DEPARTMENT OF ENGINEERING MANAGEMENT

**An equi-model matheuristic for the multi-depot
ring star problem**

Alessandro Hill & Stefan Voß

UNIVERSITY OF ANTWERP
Faculty of Applied Economics

City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

An equi-model matheuristic for the multi-depot ring star problem

Alessandro Hill & Stefan Voß

RESEARCH PAPER 2014-015
AUGUST 2014

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

**The research papers from the Faculty of Applied Economics
are also available at www.repec.org
(Research Papers in Economics - RePEc)**

D/2014/1169/015

An equi-model matheuristic for the multi-depot ring star problem

Alessandro Hill¹ and Stefan Voß²

¹ ANT/OR - Operations Research Group
Department of Engineering Management
University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium
alessandro.hill@uantwerpen.be

² Institute of Information Systems (IWI)
University of Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany
stefan.voss@uni-hamburg.de

Abstract. In the multi-depot ring star problem (MDRSP) a set of customers has to be connected to a set of given depots by ring stars. Such a ring star is a cycle graph, also called a ring, with some additional nodes assigned to its nodes by single star edges. Optional Steiner nodes can be used in the network as intermediate nodes on the rings. Depot dependant capacity limits apply to both, the number of customers in each ring star and the number of ring stars connected to a depot. The MDRSP asks for a network such that the sum of the edge costs is minimized.

In this paper we present a matheuristic that iteratively refines a solution network in a locally exact fashion. In contrast to existing approaches the optimization model that is used to explore the various structural multi-exchange neighborhoods in our algorithm is the MDRSP itself. A first class of neighborhoods considers local sub-networks for optimal improvements. Through an advanced modeling technique we are able to refine arbitrary sub-networks of suitable size induced by simple node sets. A second class aims at globally restructuring the current network after the application of different contraction techniques. For both purposes we develop an exact branch & cut algorithm for the MDRSP that efficiently solves the local optimization problems to optimality, if they are chosen reasonably in terms of size and complexity. The efficiency of the approach is shown by computational results improving known upper bounds for instance classes from the literature containing up to 1000 nodes. 91% of the known best objective values are improved up to 13% in competitive computational time.

Keywords: Multi-depot ring star problem, Hybrid heuristic, Branch & cut, Local refinement, Network design, Matheuristic

1 Introduction

In the last decade telecommunication companies have been spending billions on their physical networks and will continue their investments. Due to the enormous technological progress and the demand of high speed data connectivity of the customers, improved network structures are needed. Network providers are faced with eminent challenges concerning the extension and the replacement of existing out-dated copper cables by faster fiber-based technology. Since urban areas need an almost complete coverage with improved service and rural areas at least partially, the material costs for the expensive up-to-date cables together with the installation costs are significant. The cost-saving potential motivates the application of optimization methods to suitable mathematical models.

Generally spoken, in modern telecommunication network design a principal task is to connect given customers to one or several distributors by means of some technology. The structure of the network is defined by technological or business requirements. Many modern network design models consist of a subproblem that requires a certain core network structure combined with a component that allows assignments of customers to the core nodes. The *Steiner tree star problem* and the *connected facility location problem*, for example, ask for a cost saving allocation of the customers to potential suppliers that are connected to the distributor by a tree structure. In the *ring star problem* each customer has to be connected to a depot by a ring or may be assigned to a node of such a ring. Capacities are introduced in the *capacitated m-ring star problem* (CmRSP), where the number of customers visited by or assigned to a ring is limited as well as the number of possible rings served by the depot. In our work we consider the *multi-depot ring star problem* (MDRSP) which generalizes the CmRSP by allowing multiple depots. The desired core ring structure guarantees reliability in the sense of one-link-failure insensibility since after the removal of a single link the nodes on the ring are still connected by a path. Optional Steiner nodes may be used to reduce the network costs. These nodes do not consume capacity and can be incorporated into rings if advantageous in terms of the overall edge costs. The MDRSP is a suitable model for routing problems, too, as it generalizes the well-known *capacitated vehicle routing problem* with a heterogeneous fleet, homogeneous demands and multiple depots. Real-world instances of the MDRSP and related problems commonly involve several hundred customers that have to be supplied. As a generalization of the classical *traveling salesman problem* (TSP) the MDRSP is NP-hard.

Finding a proven optimum for reasonably sized problem instances is not possible using nowadays methods and computing power. Therefore, current practical solution approaches are mostly heuristical but may incorporate exact subroutines. In this respect recent ideas from the realm of matheuristics and related hybrid methods come into play. Moreover, the rapid improvement of integer linear programming (ILP) solvers renders possible their usage to full capacity as subroutines in heuristic solution frameworks. Within the method, setting up suitable subproblems that locally improve a given solution structure seems a important issue. On the one hand the sizes of the explored neighborhoods have

to be chosen carefully such that the computational complexity does not exceed the optimization potential of the exact method. On the other hand subproblems should be worth being solved to optimality to outperform heuristic searches. In our work we develop an efficient hybrid algorithm for the MDRSP based on a branch & cut method. We demonstrate how the integration of exact methods can be used efficiently to solve this sophisticated problem in combinatorial optimization and to improve previous best known results from literature.

The paper is organized as follows. In Section 2 we provide a formal definition of the problem and a literature review. Our branch & cut method for the MDRSP is explained in Section 3. The main procedure is illustrated in detail in Section 4. Our tests on literature instances and the comparison with existing heuristics is given in Section 5. We present numerical results together with a study analyzing our improvement performance. The paper is closed with some conclusions in Section 6.

2 The multi-depot ring star problem

2.1 Problem definition

In the following we give a formal definition of the *multi-depot ring star problem* (MDRSP) and introduce some basic notation.

Definition. We are given an undirected complete graph $G = (V, E)$. The node set V is the disjoint union of depots D , customers U and Steiner nodes W . A set of possible assignments $A \subseteq U \times (U \cup W)$ contains arcs from customers to potential suppliers. Each edge $e \in E$ is associated with a nonnegative weight c_e and each assignment $a \in A$ is associated with a nonnegative weight c_a , respectively. A cycle C in G containing exactly one depot is a ring. Paired with assignments $B \subseteq U \setminus V(C) \times V(C) \subseteq A$ we obtain a ring star. For each depot $k \in D$ a ring star limit m_k and a customer limit q_k per ring star are given. A solution to the MDRSP is a set of ring stars such that

- every customer is part of exactly one ring star,
- each Steiner node is used in at most one ring star,
- each depot k is contained in at most m_k ring stars, and
- the number of customers in a ring star does not exceed the limit q_k for its depot k .

