Models and Algorithms for Extended Network Design

Kumulative Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaften
am Institut für Wirtschaftsinformatik
an der Fakultät für Betriebswirtschaft
der Universität Hamburg
(Doctor Rerum Politicarum)

Cumulative Dissertation for the degree of
Doctor of Philosophy
Institute of Information Systems
Faculty of Business Administration
University of Hamburg

Alessandro Hill
Februar 2016
Hamburg, *Germany*

Annahmevermerk
Acceptance

Vorsitzender: *Chairman*	Prof. Dr. Malte Fliedner
Erstgutachter: *Referee #1*	Prof. Dr. Stefan Voß
Zweitgutachter: *Referee #2*	Prof. Dr. Luís Gouveia
Drittgutachter: *Referee #3*	Prof. Dr. Knut Haase
Datum der Disputation: *Defense date*	13.04.2016

Zusammenfassung

zur Dissertation

„Models and Algorithms for Extended Network Design“

von Alessandro Hill

In dieser Dissertation werden Modelle und Methoden für die Planung kostenoptimierter Netzwerke studiert. Es werden Strukturen aus der Klasse sogenannter erweiterter Netzwerke betrachtet. Diese zweistufigen zentralisierten Netzwerke vereinen zwei unterschiedliche Netzwerktopologien und ermöglichen somit integrierte Entscheidungsfindung. Zusammenfassend werden folgende Ergebnisse erarbeitet.

Einerseits wird in dieser Doktorarbeit eine neuer Typ von Netzwerkplanungsmodellen entwickelt, sogenannte Ring-Baum-Probleme. Diese verallgemeinern grundlegend unterschiedliche Steinerbaum- und Tourenplanungsprobleme unter Berücksichtigung von Kapazitätsrestriktionen. Hierdurch lassen sich simultan zentrale Kreisstrukturen sowie angrenzende Baumstrukturen kostenoptimal planen. Zudem zeigt sich, dass diese Modelle ein hohes Potential für die Ableitung von Modellvarianten und deren Einsatz in diversen Anwendungsgebieten bergen. Die in der Arbeit entwickelten exakten und heuristischen Lösungsverfahren sind hochkompetitiv und entsprechen dem aktuellen Stand entsprechender Forschungsarbeit.

Andererseits werden innovative Methoden auf Grundlage mathematischer Optimierung vorgestellt und es wird gezeigt, dass diese zum Lösen der betrachteten komplexen Modelle geeignet sind. Die konsequente Einbettung von Techniken der mathematischen Programmierung in metaheuristische Ansätze führt zu herausragenden Ergebissen im Vergleich zu den besten bekannten Methoden aus der Fachliteratur. Die vorgestellten Algorithmen vereinen exakte Schnittebenenverfahren mit iterativen lokalen Suchverfahren. Diese Methoden werden ausführlich für die betrachteten Modelle getestet, haben allerdings einen sehr allgemeinen Charakter, sodass die Anwendung auf weitere kombinatorische Optimierungsprobleme nahe liegt.

Contents

Highlights

1. **Exact cutting plane algorithms, metaheuristics, matheuristics and presolving techniques are presented for selected models in extended network design using state-of-the-art methodology.**

2. **A new class of extended network design models is developed. These ring tree models turn out to be suitable for application to various areas and fruitful to derive further related models.**

3. **Novel mathematical programming based heuristic algorithms are elaborated which are shown to be capable of handling the high complexity of the considered models. These techniques are extensively tested and can naturally be adapted for many kinds of problems in combinatorial optimization.**

Abstract

The ubiquitous interest in networks per se stems from their natural appearance in countless situations - real world kind as well as theoretical. Network models can be used to describe infrastructure, processes, and relations in general. Furthermore, the cost-driven evaluation of networks leads to the question of how to efficiently construct networks of high quality in regard to the expenses - either from scratch or by extending existing ones. The design of algorithms for efficiently finding optimal, or near optimal, structures is a field of research motivated by applications in various industries, such as telecommunications, transportation, manufacturing and mining, just to mention a few.
The problems studied in this dissertation, all of them essentially combinatorial, ask for networks of minimal overall cost obtained from the synthesis of the fundamental graph classes of cycles, trees or stars. The resulting integrated decision-making under various capacity side constraints leads to computationally challenging models in discrete optimization.
The contribution of the presented research is twofold. Firstly, new models of practical relevance are developed for which exact and heuristic state-of-the-art algorithms are elaborated. Secondly, innovative matheuristic concepts of generic type are presented and studied on these models. These techniques are able to elegantly push the frontiers of computability to efficiently approximate optimal solutions using mathematical programming.

Keywords: Network design, combinatorial optimization, mathematical programming, heuristics, matheuristics, operations research

1. Introduction

1.1 Research Classification

This collection of cohesive research works covers topics that can be assigned to multiple disciplines. However, Operations Research is probably the most suitable since it essentially embodies optimization while being application oriented. As widely accepted, one of the strengths of operations research as a discipline is its many-sidedness. Although hereby hard to properly define, it substantially overlaps with mathematics, computer science, industrial engineering and management science.
Figure 1.1 shows the overlappings of this dissertation with the most relevant fields: Network design, combinatorial optimization, mathematical programming and heuristics. Along the way we borrow concepts from graph theory, data structures, algorithm engineering, strategic planning and artificial intelligence to model and solve problems motivated by real world applications. More specifically, the predominant subject of this thesis is network

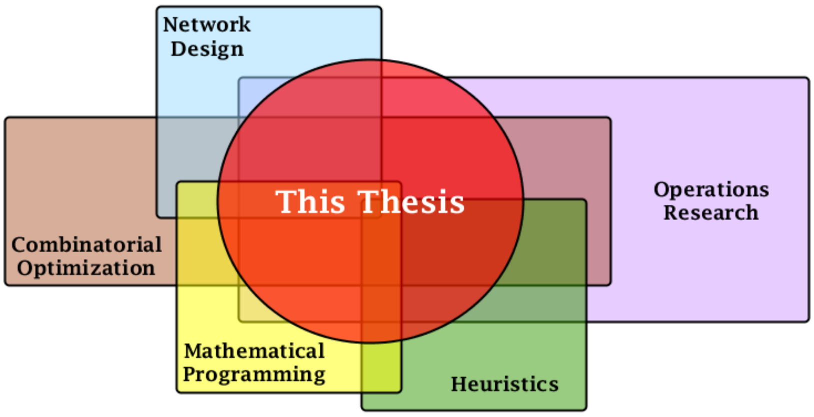


Figure 1.1: An illustration of this work's overlapping fields of research.

optimization. The three studied main models are predominantly related to their applicability in telecommunications [6, 5, 13, 30]. The developed optimization techniques are of heuristic and exact nature, but substantially strengthen the cross-linkage between classical heuristic concepts and mathematical programming.

1.2 Research Framework

In the following, the research framework is described with the included scientific works. Additionally, their relationships regarding optimization models and methodology are provided. A detailed summary of the models is given in Section 2 and the developed methodology is reviewed in Chapter 3.

