

Date of submission: 01/14/2011  
Date of first revision: 07/16/2012  
Date of second revision: 06/06/2013

# Incomplete Service and Split Deliveries in a Routing Problem with Profits

*C. Archetti*<sup>(1)</sup> *N. Bianchessi*<sup>(1)</sup> *A. Hertz*<sup>(2)</sup>  
*M.G. Speranza*<sup>(1)</sup>

<sup>(1)</sup>*Department of Quantitative Methods*

*University of Brescia, Brescia, Italy*

{archetti, bianche, speranza}@eco.unibs.it

<sup>(2)</sup>*École Polytechnique and GERAD, Montréal, Canada*

alain.hertz@gerad.ca

## Abstract

In this paper we study a variant of the Capacitated Team Orienteering Problem (CTOP), that is the problem where a fleet of vehicles, each with a constraint on the time available, is given to serve profitable customers with the objective of maximizing the collected profit. We study the variant where customers may be only partially served (incomplete service) and, if beneficial, also by more than one vehicle (split deliveries). We will analyze the maximum theoretical increase of the profit due to the incomplete service and to the split deliveries. We also computationally measure such increase on a set of instances, by means of an exact algorithm on small/medium size instances and of two heuristics on instances of larger size.

**Keywords:** Routing problems with profits, worst-case analysis, branch-and-price algorithm, tabu search, split delivery.

## 1 Introduction

The Team Orienteering Problem (TOP) is the problem where a fleet of vehicles, each with a constraint on the time available for a tour, is given to serve profitable

customers with the objective of maximizing the collected profit. The TOP belongs to the class of routing problems with profits (see the recent survey [12] and, for the case of one vehicle only, [14]) and appeared in the literature in [8] under the name Multiple Tour Maximum Collection Problem, while the definition of TOP was introduced in [9].

While in most routing problems (see [13]) all customers have to be served, in the routing problems with profits a subset of potential customers, each of which offers profit but also consumes resources or generates costs, has to be visited. The routing problems with profits have a number of applications. In particular, they model a phase of the decision process of a carrier who does not completely use the capacity of the vehicles with regular customers and aims at identifying the most profitable customers among a set of potential customers. The potential customers may be made known through the web, in specialized data bases, or by other carriers with whom a partnership agreement has been stipulated. In such applications, the capacity of the vehicles is a crucial feature of the problem, to be explicitly modeled. The Capacitated TOP (CTOP), that is the TOP with capacitated vehicles, has been introduced in [3] where exact and heuristic solution algorithms have been proposed and tested.

In the CTOP, as in most routing problems, it is assumed that a customer is visited by one vehicle only. The value of split deliveries has been investigated in several papers for several different problems. In particular, for the classical vehicle routing problem, it has been shown in [5] that allowing split deliveries, that is allowing a customer to be visited by more than one vehicle, may halve the routing cost. For a survey on routing problems with split deliveries we refer to [6]. The CTOP with split deliveries (SDCTOP) has been studied in [2], where it has been shown that allowing split deliveries may double the profit collected. Moreover, exact and heuristic algorithms have been proposed for the solution of the SDCTOP.

An underlying assumption of the TOP, of the CTOP and also of the SDCTOP is related to the service to customers. In all problems it is assumed that a customer is either served completely or not served at all. Whereas in the TOP this is always the best option because when a customer is reached all the profit can be collected, in the CTOP and in the SDCTOP the capacity constraint on the vehicles may make it beneficial to serve a customer partially, introducing a new service option. From the application point of view, a partial or incomplete service is motivated by all situations where the service of a customer may be shared between different carriers or where a customer is a geographical area for which it may be decided whether to serve it partially or completely, or not to serve it at all.

In this paper we will assess the value of allowing incomplete service in the SD-

CTOP. We will analyze the maximum theoretical increase of the profit due to the incomplete service and to the split deliveries by means of worst-case analysis. We will also computationally measure such increase on a set of instances, by means of an exact algorithm on small/medium size instances and of two heuristics on instances of larger size.

In Section 2 we will introduce the problems we study and the notation used, whereas in Section 3 we derive some properties. Section 4 is devoted to the worst-case results. In Section 5 we present an arc flow formulation and in Section 6 an exact and two heuristic approaches. Finally, in Section 7 the computational results are discussed.

## 2 The CTOP with split deliveries

We consider a complete directed graph  $G = (V, A)$ , where  $V = 1, \dots, n$  is the set of vertices and  $A$  is the set of arcs. Vertex 1 represents the depot and each vertex  $i = 2, \dots, n$  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  $(i, j) \in A$ . Travel times are assumed to satisfy the triangle inequality. A set of  $m$  vehicles is available to serve the potential customers, each with limited integer capacity  $Q$ . Let  $F$  denote the index set of 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 potential customer  $i \in V'$  may be served by more than one vehicle (split deliveries) and the profit associated with each served customer can be collected at most once. When a customer is served by more than one vehicle the profit collected by each vehicle is proportional to the demand served by the vehicle. 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.

We study the version of the SDCTOP where we allow a customer to be partially served. When a customer is partially served the profit collected is proportional to the served demand. We call this problems SDCTOP with Incomplete Service (SDCTOP-IS).

The advantage of allowing split deliveries and incomplete service are highlighted by the following example. Consider the instance depicted in Figure 1.a: the round

vertices represent customers while the square vertex corresponds to the depot. Customers are associated with demand and profit equal to 6 each,  $m = 2$ ,  $Q = 10$ ,  $t_{ij} = 1$  for each  $(i, j)$  and  $T_{max} = M \gg 0$ . The solutions of CTOP, SDCTOP and SDCTOP-IS are depicted in Figures 1.b, 1.c and 1.d, respectively.

In the following, we denote by  $z(P)$  the value of an optimal solution of problem  $P$ .

