, let Λ' be the set of customers visited in s'. Choose the customer $i \in \Lambda'$ with the lowest profit p_i and remove it from s'. Then, apply again SMART OPT.

3. $\tilde{s}$ is infeasible. This can happen because of a bad estimation of the new route duration made in constraint (46). Let $\tilde{\Lambda}$ be the set of customers for which $w_i^r = 1$ for some $r \in \mathcal{R}'$. Note that $\tilde{\Lambda} \neq \emptyset$ since s' is always feasible and, thus, $\tilde{s}$ is infeasible only if at least one customer is inserted. We choose the customer i in $\tilde{\Lambda}$ with the lowest profit p_i and we forbid the insertion of i in s' (i.e., we set $w_i^r = 0$ for each $r \in \mathcal{R}'$). Then, we apply again SMART OPT.

Thus, SMART OPT is applied iteratively and we stop the procedure when one of the following conditions is met:

- all customers in Λ' have been removed;
- no improvement has been reached in the last 20 iterations;
- a maximum time of 5 minutes has elapsed.

8 Computational results

The exact and the heuristic solution approaches have been implemented in C++, using CPLEX 10.1.1 to solve the RLMPs and the problems (34)-(40), as well as problems (43)-(54) in the hybrid heuristic. Experiments have been carried out on a 2.4 GHz Intel Dual Core Pentium IV machine with 3 GB of RAM for the hybrid heuristic presented in Section 7, while an Intel Xeon processor E5520, 2.26 GHz machine with 12 GB of RAM has been used to test the exact algorithm described in Section 6. Both approaches have been tested on known and new sets of benchmark instances.

We considered first the three sets of instances proposed in Archetti et al. [3] for the CTOP.

Set 1. This set is based on 10 VRP instances proposed by Christofides, Mingozzi and Toth [11] which consider both capacity and time constraints. The number n of vertices ranges from 51 to 200. The profit p_i of customer i has been defined as $(0.5 + b)d_i$, where b is a random number uniformly generated in the interval $[0, 1]$. An optimal solution to the CTOP has been obtained for all the instances of this set. For each instance, the optimal solution serves all the customers and collects all the possible profit. Thus, this set of instances is not interesting for the SDCTOP as no advantage can be obtained by allowing split deliveries.

Set 2. For each instance of Set 1, 9 instances are derived considering 3 different values for Q and T_{max} ($Q = T_{max} = 50, 75, 100$, respectively) and, for each of the former values, 3 different fleet size values: $m = 2, 3, 4$. We obtain 90 instances in total.

Set 3. Three instances are derived for each instance of Set 1, for $m = 2, 3, 4$. We obtain 30 instances in total.

We have tested the instances of Sets 2 and 3. As in these instances the demands of the customers are small with respect to the vehicle capacity, the split deliveries do not have a relevant impact. Therefore, we also propose a new set of benchmark instances (**Set 4**) generated on the basis of the Set 1 instances by changing the customer demands only. The technique adopted to define the new instances is the one used in Belenguer, Martinez and Mota [7] and Dror and Trudeau [14] to derive benchmark instances for the SDVRP. For each original instance, we generate 11 new instances where the customer demand is generated according to 11 scenarios ($[0.01-0.1]$, $[0.1-0.3]$, $[0.1-0.5]$, $[0.1-0.7]$, $[0.1-0.9]$, $[0.3-0.5]$, $[0.3-0.7]$, $[0.3-0.9]$, $[0.5-0.7]$, $[0.5-0.9]$, $[0.7-0.9]$). The demand of a customer in scenario $[\eta - \nu]$ is randomly generated from a uniform distribution on the interval $[\eta Q, \nu Q]$.

The overall execution time limit for the exact approach has been set to 3 hours in case of Set 2 instances and to 6 hours in all the other cases. We now report the setting of the parameters of the branch-and-price algorithm we made according to preliminary tests. These parameters are introduced and described in Archetti, Bianchessi and Speranza [2]. We did not introduce all of them in the description of the branch-and-price algorithm for the sake of clarity. We simply give here a short explanation of the meaning of each parameter. The reader is referred to Archetti, Bianchessi and Speranza [2] for more details. The parameter $\bar{n}_a$, considered while defining the subgraphs in the heuristic column generation phase, has been set to $\frac{n}{16}$. The parameters $\bar{n}_S$ and $\bar{n}_m$, that is the maximum number of column subsets and the maximum number of columns to insert in the RLMP after each solution of the pricing problem, have been set to 30 and to $\frac{1}{3}$ of the number of the RLMP rows, respectively. The time limit for each individual problem (34)-(40) solution has been set to 1800 seconds if $n \leq 101$, and to $1800 + \lceil \frac{n-101}{50} \rceil \cdot 450$ seconds otherwise. In particular, after 600 seconds without improvements in the primal bound value, the solution process stops. The parameter $\bar{n}_O$, that is the maximum number of column subsets to be included in Ω, is initially set to 75. The minimum and the maximum values for $\bar{n}_O$, that is $\bar{n}_{min}$ and $\bar{n}_{max}$, have been set to 15 and 150, respectively. The $\bar{n}_O$ variation (increase or decrease) has been set equal to 5. Finally, a feasible solution for (34)-(40) is considered close to the corresponding dual bound if the gap is less than 5%.

As far as the hybrid heuristic is concerned, according to preliminary tests, parameter L is set to 5 and parameter κ to 2. The stopping criterion is 10 minutes of running time.