Scientific papers

The papers that are part of this dissertation are listed in Table 1.1[1]. Figure 1.2 shows the embedding of the included works in both, the related network design models and the used methodology.

i	A. Hill, I. Lubić	A GRASP Algorithm for the Connected Facility Location Problem, IEEE CS Press [28]	2008
ii	A. Hill	Novel Presolving Techniques for the Connected Facility Location Problem, IEEE CS Press [27]	2012
iii	A. Hill	Modeling Techniques in Ring and Tree Bases Network Design, IEEE CS Press [26]	2012
iv	A. Hill, S. Voß	An Equi-Model Matheuristic for the Multi-Depot Ring Star Problem, *to appear in Networks* [15]	2014
v	A. Hill, S. Voß	Optimal Capacitated Ring Trees, *EURO Journal on Computational Optimization* [17]	2015
vi	A. Hill	Multi-Exchange Neighborhoods for the Capacitated Ring Tree Problem, LNCS [14]	2015
vii	A. Hill, S. Voß	Generalized Local Branching Heuristics and the Capacitated Ring Tree Problem, *submitted* [16]	2014

Table 1.1: Papers included in this thesis.

A work that considers a problem variant that is closely related to the ring tree problem studied in [17] and is not included in this thesis can be found in [1].

[1] Papers i.,ii. and iii. were published under the name Alessandro Tomazic.

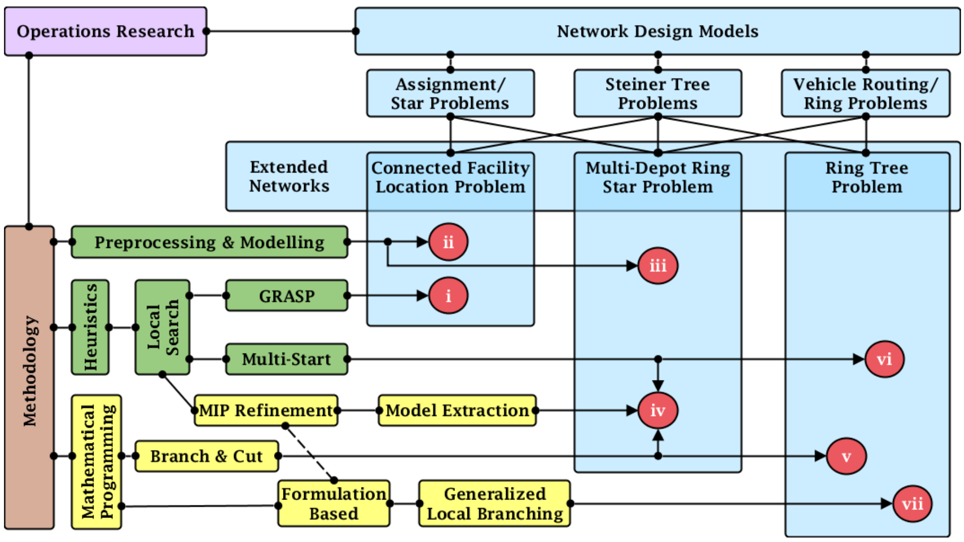


Figure 1.2: An illustration of the research framework.

2. Extended Network Design Models: Combining Stars, Trees and Rings

In this thesis, we consider models from a class of emerging network design problems which are motivated by integrated decision making. These *extended network design models* combine different classical network optimization models within more general ones. As a consequence, the solutions essentially generalize the different network topologies that are associated with the base models. These extended models allow the utilization of a major optimization potential, which arises when considering the base model decisions simultaneously, rather than separately. The overall objective in the models in this dissertation is the minimization of the network cost measured as the sum of the edge costs and eventual facility installation costs.
An overview of the studied extended network design models is given in Section 2.2 after some preliminaries in Section 2.1. Relevant model extensions and applications can be found in Sections 2.3 and 2.4.

2.1 Preliminaries

The considered networks correspond to graphs being the central concept in graph theory. An undirected graph is a pair of a set of nodes, or vertices, and a set of edges, which are unordered pairs of nodes representing links between two distinct nodes. In a directed graph, the node relations are represented by arcs which are ordered pairs of nodes, respectively. In some network design problems, a designated central node, also called *root* or *depot*, is used to mark a special location within the network. For an introduction to network based models and algorithms we refer to [18]. An extensive presentation of network flows, which are important in regard to mathematical model formulations and algorithmic solution approaches, can be found in [3].

2.2 Optimization Models

In this section, the extended network design models that are studied in this thesis are summarized. The basic models that serve as building blocks are presented beforehand. Figure 2.1 highlights the corresponding model dependencies.

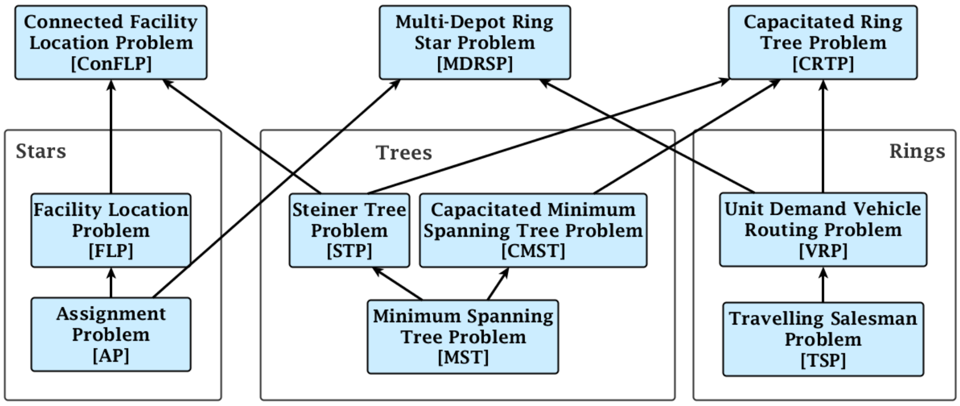


Figure 2.1: Relationships between base models (star, tree and ring topologies) and extended models (top layer).

2.2.1 Base Models

The following base models and their corresponding network topologies play central roles in the subsequent extended problems.

Tree based

A prominent tree based network design model is the Steiner tree problem (STP) [30]. The STP asks for an edge cost minimal tree that connects given *terminal nodes*, or *customers*. Nodes from a set of *Steiner nodes* can optionally be chosen to be part of the tree if this results in an overall cost reduction. The specification of a root, from which each terminal can be reached on a directed path, is necessary for the directed STP. If no Steiner nodes are present, the NP-hard STP reduces to the minimum spanning tree problem (MST) which can be solved in polynomial time.

Ring based

Vehicle routing problems in its many variations have been widely studied. A resulting solution network is the intersection of cycles, also called *tours*, *routes* or *rings*, in a designated depot node [12]. As in the STP, Steiner nodes might be available. If we restrict the number of tours to one, we obtain the prominent NP-hard traveling salesman problem (TSP) [4]. The classical problem of finding such a single tour within an arbitrary graph is known as the Hamiltonian cycle problem which is NP-hard as well.