The MDRSP asks for a solution of minimal total cost, i.e. minimized sum of edge costs and assignment costs.

Note that in our definition of the MDRSP we allow heterogeneous capacities m_k and q_k depending on the considered depot k . The latter property generalizes the model used in the literature so far. On the one hand this increases flexibility for the application to real-world problems. On the other hand we utilize this potential when formulating subproblems in our algorithm. If capacities are not

heterogeneous, we may simply skip the index referring to the depot, i.e., we use m and q as data.

A path in a solution ring together with the associated assignments is called a *path star*. For a customer v , $R(v) = \{u \in V : (v, u) \in A\}$ denotes the set of potential suppliers. Edges between depots are not considered in a solution and just carried along for simplicity. The MDRSP may turn out to be infeasible due to capacity restrictions, i.e., capacities of the potential suppliers might not allow the supply of customers such that the required customer demands are fulfilled. Here we assume feasibility to avoid technical issues complicating our descriptions.

2.2 Related work

An overview of previously studied ring-based optimization models including algorithmic approaches can be found in [8] and [4]. The *ring star problem* was introduced by [7] and simply asks for a single cost-minimal ring star for a depot connecting all the customers. The authors studied this problem from a polyhedral point of view and proposed a branch & cut algorithm. Later [3] introduced the single depot version of the capacitated problem allowing multiple ring stars and Steiner nodes: the capacitated m -ring star problem. The authors provided branch & cut procedures based on two ILP formulations and identified their two index formulation as computationally superior to the two-commodity flow formulation. Results slightly outperforming the branch & cut method were achieved by a branch & cut & price algorithm by [6]. The heuristic of [11] yields efficient results for the CmRSP. It incorporates several local search steps that are combined with a random shaking procedure. Computational comparison shows that it is on average superior to the metaheuristic approach for the CmRSP of [10]. The two index cut set formulation of [3] was extended for the MDRSP by [2] when introducing the multi-depot generalization. Three heuristic methods are proposed for the computation of upper bounds. These start with an initial solution that is the result of a heuristic solution of the vehicle routing problem obtained after dropping the Steiner nodes and star-assignments. Then heuristic ring star improvements are followed by different tabu searches. Lower bounds are derived after the application of a depot contraction argument that reduces the problem to the CmRSP which is then solved by exhaustive runs of their branch & cut method.

Several locally exact refinement-based algorithms have been developed for discrete optimization problems related to routing. For a given initial solution they aim at its iterative improvement through local search. The methods share the fundamental idea of defining a search space of neighboring solutions in each step from which a best one is chosen. The well-known k -opt improvement procedures for the TSP can be seen as very basic examples. The exploration of possible rearrangements of single route nodes for the TSP by [13] is often referenced as a starting idea in the literature. More sophisticated techniques for the *capacitated distance constrained vehicle routing problem* have been developed by [5]. They extract partial routes from an existing solution and reinsert

them according to the exact solution of an ILP-based reallocation problem. [14] carried out this technique for the vehicle routing problem. More recently, in the work of [12] the related *open vehicle routing problem* was tackled likewise involving exact local improvements using integer programming. Here the reallocation model contains an exponential number of variables representing possible route subsequences considered for reinsertion. The heuristical column generation based approach is extended by local search techniques and a perturbation mechanism. Recently, another hybrid refinement method by [1] based on ILP subproblems for an *inventory routing problem* was able to improve existing heuristic results. Our algorithm differs from the ones found in the literature in that the subproblems solved in our approach share the structure of the global problem, i.e., the subproblems are smaller versions of the global problem. This allows us to use the exact algorithm developed for the global problem to iteratively solve well-chosen subproblems. We define neighborhoods that correspond to these subproblems according to several structural patterns that we identify in the current solutions. Additionally, we develop sophisticated contraction based neighborhoods to effect structural recombinations respecting the capacity constraints. These refinement techniques are embedded in a heuristic framework that uses tabu mechanisms to minimize redundant searches. For some references regarding matheuristics the reader is referred to [9].

3 A branch & cut method

3.1 An ILP model

Branch & cut algorithms are known to be efficient for numerous combinatorial optimization problems. For the related CmRSP Baldacci et al. [3] concluded that a cut set formulation is computationally superior to a proposed two-commodity formulation. Therefore, we develop a branch & cut method for the non-compact three index formulation for the MDRSP introduced by [2]. For a node set $S \subseteq V$ we denote the set of its neighbors in G by $\delta(S)$. For a single node $v \in V$ we denote the set of its neighbors in G by $\delta(v)$.

$$\min \quad \sum_{k \in D} \sum_{e \in E} c_e x_e^k + \sum_{k \in D} \sum_{a=(i,j) \in A} c_a z_{ij}^k \quad (1)$$

$$s. \ t. \quad \sum_{e \in \delta(k)} x_e^k \leq 2m_k \quad \forall k \in D, \quad (2)$$

$$\sum_{e \in \delta(i)} x_e^k = 2z_{ii}^k \quad \forall i \in U, \forall k \in D, \quad (3)$$

$$\sum_{e \in \delta(j)} x_e^k = 2w_j^k \quad \forall j \in W, \forall k \in D, \quad (4)$$

$$\sum_{k \in D} \sum_{j \in R(i)} z_{ij}^k = 1 \quad \forall i \in U, \quad (5)$$

$$\sum_{k \in D} w_j^k \leq 1 \quad \forall j \in W, \quad (6)$$

$$\sum_{i \in U} \sum_{j \in S \cap R(i)} z_{ij}^k \leq \frac{q_k}{2} \sum_{e \in \delta(S)} x_e^k \quad \forall S \subseteq V \setminus D : S \neq \emptyset, \forall k \in D, \quad (7)$$

$$x_e^k \in \{0, 1\} \quad \forall e \in E, \forall k \in D, \quad (8)$$

$$z_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in D, \quad (9)$$

$$w_j^k \in \{0, 1\} \quad \forall j \in W, \forall k \in D. \quad (10)$$

The model uses binary variables x_e^k for all $e \in E$ to indicate the installation of the edge e in a ring star connected to the depot k . If a Steiner node $j \in W$ is used in a ring star of depot k , variable w_j^k takes the value one, zero otherwise. For the binary assignment variables $z_{ij}^k \forall (i, j) \in A$ a non-zero value corresponds to the assignment of customer i to the ring node j connected to depot k . Note that an assignment variable z_{ii}^k of value one is equivalent to i being a ring node of degree two. Constraints (2) impose that a depot k is serving at most m_k rings. The problem variant which asks for an exact predefined number of ring stars for the depots, as in the CmRSP, requires equalities here. Customers and Steiner nodes are incident to exactly two ring edges if placed on a solution ring due to (3), (4) and (6). Each customer has to be either assigned to a ring or part of a ring which is ensured by the equalities (5). The *fractional capacity inequalities* (7) forbid subtours and restrict the number of customers in the ring stars of each depot. This exponential number of constraints cannot be incorporated explicitly, hence they are added dynamically subsequent to a separation process. Again, the depot-dependent customer per ring star capacities q_k for each depot k are introduced without considerably complicating the model.