A first set of tests have been performed in order to assess the efficiency of the restricted master heuristic embedded in the branch-and-price algorithm, on one side, and of proce-

dure *Optimize(s)* for the hybrid heuristic. The tests have been performed on the instances with 51 vertices of Set 4 for the branch-and-price algorithm and on all the instances for the hybrid algorithm. The overall execution time limit for the branch-and-price algorithm has been set to 6 hours and the results are reported in Table 1. The first 5 columns of the table describe the instance. Then, columns 6-8 and 10-12 give the best upper bound ($\overline{z}$), the best heuristic solution value ($\underline{z}$) and the percentage optimality gap (gap(%)) obtained by means of the branch-and-price algorithm when the time limit for the restricted master heuristic is set to 900 seconds and to 1800 seconds, respectively. Next to the optimality gap (columns 9 and 13), the number of seconds required to find the optimal solution is reported in all cases optimality has been proved. The results show that, in most of the cases where the optimal solution is found, the branch-and-price algorithm is faster when the time limit is set to 1800 seconds rather than 900 seconds. Without making use of the restricted master heuristic, the branch-and-price algorithm is able to find the optimum for the first instance listed in Table 1 only. For the remaining instances no feasible solution is found and the dual bound does not improve. This underlines the relevance of embedding the heuristic in the branch-and-price algorithm.

Table 2 concerns the performance of the procedure *Optimize(s)* for the hybrid heuristic. The table reports, for each set of instances, the average and maximum percentage gaps between the solutions obtained running or not the *Optimize(s)* procedure. Then, the number of times the *Optimize(s)* procedure allows us to improve the final solution is reported (# best). In particular, in the last column, the total number of instances in each set is reported in parentheses. As the results show, the *Optimize(s)* procedure is on average effective in improving the solution. On the other hand, the solution obtained with the *Optimize(s)* may be worse than the solution without. This is the case, for example, in two instances of Set 4.

Table 3 shows the results on the instances of Set 2. The column CTOP gives the optimal solution value to the CTOP, which is obtained by using the code of the algorithm presented in Archetti, Bianchessi and Speranza [1] run for 3 hours. Note that for the instances of sets 3 and 4 the algorithm has been run for 6 hours, given the difficulty of the instances. The columns 'Branch-and-price' give the best upper bound ($\overline{z}$), the best heuristic solution value ($\underline{z}$) and the percentage gap (gap(%)). Moreover, the time in parentheses gives the number of seconds required to find the optimal solution in all the cases optimality has been proved, i.e. whenever the gap is 0. For the hybrid heuristic the solution value is reported ($\underline{z}_H$). Then, the percentage gap with respect to the upper bound obtained by the branch-and-price ($\frac{\overline{z}-\underline{z}_H}{\overline{z}}$) and the percentage gap with respect to the restricted master heuristic ($\frac{\underline{z}-\underline{z}_H}{\underline{z}}$) are shown. A positive value in the latter column indicates that the restricted master heuristic has obtained a better solution than the hybrid heuristic. Finally, in the last column (Imp.(%)) the percentage improvement of the best solution obtained for the SDCTOP with respect to the optimal solution to the CTOP

is given. 56% of the instances of Set 2, 50 out of 90, have been solved to optimality. The optimality gaps obtained through the primal and dual bounds of the branch-and-price algorithm are, with the exception of 4 cases, less than 1%. On these instances, the branch-and-price heuristic is more effective than the hybrid heuristic. The best solution found for the SDCTOP improves or is equal to the optimal solution to the CTOP on all but one instance. This is due to the fact that the CTOP is solved to optimality while a heuristic solution was found for the SDCTOP.

In Table 4 the results for the instances of Set 3 are presented. The structure of the table is similar to the structure of Table 3 with the difference that in the CTOP column we report the upper bound $\overline{z}$. The symbol '*' after $\overline{z}$ means that the upper bound is proved to correspond to an optimal solution (the same for Tables 5 and 6). The branch-and-price is able to solve 4 of the 30 instances. However, on 25 instances the upper bound corresponds to an optimal solution, as proved by the solution provided by the hybrid heuristic. On all instances where the upper bound is produced, the hybrid heuristic finds an optimal solution with only one exceptions, with a maximum gap of 0.18% with respect to the upper bound. The hybrid heuristic performs much better than the restricted master heuristic on this class of instances with a maximum improvement of 81.91%. No improvement has been found with respect to the CTOP solution. This is due to the fact that, in this class of instances, demands are very small with respect to Q.

Tables 5 and 6 present the results for the instances of Set 4 with $n \leq 101$ and $n > 101$, respectively. With respect to the structure of Table 3, a column t provides the time needed to solve the linear relaxation at the root node. Moreover, here two columns are presented in the last section (Imp.(%)) of the table. The first of these two columns ($\underline{z}$) gives the improvement achieved by the restricted master heuristic with respect to the best known solution to the CTOP, while the second column ($\underline{z}_H$) gives the improvement achieved by the hybrid heuristic. On these instances, the hybrid heuristic performs better than the branch-and-price heuristic. Moreover, a large increase in the profit can be observed on instances where the demands are over 50% of the capacity. We do not report the results of the branch-and-price algorithm on instances of Table 6 since these instances turned out to be very hard to solve and the branch-and-price algorithm was not able to produce any solution or upper bound in the maximum time allowed. In Table 7 we summarize the improvements with respect to the CTOP solution reported in Tables 5 and 6, by classes of demands. As can be observed, the main advantages of split deliveries are obtained on instances with high customers demands, especially when demands are just above half of the vehicle capacity. In the last two columns of the same table we report the percentage capacity used in the SDCTOP solution and the percentage time limit used. The results show that, apart for the first class of instances, split deliveries allow the efficient use of the capacity of the vehicles while the time limit constraint is not binding. This obviously depends on the instances we tested.