Star based

A star can be described as a connected graph in which all but one node, the center, are required to have degree one. Every star is a tree and can be used to model assignments of the leave nodes to the center, among others. Corresponding optimization problems are known as assignment problems, warehouse/facility location problems, bin packing problems and knapsack problems [20].

2.2.2 Trees and Stars

The terminals in the STP commonly represent potential *hub* locations from which *end customers* are served. Such an end customer could be assigned to different hubs, or even to a Steiner node which serves as a hub. Conversely, a terminal might not even be needed if no end customers are assigned to it. Finally, corresponding facility costs incur for the hubs only. Instead of solving a STP and a facility location problem separately, the connected facility location problem (ConFLP) [19] incorporates both and asks for an overall cost-optimal layout of a tree connecting selected hubs while assigning specific end customers to these hubs. A solution network consisting of 14 assignment stars, the installed hub facilities and the core Steiner tree is illustrated in Figure 2.2.

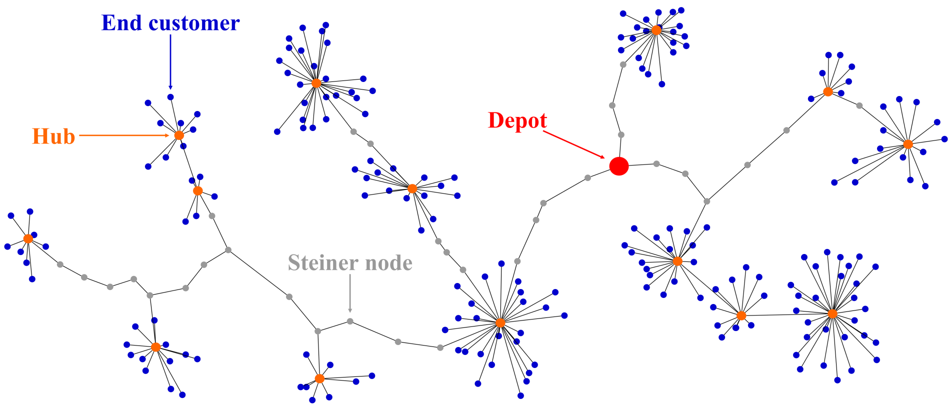


Figure 2.2: A solution for the ConFLP: A tree and star based network with customers assigned to facilities installed at hub nodes which are connected to a depot by a Steiner tree.

2.2.3 Rings and Stars

In some applications, such as transportation or reliable telecommunication networks, the core structure that interconnects the hubs and the depot in the ConFLP is naturally cycle-based. Multiple cycles (or rings) can be installed to either contain customer nodes directly, or have them assigned to its nodes to turn them into hubs. The capacitated ring star problem (CRSP) is an extended network design model that incorporates ring and star structures in this way. Upper bounds on the number of ring star structures attached to the depot as well as the number of customers served by each of them make the problems especially challenging. Figure 2.3 shows such a (single depot) ring star network that satisfies the requirements for a telecommunication network in Northern Italy [6]. The more general multi-depot ring star

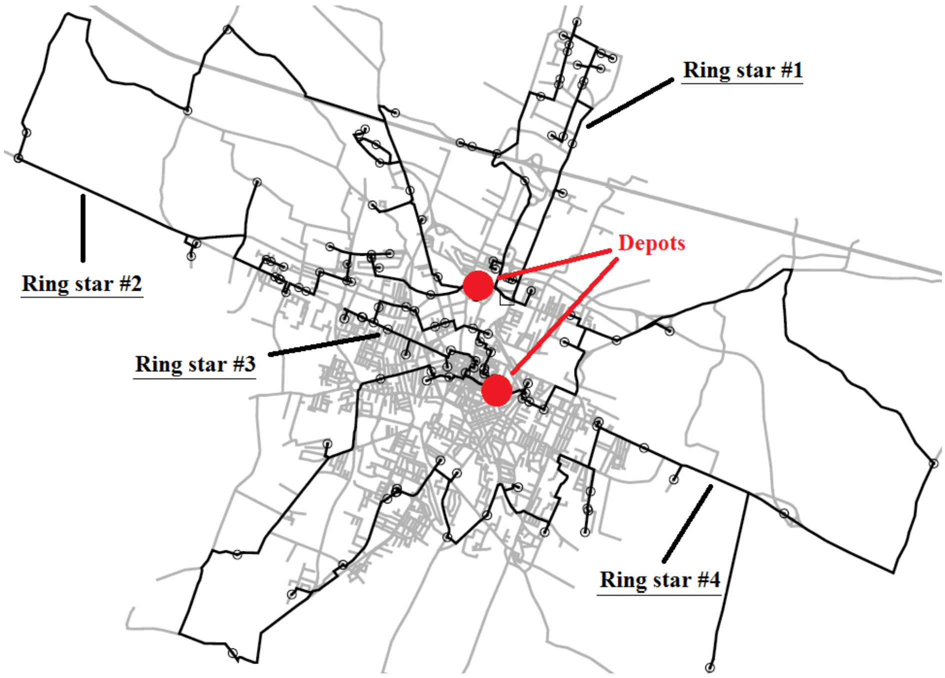


Figure 2.3: An implementation of an optimal ring star based optical network in a provincial town in Northern Italy (CRSP; from [6]).

problem [5] even allows ring star structures originating from different given depot locations. In [15] we intrduced depot dependent capacities which were useful for modeling. Two high quality solutions of diverse structure for a literature instance - obtained by algorithms developed in this thesis [15] - are given in Figure 2.4. The network implements 16 ring stars connected to 8 depots. Baldacci et al. [5] computed an upper bound of 75426 whereas we found 68566 (left) and 67792 (right) which corresponds to cost savings of about 10%.

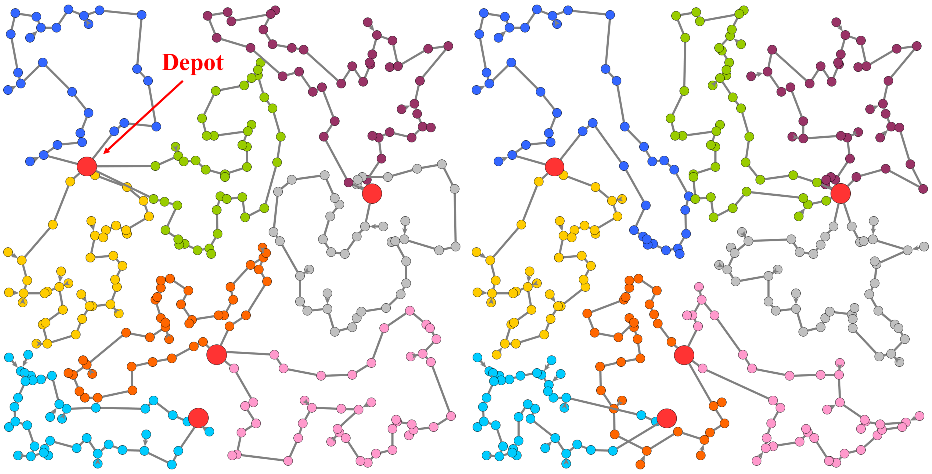


Figure 2.4: Two locally optimal MDRSP solutions for instance B-119 found by our method [15].

