Capacitated ring arborescence problems with profits

Alessandro Hill[1], Roberto Baldacci[2], and Edna Ayako Hoshino[3]

[1] School of Business
Universidad Adolfo Ibáñez
Diagonal Las Torres 2640, Santiago, Chile
alessandro.hill@uai.cl
[2] Department of Electrical, Electronic,
and Information Engineering "Guglielmo Marconi"
University of Bologna
Via Venezia 52, 47521 Cesena, Italy
r.baldacci@unibo.it
[3] Faculty of Computing
University of Mato Grosso do Sul
Campo Grande MS, Brazil
eah@facom.ufms.br

Abstract. In this work we introduce profit-oriented capacitated ring arborescence problems and present exact and heuristic algorithms. These combinatorial network design problems ask for optimized bi-level networks taking into account arc costs and node profits. Solutions combine circuits on the inner level with arborescences on the outer level of the networks. We consider the prize-collecting, the budget-constrained and the target-profit models and develop corresponding exact branch and cut algorithms based on mixed integer formulations and valid inequalities. Iterated local search heuristics based on the exploration of problem specific neighborhoods are elaborated to strengthen the upper bounds. For a set of hard literature derived instances with up to 51 nodes, we provide computational results which give evidence for the efficiency of the proposed approaches. Furthermore, we extensively analyze the performance of our methods, the obtained solution networks and the impact of the cutting planes on the obtained lower bounds.

Keywords: ring arborescence problem, vehicle routing problem, prize-collecting Steiner tree problem, network design, mathematical programming

1 Introduction

Classical vehicle routing problems ask for a set of node-disjoint routes on which given customers can be served from a depot at minimal overall travel cost. Besides applications in transportation planning such centered cycle-based structures are commonly used to model reliable telecommunication infrastructure in backbone networks or to directly provide reliable service to major end customers. In the case that simple connectivity is sufficient, more cost-efficient tree topologies are used instead. In a transportation setting, such trees can be used to represent distribution networks and its corresponding customers.

In both environments the consumers might differ in terms of the profit expected from delivering goods or service to them. Consequently, their inclusion in the network might not be lucrative due to disproportional connection costs. More detailed, the decision which customers to service is frequently guided by the achieved profits, the corresponding network costs or both.

For integrated decision making, the design of directed networks which contain both, routes and tree structures can be modeled by ring arborescence problems (RAPs). In these problems two types of customer nodes are given. Type 1 customer nodes can be situated on arborescences or circuits whereas type 2 customer nodes have to be connected to the central node via circuits. The allowed structures are circuits, arborescences and circuits which have arborescences attached to some of their nodes. They have to intersect in the depot, or central node, while fulfilling two capacity requirements: the number of customer nodes in each substructure as well as the number of these structures are limited. Furthermore, Steiner nodes can optionally be used in the network if beneficial regarding the objective function which takes into account the asymmetric cost structure. In this work, we formally introduce this RAP and develop three variants which take into account the customer profits in the objective or in a side constraint.

1.1 Literature review

The studied RAPs generalize the recent capacitated ring tree problem (CRTP). In this undirected version, structures originating in the central node can be either trees, cycles or cycles to which some trees are attached. Type 2 customers have to be nodes on cycles whereas type 1 customers can be on rings or trees. Furthermore, Steiner nodes can optionally be used in the network if beneficial regarding the overall edge costs. Exact algorithms are developed by Hill and Voß [2016] and Hoshino and Hill [2015], and heuristics are elaborated by Hill [2015] and Hill and Voß [2014]. Abe et al. [2015] considered a locational variant which generalizes the CRTP by imposing installation costs on the nodes at which tree structures intersect with rings. Moreover, the presented model and algorithms allow an additional capacity bound on the number of customers in each attached subtree. Bi-objective CRTPs are introduced together with exact ϵ-constrained methods in Hill and Schwarze [2016].

The models considered in this work are related to capacitated ring star problems (CRSPs) (Baldacci et al. [2007], Baldacci and Dell'Amico [2010]). In the CRSPs, the ring structures are only allowed to be extended by trees that consist of a single edge, leading to star-like ring extensions. The polytope of the related network design problem asking for a regular 1-arborescence, a special case of the ring tree structures in the CRTP, is studied by Balas and Fischetti [1992]. Ring star models have been proven useful in several applications, such as to model school bus routing problems (Schittekat et al. [2013]). For an overview on related models we refer to Park and Kim [2010]. A review on related tributary networks with a focus on hub location at the points where inner and outer network structures intersect is given by Klincewicz [1998].

Profit-oriented models are widely studied in the network design literature and in numerous works on transportation models. Profits, also called scores, need to be maximized in the related orienteering problem (OP) (Chao et al. [1996b], Vansteenwegen et al. [2011]). The OP asks for a path from a given start node in a network to a dedicated target node such that the collected scores are maximized and a certain time limit, w.r.t. the edge traversal times, is respected. There is also a multi-path version of the OP. The research on the so-called team orienting problem (TOP) which was introduced as the multi-path variant of the OP by Chao et al. [1996a] is summarized by Archetti et al. [2009]. The capacitated TOP (CTOP) generalizes the latter by imposing a bound on the number of individual path nodes (Archetti et al. [2013a]).

Motivated by the applications in short-term steel production scheduling, the prize-collecting travelling salesman problem (PC-TSP) was introduced by Balas [1989] and Balas and Martin [1991]. In contrast to the OP, a closing edge needs to be found in this selective TSP to establish a tour. A polyhedral analysis of the PC-TSP is conducted in Balas [1995]. A review of models with profits related to the TSP can be found in Feillet et al. [2005]. The PC-TSP can be seen as a single-tour vehicle routing problem. Models and algorithms for general vehicle routing problems with profits are surveyed by Archetti et al. [2013b] and related arc routing problems with profits are studied by Archetti et al. [2010]. In covering tour problems the set of customers visited by one (Gendreau et al. [1997]) or multiple routes (Hachicha et al. [2000]) has to fulfill additional covering constraints. The latter assure that isolated customers have at least one tour customer within a pre-specified distance range. Another ring-based network design model that takes into account edge costs and revenues is discussed together with exact methods by Gouveia and Pires [2001]. The counterpart allowing multiple rings is considered by Bautzer et al. [2015].

Several tree-based models that consider customer profits can be found in the literature. The prize-collecting generalized minimum spanning tree problem is presented by Golden et al. [2008] together with an exact algorithm and heuristics. Models that consider node profits in Steiner trees are studied by Johnson et al. [2000] and Chapovska and Punnen [2006]. One variant is the prize-collecting Steiner tree problem (PC-STP). An efficient branch and cut method is suggested by Ljubić et al. [2006]. Costa et al. [2009] elaborate exact algorithms to solve the more general PC-STP with additional budget and hop side constraints.

1.2 Contribution

In this paper, we introduce the ring arborescence problem with its asymmetric cost structure which generalizes the CRTP. We identify three profit-oriented generalizations which incorporate the well-known concepts of prize-collecting, budget-constraints and target-profits into the purely cost-oriented RAP. We use

a generic non-compact mathematical formulation to develop branch-and-cut algorithms for all three variants and use several classes of problem-specific valid inequalities to strengthen the formulation. Furthermore, we propose iterated local search heuristics that operate on various problem-specific neighborhoods to efficiently construct optimized feasible networks. The latter algorithms substantially support the exact methods and are used to tighten the bounds during the branch and bound procedures. We report first bounds achieved for a set of hard instances with asymmetric costs and challenging budget constraints and target profits, respectively. Moreover, we provide a thorough analysis of the algorithmic performance and the solution networks.
The remainder of the paper is structured as follows. After this introductory section, we provide a formal definition of the RAP and the profit-oriented variants in Section 2. In Section 3, we mathematically formulate these problems as mixed integer programs and develop corresponding valid inequalities in Section 4. Heuristics based on problem specific local search are described in Section 5. Branch and cut algorithms for the models are described in Section 6. The results of a computational study show the performance of our algorithms in Section 7.

2 The ring arborescence problem and its profit-oriented variants

In this section, we formally introduce the basic ring arborescence problem. The RAP generalizes its undirected counterpart, the CRTP [Hill and Voß, 2016]. It is defined on a digraph $G = (V', A)$, where $V' = \{d\} \cup V$ is a set of nodes and A is a set of arcs. Node d represents the *central node* (or *depot*) and node set V is partitioned into three subsets: V_S containing *Steiner nodes* (or *transit point*), and node sets V_C^1 and V_C^2, representing customers of *type 1* and *type 2*, respectively. We denote the set of all customers by V_C. Set A contains both the set of possible ring arcs and arborescence arcs. Each arc $(i, j) \in A$ is associated with an installation cost c_{ij}.

In contrast to the CRTP, the RAP considers directed representations of ring trees, called *ring arborescences* (or *pseudo-arborescences*). More formally, a ring arborescence Q is a directed graph that consists of a simple directed cycle C passing through d, or *circuit*, and disjoint directed trees $T_1, ..., T_k$, or *arborescences*, that intersect with C in their root nodes. C is called the *fundamental (directed) cycle* or the *ring* of Q. Since we allow C to be *trivial*, i.e. a cycle of order 1, ring arborescences generalize the graph class of directed trees and directed 1-trees. Note that we do not allow circuits of two arcs. A directed 1-tree can be obtained from a directed tree by adding an arc that connects a node different from the root to the root. A node of a directed 1-tree is a root if it is on the fundamental cycle and has out-degree 1. The cost of a ring arborescence is equal to the sum of its arc costs. We say that a *k-ring arborescence star* S is a network $S = (V[S], A[S])$ obtained by the union of ring arborescences $Q_1, ..., Q_k$ that only intersect in the central node d which is the root of each ring arborescence. Its cost is defined as the sum of the ring arborescence costs.

The RAP asks for a minimum cost k-ring arborescence star S on the nodes in V', centered in d, such that

(i) $V_C \subset V[S]$,
(ii) each type 2 node in $V[S]$ is part of a fundamental cycle in S,
(iii) the number of customers in each ring arborescence is at most q, and
(iv) k is at most m.