For all previous instances, we made further tests on the hybrid heuristic by increasing

the maximum time allowed to 20 minutes. All results remained identical to the case with a maximum time of 10 minutes with the exception of instance p09 with $\eta = 0.70$ and $\nu = 0.90$ (**) where there is an improvement of the solution value of 3.55%.

Finally, Table 8 reports performance indicators about the branch-and-price algorithm for instances of Set 4 with $n \leq 101$. Column 6 gives the number of nodes explored. With the exception of the instances with 51 vertices, very few nodes are explored by the algorithm, rarely above 10 for instances with 101 vertices. Columns 7 - 11 show how the total computational time (Tot.) has been used by the different components of the algorithm, i.e. solving the continuous relaxation of the master (LMPs), heuristically solving the pricing problem (PPs - heur.), solving to optimality the pricing problem (PPs - opt.), running the restricted master heuristic (RMH). The value in column (Tot.) is below 100% because of side procedures such us management of the columns pool and of the branch-and-bound tree. The table clearly shows that the most time consuming components of the algorithm are the restricted master heuristic and the exact algorithm for the pricing problem.

Conclusions

The possibility to serve customers with multiple vehicles, even if the demand is smaller than the vehicle capacity, has been explored for the traditional Vehicle Routing Problem. In this paper we investigated this possibility for the most studied routing problem with profits, the Capacitated Team Orienteering Problem. We have theoretically shown that split deliveries may double the profit collected and we have computationally shown that the profit increase due to split deliveries varies on the basis of the instance. A large increase has been observed on instances where the demands are slightly greater than half the vehicle capacity. A branch-and-price algorithm could solve to optimality several instances with up to 200 customers. A branch-and-price and a hybrid heuristic produce high quality solutions.

Future research will be devoted to investigating the advantages of split deliveries on other routing problems with profits.

Acknowledgments

We acknowledge the contribution of two reviewers that have helped us to improve a previous version of this paper.

Table 1: Branch-and-price preliminary results on Set 4, $n \leq 51$

Instance					Time limit							
					900 sec.				1800 sec.			
name	n	m	Q	T_{max}	$\overline{z}$	$\underline{z}$	gap(%)		$\overline{z}$	$\underline{z}$	gap(%)	
p06_1_10	51	10	160	200	761	761	0.00	(11")	761	761	0.00	(14")
p06_10_30					757	755	0.26		757	750	0.92	
p06_10_50					687	687	0.00	(18758")	687	687	0.00	(11148")
p06_10_70					581	581	0.00	(16178")	581	581	0.00	(7269")
p06_10_90					495	495	0.00	(15309")	495	495	0.00	(11550")
p06_30_50					538	538	0.00	(14134")	538	538	0.00	(4759")
p06_30_70					490	490	0.00	(212")	490	490	0.00	(299")
p06_30_90					433	432	0.23		433	432	0.23	
p06_50_70					433	392	9.47		433	392	9.47	
p06_50_90					400	392	2.00		400	392	2.00	
p06_70_90					339	335	1.18		339	335	1.18	

Table 2: Hybrid heuristic preliminary results

	av. gap (%)	max. gap (%)	# best
Set 2	0.04	0.24	9(30)
Set 3	0.63	3.98	54(90)
Set 4	0.21	5.88	60(100)

Table 3: Set 2 instances

Instance					CTOP[b]	SDCTOP							
						Branch-and-price				Hybrid heuristic			Imp.
name	n	m	Q	T_{max}	z^*	$\overline{z}$	$\underline{z}$	gap(%)		$\underline{z}_H$	$\frac{\overline{z}-\underline{z}_H}{\overline{z}}$	$\frac{\underline{z}-\underline{z}_H}{\underline{z}}$	(%)
p03	101	2	50	50	133	138	138	0.00	(61")	138	0.00	0.00	3.76
p03	101	3	50	50	198	203	203	0.00	(174")	203	0.00	0.00	2.53
p03	101	4	50	50	260	269	269	0.00	(170")	269	0.00	0.00	3.46
p03	101	2	75	75	208	210	208	0.95		208	0.95	0.00	0.00
p03	101	3	75	75	307	311	311	0.00	(2388")	311	0.00	0.00	1.30
p03	101	4	75	75	403	404	404	0.00	(379")	404	0.00	0.00	0.25
p03	101	2	100	100	277	278	277	0.36		275	1.08	0.72	0.00
p03	101	3	100	100	408	410	410	0.00	(2062")	410	0.00	0.00	0.49
p03	101	4	100	100	532	534	534	0.00	(6493")	532	0.37	0.37	0.38
p06	51	2	50	50	121	124	124	0.00	(2")	124	0.00	0.00	2.48
p06	51	3	50	50	177	179	179	0.00	(16")	179	0.00	0.00	1.13
p06	51	4	50	50	222	230	230	0.00	(33")	230	0.00	0.00	3.60
p06	51	2	75	75	183	186	186	0.00	(384")	186	0.00	0.00	1.64
p06	51	3	75	75	269	271	270	0.37		270	0.37	0.00	0.37
p06	51	4	75	75	349	354	353	0.28		353	0.28	0.00	1.15
p06	51	2	100	100	252	254	253	0.39		252	0.79	0.40	0.40
p06	51	3	100	100	369	375	372	0.80		371	1.07	0.27	0.81
p06	51	4	100	100	482	487	483	0.82		479	1.64	0.83	0.21
p07	76	2	50	50	126	127	127	0.00	(278")	127	0.00	0.00	0.79
p07	76	3	50	50	187	190	190	0.00	(280")	190	0.00	0.00	1.60
p07	76	4	50	50	240	246	246	0.00	(5405")	246	0.00	0.00	2.50
p07	76	2	75	75	193	198	198	0.00	(3566")	198	0.00	0.00	2.59
p07	76	3	75	75	287	295	295	0.00	(2482")	294	0.34	0.34	2.79
p07	76	4	75	75	378	388	386	0.52		386	0.52	0.00	2.12
p07	76	2	100	100	266	271	269	0.74		270	0.37	-0.37	1.50
p07	76	3	100	100	397	400	398	0.50		398	0.50	0.00	0.25
p07	76	4	100	100	521	524	524	0.00	(338")	521	0.57	0.57	0.58
p08	101	2	50	50	133	138	138	0.00	(59")	138	0.00	0.00	3.76
p08	101	3	50	50	198	203	203	0.00	(179")	203	0.00	0.00	2.53
p08	101	4	50	50	260	269	269	0.00	(168")	269	0.00	0.00	3.46
p08	101	2	75	75	208	210	208	0.95		208	0.95	0.00	0.00
p08	101	3	75	75	307	311	311	0.00	(2613")	311	0.00	0.00	1.30
p08	101	4	75	75	403	404	404	0.00	(358")	404	0.00	0.00	0.25
p08	101	2	100	100	277	278	277	0.36		275	1.08	0.72	0.00
p08	101	3	100	100	408	410	410	0.00	(1993")	410	0.00	0.00	0.49
p08	101	4	100	100	532	534	534	0.00	(6338")	532	0.37	0.37	0.38