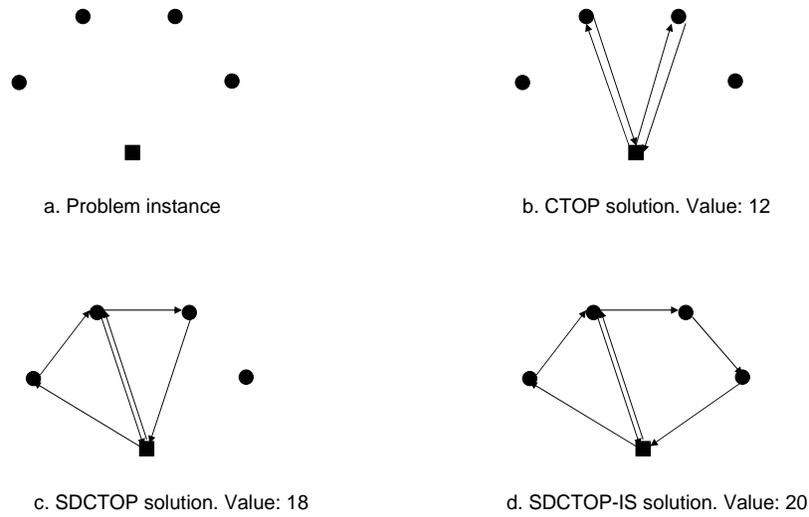


Figure 1: Advantage of split deliveries and incomplete service

### 3 Properties

In [11] an important property of the Split Delivery Vehicle Routing Problem (SD-VRP), that is the classical vehicle routing problem where split deliveries are allowed, was shown based on the following concept.

**Definition 1** Given  $k$  customers  $i_1, i_2, \dots, i_k$  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  $i_1, i_2, \dots, i_k$  is called a  $k$ -split cycle.

In [11] it was shown that an optimal solution to the SDVRP exists without  $k$ -split cycles and in [2] it was proved to be valid also for the SDCTOP. We now extend this property to the SDCTOP-IS.

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

Proof: Theorem 1 in [2] shows that the result is valid for the SDCTOP. The same proof applies to the SDCTOP-IS.  $\square$

The following property is an immediate consequence of Theorem 1.

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

**Theorem 2** *If the cost matrix satisfies the triangle inequality, an optimal solution to the SDCTOP-IS exists where each route has at most one customer with incomplete service.*

Proof: Take an optimal solution to the SDCTOP-IS. Consider one of the routes of such solution and suppose that more than one customer receives incomplete service. Take two such customers, say  $i$  and  $j$ , and assume, w.l.o.g., that  $p_i/d_i \geq p_j/d_j$ . Then, we modify the solution by moving as much as possible of the demand served of the least profitable customer  $j$  to the most profitable one  $i$ . We decrease the demand served of  $j$  to increase the demand served of  $i$ . If the demand served of  $j$  was greater than or equal to the non served demand of  $i$ , the demand of  $i$  will be fully satisfied and we reduced the number of customers with incomplete service by one. The solution obtained is not worse than the previous solution. Otherwise, if the demand served of  $j$  was smaller than the not served demand of  $i$ , the demand of  $i$  remains partially served but again we reduced the number of partially served customers by one while not worsening the value of the solution. The solution remains feasible because of the triangle inequality. We repeat this procedure until at most one customer with incomplete service remains in the route. Then we repeat the procedure on all the other routes.

□

In [5] a property of the SDVRP that creates a relation between the number of splits and the number of routes in an optimal solution was shown. Such property holds for the SDCTOP-IS.

Let  $n_i$  be the number of routes that visit customer  $i$ . We say that customer  $i$  is a customer with a split delivery if  $n_i > 1$  and that the number of splits at customer  $i$  is  $n_i - 1$ . Therefore, the total number of splits is equal to  $\sum_{i \in V'} (n_i - 1)$ .

**Theorem 3** *If the cost matrix satisfies the triangle inequality, then there exists an optimal solution to the SDCTOP-IS where the total number of splits is less than the number of routes.*

Proof: Theorem 2 in [2] shows that the result is valid for the SDCTOP. The same proof applies to the SDCTOP-IS. □

In the case of the SDCTOP-IS, in addition to customers, without or with a split delivery, that are completely served there may be customers, with or without a split delivery, that are only partially served. We say that an optimal solution has  $h$  incomplete services if the number of customers with incomplete service is  $h$ .

**Theorem 4** *If the cost matrix satisfies the triangle inequality, then there exists an optimal solution to the SDCTOP-IS where the total number of splits and incomplete services is not greater than the number of routes.*

Proof: Let us consider a solution without  $k$ -split cycles. As each route has at most one customer with incomplete service (see Theorem 2), if there are no splits the theorem is proved. Thus, let us assume that there is at least one split. Let us consider the routes with the split customer. For each possible other split customer of any of these routes we consider the other routes which share the split customer and we continue on any split customer until no new split customer is found in the involved routes. This set of routes has a number of splits that is at most equal to the number of routes in the set minus 1 by Theorem 3. We can associate to this set of routes a graph, where each route is associated with a node of the graph and two nodes of the graph are connected by an edge iff the two corresponding routes share a split customer. Note that the graph is connected and, thus, has at least a number of edges which is equal to the number of routes in the set minus 1. Consequently, the number of splits is exactly the number of routes in the set minus 1. The resulting graph is thus a tree. If this set of routes does not contain more than one incomplete service the theorem is proved. In this case we consider another tree

of routes, if any. If no tree of routes with more than one incomplete service exists, the theorem is proved. Thus, we assume that one of the trees of routes contains at least two customers with incomplete service. We now show that we can reduce by 1 the number of splits or of incomplete services without worsening the quality of the solution. Let us consider any pair of customers with incomplete service and the two corresponding routes.