Here $q \in \mathbb{N}$ specifies a given *ring arborescence capacity* and $m \in \mathbb{N}$ a *ring arborescence limit*. In the RAP it is assumed that the *total customer capacity* is sufficient to cover all the given customers: $mq \geq |V_C|$.

2.1 The prize-collecting, budget-constrained, and target profit RAP

We now define the three profit-oriented problem variants considered in this paper. In contrast to the RAP defined above, the following problems allow a solution S to span only a subset of the customers: $|V[S] \cap V_C| \leq |V_C|$. Henceforth, the total customer capacity (mq) is allowed to be less than the number of customers. We associate a *profit* $p_i \geq 0$ to each node $i \in V'$ and assume $p_i = 0\ \forall i \in V_S \cup \{d\}$. In addition, let $\hat{c}_{ij}$ be the arc installation cost associated with arc $(i, j) \in A$.

1. The *prize-collecting ring arborescence problem* (PC-RAP) aims at finding a solution S that maximizes the collected profits after the deduction of the overall arc installation costs. The PC-RAP corresponds to the RAP without requirement (i) where the arc cost c_{ij}, $(i, j) \in A$, is defined as $\hat{c}_{ij} - p_j$.
2. The *budget-constrained ring arborescence problem* (BC-RAP) asks for a solution S that maximizes the collected profits without exceeding a given installation cost *budget* B. The BC-RAP corresponds to the RAP without requirement (i) where the arc cost c_{ij}, $(i, j) \in A$, is defined as $-p_j$, with an additional budget constraint that bounds the overall arc installation costs to B.
3. The *target profit ring arborescence problem* (TP-RAP) is to minimize the arc installation costs while meeting an overall *target profit* T. The TP-RAP corresponds to the RAP without requirement (i) where the arc cost c_{ij}, $(i, j) \in A$, is defined as $\hat{c}_{ij}$, with an additional target constraint that requires the collected profit to be at least T.

Figure 1 illustrates a 3-ring arborescence star that fulfills the requirements for a solution for the three problems above. Three ring arborescences with node sets $\{1, ..., 13\}$, $\{14, 15, 19, 20, 22, ..., 26\}$ and $\{16, 17, 18, 21\}$ are implemented, servicing 11, 9 and 3 customers, respectively. Note that in a feasible solution for the RAP all customers would have to be connected to d by the network.

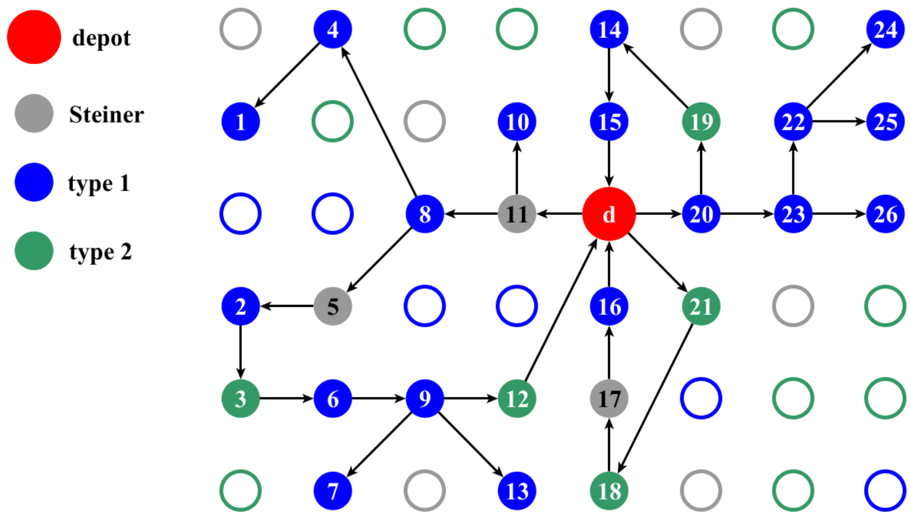

Fig. 1: A solution for the profit-oriented ring arborescence problems in which 23 of the 38 customers are serviced using 3 Steiner nodes in a 3-ring arborescence star.

3 Mathematical formulation

In this section we present a mathematical formulation for the RAP and the three problem variants considered. The formulation uses the following notation. For any $S \subset V'$, let $\delta^+(S)$ (respectively, $\delta^-(S)$) denote the set of arcs (i,j) with $i \in S$, $j \in V' \setminus S$ (respectively, with $i \in V' \setminus S$, $j \in S$). For sake of notation we use $\delta^+(i)$ ($\delta^-(i)$) instead of $\delta^+(\{i\})$ ($\delta^-(\{i\})$), $\forall i \in V'$. $\delta(S_1 : S_2)$ contains all the arcs in A with tail node in S_1 and head node in S_2 for $S_1, S_2 \subseteq V'$. In addition, we denote with $V_C(S)$, $S \subseteq V$, the set of customers in node set S, i.e. $V_C(S) = S \cap V_C$.

The formulation uses the following decision variables:

- w_i: a binary *customer service* variable which is equal to 1 if and only if customer $i \in V_C$ is visited or selected in the solution ($w_i = 1\ \forall i \in V_C$ for the RAP);
- x_a: a binary *arc* variable which is equal to 1 if and only if arc $a \in A$ is selected in the solution;
- y_a: a binary *ring arc* variable which is equal to 1 if and only if arc $a \in A$ belongs to a circuit in the solution.

We recall that the three models differ in the arc cost function and an additional constraint in case of the BC-RAP and the TP-RAP. Defining $A_2 = \bigcup_{i \in V_C^2} (\delta^-(i) \cup \delta^+(i))$, the mathematical formulation for the PC-RAP is as follows.

$$(F) \quad \min \quad \sum_{a \in A} c_a x_a \tag{1}$$

$$
\begin{aligned}
s.t. \quad & \sum_{a\in\delta^-(i)} x_a = w_i && (i \in V_C) && (2)\\
& \sum_{a\in\delta^-(i)} x_a \le 1 && (i \in V_S) && (3)\\
& \sum_{a\in\delta^+(d)} x_a \le m && && (4)\\
& \sum_{a\in\delta^-(S)} x_a \ge \frac{1}{q} \sum_{i\in V_C(S)} w_i && (S \subseteq V) && (5)\\
& \sum_{a\in\delta^-(i)} y_a - \sum_{a\in\delta^+(i)} y_a = 0 && (i \in V') && (6)\\
& y_a - x_a = 0 && (a \in A_2) && (7)\\
& y_a - x_a \le 0 && (a \in A \setminus A_2) && (8)\\
& x_{di} + x_{id} \le 1 && ((d,i) \in A) && (9)\\
& w_i \in \{0,1\} && (i \in V_C) && (10)\\
& x_a, y_a \in \{0,1\} && (a \in A) && (11)
\end{aligned}
$$

Equations (2) are the in-degree equations imposing that the in-degree of each customer node $i \in V_C$ is 1 if its belongs to a ring arborescence. Constraints (3) ensure that in-degree of each node $i \in V_S$ is at most one. The number of ring arborescences selected in the solution is limited by inequalities (4). The exponential number of *capacitated connectivity inequalities* (5) force the resulting solution to be connected while restricting each ring arborescence in solution to a cardinality of at most q. Constraints (2) together with constraints (6)-(8) impose that $y_a = 1$ if arc a is used by a circuit in the solution. Finally, constraints (9) prohibit single-node circuits in the solution.

The RAP can be formulated by imposing $w_i = 1 \ \forall i \in V_C$ in (F). Moreover, this formulation facilitates the integration of node profits that are dependent on the node's role in the network, ring node or arborescence node. More specifically, if we are given an ring profit p_i^r and a arborescence profit p_i^t for each node $i \in V$, then the following objective can be used to model this PC-RAP variant: $\sum_{(i,j)\in A}[(\hat{c}_{ij} - p_j^r)y_{ij} + (\hat{c}_{ij} - p_j^t)(x_{ij} - y_{ij})]$.

The BC-RAP can be formulated by adding the following *budget constraint* to formulation (F).

$$\sum_{a\in A} \hat{c}_a x_a \le B. \tag{12}$$

A formulation for the TP-RAP is obtained by extending (F) by the following *target profit constraint*.

$$\sum_{i\in V_C} p_i w_i \ge T. \tag{13}$$

4 Valid inequalities

In this section we describe the valid inequalities used to strengthen formulation (F). We categorize them into three different classes as follows.

4.1 Connectivity inequalities

The following *circulation inequalities* are valid and prevent from reverse ring arcs in the circuits (see Hill and Voß [2016], Ineq. 16).

$$y_{ij} \leq \sum_{a \in \delta(j:V' \setminus i)} y_a \quad ((i,j) \in A). \tag{14}$$

Considering the structure of the solution, in every Steiner node, which is not a branching node, indegree and outdegree must be equal, whereas in a branching Steiner node indegree is always less than outdegree (see Hill and Voß [2016], Ineq. 31). In particular, this forbids Steiner leaf nodes. Thus, we have:

$$\sum_{a \in \delta^-(i)} x_a \leq \sum_{a \in \delta^+(i)} x_a \quad (i \in V_S). \tag{15}$$

Removing the circuit arcs that enter node d turns the network into an arborescence. Therefore, we can impose the following *connectivity inequalities* for Steiner tree problems which enforce the arborescence structure for each node in V_C or V_S that appears in a solution.

$$\sum_{a \in \delta^-(S)} x_a \geq w_i \quad (i \in V_C, S \subseteq V : i \in S). \tag{16}$$

Note that inequalities (16) strengthen inequalities (5) if $\sum_{j \in V_C(S)} w_j < q w_i$ and $i \in V_C^2$. Obviously, they are unneeded if a customer i is not serviced, i.e. $w_i = 0$. Similarly, the following *Steiner node connectivity inequalities* are valid.

$$\sum_{a \in \delta^-(S)} x_a \geq \sum_{a \in \delta^-(i)} x_a \quad (i \in V_S, S \subseteq V : i \in S). \tag{17}$$