Table 3: Set 2 instances

Instance					CTOP[b]	SDCTOP							
						Branch-and-price				Hybrid heuristic			Imp.
name	n	m	Q	T_{max}	z^*	$\overline{z}$	$\underline{z}$	gap(%)		$\underline{z}_H$	$\frac{\overline{z}-z_H}{\overline{z}}$	$\frac{\underline{z}-z_H}{\underline{z}}$	(%)
p09	151	2	50	50	137	137	137	0.00	(3958")	137	0.00	0.00	0.00
p09	151	3	50	50	201	204	203	0.49		202	0.98	0.49	1.00
p09	151	4	50	50	262	270	268	0.74		268	0.74	0.00	2.29
p09	151	2	75	75	210	211	210	0.47		209	0.95	0.48	0.00
p09	151	3	75	75	312	313	312	0.32		312	0.32	0.00	0.00
p09	151	4	75	75	408	411	409	0.49		408	0.73	0.24	0.25
p09	151	2	100	100	279	282	280	0.71		280	0.71	0.00	0.36
p09	151	3	100	100	415	418	418	0.00	(3295")	418	0.00	0.00	0.72
p09	151	4	100	100	546	547	547	0.00	(4552")	547	0.00	0.00	0.18
p10	200	2	50	50	134	138	138	0.00	(279")	138	0.00	0.00	2.99
p10	200	3	50	50	200	204	204	0.00	(3515")	204	0.00	0.00	2.00
p10	200	4	50	50	265	269	268	0.37		268	0.37	0.00	1.13
p10	200	2	75	75	208	210	210	0.00	(1434")	210	0.00	0.00	0.96
p10	200	3	75	75	311	314	313	0.32		311	0.64	0.01	0.64
p10	200	4	75	75	411	416	414	0.48		410	1.44	0.97	0.73
p10	200	2	100	100	282	284	282	0.70		282	0.71	0.00	0.00
p10	200	3	100	100	418	420	418	0.48		419	0.24	-0.24	0.24
p10	200	4	100	100	553	555	554	0.18		553	0.36	0.18	0.18
p13	121	2	50	50	134	134	134	0.00	(21")	134	0.00	0.00	0.00
p13	121	3	50	50	193	193	193	0.00	(323")	192	0.52	0.52	0.00
p13	121	4	50	50	243	243	243	0.00	(263")	243	0.00	0.00	0.00
p13	121	2	75	75	193	193	193	0.00	(294")	193	0.00	0.00	0.00
p13	121	3	75	75	265	265	265	0.00	(5337")	265	0.00	0.00	0.00
p13	121	4	75	75	$\underline{323}$	323	323	0.00	(1989")	323	0.00	0.00	0.00
p13	121	2	100	100	$\overline{263}$	254	254	0.00	(6818")	249	1.97	1.97	-3.42
p13	121	3	100	100	-	344	338	1.74		329	4.36	2.66	-
p13	121	4	100	100	-	419	397	5.25		395	5.73	0.50	-
p14	101	2	50	50	124	124	124	0.00	(157")	124	0.00	0.00	0.00
p14	101	3	50	50	184	184	184	0.00	(161")	184	0.00	0.00	0.00
p14	101	4	50	50	241	241	241	0.00	(46")	241	0.00	0.00	0.00
p14	101	2	75	75	190	203	200	1.48		199	1.97	0.50	5.26
p14	101	3	75	75	279	299	291	2.68		291	2.68	0.00	4.30
p14	101	4	75	75	366	388	386	0.52		382	1.55	1.04	5.46
p14	101	2	100	100	271	272	271	0.37		271	0.37	0.00	0.00
p14	101	3	100	100	399	402	399	0.75		397	1.24	0.50	0.00
p14	101	4	100	100	525	526	525	0.19		514	2.28	2.10	0.00