2.2.4 Rings and Trees

Consider a ring star model from above. When used on a strategic level, the model customers that are allowed to be assigned to ring structures represent a bundle of physical consumers. These could for instance be several buildings that need to be connected to an infrastructure and share a certain proximity to each other. This grouping is usually done in a preprocessing phase to obtain a model of tractable size and the assignment costs are estimated according to the consumer locations. Once a convenient ring star layout is found the end customer groups are expanded and interconnected according to the initial estimation - using a tree structure.
In the spirit of the full optimization potential utilization in extended network design models, we develop a model that plans the network structure up to the end customers. In other words, we consider the assignment of Steiner trees to rings, leading to so-called ring trees, under similar capacity constraints as in the CRSP in Section 2.2.3. The installation of rings, which are more cost intensive than trees, is motivated by selected nodes that require ring-connectivity. This capacitated ring tree problem turns out to elegantly generalize several prominent existing network design problems. On the one hand (capacitated) spanning trees and Steiner trees are captured as special cases. On the other hand, the underlying ring structures can model vehicle routing problems and, therefore, the traveling salesman problem. A solution for a medium sized instance Q-217 from [16] implementing four ring trees is given in Figure 2.5. Figure 2.6 shows optimal capacitated ring trees for instance Q-1 from [17] while increasing the number of customers that need to be on ring structures, called type 2 customers. Clearly, the cost of an optimal network is non-decreasing when incrementally augmenting the

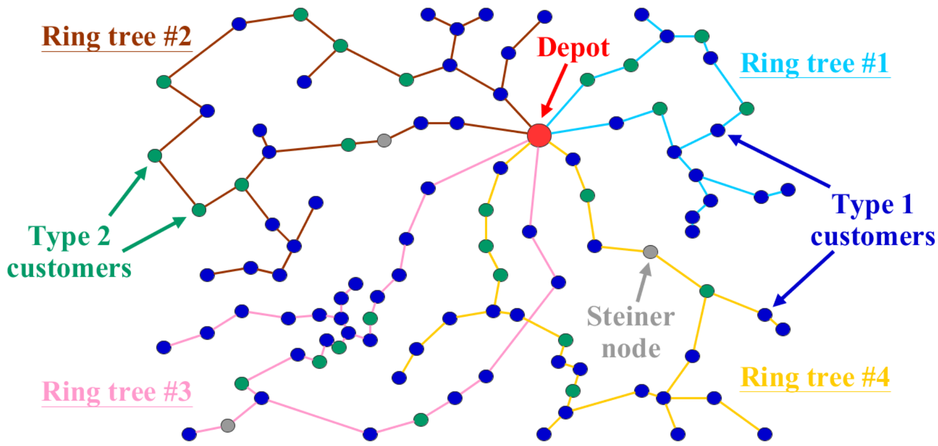


Figure 2.5: A ring-tree based solution network (instance Q-217 from [16]).

number of type 2 customers. The objective values for the optimal solutions in Figure 2.6 are 157, 210, 227, 236 and 242, respectively.
In this thesis, we extensively study this capacitated ring tree problem and present several algorithms to solve it. Certainly, its general nature does make it significantly harder than its mentioned special cases. At the same time, this fact gives rise to new algorithmic challenges. Nevertheless, the increased complexity triggers parts of our novel MIP based heuristic strategies in Section 3 which turn out to be highly effective for the CRTP.

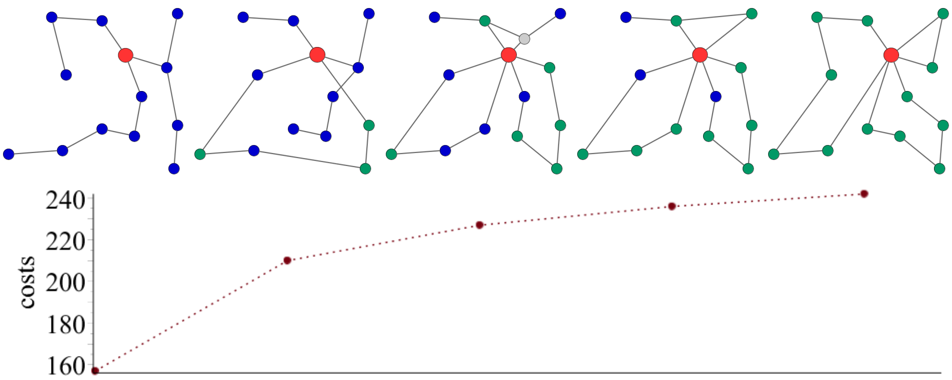


Figure 2.6: Optimal CRTP solutions for instance Q-1 and increasing type 2 customer rates: 0%, 25%, 50%, 75%, 100% (top); corresponding increasing network costs (bottom). [17].

2.3 Model Features and Extensions

Typically, network design problems incorporate multiple application-oriented requirements as constraints. In the following we highlight a relevant selection applied to the models considered in this thesis.

- *Connectivity*: We consider the connectivity r between two nodes as the number of node, or edge, disjoint paths in the network. For sets of more than two nodes this property holds if pairwise valid. Trees provide 1-connectivity by definition and appear in the ConFLP and the CRTP [28, 17]. The special case of stars can be found in [28, 15]. Cycle (sub-)structures naturally provide twice the protection compared to trees when assuming a single edge (or node) failure. Moreover, this reliability can be implicitly augmented by reducing the allowed cycle length; see also capacity bounds below. Survival network design models [25] provide more general connectivity by requiring a minimum connectivity between two network nodes, i.e. the number of disjoint paths between two nodes is greater or equal than r, without imposing structural requirements. We do not consider such networks in this thesis.

- *Oriented networks*: Depending on the application of a model, it can be necessary or convenient to use a directed network representation. This leads to the replacement of edges by arcs which allows the usage of an asymmetric cost function. The latter is commonly required in VRPs. Furthermore, the modeling becomes more powerful and computational benefits might be expected through such directed representations; see, e.g. [17], for some mathematical formulations.

- *Multiple node types*: Differentiating between customer node types in the network is important to specify eventual node-dependent requirements, e.g. reliability in the CRTP. In some models, such as the facility location problem, the roles of the customer nodes are strictly preset regarding the network structure. Others allow various node configurations, e.g. customers that can be ring nodes or they can be assigned to rings in the MDRSP.