Circuit closure inequalities (18) ensure that the circuits through type 2 nodes are closed at node d.

$$\sum_{a \in \delta^+(S)} y_a \geq w_i \quad (i \in V_C^2, S \subseteq V : i \in S). \tag{18}$$

They are also valid in the following slightly modified version for general circuit nodes in $V_S \cup V_C^1$ (see Hill and Voß [2016], Ineq. 23).

$$\sum_{a \in \delta^+(S)} y_a \geq \sum_{a \in \delta^-(i)} y_a \quad (i \in V_S \cup V_C^1, S \subseteq V : i \in S). \tag{19}$$

4.2 Multi-star inequalities

Multi-star inequalities (e.g. Letchford et al. [2002]) have been proven useful for vehicle routing problems. Out of this class, the following *ring-arb star I inequalities* are valid for $F1$.

$$\sum_{a\in\delta^-(S)} x_a \geq \frac{1}{q}\Big(\sum_{i\in V_C(S)} w_i + \sum_{a\in\delta(V_C\setminus S:S)} y_a \Big) \quad (S \subseteq V). \tag{20}$$

Inequalities (20) are related to the capacitated ring tree multi-star inequalities in Hill and Voß [2016] and take into account the number of serviced customers in S plus the number of customers outside of S that are connected to S by a circuit arc, to estimate the number of needed ring-arbs (i.e. number of arcs entering S). We recall that $y_a \leq x_a$ for each arc $a \in A$. Since a customer in $V \setminus S$ may be the tail node of multiple arcs with heads in S, we cannot count these customers by adding up the corresponding arc variables in inequalities (20). Nevertheless, using the fact that at most $\alpha = \min(q, |S|)$ arcs may enter S from a single customer, we can formulate the following related inequalities.

$$\sum_{a\in\delta^-(S)} x_a \geq \frac{1}{q}\left(\sum_{i\in V_C(S)} w_i + \frac{1}{\alpha} \sum_{a\in\delta(V_C\setminus S:S)} x_a \right) \quad (S \subseteq V). \tag{21}$$

Moreover, the following *ring-arb star II inequalities* are valid for (F) and generalize the capacitated ring closure multi-star inequalities in Hill and Voß [2016].

$$\sum_{a\in\delta^+(S)} y_a \geq \frac{1}{q}\Big(\sum_{i\in V_C^2(S)} w_i + \sum_{i\in V_C^1(S)} \sum_{a\in\delta^-(i)} y_a + \sum_{a\in\delta(S\setminus V_C^2:V_C\setminus S)} y_a + \sum_{a\in\delta(V_C^2(S):V_C\setminus S)} x_a \Big) \quad (S \subseteq V). \tag{22}$$

They take into account the number of serviced type 2 customers in S, the number of type 1 circuit customers in S, the number of customers outside of S that are connected by a circuit arc leaving S from a non type 2 node, plus the number of customers in $V \setminus S$ that are connected by an arc from a type 2 node in S, to estimate the number of needed circuits (i.e. number of circuit arcs leaving S).

4.3 Capacity inequalities

Recall that $w_i = \sum_{a\in\delta^-(i)} x_a$, $\forall i \in V_C$. For $S \subseteq V$, define $\overline{S} = V \setminus S$. For a given $S \subseteq V$, $V_C(S) \neq \emptyset$, the term $\sum_{i\in V_C(S)} w_i$ at the right-hand-side of inequalities (5) can be rewritten as follows:

$$\sum_{i\in V_C(S)} w_i = \left(|V_C| - \sum_{i\in V_C} (1 - w_i) \right) - \sum_{i\in V_C(\overline{S})} w_i \tag{23}$$

or, equivalently,

$$\sum_{i \in V_C(S)} w_i = \left(|V_C| - \sum_{i \in V_C(S)} (1 - w_i) \right) - |V_C(\overline{S})|. \tag{24}$$

We also have:

$$\sum_{i \in V_C(S)} w_i = |V_C(S)| - \sum_{i \in V_C(S)} (1 - w_i). \tag{25}$$

Inequalities (5) can be strengthened as follows:

$$\sum_{a \in \delta^-(S)} x_a \geq \left\lceil \frac{\sum_{i \in V_C(S)} w_i}{q} \right\rceil \quad (S \subseteq V), \tag{26}$$

and using equation (24), can be rewritten as:

$$\sum_{a \in \delta^-(S)} x_a \geq \left\lceil \frac{|V_C| - (\sum_{i \in V_C(S)} (1 - w_i) + |V_C(\overline{S})|)}{q} \right\rceil \quad (S \subseteq V), \tag{27}$$

or, using equation (25) as:

$$\sum_{a \in \delta^-(S)} x_a \geq \left\lceil \frac{|V_C(S)| - \sum_{i \in V_C(S)} (1 - w_i)}{q} \right\rceil \quad (S \subseteq V). \tag{28}$$

The following lemma holds.

Lemma 1. *Let v, y and b be three non-negative integer values with $v > b > 0$ and $mod(v, b) \neq 0$. Then*

$$\left\lceil \frac{v - y}{b} \right\rceil \geq \left\lceil \frac{v}{b} \right\rceil - \frac{y}{mod(v, b)}. \tag{29}$$

Proof. See Baldacci et al. [2007].□

Using the above lemma, inequalities (27) can be linearized as follows:

$$\sum_{a \in \delta^-(S)} x_a + \frac{1}{mod(|V_C|, q)} \sum_{i \in V_C(S)} (1 - w_i) \geq \left\lceil \frac{|V_C|}{q} \right\rceil - \frac{|V_C(\overline{S})|}{mod(|V_C|, q)} \quad (S \subseteq V) \tag{30}$$

Inequalities (30) can be reformulated as

$$\sum_{a \in \delta^-(S)} x_a - \gamma \sum_{i \in V_C(S)} w_i \geq \left\lceil \frac{|V_C|}{q} \right\rceil - \gamma |V_C| \quad (S \subseteq V), \tag{31}$$

where $\gamma = 1/mod(|V_C|, q)$. Since

$$\sum_{i \in V_C(S)} w_i = \sum_{i \in V_C} w_i - \sum_{i \in V_C(\overline{S})} w_i, \tag{32}$$

inequality (30) (for the given S) can be rewritten as follows:

$$\sum_{a\in\delta^-(S)} x_a + \sum_{i\in V_C(\overline{S})} \gamma w_i \geq \beta \tag{33}$$

where $\beta = \left\lceil \frac{|V_C|}{q} \right\rceil - \gamma|V_C| + \sum_{i\in V_C} \gamma w_i$. Similarly, inequalities (28) can be linearized as follows:

$$\sum_{a\in\delta^-(S)} x_a + \frac{1}{mod(|V_C(S)|, q)} \sum_{i\in V_C(S)} (1 - w_i) \geq \left\lceil \frac{|V_C(S)|}{q} \right\rceil \quad (S \subseteq V). \tag{34}$$

5 An iterated local search heuristic

The presented iterated local search heuristic (Lourenço et al. [2003]) is embedded in a multi-start framework. It first constructs a series of solutions for an (undirected) CRTP. Each of the obtained networks is transformed into a directed network, i.e. a RAP solution, and is then locally optimized by the exploration of various model-specific neighborhoods.

5.1 Initial solution pools

In the following we describe techniques to construct feasible solutions for the three profit-oriented models. We recall that, in contrast to the RAP, feasible solutions for all three profit-oriented variants may not admit a network that spans all the customers due to restrictive capacity constraints, i.e. $mq < |V_C|$. Furthermore, each RAP solution can be transformed into a solution for a CRTP by replacing the arcs between two nodes by an edge. Conversely, a CRTP solution can be transformed into a solution for the RAP as follows. Replace each ring by a circuit. Each tree in the remaining undirected forest shares exactly one node with the circuits. Uniquely turn each of these trees into an arborescence rooted in this circuit node.

PC-RAP We transform the PC-RAP into a RAP for which we compute a solution pool using the heuristic techniques for the CRTP as presented in Hill [2015]. Therefore, an edge $e = \{i, j\}$ is assigned a cost $c'_e = (c_{ij} + c_{ji})/2$ to approximate the arc cost. To overcome an eventual capacity shortage we apply a selection mechanism which aims at iteratively reducing the number of customers to mq, if necessary. Let $d(R, S)$ be the length of a shortest (directed) path connecting node set $R \subset V$ to node set $S \subset V$ in G. To estimate the impact of the inclusion of a customer i in the network we calculate its *estimated revenue* $\tilde{r}(i)$ as follows.

$$\tilde{r}(i) \;=\; \begin{cases} p_i - d(\tilde{V}_C^0, \{i\}) & if \quad i \in V_C^1 \\ p_i - d(\tilde{V}_C^0, \{i\}) - d(\{i\}, \tilde{V}_C^0) & if \quad i \in V_C^2 \end{cases}$$

Here $\tilde{V}_C^0$ is the set of the currently discarded customers including the depot d. In each step of the reduction procedure the customer that maximizes this value is removed. Additionally, the trivial zero-cost solution which only contains the depot is added to the solution pool.

BC-RAP In general, a solution for the corresponding PC-RAP is not feasible for the BC-RAP since it might exceed the budget B. However, we can attempt to stay within the available budget by the application of the following reduction technique. While B is exceeded, remove a single leaf node or squeeze out a single circuit node i. In each step choose the one with the minimal relative profit loss, i.e. the customer profit p_i divided by the resulting network cost saving. Again, we apply this technique to all the start solutions obtained for the PC-RAP.

TP-RAP To obtain feasible solutions that meet the target profit T we apply the following procedure. Sort the customers in descending profit order and solve the RAP on the first k customers $i_1, ..., i_k$ such that $\sum_{h \leq k} p_{i_h} \geq T$. We include all Steiner nodes since they are not contributing profit but could be useful to reduce the total arc costs. Again, we obtain a solution pool by solving the RAPs as CRTPs as above.