3.2 Cutting planes

In the branch & cut fashion, at each node of the branch & bound tree a lower bound is computed. The linear programming relaxation is solved and the objective value is used for pruning. Since we are dealing with an exponential number of model cuts in (7) this is achieved by an iterative process. Initially, we solve the linear program (LP) omitting the fractional capacity constraints. In a typical step we add identified violated inequalities and resolve it. A separation procedure is needed to find these required cuts. By adding inequalities that are not implied by the current model we can further reduce the solution space of the LP. Such *valid* cuts may improve the resulting lower bound and herewith allow earlier pruning in the branch & bound tree. We adopt valid inequalities for the CmRSP, among them are connectivity inequalities and multi-star inequalities. Compared to the CmRSP a specific depot does not necessarily serve all the customers in the MDRSP. Therefore, the number of customers $|U|$ is replaced by the current sum of fractional assignments $U_k = \{i \in U : \sum_{j \in R(i)} z_{ij}^k > 0\}$ for

a depot $k \in D$. The identification of violated inequalities is still of polynomial complexity and essentially based on computations of maximal flows in auxiliary networks. A more detailed description of the separation techniques for the CmRSP can be found in [3].

Fractional capacity cuts For a set of nodes $S \subseteq V \setminus D : S \neq \emptyset$ we know that the customers associated with S are $U \cap S$ and the customers in $U \setminus S$ that are assigned to a node in S . Thus at least $\sum_{i \in U} \sum_{j \in S \cap R(i)} z_{ij}^k / q_k$ rings are needed to satisfy the demand of S for each depot k . Since each ring that connects a node in S implies an entering edge and a leaving edge, S violates constraint (7) if

$$\sum_{i \in U} \sum_{j \in S \cap R(i)} z_{ij}^k \leq \frac{q_k}{2} \sum_{e \in \delta(S)} x_e^k. \quad (11)$$

The separation of these integrality cuts for the single depot case was carried out in [3]. In contrast to the CmRSP we separately look for the most violated cut for each depot k individually.

Connectivity cuts Each customer u that is situated on a ring has to be connected to the depot by two node disjoint paths which form the ring. By Menger's theorem this is equivalent to requiring each set $S \subseteq V \setminus D$ that includes u to be connected by two edges to $V \setminus S$. The resulting well-known connectivity cuts are added by separating the most violated inequality for each depot $k \in D$, similarly as in [3], out of

$$\sum_{e \in \delta(S)} x_e^k \geq 2 \sum_{j \in S} z_{uj}^k \quad S \subseteq V \setminus D, \quad u \in U \cap S. \quad (12)$$

Ring multi-star cuts The ring multi-star cuts were introduced by [3] for the single depot case and are related to multi-star cuts known for vehicle routing problems. They generalize (11) by adding capacity consumption of S obtained from counting the customers in $V \setminus S$ that are connected to nodes in S . We add the most violated inequality for each depot $k \in D$ out of

$$\sum_{e \in \delta(S)} x_e^k \geq \frac{2}{q_k} \left(\sum_{i \in U, j \in S} z_{ij}^k + \sum_{i \in U \setminus S, j \in S} x_{ij}^k \right) \quad S \subseteq V \setminus D. \quad (13)$$

Further tightening of the LP relaxation Based on a connectivity argument we add the following constraints to the initial model.

$$\sum_{k \in D, e \in \delta(S)} x_e^k \geq 2 \quad S = i \cup R(i), \quad i \in U. \quad (14)$$

We also add the following known valid inequalities as done before for related problems [7, 3].

$$x_{ij}^k \leq z_{jj} \quad j \in U, \quad i \in W, \quad k \in D, \quad (15)$$

$$x_{ij}^k \leq w_j \quad i, j \in W, \quad k \in D. \quad (16)$$

Besides this we activate the CPLEX-internal cutting techniques to increase the solver efficiency. We experienced that the lower bound computations of the solver could be speeded up by initially adding some classes of inequalities to the integer model. These constraints do not improve the obtained optimal value of the LP relaxation but tend to reduce the number of needed cuts and runs of the simplex algorithm, respectively. Inequality (11) is explicitly added for cut sets $S = R(i) \forall i \in U$, $S = i \cup R(i) \forall i \in U$ and $S = V \setminus D$ for each depot $k \in D$. These are the sets of potential customer suppliers with and without the corresponding customer and the set of non-depots. To further accelerate the cutting process we initially add the following equalities for each $k \in D$ as special cases of (12).

$$x_{ij}^k + z_{ij}^k \leq \begin{cases} z_{jj}^k & \text{if } j \in U, \\ w_j^k & \text{if } j \in W, \end{cases} \quad i \in U, \quad j \in V \setminus D, \quad i \neq j. \quad (17)$$

Since the considered subproblems in our algorithm may contain zero capacities for some instances we use simple inequalities to avoid non-depot ring stars in early fractional solutions:

$$\sum_{i \in U, j \in R(i)} z_{ij}^k \leq q_k m_k, \quad k \in D. \quad (18)$$

4 The hybrid method

Our hybrid algorithm iteratively tries to improve an existing solution for the MDRSP. For this purpose we implement several strategies to construct local search spaces. The obtained neighborhoods are explored by solving MDRSP type subproblems to optimality using our branch & cut method. These procedures are explained in Sect. 4.1. Moreover, we designed a mechanism to escape local optima based on several clustering and contraction ideas. Again the optimization is done by exact methods for the resulting vehicle routing problem variants. Section 4.2 contains the detailed techniques. Afterwards the initial construction heuristic is described in Sect. 4.3 since it is based on the latter ideas. The overall process is illustrated in Sect. 4.4.