- *Capacity bounds*: Limiting the number of nodes or customers of a certain type within specific substructures of the network is commonly needed to reflect technologically or economically motivated limits in practice. In the CRTP a restricted number of customers can be located on each of the ring trees. Moreover, the number of the ring trees is bounded. Similarly, the number of ring stars and the number of corresponding customers is bounded. In [15], we introduce the depot dependent version of these constraints for the MDRSP. Note that these capacity constraints in particular apply to the structures of the base models, e.g. rings and trees.

- *Steiner nodes*: These optional network nodes are frequently indispensable to model necessary intermediate points within the represented infrastructure. A simple example that typically leads to the introduction of a Steiner node is a street intersection at which the corresponding infrastructure can branch into some of the incident streets. It is well-known that Steiner nodes significantly increase the model complexity. The NP-hard STP even turns into the polynomially solvable MST when the instance does not include Steiner nodes at all.

- *Multi-depot*: A natural generalization of single-depot models are multi-depot models. For the TSP this can be interpreted as the (partial) coverage of given customers by multiple salesmen that are located at different home bases. These can rarely be considered independently since the assignment of the customers to the salesmen is commonly an integrated part of the problem. However, the STP, and therewith the ConFLP, admits the modeling of this feature by the introduction of an artificial depot terminal.

- *Facility location*: Facility location is a basic concept to model required additional costs when using specific network nodes. Besides physical devices, e.g. repeater or splitter in telecommunication networks, it can represent costs for servicing assigned customers. In the ConFLP these opening costs appear for potential facility nodes that have customers assigned. For the MDRSP and the CRTP these node costs can be integrated with minor effort into the mathematical formulations of the models.

- *Variable edge costs*: The cost of an edge or arc might be dependent on the sub-topology it is used for, e.g. tree edge vs. ring edge. The ConFLP separates the edges by definition, whereas in the MDRSP and the CRTP this feature does not require model changes for its integration as addressed in [6] and [17].

We further mention diameter constraints and hop constraints, node degree constraints, customer profits, customer demands and supply capacities which are common in related models. They could be integrated into the studied models in future research.

2.4 Applications

In the following, we give a brief overview of relevant fields of application for the considered optimization models. The main area of relevance for this work is strategic planning in telecommunications. Secondly, transportation problems are closely related to the considered models, but preferably in their directed variants. We note that the algorithms and techniques developed in this thesis are of generic character and can be transferred to a broad class of not necessarily related optimization problems. Furthermore, we note that the models studied in this work are recent (MDRSP) or new (CRTP) and that we are confident that their usefulness in relevant disciplines will be appreciated.

- *Telecommunications*: The design of telecommunication networks is a fruitful field for the described network optimization models [30, 13, 6]. For instance, the implementation of fiber-based technology which started about a decade ago gave rise to many research topics. Last-mile network architectures such as fiber-to-the-Curb (FTTC), fiber-to-the-home (FTTH) or more generally fiber-to-the-x (FTTX) lead to diverse optimization models depending on cost structures and additional technological or economical requirements. The extended network design models studied in this thesis are suitable to model multi-layer network planning which integrate the layout of reliable backbone networks.

- *Transportation*: Numerous vehicle routing applications can be found in the literature. The presented extended models are highly relevant for integrated modeling. Ring star models have been used to model school bus routing including the bus stop walking distances for the children [23]. An unexplored application of the ring tree models could address the integrated planning of ship routes and hinterland infrastructure of harbors.

- *Chip design*: In very large scale integration (VLSI), the terminals in the STP represent transistors that need to be connected on a chip. Besides TSP variants, variations of the STP such as the rectilinear STP or the group STP are commonly used to solve chip design problems. The layout of circuits is also fundamental in these applications. To our best knowledge, corresponding applications of the CRTP or variants have not been considered in the literature.

- *Logistics*: Logistic supply chains correspond to connected networks of various structure. Trees, stars and rings often represent sub-processes within larger systems. An example for a star based optimization problem is the warehouse location problem which can be summarized as follows. For a given set of customers we ask for the locations of facilities to which the customers will be assigned such that the assignments costs plus the location dependent facility installation costs are minimal [3].

- *Manufacturing*: It is known that production planning problems like lot-sizing problems can be modeled using Steiner trees [29]. There might be directions of research that are not sufficiently explored within the academic literature.

- *Personnel planning*: The planning of personnel typically consists of two components, which are the allocation of personnel to predefined tasks or positions and the scheduling of jobs over a time horizon. Allocating personnel to tasks corresponds to the construction of star structures and these tasks are commonly required to be performed according to some precedence constraints which are defined by a directed tree structure [3].

- *Mining*: The planning of mining leads to strategies to successively explore areas for resources or/and to mine them. The precedence constraints for operations constitute a network that may be optimized in regard to profit, time or expenses. Here, we see a major potential of extending the successful application of the base models to the presented extended models.

- *Biochemistry*: Steiner trees can be used to represent large scale biochemical data [7]. Optimization is shown to be useful for fitting and reduction methods (see Section 3.2) are necessary to facilitate an efficient network analysis.

3. Methodology: From Heuristics to Exact Methods and Back

In this thesis we solve network design problems. To do so different types of methodology may be applied. After eventually preprocessing a problem, this can be heuristic methods, exact methods as well as hybridization between the two, nowadays also called matheuristics. After some preliminaries the subsequent subsections are devoted to the different types of methods.

3.1 Preliminaries and Optimization Challenges

Algorithms to solve optimization problems can be classified into *exact methods* and *heuristics*. While exact algorithms terminate with an optimal solution, usually no satisfactory guarantee for the solution quality is provided by a heuristic. Many well-performing exact approaches for NP-hard problems are based on a prior mathematical problem formulation. Our algorithms follow this by using the framework of *Mixed Integer Programs* (MIPs). A MIP is a mathematical description of an optimization model using binary variables, integer variables, continuous variables, an optimization sense, and an objective function and inequalities linking the variables. We restrict ourselves to linear MIPs in which the objective function and the constraints describing inequalities are linear functions in the variables. MIPs serve as an interface to general purpose MIP solvers as well as a precise model description format. An introduction to the corresponding discipline of (mixed) integer programming can be found in [31]. Asurvey on metaheuristics is given, e.g., in [9].

The extended network design problems considered in this work are NP-hard since they reduce to the mentioned hard base models (see Section 2). A main aspect of this thesis is the study of their computational complexity using state-of-the-art algorithms. We think that the following characteristics are mainly responsible for the observed hardness.

- *Combined problem structures*: The extended network design models integrate decisions of different types into one optimization model. As a consequence, the complexity of the solution structures increases. Problem specific heuristics suffer from an increased number of network configurations that need to be considered. For local search algorithms, this effects neighborhoods of augmented complexity (see Section 3.4). The formulations used in exact mathematical programming based algorithms become larger and need a careful definition of the problem characteristics, e.g. for proving optimality. In Section 3.6 it is shown that generic heuristic MIP-based approaches can help to efficiently overcome this issue.

- *Model side constraints:* The integration of the model features listed in Section 2.3 into optimization problems notably increase the difficulty of solving them. In particular, Steiner nodes and capacity constraints are known to reduce the size of instances that can be solved to optimality and that they complicate heuristic search.