5.2 Local search techniques

In the following we present operators to identify local improvements for a give feasible initial solution for the PC-RAP. Note that for the BC-RAP and the TP-RAP further feasibility checks regarding budget and target profit need to be included. Moreover, the necessary capacity restrictions are omitted for the sake of a concise description. These will be embedded in a multi-start overall strategy in Section 5.3.

Circuit 2-arc-opt (C2A) As known for the asymmetric traveling salesman problem, we consider all pairs of non-incident distinct arcs (i, j) and (k, l). They are replaced by arcs (i, k), (j, l) and the arcs on the path from k to j are reversed if this results in a circuit of reduced cost.

Circuit reversal (CR) A circuit in the solution is reversed if this results in reduced network costs.

Leaf removal (LR) A leaf customer is removed if the costs for its connecting arc exceed its profit.

Leaf addition (LA) An unused type 1 customer is connected to the existing network if profitable, i.e. the connecting arc costs are less that its profit. In a recursive fashion we consider further leaf additions to the added leaf nodes before evaluating profitability.

Leaf-reconnection (LC) A leaf customer is reconnected to a different network node, if this results in a cost reduction. As for (LA), we also consider the recursive attachment of multiple unconnected type 1 customers to the reconnected leaf node.

Circuit cancellation (CC) We consider a circuit that does not contain type

2 customers and the deletion of one of its arcs. The directed path starting from the arc's head needs to be reversed to maintain feasibility. If the current network does not allow further rings, only the first and the last arc of the circuit can be removed.

1-arb-arc-opt (1A) We consider the removal of an arc from a arborescence structure of a ring. To reestablish a feasible solution we reconnect the isolated arborescence to the ring containing the depot by a different arc and reverse the arcs as needed.

Ring removal (OR) We remove a complete ring if neneficial for the objective value.

Arb addition (AA) We compute a shortest directed path from the nodes of the current network to an unconnected type 1 customer. If this network expansion is profitable after the application of (LA) and (LR) above, we add the corresponding arborescence to the solution.

Circuit from tree (CFT) We consider each node i of a pure tree and the unique directed path to u. A circuit including an unconnected type 2 customer j can be created by the insertion of arcs (i, j) and (j, d). We perform this transformation if capacity feasible and profitable after performing the local searches C2A, RC, LA and LR.

Circuit customer insertion (CCI) For each circuit arc (i, k) we consider the insertion of an unconnected type 2 customer j, if capacity-feasible. (i, k) is replaced by arcs (i, j) and (j, k). Additionally, we evaluate this insertion into the reverse circuit.

Ring-arb addition (RAA) We attempt to install one more ring. For an unused type 2 customer i we consider the circuit obtained from the concatenation of the disjoint shortest directed path from d to i and from i to d, using intermediate nodes that are not part of the solution yet. Similarly, we consider paths calculated in the opposite order: from i to d and then from d to i. We evaluate the ring's profitability after the application of (LA) and (LR) as described above.

Arborescence customer insertion (ACI) Similar to (CCI) we consider the insertion of an unconnected type 1 customer by splitting a solution arc and apply (LA).

Circuit node removal (CNR) For each circuit of length at least four we consider the removal of a single node that has no arborescence attached, and the insertion of the arc that reconnects the two circuit fragments. We also evaluate this modification on the reversed circuit.

Steiner node removal (SNR) We consider the removal of a Steiner node of out-degree less or equal to one.

Steiner node insertion (SNI) We consider the replacement of an arc (i, k) by arcs (i, j) and (j, k) for an unused Steiner node j. Furthermore, if (i, k) is a circuit are, we consider the splitting of the corresponding reverse arc (k, i) in the reversed circuit.

Improvements for (C2A), (LR), (LA), (LC), (1A), (TA), (CCI), (RAA), (ACI), (CNR), are carried out in best fit fashion whereas for (CR), (CC), (OR), (CFT), (SNR), (SNI) first fit is applied.

5.3 Multi-start strategy

Solution pools for the three models are computed as described in Section 5.1. We apply the local search operators to each of the pool networks. Each of them is repeated until no improvement could be found. Globally, the search terminates once no solution could be improved. Table 1 shows the order in which the techniques are applied for each problem variant. A network out of the initial solution pool that minimizes the corresponding objective function is returned.

Table 1: The local search techniques for each optimization model in the order that they are applied.

Model	Local Search Operators
PC-RAP	(CR), (C2A), (LR), (LA), (LC), (1A), (TA), (CCI), (RAA), (ACI), (CNR), (CC), (OR), (CFT), (SNR), (SNI)
BC-RAP	(CR), (CC), (LA), (LR), (CNR), (SNR), (SNI)
TP-RAP	(CR), (C2A), (LR), (LC), (CC), (1A), (SNR), (SNI)

6 Branch-and-cut

In this section we describe the exact algorithms that we developed to solve PC-RAP, BC-RAP and TP-RAP. After explaining our algorithmic framework, we provide the separation routines needed for the valid inequalities from Section 4.

6.1 Exact algorithms

We implemented branch and cut algorithms for the three models based on the formulations presented in Section 3. Initial upper bounds are computed by the corresponding heuristics in Section 5. Additionally, the local search techniques are applied whenever the incumbent is replaced during the branch and bound.

6.2 Cutting planes

In the following we describe the incorporation of the valid inequalities presented in Section 4 into our exact algorithm. Inequalities (14) and (15) can be added to (F) at the root node of the branch and bound algorithm. Connectivity inequalities ((5), (16), (17), (18), (19)), multi-star inequalities ((20), (21), (22)) and capacity inequalities ((30), (34)) are separated in each node instead.

Let x_a^*, w_i^* and y_a^* be the fractional values of the corresponding variables in a solution for the linear relaxation of (F) for $i \in V$ and $a \in A$. The support digraph H^* for x_a^* is the directed graph with node set $V^* = \{i \in V' : \sum_{a \in \delta^-(i)} x_a^* > 0\}$ and arc set $A^* = \{a \in A : x_a^* > 0\}$. Let $H_+^* = (V^* \cup \{s\}, A^* \cup \{(i,s),(s,i) : i \in V^* \setminus \{d\}\})$ be an auxiliary digraph with an artificial source (or sink) node s. For $l \neq r \in V^* \cup \{s\}$, we denote by $\delta_{min}(l : r)$ a cut (L, R) in H_+^* with $l \in L$ and $r \in R$ of minimal weight $c^*(L,R) = \sum_{a \in \delta(L:R)} c_a^*$ with respect to an arc weight function c^*. $\delta_{min}(l : r)$ can be computed in polynomial time.

Then the exact separation procedures either return a most violated inequality or prove that no violated inequality exists. In each node of the branch and bound tree, we identify a most violated inequality for each type, if it exists, and add it to the current formulation before resolving the linear program. Note that inequalities (30) are separated heuristically as described at the end of this section. The individual separation problems can be solved by the computation of cuts of minimal weight in H_+^* using parameters given in Table 2. More detailed, we list the arc weight function (c_a^*) for H_+^*, the nodes l and r, the violation criterion (Φ), and the corresponding cut set ($S \in \{L, R\}$) for each inequality $Ineq.$ (see Section 4).

Table 2: The input for the separation routines based on minimal weight cut calculations in the auxiliary digraph H_+^*.

Ineq.	**Arc weights ($\mathbf{c_a^*}$)**	**l, r**	**Violation (Φ)**	**S**
(5)	$c_a^* = x_a^*\ \forall a \in A^*;\ c_{js}^* = w_j/q\ \forall j \in V_C \cap V^*$	d, s	$c^*(L,R) < \sum_{j \in V_C} w_j^*/q$	R
(16)	$c_a^* = x_a^*\ \forall a \in A^*$	d, i	$c^*(L,R) < w_i^*$	R
(17)	$c_a^* = x_a^*\ \forall a \in A^*$	d, i	$c^*(L,R) < \sum_{j \in V \setminus \{i\}} x_{ji}^*$	R
(18)	$c_a^* = y_a^*\ \forall a \in A^*$	i, d	$c^*(L,R) < w_i^*$	L
(19)	$c_a^* = y_a^*\ \forall a \in A^*$	i, d	$c^*(L,R) < \sum_{j \in V \setminus \{i\}} y_{ji}^*$	L
(20)	$c_a^* = x_a^* - y_a^*/q\ \forall a \in V_C \times V' \cap A^*$ $c_a^* = x_a^*\ \forall a \in V' \setminus V_C \times V' \cap A^*$ $c_{js}^* = w_j/q\ \forall j \in V_C \cap V^*$	d, s	$c^*(L,R) < \sum_{i \in V_C} w_i^*/q$	R
(21)	$c_a^* = x_a^* - x_a^*/q^2\ \forall a \in V_C \times V' \cap A^*$ $c_a^* = x_a^*\ \forall a \in V' \setminus V_C \times V' \cap A^*$ $c_{js}^* = w_j/q\ \forall j \in V_C \cap V^*$	d, s	$c^*(L,R) < \sum_{i \in V_C} w_i^*/q$	R
(22)	$c_a^* = y_a^* - y_a^*/q\ \forall a \in (V_C^1 \cup V_S) \times V_C \cap A^*$ $c_a^* = \min\{y_a^* - x_a^*/q, 0\}\ \forall a \in (V_C^2 \cup \{d\}) \times V_C \cap A^*$ $c_a^* = x_a^*\ \forall a \in V' \times V' \setminus V_C \cap A^*$ $c_{sj}^* = \sum_{a \in \delta^-(j)} x_a/q\ \forall j \in V_C^1 \cap V^*$	s, d	$c^*(L,R) < \sum_{i \in V_C^2} w_i^*/q$ $+ \sum_{i \in V_C^1} \sum_{a \in \delta(i)} y_a$	L
(30)	$c_a^* = x_a^*\ \forall a \in A^*$ $c_{js}^* = \gamma w_j\ \forall j \in V_C \cap V^*$	d, s	$c^*(L,R) < \beta$	R

The separation routine for each inequality given in column $Ineq.$ is as follows. In the auxiliary digraph H_+^*, we associate weights c^* given in the second column to its arcs. Remaining arcs are assigned zero weights. Compute a minimum weight cut $\delta_{min}(l:r)$ in the latter digraph using column $'l,r'$. Column S provides the cut set that is relevant for the inequality and it refers to one side of the cut (L,R). Note that inequalities (16), (17), (18) and (19) are separated for each relevant node i separately, resulting in $|V_C|+|V_S|+|V_C^2|+|V_S \cup V_C^1|$ separation routine calls per node.