Let us denote by 1 the route with the least (over the two routes) profitable customer with incomplete service (that is the customer with smallest ratio between profit and demand). Let us denote the other route as route 2. Let  $j_1$  and  $j_2$  be the customers with incomplete service on route 1 and 2, respectively. Now we construct a not worse solution with one split or one incomplete service less. We consider the path connecting the nodes associated with routes 1 and 2 in the tree and reduce by as much as possible the demand of the least profitable customer  $j_1$  served by route 1 to increase by the same amount the demand of customer  $j_2$  served by route 2. This can be achieved by increasing and decreasing by the same amount the intermediate customers in the path, keeping those customers completely served. Let  $S$  be the set of split customers along the path and  $x_i^r$  be the quantity delivered to  $i$  by route  $r$ . Consider the quantity  $q = \min\{x_{j_1}^1, d_{j_2} - x_{j_2}^2, \min_{i \in S}\{x_i^r\}\}$  for each  $i \in S$ . Then,  $x_{j_1}^1$  is decreased by  $q$ , the quantity served to the split customer in  $S$  served by route 1 is increased by  $q$  while the quantity served to the same customer by the other route visiting it is decreased by  $q$ . The procedure is repeated on all split customers along the path until, at the end,  $x_{j_2}^2$  is increased by  $q$ . By doing so, either customer  $j_1$  is not served by route 1 or customer  $j_2$  is completely served, and thus the number of incomplete services is reduced by 1, or the quantity served to a split customer along the path by one route visiting it is set to 0, and thus the number of splits is reduced by 1.

For each set of routes defining a tree, we can repeat the above procedure. Finally we obtain that the total number of splits and incomplete services is not greater than the number of routes.

□

## 4 Worst-case analysis

In this section we study how much additional profit can be collected if we relax the assumption that each customer must be either not served or served completely, and allow the incomplete service of a customer. As we will compare the CTOP and SDCTOP-IS, we assume that  $d_i \leq Q$ ,  $i \in V'$ . Moreover, we suppose that the

triangle inequality is satisfied.

**Theorem 5**

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

*and this bound is tight.*

Proof: Consider an optimal solution to the SDCTOP-IS that satisfies the claim of Theorem 2, that is a solution where in each route at most one customer is partially served. We modify this solution to obtain a feasible solution of the SDCTOP. We do not modify the possible routes where all customers are completely served. We take a route of this optimal solution with one incomplete service. We consider two possible ways to modify this route and obtain a route where all customers are fully served. The first modified route that we consider is the one where all customers with complete service only are included. In this case the profit of the partially served customer is lost. The second route that we consider is the route where only the previously partially served customer is served. Note that both satisfy the capacity and time constraints. We take the most profitable of the two. The profit of this route is greater than or equal to half of the profit of the route of the SDCTOP-IS. We repeat the procedure on all routes.

To show that the bound is tight, take an instance with two customers each with demand and profit equal to  $\frac{Q}{2} + 1$ , one vehicle with capacity  $Q$  and large time limit. The SDCTOP can serve only one customer while the SDCTOP-IS can serve  $\frac{Q}{2}$  of each of the two. When  $Q$  tends to infinity the ratio tends to  $1/2$ . □

By combining the results of Theorem 3 in [2] and Theorem 5 we would obtain  $\frac{z(CTOP)}{z(SDCTOP-IS)} \geq \frac{1}{4}$ . However, this bound is not tight whereas the following result holds.

**Theorem 6**

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

*and this bound is tight.*

Proof: Consider an optimal solution of the SDCTOP-IS which satisfies the claim of Theorem 4. For each incomplete service we remove the corresponding customer from the solution and we create an out-and-back tour to serve it. We repeat the same operation for every split customer. All routes satisfy the time limit constraint

because of the triangle inequality. Due to Theorem 4, the number of out-and-back tours created is at most equal to the number of routes in the SDCTOP-IS solution. The constructed solution is possibly infeasible for the CTOP because the number of routes may be greater than  $m$ . 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 of the SDCTOP-IS.

The tightness of the bound follows from Theorem 5.

□

## 5 Problem formulation

In the proposed arc flow formulation for the SDCTOP-IS we use the following notation. Let  $V^+(i) = \{j \in V \mid (i, j) \in A\}$  and  $V^-(i) = \{j \in V \mid (j, i) \in A\}$  be the set of successors and predecessors of  $i \in V$ , respectively. Moreover, let  $\bar{d}_i = \min\{d_i, Q\}$  denote the maximum quantity that can be delivered to customer  $i$  by a single vehicle.

Using this notation, the formulation for the SDCTOP-IS is the following:

$$\max \sum_{f \in F} \sum_{i \in V'} \frac{p_i}{d_i} \delta_i^f \quad (1)$$

$$\sum_{f \in F} \delta_i^f \leq d_i \quad i \in V' \quad (2)$$

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

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

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

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

$$\sum_{j \in V^-(1)} x_{j1}^f = 1 \quad f \in F \quad (7)$$

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

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

$$\sum_{i \in V'} \delta_i^f \leq Q \quad f \in F \quad (10)$$

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

$$\delta_i^f \geq 0, \text{ integer} \quad f \in F, i \in V' \quad (12)$$

$$x_{ij}^f \in \{0, 1\} \quad f \in F, (i, j) \in A \quad (13)$$

where  $x_{ij}^f$  is a binary variable equal to 1 if vehicle  $f \in F$  traverses arc  $(i, j) \in A$  and  $\delta_i^f$  is a non-negative variable representing the quantity delivered by vehicle  $f \in F$  to customer  $i \in V'$ .

The objective function (1) calls for the maximization of the collected profit. Constraints (2) impose that the total quantity delivered to each customer cannot exceed its demand. Constraints (3) follow from Corollary 1. Constraint (4) imposes the upper bound of  $m$  on the number of used vehicles. Flow conservation constraints

and subtour elimination constraints are stated in (5)-(7) and (8), respectively. Constraints (9) limit the time duration of each route, whereas constraints (10) are the vehicle capacity constraints. Consistency between variables  $\delta_i^f$  and  $x_{ij}^f$  is imposed in constraints (11). Finally, (12) and (13) are the non-negativity and integrality constraints on the problem variables.