Ways to reduce complexity and size of an instance, so-called preprocessing or reduction techniques, are addressed in Section 3.2 whereas Sections 3.3, 3.4, 3.5 and 3.6 deal with concrete solution techniques.

3.2 Reduction Techniques

The formulation of the underlying practical problem in terms of an optimization model can cause significant increase of the problem size. Especially for large-scale instances of network design problems, the reduction of the input information, e.g. edges and nodes, can be necessary for the application of suitable solution methods. These *preprocessing* techniques aim at detecting relationships that lead to beforehand eliminations of input decision variables.

Certainly, these procedures cannot resolve the overall problem unless it is easy, or the preprocessing effort is sufficiently high. With an increasing effort for detecting these eliminations that can be tracked back, the techniques typically get more powerful. Therefore, the term *presolving* is sometimes used in the literature. For very large scale instances, preprocessing in fact serves as a first part of a heuristic approach. In [27], we develop such extended techniques for the ConFLP, which generalize known procedures for the STP [10]. Advanced tests involving shortest path and minimal cut computations are provided as well. To avoid redundant searches, we analyze the complex interactions, before embedding them into a tabu search guided framework. Finally, the techniques presented in [27] are shown to reduce the size of the input node set by up to 36%, and the edge set by up to 86%, respectively, for the considered set of test instances.

3.3 Constructive Approaches

Regarding the considered models it is easy to find a network that satisfies the stated requirements, so that feasibility is not an issue, albeit, a naively generated solution can be of exorbitant cost. In many cases heuristics are conceptually a composition of improvement strategies which are initialized by such a start solution. This initial start solution usually has a significant impact on the effectiveness and the behavior of subsequent optimization strategies.

3.4 Polynomial Local Search

An important ingredient for effective local search algorithms is the definition of appropriate *neighborhoods*. A neighborhood with respect to an existing reference solution can be defined as a set of solutions. Typically, they are related to the reference solution by some problem specific distance measure or a transformation technique. In local search, these solutions are dynamically constructed and compared to the reference solution regarding their costs. If an improving solution is found, then this solution is considered to replace the incumbent solution before the search continues according to a high level strategy. The developed MIP based refinement techniques in Section 3.6 are also based on this generic concept.

The benefit of the exploration of a neighborhood is typically problem dependent and, unfortunately, also depends on the considered instance class. To efficiently explore such a partial solution space, two concepts are commonly used. Either the neighborhood is completely examined to improve solutions with respect to the objective function (I) [24], or it is partially traversed by some search heuristic (II) [2]. We say that either technique is a polynomial local search if the used exploration algorithm has polynomial time complexity.

Exact local search (I) is usually suitable for relatively simple neighborhoods, e.g. shift, swap, re-assign moves, and collapses to polynomial local search. Neighborhoods of advanced structure require more complex exploration techniques (e.g. MST [14]) and are thus sensitive to the instance size. The MIP techniques introduced in Section 3.6 extend this approach to neighborhoods of arbitrary structure and time complexity.

Techniques based on (II) allow the definition of larger neighborhoods which, depending on the used heuristic, might lead to larger improvements. It is up to the exploration heuristic to efficiently identify these in turn. For the models considered in this work, it is shown that sub-routines of type (I) and (II) lead to powerful tools [14] to design iterative improvement algorithms. In order to escape local optimality, we utilize structural and random based multi-start approaches [28, 15, 14] and support these by problem-tailored (multi-exchange) search operators. These techniques provide decent solutions, even for large instances. However, the mentioned neighborhood structures arising in the extended network design models are hard to capture since they quickly become complex due to the network topology and the additional model features; see Section 3.1. This motivates the development of the subsequent generic algorithms which lead to further improved results.

3.5 Exact Mixed Integer Programming Based Algorithms

Many state-of-the-art algorithms that solve NP-hard optimization problems to optimality are based on branch and bound with integrated linear programming techniques. This thesis contains two such approaches which are developed for the MDRSP [15] and the CRTP [17]. The presented exact approach for the MDRSP is based on the three index arc formulation suggested in [5]. We develop the first exact algorithm which incorporates additional cutting planes to increase its efficiency. It is known that multi-depot models are significantly harder than their single-depot counterparts. Nevertheless, we are able to solve most instances with up to 50 customers, which is enough to use the algorithm within our heuristic framework presented in [15].

For the CRTP, we present the first MIP formulation in [17]. The formulation combines a non-compact arc based STP formulation with a circulation flow model. Among others, we elaborate sophisticated valid multi-star like inequalities. Furthermore, heuristic solution techniques [14] are used to compute initial primal bounds and to polish feasible intermediate solutions. This leads to a highly efficient exact method for the CRTP. Purely tree structured problems, or capacitated Steiner tree problems, up to 100 nodes are solved within a few seconds; see [17]. Optimal purely ring-based solutions and ring tree structures are computed in most cases for instances with up to 50 customers, using a 1 hour time limit. A TSPlib based instance set of 225 instances [17] is derived to study lower bounds and corresponding optimality gaps if optimality could not be proved. Furthermore, the impact of the distribution of the given customer types is investigated in [17]. It turns out that CRTP instances with balanced numbers of randomly distributed customers of the two types are the hardest to solve.

3.6 Mixed Integer Programming Refinement Methods

The search for improving neighboring solutions using mathematical programming techniques primarily to completely explore a neighborhood deserves special attention. The approaches developed in this thesis operate within structured frameworks to navigate through the search space in which the actual exploration of the considered neighborhoods is carried out by a MIP solver. The following two aspects need to be mentioned in this context.

1. No local search algorithm that is tailored to a specific neighborhood is required.
2. The performance of the used black-box MIP solver contributes to the overall efficiency of the algorithm.

Only during the last decade the incorporation of mathematical programming based techniques into (meta-)heuristic frameworks gained increased popularity [22]. One reason is the continuous improvement of general purpose MIP solvers. Significant speedups could not just be achieved by the rapidly increasing computing power but also by superior dual and primal bounding techniques as well as the implemented search strategies. Heuristics that embody components borrowed from mathematical optimization found their way into the academic literature as *matheuristics* [22]. This class of methods is a subclass of more generic *hybrid algorithms* which are combinations of different algorithmic approaches [8].

3.6.1 Model Extraction

If the instance of the optimization problem that we intend to solve is small enough then it can certainly be solved to optimality by an exact algorithm (see Section 3.5). For larger instances this is generally not possible in an efficient way. For the models considered in this work an instance including 70-100 nodes may exceed the capabilities of our exact approaches to be solved within reasonable time limits, unless of notably simple structure.