Inequalities (34) cannot be separated using the techniques above since their right hand side is dependent on the cut set S. Therefore, we use the following heuristic separation procedure. In each separation phase, we consider all the cut sets that are identified for inequalities in Table 2 and remove d. We extend this set of node sets by the sets obtained by inserting two nodes, removing two nodes, or exchanging a node. Then we add at most three most violated cuts with respect to the absolute discrepancy of the left hand side and the right hand side of inequality (34).

7 Computational Study

In this section we report and analyze the results obtained by our algorithms. They were implemented in c++ and cplex 12.6 with default parameters was used as linear programming solver, and as branch and bound framework. The computations were carried out on a standard personal computer with an i7-3667U 2GHz processor.

7.1 Instances

We derived a total of 315 instances[4] for the PC-RAP (105), the BC-RAP (105) and the TP-RAP (105) from 105 of the CRTP instances introduced in Hill and Voß [2016]. We limited ourselves to problems with up to 51 nodes. The customer types in these instances were assigned uniformly according to the type 1 customer rates 0, 0.25, 0.5, 0.75 and 1. To incorporate an asymmetric cost structure, we applied the following transformation. For each potential edge $e=\{i,j\}$ of cost c_e in the original CRTP instance, we set the cost of the two potential arcs (i,j) and (j,i) in a RAP to $\lceil c_e/2\rceil+\rho_1$ and $\lceil c_e/2\rceil+\rho_2$, respectively. Here ρ_1 and ρ_2 are random integers in the interval $[0,\lceil c_e/2\rceil]$. Note that the original edge cost function c is not Euclidean, but slightly deviates from this metric. We set the budget B for the BC-RAP to $z/2$ where z is the best known upper bound for the corresponding CRTP reported in Hill and Voß [2016]. For an instance with $|V_C|$ customers we assign a profit of $\lceil z/(4|V_C|)\rceil+\rho_3$ to each customer node with ρ_3 a random integer in $[0,\lceil z/(2|V_C|)\rceil]$. For the TP-RAP we use a target profit $T=\lceil\sum_{i\in V_C} p_i/2\rceil$. This results in an average relative potential (i.e., $100*T/[\sum_{i\in V_C} p_i]$) of about 70%. With this parameterization we expect a coverage of roughly 50% of the customers in the BC-RAP and in the PC-RAP. We tighten the capacities to increase the difficulty of the instances by reducing the ring tree capacity q to the smallest integer such that $mq \geq \lceil 0.6|V_C|\rceil$.

[4] The instances and the solutions can be requested from the corresponding author.

7.2 Results

In the following we provide the results for the instances obtained by our algorithms. Table 3 shows properties and PC-RAP results for each instance identified in column P. The numbers of given nodes ($|V'|$,$|V_C^2|$, $|V_C^1|$, $|V_S|$) and the capacity bounds (q, m) are given in the corresponding columns. The final bounds are given in columns lb and ub within the time limit of one hour. The initial upper bound that was found by our heuristic is given in ub_0. Column δ contains the optimality gap $(100(ub - lb)/lb)$ and the collected profit is given in a dedicated column. Whenever $\delta = 0$ then the corresponding instance has been solved to optimality. The number of branching nodes and run time is given in column # and column t, respectively. Results for the BC-RAP and the TP-RAP are listed in Table 4. In addition to the information descrived above, the budget (B), target profit (T), and the network costs are provided.

P	$\|V'\|$	$\|V_C^2\|$	$\|V_C^1\|$	$\|V_S\|$	q	m	lb	ub	ub_0	δ	profit	#	t
1	26	0	12	13	3	3	41	41	41	0	82	1	1
2	26	3	9	13	3	3	59	59	59	0	152	1	1
3	26	6	6	13	3	3	37	39	39	0	140	303	6
4	26	9	3	13	3	3	19	19	19	0	59	35	3
5	26	12	0	13	3	3	46	51	51	0	204	53	3
6	26	0	12	13	2	4	36	42	42	0	93	1	1
7	26	3	9	13	2	4	60	60	60	0	136	1	1
8	26	6	6	13	2	4	46	46	46	0	103	1	1
9	26	9	3	13	2	4	39	39	39	0	115	4	2
10	26	12	0	13	2	4	31	31	31	0	143	1	1
11	26	0	12	13	2	5	50	50	50	0	122	1	1
12	26	3	9	13	2	5	84	84	84	0	189	4	2
13	26	6	6	13	2	5	46	50	50	0	184	1	1
14	26	9	3	13	2	5	54	54	54	0	181	18	3
15	26	12	0	13	2	5	47	55	55	0	212	1	1
16	26	0	18	7	4	3	39	42	42	0	151	67	5
17	26	4	14	7	4	3	38	42	42	0	160	81	10
18	26	9	9	7	4	3	35	39	39	0	145	121	13
19	26	13	5	7	4	3	25	43	43	0	206	3440	210
20	26	18	0	7	4	3	31	41	41	0	218	403	18
21	26	0	18	7	3	4	61	61	61	0	149	46	1
22	26	4	14	7	3	4	46	47	47	0	137	73	2
23	26	9	9	7	3	4	41	50	50	0	174	233	7
24	26	13	5	7	3	4	39	49	49	0	181	192	5
25	26	18	0	7	3	4	31	31	31	0	100	39	1
26	26	0	18	7	3	5	62	62	62	0	171	108	1
27	26	4	14	7	3	5	36	36	36	0	136	1	2
28	26	9	9	7	3	5	62	62	62	0	200	3050	104
29	26	13	5	7	3	5	68	70	70	0	269	32000	2795
30	26	18	0	7	3	5	67	75	75	0	254	39	2
31	26	0	25	0	5	3	29	39	39	0	153	1	0
32	26	6	19	0	5	3	34	36	36	0	154	2165	46
33	26	12	13	0	5	3	47	51	51	0	217	10841	851
34	26	18	7	0	5	3	43	45	45	0	201	871	17
35	26	25	0	0	5	3	36	42	42	0	152	374	5
36	26	0	25	0	4	4	62	75	75	0	199	36	4
37	26	6	19	0	4	4	66	77	77	0	216	287	28
38	26	12	13	0	4	4	53	53	53	0	177	1597	105
39	26	18	7	0	4	4	41	48	48	0	195	6721	296
40	26	25	0	0	4	4	34	39	39	0	193	280	14
41	26	0	25	0	3	5	51	64	64	0	179	68	5
42	26	6	19	0	3	5	68	74	74	0	210	185	15
43	26	12	13	0	3	5	56	59	59	0	177	1147	73
44	26	18	7	0	3	5	41	41	41	0	178	630	35
45	26	25	0	0	3	5	21	25	25	0	152	1	1
46	51	0	12	38	3	3	27	28	28	0	96	65	11
47	51	3	9	38	3	3	21	28	28	0	129	122	36
48	51	6	6	38	3	3	33	39	39	0	170	13	24
49	51	9	3	38	3	3	34	36	36	0	185	357	138
50	51	12	0	38	3	3	30	44	44	0	191	258	49
51	51	0	12	38	2	4	27	27	27	0	70	1	5
52	51	3	9	38	2	4	46	52	52	0	132	1	12
53	51	6	6	38	2	4	41	41	41	0	165	1	8
54	51	9	3	38	2	4	34	35	35	0	127	1	9
55	51	12	0	38	2	4	24	26	26	0	131	1	5
56	51	0	12	38	2	5	52	52	52	0	141	1	2
57	51	3	9	38	2	5	41	47	47	0	151	74	26
58	51	6	6	38	2	5	35	41	41	0	125	12	23
59	51	9	3	38	2	5	28	41	41	0	230	147	35

60	51	12	0	38	2	5	41	41	41	0	190	1	7
61	51	0	25	25	5	3	27	31	31	0	139	70	21
62	51	6	19	25	5	3	74	74	74	0	223	3	22
63	51	12	13	25	5	3	41	60.2	41	46.9	221	1918	3600
64	51	18	7	25	5	3	28	46.3	28	65.2	146	2503	3600
65	51	25	0	25	5	3	35	41	41	0	220	524	62
66	51	0	25	25	4	4	58	61	61	0	196	96	297
67	51	6	19	25	4	4	65	69.6	65	7	186	1929	3600
68	51	12	13	25	4	4	88	88	88	0	262	730	1027
69	51	18	7	25	4	4	47	67.9	47	44.4	258	1752	3600
70	51	25	0	25	4	4	46	54	54	0	261	161	206
71	51	0	25	25	3	5	54	55	55	0	181	52	169
72	51	6	19	25	3	5	45	47	47	0	173	868	1568
73	51	12	13	25	3	5	65	65	65	0	231	1	72
74	51	18	7	25	3	5	50	53	53	0	197	1903	1816
75	51	25	0	25	3	5	28	31	31	0	149	70	107
76	51	0	37	13	8	3	78	83	83	0	255	157	414
77	51	9	28	13	8	3	115	124	124	0	294	742	1345
78	51	18	19	13	8	3	87	115.1	87	32.3	281	1102	3600
79	51	27	10	13	8	3	68	94.5	68	39	275	1660	3600
80	51	37	0	13	8	3	75	87	87	0	299	253	468
81	51	0	37	13	6	4	71	76	76	0	245	891	648
82	51	9	28	13	6	4	48	57.2	48	19.2	178	1524	3600
83	51	18	19	13	6	4	71	98.3	71	38.5	286	1193	3600
84	51	27	10	13	6	4	47	72.5	47	54.3	209	1844	3600
85	51	37	0	13	6	4	43	63.1	55	14.7	227	3440	3600
86	51	0	37	13	5	5	60	73	73	0	263	1387	845
87	51	9	28	13	5	5	112	126	126	0	295	836	2133
88	51	18	19	13	5	5	66	88.3	66	33.7	243	1207	3600
89	51	27	10	13	5	5	47	68.7	47	46.3	292	1945	3600
90	51	37	0	13	5	5	57	86	86	0	291	4950	2976
91	51	0	50	0	10	3	47	53	53	0	231	563	22
92	51	12	38	0	10	3	85	120	85	41.1	310	1614	3600
93	51	25	25	0	10	3	61	95.5	61	56.5	290	1539	3600
94	51	37	13	0	10	3	47	84.6	52	62.7	312	1563	3600
95	51	50	0	0	10	3	74	80	80	0	312	1046	61
96	51	0	50	0	8	4	38	50.6	43	17.8	226	5454	3600
97	51	12	38	0	8	4	95	115.9	109	6.3	329	2283	3600
98	51	25	25	0	8	4	64	103.8	64	62.2	322	1116	3600
99	51	37	13	0	8	4	105	151.2	105	44	371	1163	3600
100	51	50	0	0	8	4	50	65.4	61	7.3	318	4603	3600
101	51	0	50	0	6	5	35	43.4	39	11.3	224	4824	3600
102	51	12	38	0	6	5	62	91.1	62	46.9	227	1342	3600
103	51	25	25	0	6	5	83	123	83	48.2	365	1133	3600
104	51	37	13	0	6	5	63	111.7	63	77.3	288	1213	3600
105	51	50	0	0	6	5	69	97	75	29.3	337	3628	3600