Constraints (3) and the lower bound on the  $\delta$  variables in (11) are included in the model to strengthen it but are not necessary for the sake of correctness. Similarly, only one of the sets (5) and (7) is necessary. Moreover, the integrality condition in constraints (12) may be relaxed if the customer demands and the vehicle capacity are integer. Actually, in this case there exists an optimal solution to the problem where the  $\delta$  variables assume integer values (see [4] for a similar proof).

## 6 Solution approaches

We present in this section both exact and heuristic algorithms to solve the problem.

### 6.1 A branch-and-price algorithm

The exact approach is an adaptation of the branch-and-price algorithm proposed in [2] for the SDCTOP.

By applying the Dantzig-Wolfe decomposition principle to the arc flow formulation (1)-(13), the SDCTOP-IS is decomposed into a set packing master problem and a subproblem, also called pricing problem. Then, the master problem is solved by dynamically generating the variables associated with the extreme points of the feasible region of the pricing subproblem, that is, feasible routes associated with extreme delivery patterns. The feasible routes are circuits in graph  $G$  starting and ending at the depot and with total time not exceeding the given limit  $T_{max}$ . For a given route, the associated extreme delivery patterns assign delivery quantities to the customers visited in the route in such a way that at most one customer receives a split delivery greater than 1 and lower than  $\bar{d}_i$  and the total quantity delivered is at most  $Q$ . Due to the combinatorial structure of the extreme points, in the master problem integrality requirements cannot be imposed directly on the dynamically generated variables. Additional problem variables have to be considered.

Let  $R = R^s \cup R^t$  be the set of all feasible routes, where  $R^s$  and  $R^t$  are the sets of routes visiting single and multiple customers, respectively. The empty route is denoted by 0. Then, let  $W_r$  represent the set of all feasible extreme delivery patterns associated with route  $r \in R$ .

Then, the master problem can be formulated as follows:

$$\max \sum_{r \in R} \sum_{w \in W_r} \left( \delta_{irw} \frac{p_i}{d_i} \right) \theta_{rw} \quad (14)$$

$$\text{s.t.:} \sum_{r \in R} \sum_{w \in W_r} \delta_{irw} \theta_{rw} \leq d_i \quad \forall i \in N \quad (15)$$

$$\sum_{r \in R} \sum_{w \in W_r} (b_{ijr} + b_{jir}) \theta_{rw} \leq 1 \quad \forall i, j \in N, j > i \quad (16)$$

$$\sum_{r \in R \setminus \{0\}} \sum_{w \in W_r} \theta_{rw} \leq m \quad (17)$$

$$\theta_{rw} \geq 0 \quad \forall r \in R, \forall w \in W_r \quad (18)$$

$$\theta_r = \sum_{w \in W_r} \theta_{rw} \quad \forall r \in R \quad (19)$$

$$\theta_r \text{ integer} \quad \forall r \in R^s \quad (20)$$

$$\theta_r \in \{0, 1\} \quad \forall r \in R^t \quad (21)$$

where  $\theta_{rw}$  variables are the dynamically generated variables representing the number of vehicles assigned to route  $r \in R$  and delivery pattern  $w \in W_r$ , and  $\theta_r$  is a non-negative integer (resp. binary) variable indicating the number of vehicles assigned to route  $r \in R^s$  (resp.  $R^t$ ). For each  $r \in R$ ,  $\theta_r$  is defined in constraints (19) as the convex combination of the extreme delivery patterns (i.e. the  $\theta^{rw}$  variables) associated with the route  $r$ . Given this definition, imposing binary and integrality requirements (20) and (21) on the  $\theta_r$  variables and letting the  $\theta_{rw}$  variables to be continuous allows us to guarantee that routes can contain more than one split customer (see [10]). The objective function (14) and constraints (15)-(17) are the reformulation in terms of the  $\theta_{rw}$  variables of the objective (1) and the linking constraints (2)-(4). In particular,  $b_{ijr}$  is a binary parameter equal to 1 if arc  $(i, j) \in A$  is traversed by route  $r \in R$ , 0 otherwise; and parameter  $\delta_{irw}$  indicates the quantity delivered to customer  $i \in N$  in delivery pattern  $w \in W_r$ ,  $r \in R$ .

The subproblem is modeled over an expanded graph (derived from  $G$ ) as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). The expanded graph considers a set of vertices for each customer, each of them representing the customer served with a feasible integer quantity. With each arc are associated load and time consumptions. As finding elementary paths is very cumbersome, we allowed paths that contain cycles. This does not affect the correctness of the overall approach and generates a remarkable decrease of the computational time. The resulting Shortest Path Problem with Resource Constraints (SPPRC) is solved by

means of a label setting dynamic programming algorithm. In particular, an additional resource is taken into account to limit to one the number of customers in the path which receive a split delivery greater than 1 and less than  $\bar{d}_i$ . In order to accelerate the solution process of the linear master problem, 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 the expanded graph. The subgraphs restrict the dynamic programming algorithm to consider only paths with full delivery to each customer or paths making a partial delivery only to the customer reached directly from the depot. In particular, some of the subgraphs are also built taking into account the geographic closeness of the customers.

When branching is required, the four branching rules proposed in [1] are hierarchically evaluated. Branching is attempted on the number of vehicles, on the number of visits to each customer, on the flow on an arc and on the possibility to use to arcs consecutively in a path. In particular, when a single visit is imposed to a customer  $i$  no reduction can be applied to the expanded graph; actually the quantity delivered can be any quantity lower than  $d_i$  as  $i$  can be only partially served. This is different from what happens in the SDCTOP addressed in [2] where, when a single visit is imposed, the expanded graph can be reduced by considering only the vertex associated with the quantity  $d_i$ .