4.1 Exact local refinement procedure

In the following we introduce various so-called *reallocation* models which refer to the modification of customer positions in routing structures. For the MDRSP this may correspond to rearrangements of customers within a ring star but also among several ring stars served by distinct depots. We accomplish local optimization by a clustering step followed by the solution of a MDRSP. We present a generic technique to build the subproblems from clusters containing nodes in the solution. The key ingredients are the different cluster generation ideas. Changing cluster strategy means exploring diverse neighborhoods with respect

to the current overall solution. Our ideas of building problem clusters are explained in Sect. 4.1 using a general framework. Afterwards we elaborate the actual construction of the corresponding MDRSP based on such a cluster. Although the local application of exact methods is equivalent to solving the problem after complementary variable fixing, we explicitly extract substructures which is more efficient since we avoid the solver overhead due to the unchanged solution structure.

Clustering For the success of the local refinement idea, the choice of clusters is crucial. Generally spoken, a cluster $C \subseteq U \cup W$ that will be used to create a subproblem is a set that contains customers and Steiner nodes. We construct clusters using different strategies to obtain diverse structured neighborhoods. We develop the following constructions based on different neighborhood concepts.

Ring edge The idea for this cluster type is to avoid expensive ring edges passing by close customers that could be collected or assigned to the ring. Starting with a current ring edge uv we add a current solution node x if $c_{vx} \leq c_{uv}$ and $c_{ux} \leq c_{uv}$. Nodes from different solution rings may be part of the cluster. In a selection procedure we add current solution nodes according to a low value of the sum of the squared distances to the edge's ends $c_{vx}^2 + c_{ux}^2$ until a cluster size limit is reached. Figure 1 shows an example for such a cluster.

Ball The basic idea is to select nodes contained in the neighborhood of a central node without considering the current solution structure. For a given cluster-center node v we build a subproblem cluster C_v consisting of customers and Steiner nodes according to two parameters. The number of total nodes and the number of customers is limited. After sorting the nodes in $(U \cup W) \setminus \{v\}$ according to their distance to v we iteratively add close nodes until one of the capacities is utilized. An example of such a clustering is given in Figure 2.

Path star We extract a path on a solution ring with its assigned customers and close unused Steiner nodes. The number of admitted customers depends on a limiting parameter. We give an example of such a clustering in Figure 3.

Ring star A complete ring star from the current solution is added to the cluster if the number of nodes does not exceed a cluster size limit. Figure 3 illustrates such a selection.

Path star exchange Following the path star idea we build two close clusters on distinct rings and join them. See Figure 4 for an example of such a clustering.

To enable the following problem construction, some cleaning steps for the cluster nodes are needed. Let S denote a current solution and C a selected cluster:

Fig. 1. Example for ring edge clustering

- We call a non-cluster path (star) in S connecting C to itself without containing a depot an *ear* (*star*). Ears are added to C and their structure is forced to be unchanged to reduce the computational burden for the branch & cut method.
- Customers in the cluster that are assigned to non-cluster suppliers are removed.
- We add customers in $U \setminus C$ that are assigned to nodes in the cluster.
- If just one node of a ring is in the cluster, add its non-depot neighbors in S to enable improving exchanges.

These modifications might lead to an empty cluster or a predefined cluster size limit may be exceeded. In such a case we build up the next scheduled cluster.

Problem construction In the following we describe a general procedure to define the local problem based on a cluster C that is a MDRSP itself. For a solution of such a subproblem. We recall that each solution for such a subproblem embedded in the complementary global solution corresponds to a solution subspace for the overall problem. Basically this set of feasible solutions represents our considered local neighborhood.

An edge or a node might need to be forced to be part of the solution in the modeling process. Independent of the exact solution method, we can incorporate such features by modifying the edge costs. For example, forbidding an edge e is achieved by setting its cost c_e to big M . A suitable value for M could be the

Fig. 2. Example for ball clustering

Fig. 3. Example for path star/ring star clustering

Fig. 4. Example for path star exchange clustering

sum of all the routing and assignment costs plus one. Since we employ a branch & cut algorithm, variable fixing could also be done in the ILP solver by the modification of bounds for the corresponding variable.

The customers and Steiner nodes of the local MDRSP are the ones in the cluster. A potential assignment in the original problem is considered in the subproblem if the customer and the potential supplier are elements of C . The costs of assignments and ring edges are inherited from the global problem. Since C does not contain the depot C induces a set of path trees in the current overall solution S . For each path tree T we introduce an artificial depot k' and set the edge costs to connect it to the the ends of the partial ring in T to zero. We call such such a substituting edge a *leg*. At the same time we forbid all the edges connecting k' to other cluster nodes. Figure 5 illustrates such a construction. If a path star T' that is substituted by a leg does not contain customers, we allow cluster reconnections: the cost of the edge connecting k' and C is set to the cost of T' . Herewith ring stars may be redesigned starting from the depot (see Figure 6). The ring star limits of the depots are set to the number of its involved ring stars and forced to be tight in the ILP for ring star fragments that do not have two reconnectable legs. Ring stars may completely disappear during the local optimization followed by the removal of the leg Steiner nodes. Any local redesign has to be valid with respect to the global customer per ring capacities of the depots. Let k be the depot in S that currently serves the customers of a cluster path star that is part of the ring star R . We set the ring customer limit for the corresponding artificial depot to the capacity of the current solution ring

Fig. 5. Example cluster leading to a 2-depot ring star problem

star depot k minus the number of ring star customers in R not in C . In the following paragraphs we consider the special case of allowed reconnections. If the cluster contains nodes of two or more ring stars of the same depot we pursue as described in the ring star management paragraph below.

Fig. 6. Example cluster with reconnectable zero-customer legs and their modeling

Depot ring star interchanges A cluster in general contains multiple ring star fragments, say r , connected to the same depot k . To respect the depot capacities when recombining ring star interchanges we could underestimate the customer per ring star capacity of k or introduce $\binom{r}{2} - r$ artificial depots to model all

the possible recombinations or add leg customers until the numbers of supplied customers on each leg are equal and force them to be part of a solution. We decided to integrate this one by one in our subproblem. Consider the case that different customer per ring star capacities would apply for two rings of a depot that serve cluster customers due to non identical numbers of customers on the legs. We model one of the rings as associated with another artificial depot which is added to the subproblem with corresponding capacities and connection costs. So a depot in a subproblem of type MDRSP is just allowed to serve multiple rings if the path star customer numbers are equal. The consequence is that in general the number of depots increases. This technical step needs to be reverted when interpreting the solution of the subproblem globally.