Table 3: Instance properties, and results for the PC-RAP.

	BC								**TP**							
P	**B**	**lb**	**ub**	$\mathbf{ub_0}$	δ	**costs**	**#**	**t**	**T**	**lb**	**ub**	$\mathbf{ub_0}$	δ	**profit**	**#**	**t**
1	79	112	112	107	0	79	587	5	84	51	51	65	0	90	8	1
2	105	152	152	152	0	98	1293	36	111	65	65	104	0	116	3	3
3	114	145	145	127	0	113	1183	53	116	85	85	133	0	116	99	5
4	118	123	123	96	0	118	3119	183	116	112	112	143	0	119	1093	40
5	121	156	156	119	0	121	946	14	136	95	95	160	0	136	3	3
6	82	115	115	95	0	81	170	3	84	51	51	71	0	93	1	0
7	104	152	152	137	0	100	62	6	111	65	65	89	0	114	1	1
8	120	145	145	100	0	112	348	11	116	79	79	135	0	118	1	1
9	125	159	159	99	0	123	93	4	136	104	104	172	0	136	50	6
10	126	144	144	75	0	126	360	4	136	112	112	172	0	143	1	2
11	85	129	129	122	0	85	116	2	90	51	51	62	0	91	5	1
12	121	189	189	189	0	105	1259	36	136	74	74	121	0	145	1	1
13	126	165	165	141	0	126	195	7	136	89	89	140	0	138	1	1
14	140	186	186	106	0	137	245	6	150	106	106	126	0	153	26	3
15	140	167	167	156	0	124	2225	27	150	110	110	152	0	155	1	1
16	104	146	146	130	0	104	192	9	108	69	69	102	0	108	1	1
17	128	165	165	152	0	124	918	49	134	94	94	136	0	134	91	11
18	137	160	160	145	0	137	5540	315	134	106	106	140	0	134	468	35
19	146	181	181	156	0	145	3564	128	162	132	132	154	0	163	840	48
20	151	186	186	155	0	150	1751	53	162	132	132	166	0	165	1477	46
21	109	162	162	154	0	109	1314	8	123	70	70	88	0	123	14	2
22	143	179	179	154	0	143	493	41	134	90	90	182	0	136	158	5
23	157	194	194	178	0	157	7303	313	162	121	121	155	0	171	212	12
24	167	212	212	151	0	166	835	42	171	129	129	211	0	178	117	5
25	170	188	188	128	0	170	1752	17	171	149	149	221	0	171	344	8
26	114	173	173	161	0	113	445	5	123	70	70	88	0	123	66	1
27	139	167	167	153	0	138	495	11	134	100	100	150	0	136	12	4
28	168	220	220	201	0	167	23808	2779	171	117	117	188	0	176	490	22
29	181	244	244	217	0	181	20882	2868	204	143	143	153	0	208	2285	70
30	188	254	254	144	0	179	461	8	204	137	137	220	0	205	13	3
31	123	156	156	135	0	123	441	5	123	90	90	130	0	123	15	1
32	147	179	179	140	0	147	4947	425	151	118	118	197	0	153	2351	151

33	157	206	206	156	0	156	12910	2146	174	126	126	190	0	175	1074	45
34	164	202	202	149	0	161	5020	141	174	138	138	211	0	176	1447	49
35	164	198	198	159	0	164	2076	27	174	144	144	176	0	175	8349	151
36	126	200	200	170	0	125	354	10	151	89	89	128	0	151	12	2
37	156	230	230	200	0	156	872	78	174	105	105	153	0	175	53	8
38	173	215.8	214	187	0.8	173	18733	3600	174	124	124	176	0	174	856	55
39	179	218.1	210	195	3.9	170	30487	3600	191	147	147	217	0	195	1396	77
40	181	211	211	161	0	180	6696	122	191	154	154	196	0	193	182	12
41	127	190	190	158	0	127	19	2	151	92	92	133	0	151	6	2
42	160	226	226	216	0	160	58	6	174	106	106	131	0	175	397	19
43	185	234.4	231	199	1.5	185	17865	3600	191	133	133	239	0	191	683	65
44	189	214	214	171	0	188	23099	1540	191	156	156	225	0	194	3518	166
45	198	205	205	161	0	195	2402	53	191	173	173	228	0	193	35	6
46	78	96	96	95	0	76	10502	498	78	54	54	82	0	79	606	23
47	95	116	116	86	0	95	441	145	95	76	76	115	0	99	171	54
48	107	135	135	117	0	107	648	103	113	81	81	112	0	113	52	18
49	111	136.7	129	90	6	107	3858	3600	123	90	90	142	0	123	73	19
50	121	154	154	133	0	121	1799	275	131	94	94	136	0	132	20	12
51	80	103	103	96	0	80	170	14	78	53	53	82	0	79	29	5
52	105	151	151	137	0	104	48	18	113	66	66	99	0	113	56	18
53	115	147	147	127	0	115	154	25	123	87	87	158	0	123	9	19
54	119	144	144	91	0	118	146	22	123	92	92	166	0	127	3	16
55	126	139	139	93	0	126	3414	562	131	105	105	153	0	131	1	7
56	85	128	128	110	0	84	192	7	95	59	59	68	0	95	10	6
57	102	137	137	120	0	98	5746	1340	113	76	76	110	0	113	18	18
58	126	155	155	149	0	122	975	287	131	99	99	147	0	133	234	118
59	139	169	169	138	0	139	1093	278	137	106	106	153	0	140	69	27
60	140	170	170	119	0	140	757	61	137	104	104	133	0	143	1	5
61	123	152	152	123	0	123	166	17	126	98	98	155	0	127	301	20
62	151	218.6	217	180	0.7	148	1802	3600	172	111	111	167	0	173	126	54
63	156	216.2	181	157	19.5	156	1342	3600	172	122.2	134	158	8.8	181	1688	3600
64	161	209.6	153	153	37	147	1581	3600	172	136.5	141	194	3.2	175	1974	3600
65	164	200	200	137	0	164	2092	399	172	141.1	144	182	2	176	9535	3600
66	126	184	184	168	0	126	562	464	151	99	99	132	0	152	200	361
67	152	223.7	204	192	9.6	146	1425	3600	172	111	111	167	0	172	733	1569
68	176	268.4	209	208	28.4	175	1083	3600	200	125.4	129	208	2.8	206	1251	3600
69	179	247	207	185	19.3	179	1580	3600	200	147.3	163	207	9.6	201	1316	3600
70	181	226	226	155	0	181	3647	1525	200	156.5	167	235	6.3	210	3543	3600
71	127	181	181	158	0	127	307	333	151	102	102	142	0	151	147	316
72	168	212.7	197	197	8	168	1386	3600	172	125.5	132	177	4.9	174	903	3600
73	185	239.2	225	190	6.3	183	1157	3600	200	148.7	159	237	6.5	209	955	3600
74	194	251	235	166	6.8	190	1449	3600	200	148.7	162	235	8.2	214	1526	0
75	195	217.1	215	157	1	194	6441	3600	200	172	172	280	0	202	242	254
76	152	230	230	188	0	152	459	603	187	121	121	177	0	187	1172	822
77	188	303.6	260	243	16.8	186	1189	3600	223	123.4	125	228	1.3	223	3474	3600
78	188	300.5	231	231	30.1	182	1156	3600	223	132	132	238	0	224	723	1609
79	190	284.1	210	207	35.3	186	1316	3600	223	145.4	188	254	22.6	223	990	0
80	190	266.4	264	194	0.9	187	4355	3600	223	155.2	165	233	5.9	223	3541	3600
81	154	225	225	186	0	154	1651	1027	187	124	124	178	0	187	3823	1204
82	182	237.7	222	204	7.1	180	1230	3600	187	139	139	225	0	188	1154	3146
83	200	299.7	243	230	23.4	197	776	3600	223	141	172	249	18	228	943	3600
84	202	273.3	205	194	33.3	202	1478	3600	223	159.8	225	263	29	224	1152	3600
85	205	265.4	248	204	7	203	3056	3600	223	167	167	241	0	223	2053	2218
86	157	224	224	188	0	157	3641	1490	187	127	127	192	0	187	3036	1225
87	204	317.2	304	278	4.3	202	1192	3600	223	119	119	210	0	223	422	830
88	216	305.9	235	225	30.2	215	836	3600	223	147.7	171	286	13.6	227	1100	3600
89	218	289.7	233	233	24.3	213	1209	3600	223	161.9	205	275	21	229	1266	3600
90	223	309.6	298	235	3.9	217	2494	3600	252	176.1	180	229	2.1	253	5232	3600
91	188	239	239	207	0	188	1312	56	191	142	142	199	0	192	115	11
92	214	323.9	258	258	25.6	213	1425	3600	253	150.7	178	251	15.3	254	1983	3600
93	223	316.1	250	250	26.4	223	1256	3600	253	170.5	204	291	16.4	253	1862	3600
94	226	306.7	288	207	6.5	224	1744	3600	253	176.3	281	281	37.3	259	1241	3600
95	231	310	310	175	0	231	3156	428	253	186	186	303	0	253	9178	941
96	192	241.1	238	204	1.3	192	5246	3600	191	147	147	200	0	191	6547	1391
97	229	338.3	276	258	22.6	229	1053	3600	253	157.1	164	274	4.2	255	1530	3600
98	247	341.3	262	262	30.3	246	707	3600	253	165.4	287	287	42.4	259	1172	3600
99	251	401.1	295	295	36	243	1006	3600	298	171.1	252	291	32.1	306	974	3600
100	247	311.3	297	210	4.8	247	4333	3600	253	189.2	224	316	15.6	255	1905	3600
101	195	236.6	227	198	4.2	195	7338	3600	191	152.1	155	216	1.9	191	11300	3600
102	246	320.1	247	247	29.6	244	1102	3600	253	170.4	193	302	11.7	255	1255	3600
103	263	375.9	281	281	33.8	263	846	3600	298	186.2	288	336	35.4	299	1023	3600
104	263	371.9	286	286	30	258	940	3600	298	200	281	342	28.8	299	1141	3600
105	263	360.1	309	243	16.5	263	3065	3600	298	207.6	238	297	12.8	305	3229	3600