In [15] extraction-based techniques are developed that allow us to apply our exact algorithms restricted to parts of the instance to refine a given solution. If an improving solution could be identified, it is used to replace the current network substructure that is considered in this refinement step. We are able to develop techniques to use the overall model, the MDRSP, to solve these significantly smaller subproblems to optimality. They are constructed according to differently structured problem-tailored node clustering concepts. More specifically, we select sets of network nodes, so-called *clusters*, which induce substructures within the incumbent solution network. Two approaches are used to find these clusters. The first concept identifies node sets based on the edge cost structure only, by minimizing the maximal edge costs to connect two nodes in the set. Apart from these *ball-type* clusters, nodes are selected according to various substructures in the current reference solution, such as paths, pairs of paths or rings. In both cases multi-exchange neighborhoods are captured by considering nodes from multiple ring stars for refinement. Figure 3.1 illustrates some of the used cluster

types used for the MDRSP.
Rather than decomposing an existing solution into such refinement problems, we build them on overlapping node set covers - just as large such that they can still be solved efficiently. The techniques are embedded in an iterative improvement strategy. These approaches are *template based* whereas the approach in Section 3.6.3 generalizes this idea for the CRTP and does not require a specific model for carrying out the MIP refinements.

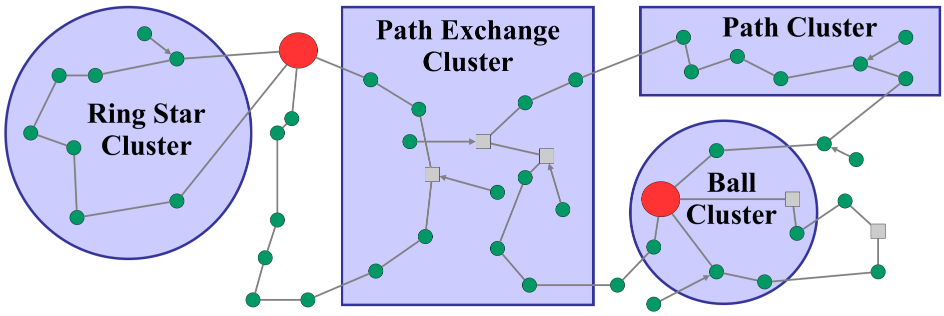


Figure 3.1: Some cluster types used for selecting refinement model nodes in the MDRSP [15].

3.6.2 Contraction Techniques

The concept of local search describes a search procedure that aims at improving a solution by the partial exploration of the complete solution space. In general, this is not sufficient to guarantee global optimality of the resulting improved solution. Therefore, search strategies usually include perturbation mechanisms which are applied whenever no local improvement could be found [21]. Instead of commonly used random strategies, we apply contraction techniques [15]. These are used to identify network substructures that may be ignored when considering the restructuring of the overall network. To do so we first cluster nodes in an existing network to reduce the number of nodes. This is done using different metrics to diversify the set of obtained reduced networks. Afterwards, we apply our exact algorithm on the induced inputs for a suitable optimization model. For instance, applying it to the MDRSP, we are able to use the overall problem to solve the resulting vehicle routing problem variants. Even if no immediate improvement of the reduced structure is observed, we further optimize the best found expanded network before comparing it to the incumbent. It turns out that this technique can also be used to construct promising start solutions, i.e., in the case that no incumbent network exists. By combining these techniques with the ones in Section 3.6.1 the heuristic results reported in [5] could be improved significantly [15]; see also Figure 2.4.

3.6.3 Mathematical Formulation Based Techniques

The methods described in the previous section utilize mathematical programming to solve the extracted refinement models. A disadvantage of these approaches is the need of suitable refinement models and the transformation effort necessary to feed these models. To overcome the latter, one can limit the modeling effort by operating on a selection of *refinement variables* in the overall mathematical model without thinking about an appropriate refinement model. Here, we assume that we have a formulation of the overall problem together with an exact algorithm at hand. In that case, the refinement search can be executed by the modification of this master MIP such that only changes of the refinement variables are taken into consideration during optimization. To forbid changes of selected non-refinement variables within the MIP we apply *variable fixing*. This fixing of variables can be seen as *projection techniques* which reduce the effective dimension of the solution space to the number of refinement variables.

3.6.4 Generalized Local Branching Techniques

The formulation based MIP refinement techniques developed in Section 3.6.3 get even stronger when they are combined with the concept of *local branching* [11]. The latter was originally introduced as a strategy to improve, or *polish*, intermediate solutions found during branch and bound algorithms for general MIPs. Local search is performed with respect to a reference solution by solving the overall MIP with additional Hamming distance constraints. These are enforced by *local branching cuts* which limit the sum of the variable value changes, typically *binary variable flips*, with respect to the reference solution. As for the refinement techniques above, the key idea is to obtain a MIP that is significantly easier to solve than the overall problem, and can still be solved efficiently to optimality to possibly detect a solution of lower cost.

In this thesis, we develop a novel generic algorithmic framework to efficiently compute high quality solutions for extended network design problems [16]. We introduce a new local branching parameter which describes the number of edge variables to which a local branching cut is applied. The remaining variables are fixed to their values in the reference solution and not considered for change. This allows the consideration of a broad class of refinement subproblems. By increasing the latter parameter we can decrease the Hamming bound so that the resulting problems can be solved efficiently. Conversely, we can reduce the set of considered variables to allow a large variable value change in an efficiently solvable subproblem. In [16] we suggest an iterative strategy leading to *generalized local branching* (GLB). This concept generalizes both, the classical local branching as well as the model extraction based refinement techniques in Section 3.6.1. Figure 3.2 illustrates two GLB re-

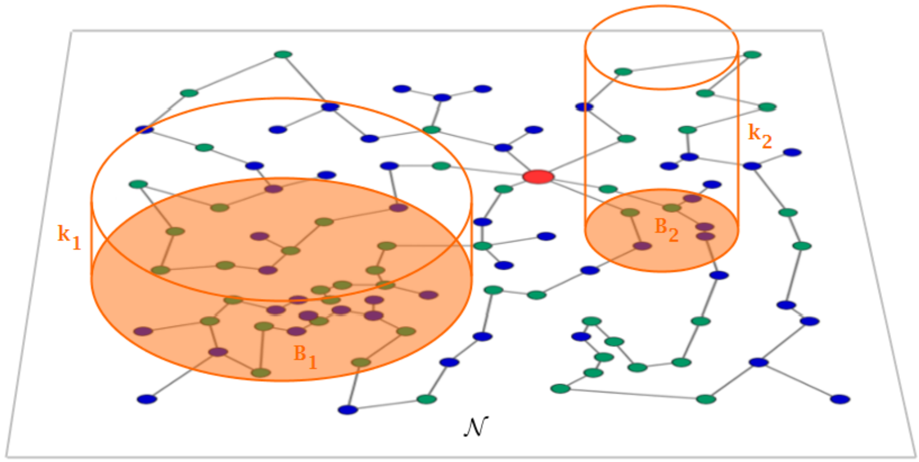


Figure 3.2: Two GLB refinement neighborhoods that differ in the number of variables considered, i.e. $|B_1|$ and $|B_2|$, and the variable value changes allowed, i.e. k_1 and k_2 (from [16]).

finement problems which are represented by cylinders of different volumes. The two GLB parameters, i.e. $(|B_1|, k_1)$ and $(|B_2|, k_2)$, correspond to the diameter and the height of each cylinder.
Using this heuristic framework, we are able to improve 36% of the best known solutions [17], 65% of the solutions obtained by the local search heuristics in [14] and we reduce the optimality gap by 10% on average. We show that the approach clearly outperforms the pure local branching technique and the model extraction based method in Section 3.6.1 regarding the obtained solution quality.