Cluster reconnection Leg substituting edges in the local problems are weighted according to the path lengths. For a leg that does not contain any customers, the corresponding ring star fragment connector x does not necessarily have to be preserved during optimization. In the case that both legs of a cluster ring star fragment are reconnectable, we certainly include all ring star customers by our clustering technique. Besides this case, a single leg might be flexible or none. In [3] a depot k is modeled by a source s and a sink t in a network flow problem fashion. This allows a simple fixing of a leg substituting edge. The additional variables representing edges incident to t provide the flexibility we need here. Note that fixing a single depot-connector edge cannot be done just by forbidding the depot edges, since the second leg is flexible. But we achieve this by the following transformation. Insert an artificial Steiner node u and force the edge kx to be present in the solution. Set zero cost for the edge xu and big M cost for other edges incident to x . For any other cluster node v let $c_{uv} = c_{xv}$. The integration of this aspect is significant when considering clusters containing ring star fragments that are close to depots. Figure 7 illustrates such a transformation.

Fig. 7. Enabling one leg to be flexible in the subproblem

Ring star management By our default construction, the number of ring stars per depot is preserved. To enable the closure of a ring star R we take into account the savings due to leg removals. This is just possible if we are dealing with a completely reconnectable ring star fragment as explained above. So far, no additional ring stars may be added by the cluster optimization. The depots not appearing in the local problem are potential ring star suppliers for the cluster customers. To model this, we add the cluster-uninvolved depot k and connect it to all the cluster nodes by original routing costs. Its ring star potential is set to m_k minus its number of ring stars in the current solution. If no ring star capacity is available, k may be dropped. Figure 8 illustrates this model modification.

Fig. 8. Enabling ring stars from depots not supplying the cluster customers

Extending existing heuristic local search By setting up appropriate subproblems, we cover existing heuristic local search techniques. For instance, the two customer exchanges, the customer transfer and the squeeze searches by [11] are covered by our cluster selection techniques.

Every time the branch & cut method is called to solve a local problem, we pass the current local solution to the solver to accelerate the exact solution process. In the case of an improvement, the current solution is updated according to the exact solution of the local subproblem. If cluster reconnections are implied by the solution of the subproblem, the former Steiner legs have to be deleted in the global solution and the proposed connection legs have to be installed.

Refinement strategies The subproblem optimization has varying structural impact on the solution depending on the cluster type. A path star based MDRSP will not affect the global ring star partition of the customers. However, a local ball refinement may imply several customer interchanges between ring stars. Our

method groups a sequence of local refinement attempts based on the same cluster structure. So when applying a local optimization technique, we actually perform multiple locally exact searches successively differing in their parametrization. Since we allow clusters of a certain type to be overlapping, a new cluster can not be built until the previous optimization has terminated. To avoid redundant local problems we use tabu lists containing representatives for each processed structure. For instance, a ring star is identified by a ring node adjacent to the corresponding depot during a phase. In the edge cluster based local refinement phase we repeatedly select an unused ring edge e of maximal length. e is then used for clustering, the derived subproblem is solved and e is set tabu for the subsequent iterations in this phase. After a limit of local search steps is reached the tabu list is cleared. Similarly, we produce an iterative covering of each ring star with path stars allowing a certain overlapping in a path star optimization phase. The ring star cluster based subproblems are CmRSPs. In such a phase all the current ring stars are processed. The ball type subproblem parameters are the central nodes. We generate a set of centers heuristically first, maximizing their distance sum. The number of centers is calculated dividing the number of customers by a scaled ball cluster customer limit. Based on this selection we build and solve the parameterized subproblems consecutively.

The path star exchange cluster sequence is constructed as follows. First select a non-tabu path star, then find the closest customer v that is not part of the same ring star. The second path star is built such that v is as central as possible depending on its depot distance. The primary path star is set tabu and to exclude remote path stars not worth considering, a local problem is just built if the path star distance is less than the network diameter scaled with a positive parameter lower or equal to one. Best improvement strategies are not efficiently applicable in this setting, because the dependencies of the subproblems would remarkably increase the computational effort. The repeated solution of all scheme subproblems would be the cost of such a look ahead strategy.

4.2 Defocus improvement procedure

Our local optimization technique is generally not able to escape local minima by changing the global structure. We do not use random sampling techniques to effect perturbation. Instead, after getting stuck locally we switch to adequate sophisticated neighborhoods involving solution nodes that were not combined in previous local clusters. Therefore, we first try to identify substructures that are unlikely to break apart after further optimization. This is achieved by clustering appropriate customers and Steiner nodes according to their distance. Then the nodes contained in such clusters are contracted into single nodes. Afterwards, subproblems are created by considering the structure induced by the nodes corresponding to the contracted clusters. On the reduced solution structure the optimization is carried out by solving vehicle routing problem variants to optimality. Namely capacitated multi-depot vehicle routing problems with heterogeneous (CMDVRP_{het}) and homogeneous demands (CMDVRP_{hom}) with an underlying homogeneous vehicle fleet. The latter reduces to the classical TSP

in the case of a single depot. Given a MDRSP solution S we build a set of ring stars R by either adding

- two ring stars,
- all the ring stars connected to a depot, or
- all the ring stars in S .

In special cases the selection rules may return the same set of rings, e.g., when having at hand a solution containing just one depot or when installing two rings in total. Note that if S consists of a unique ring star, our technique is still applicable with a slight modification. For the chosen ring stars in R we apply one of the two clustering procedures below to the set of involved customers U_R . Since we intend to solve an NP-hard combinatorial optimization problem again, a customer limit max_U is used to keep the computational effort manageable.

Fixed cluster size We contract customer sets of fixed size on the selected substructure. The fixed cluster size r is computed by $\lceil |U_R|/max_U \rceil$. For each ring star in U_R we iterate over the customers, either on the ring or assigned to it, and successively fill clusters. The cardinality of the last attained ring star cluster may be less than r leading a certain inaccuracy that has to be handled afterwards if complicating. A CMDVRP_{hom} is set up by using the depots in R and a customer for each cluster. Vehicle depots are considered if contained in a ring star of R . The vehicle limit for a depot k is set to m_k minus the number of solution ring stars of k not intersecting with R . The vehicle capacities are set to $\lceil q_k/r \rceil$. Routing costs are estimated by the minimum over the customer distances between two clusters or the depots, respectively.