Table 4: Results for the BC-RAP and TP-RAP instances.

7.3 Performance analysis

In the following we analyze and illustrate the performance of our algorithms. Our methods solve all the 45 instances with 26 nodes for both, the PC-RAP and the

TP-RAP. They fail to solve 3 small BC-RAP instances to optimality. 35/24/31 out of the 60 instances with 51 nodes (58.3%/40.0%/51.7%) can be solved for the PC-RAP, the BC-RAP, and the TP-RAP, respectively. Figure 2 (left) shows the percentage of 51 node instances that could be solved to optimality within the time limit (1 hour) for different type 1 customer rates for all the models. Instances with customers of both types seem to be harder than purely ring or tree based ones, which is also observed in Hill and Voß [2016]. A more detailed picture of the obtained results is given in Figure 2 (right). It can be seen that the average optimality gaps for the PC-RAP are the largest (15.9%), compared to the ones for the BC-RAP (10.4%) and the TP-RAP (7.0)%. The quality of

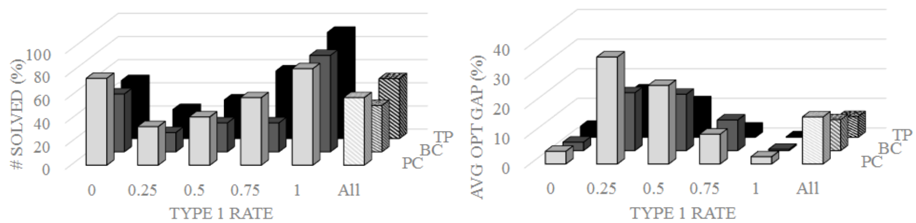

Fig. 2: The relative numbers of 51 node instances solved to optimality with respect to the number of customers and the customer type rate (left). The obtained gaps after one hour (right).

the start solutions that our heuristics in Section 5 find is shown in Figure 3. For instances with 26 and 51 nodes, the average relative optimality gaps with respect to the heuristic upper bound and the best lower bound calculated by the exact algorithm is shown. The results are comparable for both instance sizes. While relatively tight upper bounds (7.7%) can be found for the PC-RAP, BC-RAP gap sizes are about half as strong (15.2%). The gaps for the TP-RAP are notably larger (29.5%). We also observe that the local search techniques are more effective for instances with both customer types. Furthermore, the runtimes and

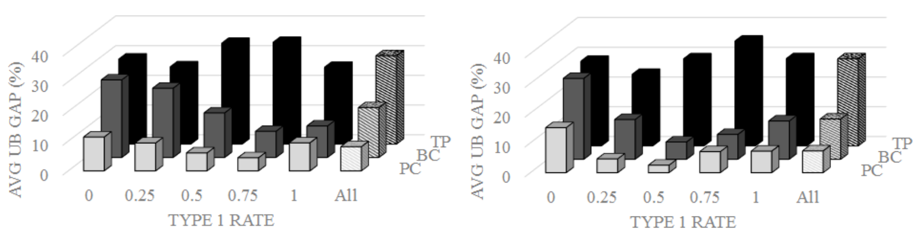

Fig. 3: The average relative optimality gaps with respect to our heuristic start solutions and the best lower bounds found by the exact methods (Left: 26 nodes; right: 51 nodes.)

the number of branching nodes are illustrated in Figure 4 for the three models. Reduced runtimes for instances with one customer type only are consistent with the hardness observations related to Figure 2 above. Finally, the sum of all runtimes for the PC-RAP is about 30 hours, 45 hours for the BC-RAP, and 32 hours for the TP-RAP. The average number of branching nodes are different for small instances (26 nodes) and larger instances (51 nodes). We observe a pattern due to increased hardness of the mixed-customer-type problems for the former instances. The opposite pattern can be seen for the larger instances: A higher average number of branching nodes is solved for the single-customer-type instances, which is due to the increasing effort of solving the linear programs with augmenting instance size.

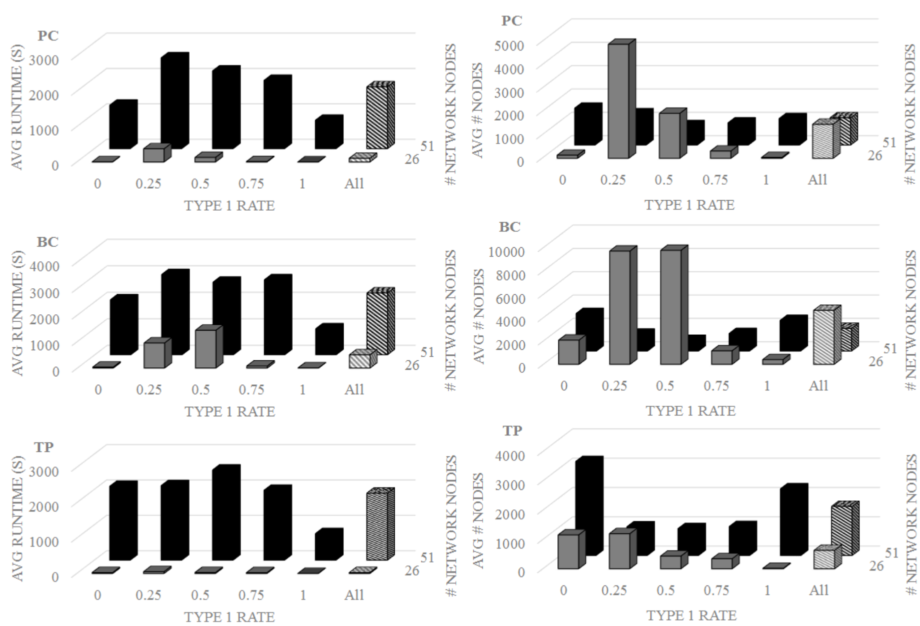

Fig. 4: Average runtimes and branching nodes for the PC-RAP, the BC-RAP and the TP-RAP for different type 1 customer rates and instance sizes.

7.4 Solution network analysis

To provide further insight into the problem instances and the solutions found by our algorithms we analyze the obtained networks in this Section. Since our models allow that only a subset of all the customers appears in the solution network, we illustrate the distribution of the percentage of the number of connected customers with the box plots in Figure 5. We also show the network cost in relation to the total customer profit for the PC-RAP, i.e. the network cost divided by the collected profit. For the BC-RAP and the TP-RAP, the diagrams show the relative network budget used, and the relative profit achieved beyond the target profit, respectively. The average numbers of customers (50%)

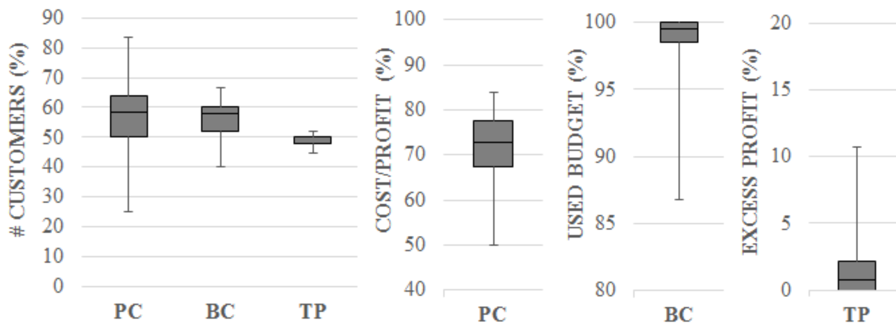

Fig. 5: The relative numbers of network customers served with respect to the total number of customers. The network cost compared to the collected profit. The relative used network budget. The relative profit collected beyond the taget. (From left to right.)

in the solution networks correspond to our expectations when constructing the instances (see Section 7.1). Additionally, the number of customers is about 84.9%/84.6%/73.6% of the customer capacity allowed by the total capacity (qm) on average (PC/BC/TP). On average, the network cost is about 71.9% of the collected profit for the PC-RAP. In 50 of the 105 best solutions for the BC-RAP, the budget is completely utilized. 35 TP-RAP solutions exactly meet the target profit.