Table 3: Set 2 instances

Instance					CTOP[b]	SDCTOP							
						Branch-and-price				Hybrid heuristic			Imp.
name	n	m	Q	T_{max}	z^*	$\overline{z}$	$\underline{z}$	gap(%)		$\underline{z}_H$	$\frac{\overline{z}-z_H}{\overline{z}}$	$\frac{\underline{z}-z_H}{\underline{z}}$	(%)
p15	151	2	50	50	134	138	138	0.00	(1821")	138	0.00	0.00	2.99
p15	151	3	50	50	200	209	209	0.00	(1616")	209	0.00	0.00	4.50
p15	151	4	50	50	266	276	276	0.00	(1667")	276	0.00	0.00	3.76
p15	151	2	75	75	211	213	212	0.47		212	0.47	0.00	0.47
p15	151	3	75	75	315	318	316	0.63		316	0.63	0.00	0.32
p15	151	4	75	75	415	418	417	0.24		417	0.24	0.00	0.48
p15	151	2	100	100	282	284	284	0.00	(1074")	284	0.00	0.00	0.71
p15	151	3	100	100	418	418	418	0.00	(1943")	416	0.48	0.48	0.00
p15	151	4	100	100	549	550	549	0.18		549	0.18	0.00	0.00
p16	200	2	50	50	137	141	141	0.00	(915")	141	0.00	0.00	2.92
p16	200	3	50	50	203	210	209	0.48		209	0.48	0.00	2.96
p16	200	4	50	50	269	279	279	0.00	(2973")	279	0.00	0.00	3.72
p16	200	2	75	75	212	215	214	0.47		214	0.47	0.00	0.94
p16	200	3	75	75	317	320	319	0.31		318	0.63	0.31	0.63
p16	200	4	75	75	420	423	422	0.24		422	0.24	0.00	0.48
p16	200	2	100	100	285	285	285	0.00	(6108")	285	0.00	0.00	0.00
p16	200	3	100	100	423	424	424	0.00	(1201")	423	0.24	0.24	0.24
p16	200	4	100	100	558	558	558	0.00	(1164")	555	0.54	0.54	0.00

b: Archetti, Bianchessi and Speranza [1].

Table 4: Set 3 instances

Instance					CTOP[b]	SDCTOP						
						Branch-and-price				Hybrid heuristic		
name	n	m	Q	T_{max}	$\overline{z}$	$\overline{z}$	$\underline{z}$	gap(%)		$\underline{z}_H$	$\frac{\overline{z}-\underline{z}_H}{\overline{z}}$	$\frac{\underline{z}-\underline{z}_H}{\underline{z}}$
p03	101	2	200	200	536*	536	536	0.00	(19446")	536	0.00	0.00
p03		3			762*	762	721	5.38		762	0.00	-5.59
p03		4			950	950	859	9.58		950	0.00	-10.59
p06	51	2	160	200	403*	403	401	0.50		403	0.00	-0.50
p06		3			565*	565	546	3.36		565	0.00	-3.48
p06		4			683*	683	631	7.61		683	0.00	-8.24
p07	76	2	140	160	377*	377	377	0.00	(2493")	377	0.00	0.00
p07		3			548*	548	548	0.00	(909")	548	0.00	0.00
p07		4			707*	707	706	0.14		707	0.00	-0.14
p08	101	2	200	230	536*	536	536	0.00	(20453")	536	0.00	0.00
p08		3			762*	762	714	6.30		762	0.00	-6.30
p08		4			950	950	768	19,16		950	0.00	-23.70
p09	151	2	200	200	548	548	535	2.37		548	0.00	-1.50
p09		3			797*	797	765	4.02		797	0.00	-4.18
p09		4			1033	1033	991	4.07		1033	0.00	-4.24
p10	200	2	200	200	556*	556	550	1.08		556	0.00	-1.09
p10		3			816	816	750	8.09		816	0.00	-8.80
p10		4			1064	-	-	-		1064	-	-
p13	121	2	200	720	513	513	282	45.03		513	0.00	-81.91
p13		3			749	-	-	-		727	-	-
p13		4			1125	-	-	-		908	-	-
p14	101	2	200	1040	534*	534	502	5.99		534	0.00	-6.37
p14		3			770*	770	715	7.14		770	0.00	-7.69
p14		4			975*	975	857	12.10		975	0.00	-13.76
p15	151	2	200	200	550	551	550	0.18		550	0.18	0.00
p15		3			802*	802	763	4.86		802	0.00	-4.86
p15		4			1031	1031	938	9.02		1031	0.00	-9.91
p16	200	2	200	200	558	558	542	2.87		558	0.00	-2.95
p16		3			822	822	747	9.12		822	0.00	-10.04
p16		4			1076	-	-	-		1073	-	-

b: Archetti, Bianchessi and Speranza [1].