Finally, the branch-and-price algorithm makes use of a restricted master heuristic. It consists in the solution of a MILP problem by means of a commercial solver. The MILP problem is defined on the basis of the columns evaluated to solve the linear relaxation of the master problem associated with the current node tree. The heuristic plays the role of a primal bound heuristic within the branch-and-price algorithm. The solution found by the heuristic 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. At the root node of the tree, the restricted master heuristic is run before each exact solution of the pricing problem, whereas at non-root nodes it is run after the solution of the linear master problem.

For more details on the algorithm, we refer the reader to [2].

## 6.2 Heuristic algorithms

In order to deal with large size instances that cannot be handled by the exact approach, we propose two solution algorithms for the solution of the SDCTOP-IS: a Variable Neighborhood Search (VNS) and a Tabu Search (TS) algorithm. Both algorithms are adaptations of the Variable Neighborhood Search and the Tabu Search

Feasible algorithms proposed in [3] for the CTOP. In this paper a third algorithm was proposed called Tabu Search Admissible, where also infeasible solutions are explored. We have decided to consider feasible solutions only, the reason being that split deliveries and incomplete service allow a wider range of moves inside the feasible solution space. Thus, we inherited only the Variable Neighborhood Search and the Tabu Search Feasible algorithms.

In the following we describe the general scheme of the algorithms. For more details, we refer the reader to [3]. The main idea of both algorithms is that all customers are organized into routes which are ordered on the basis of their profit. At the end, only the  $m$  best routes will form the final solution.

Let  $q_r^i$  be the quantity delivered to customer  $i$  in route  $r$ . The profit of a route  $r$  is defined as  $P(r) = \sum_{i \in V'} \frac{p_i}{d_i} q_r^i$  while  $T(r)$  is the total duration. For a set  $R$  of routes,  $P(R) = \sum_{r \in R} P(r)$  is defined as the total profit in  $R$ . A *solution*  $s$  is defined as a set of routes such that each route starts and ends at the depot, and the demand of each customer is satisfied. We denote by  $R_{SDCTOP-IS}(s)$  the set of  $m$  most profitable routes in  $s$ , and by  $R_{NSDCTOP-IS}(s)$  the set of all remaining routes.

Both the TS and the VNS use two kinds of moves to explore the neighborhood, namely, the *1-move* and the *swap-move*. These moves have been changed with respect to their definition in [3] in order to take into account the possibility of making split deliveries. They are now defined as follows:

- *1-move*: Consider a customer  $c$  visited in route  $r$ . Insert  $c$  in route  $r' \neq r$ . Note that  $r'$  can be an empty route or, also, a route already visiting  $c$ . The quantity delivered to  $c$  in  $r'$  becomes  $q_{r'}^c = q_r^c + \min\{q_r^c, \delta_{r'}\}$ , where  $q_r^c$  is the quantity delivered to  $c$  in  $r$  and  $\delta_{r'}$  is the residual capacity of  $r'$ . If  $\min\{q_r^c, \delta_{r'}\} = q_r^c$  then  $c$  is removed from  $r$  by joining the predecessor with the successor. If  $q_{r'}^c = 0$  (and, thus,  $r'$  does not visit  $c$  in the current solution), then  $c$  is inserted in  $r'$  with the cheapest insertion method.
- *swap-move*: Let  $c$  and  $c'$  be two customers on two different routes,  $r$  and  $r'$ , respectively. A swap-move consists in inserting  $c$  in  $r'$  and  $c'$  in  $r$ . The quantity delivered to  $c$  in  $r'$  becomes  $q_{r'}^c = q_{r'}^c + \min\{q_r^c, \delta_{r'}\}$  while the quantity delivered to  $c'$  in  $r$  becomes  $q_r^{c'} = q_{r'}^{c'} + \min\{q_{r'}^{c'}, \delta_r\}$ . If  $\min\{q_r^c, \delta_{r'}\} = q_r^c$  ( $\min\{q_{r'}^{c'}, \delta_r\} = q_{r'}^{c'}$ ) then  $c$  ( $c'$ ) is removed from  $r$  ( $r'$ ) by joining the predecessor with the successor. If  $q_{r'}^c = 0$  ( $q_r^{c'} = 0$ ) then  $c$  ( $c'$ ) is inserted in  $r'$  ( $r$ ) with the cheapest insertion method.

A temporary tabu status forbids customers to be inserted into routes from which they have previously been removed.

The solution quality is measured by combining the four following functions:

- $f_1(s)$  : the total profit  $P(R_{SDCTOP-IS}(s))$  of the routes in  $R_{SDCTOP-IS}(s)$ ;
- $f_2(s)$  : the total duration  $\sum_{r \in R_{SDCTOP-IS}(s)} T(r)$  of the routes in  $R_{SDCTOP-IS}(s)$ ;
- $f_3(s)$  : the number of non empty routes in  $s$ ;
- $f_4(s)$  : the total duration  $\sum_{r \in R_{NSDCTOP-IS}(s)} T(r)$  of the routes in  $R_{NSDCTOP-IS}(s)$ .

A solution  $s$  is considered as better than a solution  $s'$  if  $f_1(s) > f_1(s')$ , or  $f_1(s) = f_1(s')$  and  $f_2(s) < f_2(s')$ , or  $f_1(s) = f_1(s')$ ,  $f_2(s) = f_2(s')$  and  $f_3(s) < f_3(s')$ , or  $f_1(s) = f_1(s')$ ,  $f_2(s) = f_2(s')$ ,  $f_3(s) = f_3(s')$  and  $f_4(s) < f_4(s')$ .

In order to explore various regions of the solution space, we use two kinds of *jumps* to escape a local optimum. One consists in performing a series of 1-moves from  $R_{NSDCTOP-IS}(s)$  to  $R_{SDCTOP-IS}(s)$ , while the other makes a series of swap-moves between  $R_{NSDCTOP-IS}(s)$  and  $R_{SDCTOP-IS}(s)$  such that the profit of the routes in  $R_{SDCTOP-IS}(s)$  increases. A solution  $s'$  resulting from a jump possibly contains infeasible routes. In this case, to remove the infeasibility in a subset  $R$  of routes, we use a repair procedure that performs a series of 1-moves which strictly reduce the infeasibility in  $R$  and do not modify the other existing routes. Notice that such 1-moves always exist since it is always possible to remove a customer from an infeasible route and to insert it into a new route.