7.5 Cut impact

To better understand the benefit of the cutting planes described in Section 4, we provide an experimental evaluation in this section. We compare the achieved lower bound improvements for the test problems with respect to the three main cut classes: connectivity inequalities, multi-star inequalities and capacity inequalities. To measure the strength of the formulation, we limit ourselves to the value of the linear relaxation at the root node. Let $\mathcal{C}_{CONN}$ be the set of cutting planes that correspond to the connectivity inequalities (see Section 4.1, inequalities (14)-(19)), $\mathcal{C}_{MS}$ the cutting planes that correspond to the multi-star inequalities (see Section 4.2, inequalities (20)-(22)), and $\mathcal{C}_{CAPA}$ the ones that correspond to the capacity inequalities (see Section 4.3, inequalities (23)-(34)). We denote the set of generic cutting planes that are generated by cplex IBM [2013] by $\mathcal{C}_{CPLEX}$. Then let $\mathcal{C}_{\emptyset} = \emptyset$, $\mathcal{C}_I = \mathcal{C}_{CONN}$, $\mathcal{C}_{II} = \mathcal{C}_I \cup \mathcal{C}_{MS}$, $\mathcal{C}_{III} = \mathcal{C}_{II} \cup \mathcal{C}_{CAPA}$ and $\mathcal{C}_{IV} = \mathcal{C}_{III} \cup \mathcal{C}_{CPLEX}$. Figure 6 shows the relative lower bounds at the root node for these cutting plane configurations and the different type 1 customer rates. For each instance, a value of 100% represents the lower bound obtained by adding all the cut sets ($\mathcal{C}_{IV}$). The average relative lower bounds over the instances for each model are 75.4%/94.9%/89.1% ($\mathcal{C}_{\emptyset}$), 93.6%/98.4%/96.9% ($\mathcal{C}_{I}$), 97.4%/99.3%/97.9% ($\mathcal{C}_{II}$), 97.5%/99.4%/97.9% ($\mathcal{C}_{III}$), and 100.0% ($\mathcal{C}_{IV}$) (PC/BC/TP). The computation times for $\mathcal{C}_{IV}$ are 2.0/1.5/2.2 and 47.8/47.3/51.1 seconds on average for instances with 26 and 51 nodes, respectively. We observe a 25%/5%/11% drop on average when omitting

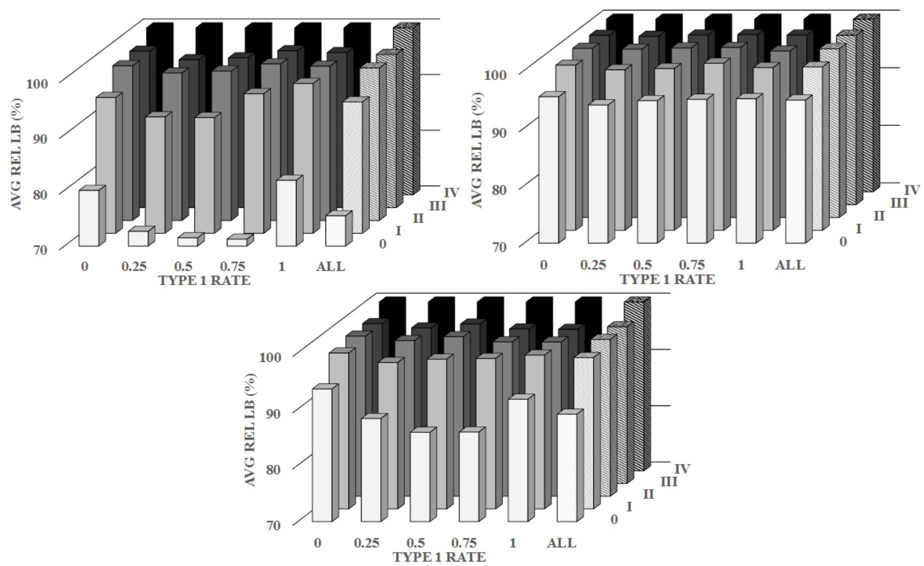

Fig. 6: Relative lower bounds when incorporating the cut classes $\mathcal{C}_{I}$, $\mathcal{C}_{II}$, $\mathcal{C}_{III}$ and $\mathcal{C}_{IV}$ (100%) for the PC-RAP, the BC-RAP and the TP-RAP.

all cuts. The largest contribution is due to connectivity inequalities $\mathcal{C}_{CONN}$. In our study, the sole addition of cuts from class $\mathcal{C}_{I}$ contributes 74%/69%/71% to the rise of the lower bounds on average. We note that the capacity cuts included in $\mathcal{C}_{III}$ improve the lower bound up to 6% for selected instances, though not being significantly effective on average.

8 Conclusion

In this work we introduced four new network optimization models that generalize the capacitated ring tree problem (CRTP). Based on the definition of the ring arborescence problem (RAP), which corresponds to a the CRTP with an asymmetric edge cost structure, we present problems that incorporate customer profits. We formulate these prize-collecting, budget-constrained and target profit variants as non-compact mixed-integer programs. Several classes of valid inequalities are developed to strengthen the formulations. A branch and cut algorithm

is described. Moreover, we elaborate heuristics based on iterated local search for the three problems. In a computational study we provide computational results for our algorithms for a set of 315 hard problem instances. It turns out that we are able to solve almost all the profit-oriented problems with up to 26 nodes to optimality efficiently (132 out of 135), and 50% of the 51 node instances (90 out of 180). Furthermore, we analyze the impact of our cutting planes to show their effectiveness.
We see two main topics for future research related to solving these hard problems in discrete optimization. Further metaheuristics, or matheuristics could be developed to eventually strengthen the upper bounds. In preliminary experiments, we observe that alternative mathematical formulations are promising in order to derive stronger lower bounds.

References

F. H. N. Abe, E. A. Hoshino, and A. Hill. The ring tree facility location problem. *Electronic Notes in Discrete Mathematics*, 50:331–336, 2015.

C. Archetti, D. Feillet, A. Hertz, and M. G. Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6):831–842, 2009.

C. Archetti, D. Feillet, A. Hertz, and M. G. Speranza. The undirected capacitated arc routing problem with profits. *Comput. Oper. Res.*, 37(11):1860–1869, 2010.

C. Archetti, N. Bianchessi, and M. G. Speranza. The capacitated team orienteering problem with incomplete service. *Optimization letters*, 7(7):1405–1417, 2013a.

C. Archetti, M. G. Speranza, and D. Vigo. Vehicle routing problems with profits. Technical report, Department of Economics and Management, University of Brescia, 2013b.

E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1989.

E. Balas. The prize collecting traveling salesman problem: II. polyhedral results. *Networks*, 25(4):199–216, 1995.

E. Balas and M. Fischetti. The fixed-outdegree 1-arborescence polytope. *Mathematics of Operations Research*, 17(4):1001–1018, 1992.

E. Balas and C. H. Martin. Combinatorial optimization in steel roling. In *Workshop on combinatorial optimization in science and technology*, COST, 1991.

R. Baldacci and M. Dell'Amico. Heuristic algorithms for the multi-depot ring-star problem. *European Journal of Operational Research*, 203(1):270 – 281, 2010.

R. Baldacci, M. Dell'Amico, and J. J. Salazar. The capacitated m-ring-star problem. *Oper. Res.*, 55(6):1147–1162, 2007.

A. Bautzer, L. Gouveia, A. Paias, and J. M. Pires. Models for a steiner multi-ring network design problem with revenues. *TOP*, pages 1–21, 2015.

I.-M. Chao, B. L. Golden, and E. A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88(3):464 – 474, 1996a.

I.-M. Chao, B. L. Golden, and E. A. Wasil. A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, 88(3):475 – 489, 1996b.

O. Chapovska and A. P. Punnen. Variations of the prize-collecting steiner tree problem. *Networks*, 47(4):199–205, 2006.

A. M. Costa, J.-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks*, 53(2):141–159, 2009.

D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205, 2005.

M. Gendreau, G. Laporte, and F. Semet. The covering tour problem. *Operations Research*, 45(4):568–576, 1997.

B. Golden, S. Raghavan, and D. Stanojević. The prize-collecting generalized minimum spanning tree problem. *Journal of Heuristics*, 14(1):69–93, 2008.

L. Gouveia and J. M. Pires. Models for a steiner ring network design problem with revenues. *European Journal of Operational Research*, 133(1):21 – 31, 2001.

M. Hachicha, M. J. Hodgson, G. Laporte, and F. Semet. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, 27(1):29 – 42, 2000.

A. Hill. Multi-exchange neighborhoods for the capacitated ring tree problem. *LNCS, Volume 8962*, pages 85–94, 2015.

A. Hill and S. Schwarze. Exact algorithms for bi-objective ring tree problems with reliability measures. *Preprint, Optimization-Online*, 2016.

A. Hill and S. Voß. Generalized local branching heuristics and the capacitated ring tree problem. *Research paper, Department of Engineering Management, University of Antwerp*, 2014.

A. Hill and S. Voß. Optimal capacitated ring trees. *EURO Journal on Computational Optimization*, 4(2):137–166, 2016.

E. A. Hoshino and A. Hill. A branch and price algorithm for the capacitated ring tree problem. *Manuscript*, 2015.

IBM ILOG CPLEX Optimization Studio documentation V12.6.0. IBM, 2013.

D. S. Johnson, M. Minkoff, and S. Phillips. The prize collecting steiner tree problem: Theory and practice. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 760–769, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

J. G. Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6(14):307 – 335, 1998.

A. N. Letchford, R. W. Eglese, and J. Lysgaard. Multistars, partial multistars and the capacitated vehicle routing problem. *Mathematical Programming*, 94(1):21–40, 2002.

I. Ljubić, R. Weiskircher, U. Pferschy, G. W. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, 105(2-3):427–449, 2006.

H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search. In *Handbook of metaheuristics*, pages 320–353. Springer, 2003.

J. Park and B.-I. Kim. The school bus routing problem: A review. *European Journal of Operational Research*, 202(2):311 – 319, 2010.

P. Schittekat, J. Kinable, K. Sörensen, M. Sevaux, F. Spieksma, and J. Springael. A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, 229(2):518 – 528, 2013.

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