Table 5: Set 4 instances; $n \leq 101$

Instance					CTOP[b]	SDCTOP									
						Branch-and-price					Hybrid heuristic			Imp (%)	
name	n	m	Q	T_{max}	$\overline{z}$	t	$\overline{z}$	$\underline{z}$	gap(%)		$\underline{z}_H$	$\frac{\overline{z}-\underline{z}_H}{\overline{z}}$	$\frac{\underline{z}-\underline{z}_H}{\underline{z}}$	$\underline{z}$	$\underline{z}_H$
p03_1_10	101	15	200	200	1409*	348	1409	1409	0.00	(351")	1409	0.00	0.00	0.00	0.00
p03_10_30					1305*	1423	1305	1283	1.69		1305	0.00	-1.71	-1.69	0.00
p03_10_50					1117*	1528	1117	1110	0.63		1117	0.00	-0.63	-0.63	0.00
p03_10_70					961*	1492	961	961	0.00	(1492")	961	0.00	0.00	0.00	0.00
p03_10_90					1005*	1881	1005	1005	0.00	(7774")	1004	0.10	0.10	0.00	-0.10
p03_30_50					892*	1862	928	911	1.83		927	0.11	-1.76	2.13	3.92
p03_30_70					807*	2068	811	808	0.37		810	0.12	-0.25	0.12	0.37
p03_30_90					704*	2374	755	753	0.26		755	0.00	-0.27	6.96	7.24
p03_50_70					549*	2653	741	687	7.29		739	0.27	-7.57	25.14	34.61
p03_50_90					517*	2548	643	618	3.89		643	0.00	-4.05	19.54	24.37
p03_70_90					517*	2655	592	585	1.18		585	1.18	0.00	13.15	13.15
p06_1_10	51	10	160	200	761*	14	761	761	0.00	(14")	761	0.00	0.00	0.00	0.00
p06_10_30					757*	949	757	750	0.92		757	0.00	-0.93	-0.92	0.00
p06_10_50					687*	996	687	687	0.00	(11148")	687	0.00	0.00	0.00	0.00
p06_10_70					581*	519	581	581	0.00	(7269")	581	0.00	0.00	0.00	0.00
p06_10_90					493*	520	495	495	0.00	(11550")	495	0.00	0.00	0.41	0.41
p06_30_50					504*	693	538	538	0.00	(4759")	538	0.00	0.00	6.75	6.75
p06_30_70					477*	299	490	490	0.00	(299")	490	0.00	0.00	2.73	2.73
p06_30_90					409*	289	433	432	0.23		432	0.23	0.00	5.62	5.62
p06_50_70					289*	1100	433	392	9.47		428	1.15	-9.18	35.64	48.10
p06_50_90					308*	1095	400	392	2.00		396	1.00	-1.02	27.27	28.57
p06_70_90					289*	280	339	335	1.18		335	1.18	0.00	15.92	15.92
p07_1_10	76	20	140	160	1327*	120	1327	1327	0.00	(121")	1327	0.00	0.00	0.00	0.00
p07_10_30					1327*	66	1327	1327	0.00	(66")	1327	0.00	0.00	0.00	0.00
p07_10_50					1292*	1013	1292	1278	1.08		1292	0.00	-1.10	-1.08	0.00
p07_10_70					1180*	1060	1180	1179	0.08		1180	0.00	-0.08	-0.08	0.00
p07_10_90					1075*	1035	1077	1064	1.21		1076	0.09	-1.13	-1.02	0.09
p07_30_50					1076*	1004	1142	1117	2.19		1142	0.00	-2.24	3.81	6.13
p07_30_70					966*	1120	980	962	1.84		980	0.00	-1.87	-0.41	1.45
p07_30_90					852*	1141	894	894	0.00	(4372")	894	0.00	0.00	4.93	4.93
p07_50_70					631*	1161	884	814	7.92		884	0.00	-8.60	29.00	40.10
p07_50_90					627*	1200	813	789	2.95		811	0.25	-2.79	25.84	29.35
p07_70_90					619*	1301	728	723	0.69		723	0.69	0.00	16.80	16.80
p08_1_10	101	15	200	230	1409*	1085	1409	1409	0.00	(1088")	1409	0.00	0.00	0.00	0.00
p08_10_30					1326*	1248	1326	1302	1.81		1326	0.00	-1.84	-1.81	0.00
p08_10_50					1158*	1497	1159	1152	0.60		1158	0.09	-0.52	-0.52	0.00
p08_10_70					1045*	1573	1045	1045	0.00	(12930")	1045	0.00	0.00	0.00	0.00
p08_10_90					909*	1920	910	888	2.42		910	0.00	-2.48	-2.31	0.11
p08_30_50					893*	2200	937	921	1.71		936	0.11	-1.63	3.14	4.82
p08_30_70					805*	3922	839	823	1.91		838	0.12	-1.82	2.24	4.10
p08_30_90					750*	2810	777	770	0.90		777	0.00	-0.91	2.67	3.60
p08_50_70					517*	3442	732	669	8.61		725	0.96	-8.37	29.40	40.23
p08_50_90					517*	3147	680	657	3.38		678	0.29	-3.20	27.08	31.14
p08_70_90					517*	3111	594	585	1.52		585	1.52	0.00	13.15	13.15
p14_1_10	101	10	200	1040	1710*	2167	1710	1710	0.00	(2172")	1710	0.00	0.00	0.00	0.00
p14_10_30					1319*	1456	1319	1277	3.18		1319	0.00	-3.29	-3.18	0.00
p14_10_50					1040*	2453	1041	1040	0.10		1040	0.10	0.00	0.00	0.00
p14_10_70					930*	2076	931	930	0.11		930	0.11	0.00	0.00	0.00
p14_10_90					822*	2964	824	808	1.94		823	0.12	-1.86	-1.70	0.12
p14_30_50					835*	4653	863	840	2.67		862	0.12	-2.62	0.60	3.23
p14_30_70					732*	4213	755	745	1.32		754	0.13	-1.21	1.78	3.01
p14_30_90					611*	8058	650	647	0.46		647	0.46	0.00	5.89	5.89
p14_50_70					418*	-	-	562	-		619	-	-10.14	34.45	48.09
p14_50_90					414*	10491	564	527	6.56		561	0.53	-6.45	27.29	35.51
p14_70_90					407*	-	-	470	-		498	-	-5.96	15.48	22.36

b: Archetti, Bianchessi and Speranza [1].