The algorithms make use of the classical 2-opt procedure to improve the current solution in case the instance is symmetric.

The TS algorithm typically visits a limited number of different regions of the solution space, each one being deeply explored before jumping to the next one. On the contrary, the variable neighborhood search algorithm performs much more jumps, which results in an exploration of a larger number of different regions of the solution space, but at the expense of a less intensive search in each region.

## 7 Computational results

The exact and the heuristic solution algorithms were implemented in C++, using CPLEX 10.1.1. The experiments were carried out on a 2.4 GHz Intel Dual Core Pentium IV machine with 3 GB of RAM for the heuristic algorithms, while an Intel Xeon processor E5520, 2.26 GHz machine with 12 GB of RAM was used to test the

exact algorithm. Both algorithms were run on Windows 7 operating system and compiled under Visual C++ 2010 Express Edition.

The overall execution time limit for the exact algorithm was set to 6 hours, while the time for each of the heuristic algorithms was set to 10 minutes. All the parameters were set as in [2].

After some preliminary tests, we have decided to test all the approaches on Set 4 of the instances proposed in [2], as the increase in the profit due to incomplete service is larger on Set 4 than on the other sets. The number  $n$  of vertices ranges from 51 to 200. The profit  $p_i$  of customer  $i$  was defined as  $(0.5 + b)d_i$ , where  $b$  is a random number uniformly generated in the interval  $[0, 1]$ . Customer demands were generated using a technique proposed in Belenguer, Martinez and Mota [7] and Dror and Trudeau [11] to derive benchmark instances for the SDVRP. For each original instance, we generated 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 results of the experiments are summarized in Tables 1 and 2. Table 1 is devoted to the instances with a number of vertices  $n \leq 101$ , whereas Table 2 to the instances with  $n > 101$ .

The first section of Table 1 describes the instances, with the name of the instance, number of vertices  $n$ , number of vehicles  $m$ , capacity  $Q$  and time limit on each tour  $T_{max}$ .

The column SDCTOP  $\underline{z}^*$  gives the lower bound of the solution to the SDCTOP which is the best heuristic solution found in [2]. An \* means that the solution is optimal.

The further sections provide the results for the SDCTOP-IS. The columns in the section 'branch-and-price' provide the best upper bound ( $\bar{z}^*$ ), the best heuristic solution ( $\underline{z}^*$ ) 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 reached, i.e. whenever the gap is 0. A sign '-' means that an upper bound could not be found within the time limit. While only few instances could be solved to optimality, the optimality gap is very small, being in most cases negligible and in 3 cases only above 1%.

In the section 'Heuristic' we report the value of the best heuristic solution obtained by the VNS and the TS ( $\underline{z}_H^*$ ). We note that both algorithms always find the same solution (on all instances reported in Tables 1 and 2) except for instance p16\_30\_50 where the solution of the TS is 1518.463 while the solution of the VNS

is 1518.461. In the following columns, the percentage gap with respect to the upper bound obtained by the branch-and-price ( $\frac{z^* - z_H^*}{z^*}$ ) and the percentage gap with respect to the branch-and-price heuristic ( $\frac{z^* - z_H^*}{z^*}$ ) are shown. A negative value in the latter column indicates that the best solution between VNS and TS is a better solution than the branch-and-price heuristic solution. Finally, the last column (Imp(%)) gives the percentage improvements of the best solution obtained for the SDCTOP-IS by the TS and the VNS with respect to the best known solution of the SDCTOP. It is interesting to observe that, while the worst-case analysis has indicated a maximum potential improvement of 50%, from the computational point of view, allowing incomplete service does not offer a significant improvement with respect to the complete service. Although the improvement is calculated by using heuristic values, the quality of the heuristics, shown by means of comparisons with the optimum, makes such values very reliable.

The first section of Table 2 describes the instances. Then, in the SDCTOP column we report the best known solution of the SDCTOP ( $z^*$ ), which is the heuristic value obtained in [2]. In the section SDCTOP-IS we provide the value of the best solution obtained by the TS and the VNS ( $z_H^*$ ) and the percentage improvement (Imp (%)) of the solution with respect to the best solution to the SDCTOP. Also on these instances of bigger size allowing incomplete service does not offer a significant increase of the profit. Only on the instance p09\_70\_90 we observed a relevant improvement of the profit (5.17%).

The results in Tables 1 and 2 show only a slight improvement of SDCTOP-IS with respect to the solution of the SDCTOP. Thus, we decided to run a second set of tests in order to investigate possible instance configurations giving higher gainings. To this purpose, we took the Solomon benchmark instances R101, R201, C101, C201, RC101 and RC201. We kept the data concerning the coordinates of the vertices and the vehicle capacity, and changed customer demands, as previously done, with the following values of  $\eta$  and  $\nu$ :  $[0.7 - 0.75]$ ,  $[0.7 - 0.8]$ . The  $T_{max}$  value has been set to a large value so that it has no impact on the solution. The number of customers is equal to 100. The results are summarized in Table 3. Column headings have the same meaning as before. The solution of the SDCTOP was obtained by applying the heuristic algorithm proposed in [2]. The results show a much higher gaining with respect to the previous tests when  $m = 2$ , with an average improvement of 29.72% and a maximum improvement of 34.23%. When  $m = 3$ , the average improvement is 0.22% and the maximum improvement is 0.67%. This is obviously due to the way the instances are constructed. In particular, when  $m = 2$ , given that  $\eta = 0.7$ , for each vehicle a fraction of capacity equal to at least  $(1 - \nu)$  remains unused in

a SDCTOP solution, while a SDCTOP-IS solution can fully exploit the capacity of both vehicles. Note that it is possible to achieve larger improvements also for the case of  $m = 3$  by choosing appropriate values of  $\eta$  and  $\nu$ . For example, when  $\eta = 0.8$  and  $\nu = 0.9$ , at least 10% of the capacity of each vehicle remains unused in a SDCTOP solution.