Flexible clusters Different cluster sizes are considered depending on the customer distances. We initialize the construction with disjoint clusters that contain exactly one customer of U_R . The cluster distances are the ring edge costs of the customer connecting edges, if adjacent. Iteratively, we merge clusters using the minimum distance criterion until their number is lower or equal to max_U . Additionally, the clustering is refined by 2-opt moves. To balance the cluster sizes we limit the cardinalities by $|U_R|/4$. A CMDVRP_{het} is built as follows. From every cluster we derive a customer with a demand equal to the number of contracted customers. Vehicle depots are considered if contained in a ring star of R . The vehicle capacities are inherited from the master MDRSP and the distance of two customers is set to the minimum distance between their contracted customers. The vehicle limit for a depot k is set to m_k minus the number of solution ring stars of k not in R . Figure 9 depicts examples of such contraction types for the simple single ring star case. The key idea is to profit from the gathered solution by grouping close customers that are likely to be part of one ring star after restructuring. To further increase the considered recombinations two variants are added that are not based on the underlying current solution. Again, we build clusters of fixed and flexible size, respectively, but not depending on the existing ring stars. They are grouped according to their distances in G . The resulting optimization problems do not differ from the ones above. Given a cluster size, we

Fig. 9. Examples for the contraction ideas on given solutions and the induced current routing structure: fixed cluster size 4 (left) and flexible cluster size (right)

greedily select $\lceil |U_R|/max_U \rceil$ distributed customers of $|U_R|$ by repeatedly adding the farthest customer. Starting with these singleton clusters the closest unused customer to a cluster is added while preserving the size limit.

During these consolidation processes the Steiner nodes used in the current solution are completely ignored. The obtained solutions are rather used for the assignment of customers to the depots. However, convenient solutions will be reconsidered in a subsequent local refinement phase before being evaluated. If the created problem turns out to be infeasible due to the capacity restrictions we skip this defocus step. Likewise, if the solution of a routing subproblem corresponds to the structure retrieved from the current set of ring stars, we proceed to the next local or defocus optimization (if clustering was based on the current solution). In any other case an overall solution to the MDRSP is constructed by replacing the current ring star structure by the routing solution after performing a heuristical expansion. Note that we obtain rings without customer assignments and the previous star structures in the clusters are not reusable in general. Therefore, we solve a TSP on each ring first. This is done greedily followed by 2-opt and 3-opt searches before applying the exact local improvement procedures. We first run the locally exact optimization using ring star, path star, ball and path star exchange locally as introduced in the previous section. Once the improvement leads to the overall starting solution we stop to prevent from redundant search in a local minimum. If the reoptimized solution finally yields an improved objective value, the current solution structure is replaced.

4.3 Initial solution and correctness

Our improvement ideas, as representatives of local search in general, operate on an initial solution for the MDRSP. The used procedure to generate such a set of ring stars is a special case of the defocus method. The latter is applied on the set of customers U using clusters of fixed size and no use of a given solution structure. Since this does not necessarily yield a feasible solution at all we embed this idea into a multistart procedure. Consider, for example, a given MDRSP with $|U| = 45$, $|D| = 3$ and capacities $q = 15$, $m = 1$. Depending on the

computational power available, assume a routing subproblem customer limit of 12. Then the fixed cluster size is determined by $\lceil |U_R|/max_U \rceil = \lceil 45/12 \rceil = 4$. The resulting defocus subproblem turns out to be infeasible. Therefore, we repeat the attempt with a decremented parameter $max_U = 11$. This increases the fixed cluster size to 5 and produces a feasible subproblem. In the case of a repeated failure we iteratively retry with a still reduced parameter. Although we succeeded in this illustrative example, this is generally false. In the case of failure we use a simple cluster first - route second approach to guarantee a feasible starting solution. According to the depot-customer distances we first assign customers to depots greedily respecting their customer capacities. Then the customers are grouped to route nodes using a clustering scheme. Basically we successively route customers in the nearest neighbor fashion until the ring capacities are utilized or all the depot customers are assigned to rings. However, on the tested literature instances we did not encounter the need of such a heuristical starting solution. Note that the initial solution consists of rings only, not involving any assignments yet. Through our modular overall strategy explained in Sect. 4.4 we do not apply any further improvements here. Since we assured the construction of at least one feasible solution for the MDRSP that can be returned it follows that our algorithm is correct.

4.4 Overall strategy

We distinguish between techniques changing the structure of a single ring star and those allowing ring star interactions. The computational complexity, i.e. the number of variables and constraints in the mathematical model, increases dramatically with every additional depot. Hence, we execute the complex searches involving multiple ring stars subsequent to less extensive steps. The process is interrupted and restarted every time an overall improvement is achieved and the incumbent solution is updated. After an iteration limit is reached or the solution could not be further improved by any search technique, the current best set of ring stars is returned. Our detailed overall strategy combining the various local searches is illustrated in Figure 10.

5 Computational study

5.1 Implementation details

The algorithm was implemented in C++ using CPLEX 12.2 as ILP solver carrying out the branch & cut framework. Computations were done on an Intel i5 U470 1.33GHz processor unit with 4 GB working memory available. CPLEX was set to run in the single thread mode. The memory usage for computations of our implementation did not exceed 100 MB in the performed tests. The parametrization of our method is of major importance for its overall performance. Parameter calibration is required based on the used ILP solver and the computing power. For each ILP we applied a time limit of 90 seconds to avoid a runtime explosion due to solver-hard subproblems. This was encountered rarely, but mostly

Fig. 10. Overall optimization process

when solving the routing subproblems for the defocus techniques. Optimal network flows in the branch & cut separation steps are efficiently computed by the network simplex algorithm. The subproblems arising in the contraction phases, $\text{CMDVRP}_{\text{hom}}$ and $\text{CMDVRP}_{\text{het}}$ were solved by a branch & cut algorithm. Clustering was done using the minimum distance criterion for adding customers. The number of overall iterations was limited to 25, which was not reached in our experiments. If a subproblem turned out to be infeasible, we simply skipped it. This was just encountered in the defocus optimization as mentioned above due to unsatisfiable capacity restrictions.

5.2 Parameter calibration

We determine the parameters limiting the subproblem sizes through running a series of tests on our machine. Starting with modest limits we repeatedly apply our method on the set of test instances while incrementing these bounds. Once the optimization of a specific type of subproblem exceeds the runtime of two seconds for five times in an instance we fix the corresponding parameter. Since size limit variations mutually impact the subproblem behaviour, we rather use this approach as a rough solver capacity estimation. Our obtained bounds for the subproblem sizes are as follows. For the defocus searches we allow a maximum of 20 cluster contractions. The local search procedures built problems respecting the following limits on the number of customers: RingStar (55), Ball (22), PathStar (20), PathStarXchg (17).