Table 6: Set 4 instances; $n > 101$

Instance					CTOP[b]	SDCTOP	
						Hybrid heuristic	Imp (%)
name	n	m	Q	T_{max}	$\overline{z}$	z_H	
p09_1_10	151	10	200	200	2194*	2194	0.00
p09_10_30					1417	1417	0.00
p09_10_50					1136*	1136	0.00
p09_10_70					920*	920	0.00
p09_10_90					973*	973	0.00
p09_30_50					766*	814	6.27
p09_30_70					702*	750	6.84
p09_30_90					653*	667	2.14
p09_50_70					357*	582	63.03
p09_50_90					379*	539	42.22
p09_70_90					357*	423**	18.49
p10_1_10	200	20	200	200	3048	3048	0.00
p10_10_30					2376	2376	0.00
p10_10_50					1975*	1975	0.00
p10_10_70					1616*	1616	0.00
p10_10_90					1578*	1578	0.00
p10_30_50					1378*	1450	5.30
p10_30_70					1260*	1278	1.43
p10_30_90					1219*	1239	1.64
p10_50_70					782*	1091	39.51
p10_50_90					773*	999	29.24
p10_70_90					738*	886	20.05
p13_1_10	121	15	200	720	1287*	1287	0.00
p13_10_30					1076*	1076	0.00
p13_10_50					884	884	0.11
p13_10_70					761*	761	0.00
p13_10_90					721*	722	0.14
p13_30_50					649*	679	4.62
p13_30_70					597*	617	3.35
p13_30_90					552*	572	3.62
p13_50_70					342*	503	47.08
p13_50_90					345*	438	26.96
p13_70_90					337*	421	24.93
p15_1_10	151	15	200	200	2159*	2159	0.00
p15_10_30					1695*	1695	0.12
p15_10_50					1341*	1341	0.07
p15_10_70					1264*	1264	0.00
p15_10_90					1064*	1065	0.09
p15_30_50					990*	1046	5.66
p15_30_70					907*	934	2.98
p15_30_90					773*	837	8.28
p15_50_70					552*	806	46.01
p15_50_90					552*	773	40.04
p15_70_90					552*	659	19.38
p16_1_10	200	15	200	200	3066	3066	0.00
p16_10_30					2386*	2386	0.00
p16_10_50					1900*	1900	0.00
p16_10_70					1731*	1731	0.00
p16_10_90					1606*	1606	0.00
p16_30_50					1449*	1518	4.76
p16_30_70					1336*	1358	1.65
p16_30_90					1109*	1177	6.13
p16_50_70					747*	1122	50.20
p16_50_90					753*	964	28.02
p16_70_90					747*	911	21.95

b: Archetti, Bianchessi and Speranza [1].

Table 7: Summary of improvements on CTOP solution for instances of Set 4

η	ν	% gap SDCTOP/CTOP	% capacity used	% time limit used
0.01	0.10	0	45.15	51.94
0.10	0.30	0.01	97.45	47.02
0.10	0.50	0.02	99.87	48.37
0.10	0.50	0	99.88	47.75
0.10	0.90	0.08	99.92	46.49
0.30	0.50	5.15	99.83	42.86
0.30	0.70	2.79	99.88	43.03
0.30	0.90	4.91	99.90	41.95
0.50	0.70	45.7	99.55	40.41
0.50	0.90	31.54	99.83	41.02
0.70	0.90	18.62	98.81	37.69