Table 1:  $n \leq 101$

Instance				SDCTOP[2]		SDCTOP-IS				Imp (%)	
name	$n$	$m$	$Q$	$T_{max}$	$z^*$	$z^*$	Branch-and-price gap(%)	$z^*$	Heuristic $\frac{z^* - \hat{z}H}{z^*}$	$\frac{z^* - \hat{z}H}{z^*}$	Imp (%)
p03-1-10	101	15	200	200	1409*	1409.000	0.00	1409.000	0	0.00	0.00
p03-10-30					1305*	1305.193	1.54	1305.193	0	-1.56	0.01
p03-10-50					1117*	1117.449	0.21	1117.449	0	-0.23	0.04
p03-10-70					961*	961.310	0.03	961.310	0	0.00	0.03
p03-10-90					1005*	1006.217	0.01	1006.217	0	-0.03	0.12
p03-30-50					927	928.876	0.03	928.876	0	-0.03	0.20
p03-30-70					810	811.357	0.01	811.357	0	-0.01	0.16
p03-30-90					755*	755.673	0.01	755.673	0	-0.01	0.09
p03-50-70					739	741.633	0.04	741.633	0	-0.04	0.35
p03-50-90					643*	643.794	0.01	643.794	0	-0.01	0.12
p03-70-90					585	592.800	0.01	592.800	0	-0.01	1.33
p06-1-10	51	10	160	200	761*	761.000	0.00	761.000	0	0.00	0.00
p06-10-30					757*	758.333	0.57	758.333	0	-0.57	0.18
p06-10-50					687*	689.298	0.08	689.298	0	-0.08	0.33
p06-10-70					581*	581.060	0.02	581.060	0	-0.02	0.54
p06-10-90					495*	497.640	0.03	497.640	0	-0.03	0.53
p06-30-50					538*	539.600	0.05	539.600	0	-0.05	0.30
p06-30-70					490*	490.563	0.01	490.563	0	-0.01	0.11
p06-30-90					432	435.342	0.01	435.342	0	-0.01	0.77
p06-50-70					428	434.511	0.02	434.511	0	-0.02	1.52
p06-50-90					396	400.875	0.01	400.875	0	-0.01	1.23
p06-70-90					335	342.993	0.01	342.993	0	-0.01	2.37
p07-1-10	76	20	140	160	1327*	1327.000	0.00	1327.000	0	0.00	0.00
p07-10-30					1327*	1327.000	0.00	1327.000	0	0.00	0.00
p07-10-50					1292*	1284.100	0.62	1292.115	0	-0.62	0.01
p07-10-70					1180*	1180.514	0.09	1181.565	0	-0.09	0.13
p07-10-90					1076	1077.755	0.03	1077.755	0	-0.03	0.16
p07-30-50					1132*	1142.215	0.22	1142.215	0	-0.22	0.02
p07-30-70					980*	981.125	0.04	981.125	0	-0.04	0.11
p07-30-90					894*	895.000	0.01	895.000	0	-0.01	0.11
p07-50-70					884*	884.350	0.08	884.350	0	-0.08	0.04
p07-50-90					811	813.650	0.12	813.650	0	-0.12	0.33
p07-70-90					723	728.840	0.01	728.840	0	-0.01	0.71
p08-1-10	101	15	200	230	1409*	1409.000	0.00	1409.000	0	0.00	0.00
p08-10-30					1326*	1326.790	2.45	1326.790	0	-2.51	0.06
p08-10-50					1158	1159.440	0.84	1159.440	0	-0.85	0.12
p08-10-70					1045*	1046.294	0.22	1046.294	0	-0.22	0.12
p08-10-90					910*	910.500	0.15	910.500	0	-0.15	0.06
p08-30-50					896	937.786	0.04	937.786	0	-0.04	0.19
p08-30-70					828	829.148	0.04	829.148	0	-0.04	0.14
p08-30-90					725	777.873	0.04	777.873	0	-0.04	0.11
p08-50-70					717*	733.288	0.29	733.288	0	-0.29	1.14
p08-50-90					678	680.646	0.00	680.646	0	0.00	0.39
p08-70-90					585	594.966	0.01	594.966	0	-0.01	1.70
p14-1-10	101	10	200	1040	1710*	1710.000	0.00	1710.000	0	0.00	0.00
p14-10-30					1319*	1319.769	2.11	1319.769	0	-2.16	0.06
p14-10-50					1040	1041.379	0.11	1041.379	0	-0.11	0.13
p14-10-70					930	931.878	0.00	931.878	0	0.00	0.20
p14-10-90					823	824.943	0.01	824.943	0	-0.01	0.24
p14-30-50					752	863.921	0.01	863.921	0	-0.01	0.21
p14-30-70					752	755.768	0.01	755.768	0	-0.01	0.23
p14-30-90					647	650.303	0.01	650.303	0	-0.01	0.51
p14-50-70					619	617.523	-	625.227	-	-0.08	1.01
p14-50-90					561	564.154	-	564.796	-	-1.25	0.68
p14-70-90					498	500.065	-	500.103	-	-0.01	0.42