5.3 Results on literature instances

To assess the quality of our approach we use random instances that were generated by [2] to test their heuristic approaches. Given uniformly distributed node coordinates in the plane, the distances are set to be Euclidean. These medium sized problems contain up to 300 customers, 750 Steiner nodes and 5 depots. Various combinations of q , m and $|D|$ are considered. The instance set is divided into classes A and B where q is 30 and 40, respectively. For our studies we draw conclusions from 276 instances (kindly provided by Roberto Baldacci). Attempts to compute the LP lower bounds for the instances through our branch & cut method failed due to exhaustive memory allocation of the solver. Therefore, we compare our results with the best results achieved by three heuristics of [2].

Tables 1 and 2 show the objective function values of our solutions (obj) for several instances listed in [2]. Additionally, the relative improvements in percent compared to the best results obtained by [2] are given in column Δ . Out of 276 instances in total we are able to find improved solutions in 251 cases, about 91%, respectively. In instance class B we compute higher upper bounds for just two problem instances. However, we do not exceed the known upper bound value by more than 1.1%. The best solution improvement our algorithm achieves is 13%. Restricted to problems of class A we could still observe 8%. On average we improve the current best solutions by 2.6%. More specifically, we observe a superior performance on the instances with more than 200 customers. These instances are improved by 1.9/3.9% (A/B) compared to the smaller ones (2.2/2.7%). Our method is slightly more effective on instances with an increased number of depots and higher rings per depot capacities. We note that this is not necessarily related to the hardness of the instances because we use existing results for benchmarking. However, we believe that one of the strengths of our approach is the ability to perform improving multiple ring star movements of customers which pays off on those instances. The runtime of our method is on average slightly less than the cumulated runtimes reported by [2]. All the instances could be solved within 1500 seconds. Since cluster selection heuristics are at most of quadratic complexity by far the most of the computational time is used for the solution (or detection of infeasibility) of up to 4000 ILPs. The number of improvements in locally exact search steps is about 75 and defocus changes usually less than 5. As illustrated exemplarily in Figure 11 our method is able to identify ring-in-ring structures and beware from it if inconvenient. Moreover, the number of ring stars and depots is in motion throughout the optimization process before commuting in the last iteration. Since the test instance capacity restrictions m and q are quite tight, we do not observe omitted depots, although algorithmically feasible.

5.4 Performance analysis

From their results, [2] conclude that class A instances are harder to solve due to the lower ring star customer capacity (30). We agree on this point, arguing

Table 1. Results for instance class A ($q = 30$) and relative improvement (Δ) compared to [2]

Instance	$ U $	$ W $	$ D $	m	obj	Δ
A-1	100	200	2	2	47051	1.87
A-2	100	200	2	2	42185	4.15
A-3	100	200	2	2	40115	0.87
A-10	100	250	2	2	41797	0.61
A-11	100	250	2	2	43800	0.56
A-12	100	250	2	2	42120	1.44
A-28	150	300	2	4	54302	-0.21
A-29	150	300	2	4	54540	1.36
A-30	150	300	2	4	53513	1.73
A-37	150	375	4	2	49580	8.01
A-38	150	375	4	2	48518	7.89
A-39	150	375	4	2	47878	5,37
A-55	200	400	2	4	66569	1.88
A-56	200	400	2	4	63215	2.60
A-57	200	400	2	4	65426	-0,93
A-58	200	400	5	2	62155	4,56
A-59	200	400	5	2	62839	6.06
A-60	200	400	5	2	60863	4,43
A-64	200	400	2	4	70434	0,46
A-65	200	400	2	4	64362	1.75
A-66	200	400	2	4	64558	0,24
A-67	200	500	4	2	62417	2,65
A-68	200	500	4	2	58520	3,09
A-69	200	500	4	2	67930	0,35
A-70	200	500	3	3	60328	2,00
A-71	200	500	3	3	60856	4.48
A-72	200	500	3	3	64508	1,02
A-79	200	500	3	3	64374	0,00
A-80	200	500	3	3	59534	-0,72
A-81	200	500	3	3	60452	2,85
A-82	200	500	2	4	69623	0.86
A-83	200	500	2	4	68559	-0,52
A-84	200	500	2	4	64874	-0,10
A-85	250	500	5	2	67004	2.24
A-86	250	500	5	2	76053	-1,15
A-87	250	500	5	2	66997	3,64
A-91	250	500	4	3	68793	1,01
A-92	250	500	4	3	66586	4.70
A-93	250	500	4	3	70588	5,14
A-94	250	500	3	4	69220	4,56
A-95	250	500	3	4	81000	1.08
A-96	250	500	3	4	69019	3,04
A-97	250	625	5	2	67496	5,86
A-98	250	625	5	2	66055	-1,10
A-99	250	625	5	2	65457	6.18
A-106	250	625	3	4	76479	2,23
A-107	250	625	3	4	72660	0,35
A-108	250	625	3	4	70744	1.55

Table 2. Results for instance class B ($q = 40$) and relative improvement (Δ) compared to [2]

Instance	$ U $	$ W $	$ D $	m	obj	Δ
B-1	100	200	2	2	43470	0.39
B-2	100	200	2	2	40462	2.82
B-3	100	200	2	2	39923	4.24
B-10	100	250	2	2	41287	1.86
B-11	100	250	2	2	41348	2.03
B-12	100	250	2	2	41762	0.79
B-19	150	300	2	2	51536	0.16
B-20	150	300	2	2	51871	1.97
B-21	150	300	2	2	53041	0.78
B-28	150	300	2	3	50859	2.01
B-29	150	300	2	3	49960	3.17
B-30	150	300	2	3	50795	3.74
B-37	150	375	2	3	51008	4.46
B-38	150	375	2	3	52529	2.56
B-39	150	375	2	3	53074	0.80
B-46	200	400	2	3	60996	3.82
B-47	200	400	2	3	62090	1.74
B-48	200	400	2	3	57822	0.03
B-55	200	400	2	4	66626	4.17
B-56	200	400	2	4	63585	6.36
B-57	200	400	2	4	61888	4.81
B-64	200	500	2	3	63006	1.56
B-65	200	500	2	3	61265	1.86
B-66	200	500	2	3	61666	1.58
B-67	200	500	4	2	59712	7.08
B-68	200	500	4	2	59410	7.27
B-69	200	500	4	2	59065	2.43
B-76	200	500	2	4	63290	2.64
B-77	200	500	2	4	66853	0.77
B-78	200	500	2	4	61833	2.40
B-82	250	500	2	4	73260	3.16
B-83	250	500	2	4	69885	3.24
B-84	250	500	2	4	65924	3.77
B-91	250	500	2	4	68694	4.41
B-92	250	500	2	4	72748	2.46
B-93	250	500	2	4	68166	3.44
B-94	250	625	4	2	66266	6.24
B-95	250	625	4	2	61705	2.80
B-96	250	625	4	2	65863	3.96
B-103	250	625	3	3	68965	2.81
B-104	250	625	3	3	66054	2.49
B-105	250	625	3	3	67538	3.06

that it is harder to improve their results by our approach on average. However, we do not exclude that their results could have been quite good and therefore closer to ours.