Table 8: Branch-and-price algorithm performance indicators

Instance					Nodes	Sol. time (%) for				
name	n	m	Q	T_{max}		LMPs	PPs - heur.	PPs - opt.	RMH	Tot.
p03_1_10	101	15	200	200	1	1,21	21.66	17.35	58.11	98.33
p03_10_30					11	0.07	3.24	31.63	64.93	99.87
p03_10_50					10	0.02	1.89	42.85	55.19	99.95
p03_10_70					1	0.01	2.44	53.8	43.71	99.96
p03_10_90					7	0.01	2.42	87.5	10.04	99.97
p03_30_50					6	0.01	0.68	60.4	38.42	99.51
p03_30_70					5	0.00	0.64	66.51	32.33	99.48
p03_30_90					5	0.01	0.72	70.32	28.92	99.97
p03_50_70					5	0.00	0.61	72.94	26.43	99.98
p03_50_90					5	0.00	0.65	72.87	26.45	99.97
p03_70_90					4	0.00	0.60	96.27	3.11	99.98
p06_1_10	51	10	160	200	1	0.33	23.08	45.04	30.22	98.67
p06_10_30					27	0.03	1.16	8.22	90.09	99.50
p06_10_50					103	0.04	3.61	66.38	29.87	99.90
p06_10_70					53	0.02	2.75	91.59	5.60	99.96
p06_10_90					67	0.01	1.77	71.52	26.66	99.96
p06_30_50					33	0.04	1.28	54.14	44.43	99.89
p06_30_70					1	0.01	1.37	59.45	39.14	99.97
p06_30_90					77	0.01	1.19	80.67	18.10	99.97
p06_50_70					14	0.00	0.33	20.5	79.14	99.97
p06_50_90					13	0.00	0.29	23.87	75.74	99.90
p06_70_90					58	0.02	1.15	93.84	4.95	99.96
p07_1_10	76	20	140	160	1	0.31	7.11	14.76	76.90	99.08
p07_10_30					1	0.14	12.31	85.35	1.65	99.45
p07_10_50					15	0.02	1.08	10.91	87.65	99.66
p07_10_70					28	0.01	1.53	28.42	69.98	99.94
p07_10_90					13	0.01	0.92	21.93	77.07	99.93
p07_30_50					15	0.02	0.50	13.94	84.99	99.45
p07_30_70					13	0.01	0.52	25.79	73.67	99.99
p07_30_90					5	0.01	0.47	27.16	72.33	99.97
p07_50_70					12	0.01	0.37	29.32	70.12	99.82
p07_50_90					12	0.01	0.42	31.39	68.15	99.97
p07_70_90					12	0.01	0.38	43.02	56.53	99.94
p08_1_10	101	15	200	230	1	0.77	10.62	4.99	82.96	99.34
p08_10_30					12	0.06	2.53	25.68	71.69	99.96
p08_10_50					9	0.02	1.86	42.44	55.44	99.76
p08_10_70					17	0.01	2.79	89.2	7.95	99.95
p08_10_90					7	0.01	1.49	60.25	38.17	99.92
p08_30_50					6	0.01	0.65	64.8	34.35	99.81
p08_30_70					3	0.00	0.45	82.64	16.88	99.97
p08_30_90					5	0.01	0.71	68.67	30.59	99.98
p08_50_70					4	0.00	0.48	78.97	20.54	99.99
p08_50_90					4	0.00	0.60	78.5	20.89	99.99
p08_70_90					4	0.00	0.52	82.01	17.46	99.99
p14_1_10	101	10	200	1040	1	2.05	52.22	3.51	41.58	99.36
p14_10_30					9	0.05	3.38	47.36	49.10	99.89
p14_10_50					11	0.01	2.77	87.77	9.42	99.97
p14_10_70					8	0.01	1.86	97.33	0.79	99.99
p14_10_90					4	0.00	1.54	73.44	25.01	99.99
p14_30_50					2	0.00	0.33	90.2	9.43	99.96
p14_30_70					1	0.00	0.49	91.18	8.30	99.97
p14_30_90					1	0.00	0.31	97.07	2.60	99.98
p14_50_70					0	0.00	0.09	95.76	4.11	99.96
p14_50_90					1	0.00	0.19	95.65	4.14	99.98
p14_70_90					0	0.00	0.07	97.13	2.73	99.93

References

[1] Archetti, C., Bianchessi, N., Speranza, M.G. (2012), Optimal solutions for routing problems with profits, *Discrete Applied Mathematics*, doi: 10.1016/j.dam.2011.12.021.

[2] Archetti, C., Bianchessi, N., Speranza, M.G. (2011), A column generation approach for the Split Delivery Vehicle Routing Problem, *Networks*, 58 (4), 241-254.

[3] Archetti, C., Feillet, D., Hertz, A., Speranza, M.G. (2009), The capacitated team orienteering and profitable tour problem, *Journal of the Operational Research Society* 60, 831-842.

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

[5] Archetti, C., Hertz, A., Speranza, M.G. (2007), Metaheuristics for the team orienteering problem, *Journal of Heuristics* 13, 49-76.

[6] Archetti, C., Savelsbergh, M., Speranza, M.G.(2006), Worst-case analysis for split delivery vehicle routing problems, *Transportation Science* 40, 226-234.

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

[8] Boussier, S., Feillet, D., Gendreau, M. (2007), An exact algorithm for team orienteering problems, *4OR* 5, 211-230.

[9] Butt, S.E., Cavalier, T.M. (1994), A heuristic for the multiple tour maximum collection problem, *Computers and Operations Research* 21, 101-111.

[10] Chao, I-M., Golden, B., Wasil, E.A. (1996), The team orienteering problem, *European Journal of Operational Research* 88, 464-474.

[11] Christofides, N., Mingozzi, A., Toth, P. (1979), The vehicle routing problem, in Christofides, N., Mingozzi, A., Toth, P., Sandi, C., editors, *Combinatorial Optimization*, 315-338, Wiley, Chichester.

[12] Desaulniers, G. (2010), Branch-and-price-and-cut for the split delivery vehicle routing problem with time windows, *Operations Research* 58, 179-192.

[13] Desrochers, M., Desrosiers, J., Solomon, M. (1992), A new optimization algorithm for the vehicle routing problem with time windows, *Operations Research* 40, 342-354.

[14] Dror, M., Trudeau, P. (1989), Savings by split delivery routing, *Transportation Science* 23, 141-145.

[15] Feillet, D., Dejax, P., Gendreau, M. (2005), Traveling salesman problems with profits, *Transportation Science* 39, 188-205.

[16] Jin, M., Liu, K., Eksioglu, B. (2008), A column generation approach for the split delivery vehicle routing problem, *Operations Research Letters* 36, 265-270.

[17] Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., Vanderbeck, F. (2010), Column generation based primal heuristics, *Electronic Notes in Discrete Mathematics* 36, 695-702.

[18] Lin, S. (1965), Computer solutions of the traveling salesman problem, *Bell System Technical Journal* 44, 2245-2269.

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

[20] Tang, H., Miller-Hooks, E. (2005), A tabu search heuristic for the team orienteering problem, *Computers and Operations Research* 32, 1379-1407.

[21] Tsiligirides, T. (1984), Heuristic methods applied to orienteering, *Journal of the Operational Research Society* 35, 797-809.

[22] Vansteenwegen, P., Souffriau, W., Van Oudheusden, D. (2011), The orienteering problem: A survey, *European Journal of Operational Research* 209, 1-10.