Table 2:  $n > 101$ 

Instance					SDCTOP[2]	SDCTOP-IS	
name	$n$	$m$	$Q$	$T_{max}$	$z^*$	Heuristic $z_H^*$	Imp (%)
p09_1_10	151	10	200	200	2194	2194.000	0.00
p09_10_30					1417	1418.491	0.11
p09_10_50					1136	1138.204	0.19
p09_10_70					920	921.526	0.17
p09_10_90					973	974.886	0.19
p09_30_50					814	816.492	0.31
p09_30_70					750	751.920	0.26
p09_30_90					667	669.500	0.37
p09_50_70					582	588.674	1.15
p09_50_90					539	542.940	0.73
p09_70_90					423	444.879	5.17
p10_1_10	200	20	200	200	3048	3048.000	0.00
p10_10_30					2376	2376.842	0.04
p10_10_50					1975	1977.051	0.10
p10_10_70					1616	1617.350	0.08
p10_10_90					1578	1579.023	0.06
p10_30_50					1450	1451.892	0.13
p10_30_70					1278	1278.358	0.03
p10_30_90					1239	1240.288	0.10
p10_50_70					1091	1094.230	0.30
p10_50_90					999	1000.250	0.13
p10_70_90					886	887.929	0.22
p13_1_10	121	15	200	720	1287	1287.000	0.00
p13_10_30					1076	1077.636	0.15
p13_10_50					884	884.807	0.09
p13_10_70					761	761.353	0.05
p13_10_90					722	722.571	0.08
p13_30_50					679	679.923	0.14
p13_30_70					617	617.864	0.14
p13_30_90					572	572.163	0.03
p13_50_70					503	505.321	0.46
p13_50_90					438	439.412	0.32
p13_70_90					421	421.105	0.03
p15_1_10	151	15	200	200	2159	2159.000	0.00
p15_10_30					1695	1696.500	0.09
p15_10_50					1341	1341.794	0.06
p15_10_70					1264	1265.603	0.13
p15_10_90					1065	1065.677	0.06
p15_30_50					1046	1047.112	0.11
p15_30_70					934	934.218	0.02
p15_30_90					837	840.565	0.43
p15_50_70					806	809.081	0.38
p15_50_90					773	777.808	0.62
p15_70_90					659	661.559	0.39
p16_1_10	200	15	200	200	3066	3066.000	0.00
p16_10_30					2386	2386.063	0.00
p16_10_50					1900	1900.871	0.05
p16_10_70					1731	1731.630	0.04
p16_10_90					1606	1607.088	0.07
p16_30_50					1518	1518.463	0.03
p16_30_70					1358	1359.733	0.13
p16_30_90					1177	1179.292	0.19
p16_50_70					1122	1123.944	0.17
p16_50_90					964	965.441	0.15
p16_70_90					911	915.262	0.47

Table 3: Test on additional instances

Instance					SDCTOP	SDCTOP-IS	
name	$m$	$Q$	$\eta$	$\nu$	$z^*$	Heuristic $z_H^*$	Imp (%)
C101	2	200	0.7	0.75	298	400	34.23
	2		0.7	0.8	318	400	25.79
	3		0.7	0.75	596	600	0.67
	3		0.7	0.8	600	600	0.00
C201	2	700	0.7	0.75	1048	1400	33.59
	2		0.7	0.8	1118	1400	25.22
	3		0.7	0.75	2095	2100	0.24
	3		0.7	0.8	2100	2100	0.00
R101	2	200	0.7	0.75	298	400	34.23
	2		0.7	0.8	318	400	25.79
	3		0.7	0.75	596	600	0.67
	3		0.7	0.8	600	600	0.00
R201	2	1000	0.7	0.75	1498	2000	33.51
	2		0.7	0.8	1594	2000	25.47
	3		0.7	0.75	2994	3000	0.20
	3		0.7	0.8	3000	3000	0.00
RC101	2	200	0.7	0.75	298	400	34.23
	2		0.7	0.8	318	400	25.79
	3		0.7	0.75	596	600	0.67
	3		0.7	0.8	600	600	0.00
RC201	2	1000	0.7	0.75	1498	2000	33.51
	2		0.7	0.8	1597	2000	25.23
	3		0.7	0.75	2993	3000	0.23
	3		0.7	0.8	3000	3000	0.00

## 8 Conclusions

In this paper we have studied the Capacitated Team Orienteering Problem with Split Deliveries and Incomplete Service (SDCTOP-IS). We have shown that allowing incomplete service may offer an improvement of the collected profit up to 50%. The computational results, obtained on a large set of instances, show that the improvement tends to be negligible on randomly generated instances while it can be substantial when the customer demands are generated in specific ranges.

## References

- [1] Archetti, C., Bianchessi, N., Speranza, M.G. (2011), A column generation approach for the Split Delivery Vehicle Routing Problem, *Networks*, 58 (4), 241-254.
- [2] Archetti, C., Bianchessi, N., Hertz, A., Speranza, M.G. (2013), The split delivery capacitated team orienteering problem, *Networks*, to appear.
- [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., Savelsbergh, M., Speranza, M.G. (2006), Worst-case analysis for split delivery vehicle routing problems, *Transportation Science* 40, 226-234.
- [6] Archetti, C., Speranza, M.G. (2008), The split delivery vehicle routing problem: A survey, in: *The Vehicle Routing Problem Latest Advances and New Challenges*, B. Golden, R. Raghavan, E. Wasil (eds.), Operations Research/Computer Science Interfaces Series 43, 103-122, Springer-Verlag.
- [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] Butt, S.E., Cavalier, T.M. (1994), A heuristic for the multiple tour maximum collection problem, *Computers and Operations Research* 21, 101-111.
- [9] Chao, I-M., Golden, B., Wasil, E.A. (1996), The team orienteering problem, *European Journal of Operational Research* 88, 464-474.
- [10] Desaulniers, G. (2010), Branch-and-price-and-cut for the split delivery vehicle routing problem with time windows, *Operations Research* 58, 179-192.
- [11] Dror, M., Trudeau, P. (1989), Savings by split delivery routing, *Transportation Science* 23, 141-145.
- [12] Feillet, D., Dejax, P., Gendreau, M. (2005), Traveling salesman problems with profits, *Transportation Science* 39, 188-205.

- [13] Toth, P., Vigo, D. (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002).
- [14] Vansteenwegen, P., Souffriau, W., Van Oudheusden, D. (2011), The orienteering problem: A survey, *European Journal of Operational Research* 209, 1-10.