In the following we try to give an explanation for the obtained solution improvements. The constraints leading to the hardness of the MDRSP are basically the customer limits q_k and ring star limits m_k for each depot k . Therefore, finding a good ring star partition is the hard task. The construction of the ring stars can be handled efficiently by our exact local searches or known routing-based algorithms from the literature. Changing the ring star partition is commonly done heuristically by *swap* move variants that perform node exchanges between ring stars. After a possible heuristic ring star improvement phase the overall solution is evaluated and compared to the incumbent.

In our approach we follow a more general idea that allows multiple node exchanges between multiple ring stars and simultaneously considers customer assignments. Certainly, we are not able to explore that many neighborhoods but benefit from the sophisticated neighborhood structure. Additionally, we explore the obtained subproblems in depth which further increases the improvement potential. Our studies with decreased subproblem sizes result in inferior solution quality emphasizing the importance of the ILP solver efficiency. Achieving perturbation through more extensive changes of the ring star partition is essential in our algorithm.

The proposed contraction techniques turn out to be very effective. However, to find improving global structural changes the subsequent local improvement phase is required. After either dropping the latter or even skipping the contraction steps we observe a substantial loss of solution quality. The defocus procedures certainly weaken the impact of the starting solution since they enable a global repartitioning. We like to point out that our techniques differ from typical perturbation steps used in large neighborhood searches since no randomness applies. In our opinion a random impact would be inconvenient in this context because of the rather expensive exact subproblem solution.

On the one hand the overall success of our method is based on our modeling techniques that allow the application of the exact method to convenient substructures. On the other hand we rely on the careful selection of neighborhoods. We are strongly dependent on minimizing redundancy and avoiding pointless neighborhood explorations. The elaborated greedy tabu-based selection schemes seem to meet these requirements.

6 Conclusions

In this work we address the multi-depot ring star problem, a useful model in practical network design and routing based transportation. A hybrid algorithm based on local refinements using exact methods is presented. For this purpose we worked out a branch & cut algorithm for the MDRSP. The arising local refinement problems were modeled as instances of the MDRSP again. Furthermore, to enable global restructuring of the network we developed contraction

Fig. 11. Solution for instance B-100 ($|U| = 250$, $m = 2$, $q = 40$) using five depots

based techniques. Again, the corresponding optimization is done by the application of exact methods. Therefore, we solved vehicle routing problem variants to optimality. The latter techniques are combined to a construction heuristic for our starting solution. To our knowledge the combination of contraction- and extraction-based subproblems for ILP refinement has not been considered before. Our overall strategy yields an efficient heuristic delivering high quality solutions. This is shown by a computational study on literature instances. More than 90% of the existing results are improved by our approach. Using appropriate models for the local subproblems for which advanced exact methods are available seems necessary for a successful application. Although technical work has to be done for the neighborhood definition, the implementation of a competing heuristic search will not be noteworthy less costly. Like in our case a benefit can be gained from the achievements in Integer Linear Programming. We suppose that this solution framework is more suitable the more complex the problem solution structure is. The combination of rings, assignments, multiple depots and capacities seems promising. We observed the possibility of increasing the solution quality by allowing bigger subproblems that are solved to optimality. This parameterizability needs careful fine-tuning but enables some sort of adjustment of the solution quality when using variable computing power. By designing additional neighborhoods the procedure could possibly be further improved. Advanced cluster techniques as basis for contractions or inserting an improved MDRSP optimization module could also result in an enhanced overall performance. A challenging approach could be the nested parameterized application of our method. On the first level, relatively big subproblems could be allowed that are solved by our proposed heuristical method replacing the exact solver.

Acknowledgments

We are indebted to Roberto Baldacci for providing the test instances, detailed results and the correspondence.

Bibliography

- [1] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24:101–116, 2012.
- [2] 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.
- [3] R. Baldacci, M. Dell’Amico, and J. J. Salazar González. The capacitated m-ring-star problem. *Operations Research*, 55(6):1147–1162, 2007.
- [4] A. Fink, G. Schneidereit, and S. Voß. Solving general ring network design problems by meta-heuristics. In M. Laguna and J. L. Gonzalez Velarde, editors, *Computing Tools for Modeling, Optimization and Simulation*, pages 91–113. Kluwer, Boston, 2000.
- [5] R. De Franceschi, M. Fischetti, and P. Toth. A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming B*, 105(2-3):471–499, 2006.
- [6] E. A. Hoshino and C. C. de Souza. A branch-and-cut-and-price approach for the capacitated m-ring-star problem. *Electronic Notes in Discrete Mathematics*, 35:103–108, 2009. LAGOS’09 - V Latin-American Algorithms, Graphs and Optimization Symposium.
- [7] M. Labbé, G. Laporte, I. R. Martín, and J. J. Salazar González. The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, 43(3):177–189, 2004.
- [8] G. Laporte and I. R. Martín. Locating a cycle in a transportation or a telecommunications network. *Networks*, 50(1):92–108, August 2007.
- [9] V. Maniezzo, T. Stützle, and S. Voß, editors. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer, Berlin, 2009.
- [10] A. Mauttone, S. Nesmachnow, A. Olivera, and F. R. Amoza. Solving a ring star problem generalization. In *Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation*, CIMCA ’08, pages 981–986, Washington, DC, USA, 2008. IEEE Computer Society.
- [11] Z. Naji-Azimi, M. Salari, and P. Toth. A heuristic procedure for the capacitated m-ring-star problem. *European Journal of Operational Research*, 207(3):1227–1234, 2010.
- [12] M. Salari, P. Toth, and A. Tramontani. An ILP improvement procedure for the open vehicle routing problem. *Computers & Operations Research*, 37(12):2106–2120, 2010.
- [13] V. I. Sarvanov and N. N. Doroshko. The approximate solution of the travelling salesman problem by a local algorithm with scanning neighborhoods of factorial cardinality in cubic time. *Softw.: Algorithms and Progr., Math. Inst. of the Belarus. Acad. of Sci., Minsk*, 31:11–13, 1982.

- [14] P. Toth and A. Tramontani. An integer linear programming local search for capacitated vehicle routing problems. *Operations Research/Computer Science Interfaces Series*, 43(2-3):275–295, 2008.