4. Conclusions

This thesis embraces several existing as well as novel models in network design that combine different network topologies for integrated decision making. These extended network design problems are studied and algorithms are developed. We conclude this summary by highlighting two main outcomes of this work.

Firstly, a new class of extended network design models, namely ring tree problems, is developed in this thesis. Hereby, well-known vehicle routing problems and the essentially different Steiner tree problems are generalized under capacity constraints. This advance allows the simultaneous optimization of two-level networks based on core rings and peripheral tree structures. Moreover, these models turn out to be fruitful to derive further models and are suitable for application to various areas. The corresponding elaborations on exact and heuristic algorithms represent a piece of state-of-the-art methodology.

Secondly, novel mathematical programming based algorithms are elaborated which are shown to be capable of handling the high structural complexity of the considered extended network design models. Through the extensive embedding of mathematical programming techniques within metaheuristic frameworks, we obtain outstanding results compared to state-of-the-art literature methods. These methods are based on the hybridization of exact branch and cut algorithms with iterative local search strategies. After the incorporation of the suggested local branching concepts we are able to still ameliorate these results. These techniques are extensively tested for the considered models but can be adapted in a natural way for many kinds of problems in combinatorial optimization.

Bibliography

[1] 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.

[2] R.K. Ahuja, Ö. Ergun, J.B. Orlin, and A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(13):75 – 102, 2002.

[3] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[4] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2007.

[5] 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.

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

[7] N. Betzler. *Steiner Tree Problems in the Analysis of Biological Networks*. Diploma Thesis, University of Tübingen, Germany, 2006.

[8] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135 – 4151, 2011.

[9] M. Caserta and S. Voß. Metaheuristics: Intelligent problem solving. In V. Maniezzo, T. Stützle, and S. Voß, editors, *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, pages 1–38. Springer US, New York, USA, 2009.

[10] C. Duin. Preprocessing the Steiner problem in graphs. In Ding-Zhu Du, J.M. Smith, and J.H. Rubinstein, editors, *Advances in Steiner Trees*,

volume 6 of *Combinatorial Optimization*, pages 175–233. Springer US, New York, USA, 2000.

[11] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.

[12] B. Golden, S. Raghavan, and E. A. Wasil. *The vehicle routing problem: latest advances and new challenges*. Operations Research/Computer Science Interfaces Series, 43. Springer US, New York, USA, 2008.

[13] M. Grötschel, C. Raack, and A. Werner. Towards optimizing the deployment of optical access networks. *EURO Journal on Computational Optimization*, online available, DOI: 10.1007/s13675-013-0016-x, 2013.

[14] A. Hill. Multi-exchange neighborhoods for the capacitated ring tree problem. In I. Dimov, S. Fidanova, and I. Lirkov, editors, *Numerical Methods and Applications*, Lecture Notes in Computer Science Vol. 8962, pages 85–94. Springer International Publishing, 2015.

[15] A. Hill and S. Voß. An equi-model matheuristic for the multi-depot ring star problem. *Research paper, University of Antwerp, Belgium*, 2014.

[16] A. Hill and S. Voß. Generalized local branching heuristics and the capacitated ring tree problem. *Research Paper, University of Antwerp, Belgium*, 2014.

[17] A. Hill and S. Voß. Optimal capacitated ring trees. *EURO Journal on Computational Optimization*, online available, DOI: 10.1007/s13675-015-0035-x, 2015.

[18] D. Jungnickel. *Graphs, Networks and Algorithms*. Algorithms and Computation in Mathematics. Springer, Berlin, Germany, 2013.

[19] D. R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *In Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, pages 613–623, 2000.

[20] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer-Verlag, Berlin, Germany, 2004.

[21] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 320–353. Springer US, New York, USA, 2003.

[22] V. Maniezzo, T. Stützle, and S. Voß, editors. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer US, New York, USA, 2009.

[23] 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.

[24] M. Sniedovich and S. Voß. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics*, 35(3):551–578, 2006.

[25] M. Stoer. *Design of Survivable Networks*, volume 1531 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, Germany, 1992.

[26] A. Tomazic. Modeling techniques in tree and ring structure based locational network design. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, pages 322–327, Oct 2012.

[27] A. Tomazic. Novel presolving techniques for the connected facility location problem. In *Federated Conference on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proceedings*, pages 467–472, 2012.

[28] A. Tomazic and I. Ljubić. A grasp algorithm for the connected facility location problem. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 257–260, July 2008.

[29] S. Voß. The Steiner tree problem with hop constraints. *Annals of Operations Research*, 86:321–345, 1999.

[30] S. Voß. Steiner tree problems in telecommunications. In M. G. C. Resende and P. M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, pages 459–492. Springer US, New York, USA, 2006.

[31] L.A. Wolsey. *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, New York, USA, 1998.

Eigenleistung
Personal Contribution

Die in dieser Dissertation zusammengefassten wissenschaftlichen Artikel (siehe Tabelle 1.1) sind wesentlicher Bestandteil meiner Forschungsarbeit. Formal ist der substantielle eigene Beitrag durch die durchgängige Erstautorenschaft gegeben. Dieser betrifft unter anderem die Initiation des Forschungsvorhabens, die Konzeption und Implementierung von Lösungsverfahren, die Durchführung empirischer Studien sowie die Verfassung des eigentlichen Aufsatzes. Hiermit soll die Leistung der Koautoren in keiner Weise in Frage gestellt werden.

The scientific papers included in this dissertation (Table 1.1) are a central part of my research. Formally, the substantial personal contribution is given by continuous lead authorship. The latter includes initiation of the research proposal, concepts and implementations of methods, conduct of empirical studies and writing of papers. However, we underline the importance of contributions of the co-authors that made the articles possible.

Eidesstattliche Versicherung
Statutory Declaration

Hiermit erkläre ich, **Alessandro Hill**, an Eides Statt, dass ich die Dissertation mit dem Titel

"Models and Algorithms for Extended Network Design"

selbständig und ohne fremde Hilfe verfasst habe. Andere als die von mir angegebenen Quellen und Hilfsmittel habe ich nicht benutzt. Die den herangezogenen Werken wörtlich oder sinngemäß entnommenen Stellen sind als solche gekennzeichnet. Ich habe keine kommerzielle Promotionsberatung in Anspruch genommen.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Hamburg, 15.02.2016 Alessandro Hill

Danksagung
Acknowledgement

Ich bin den folgenden Personen dankbar, da sie mir wesentlich geholfen haben, diese Doktorarbeit anzufertigen.

I am indebted to the following persons for their support that helped me writing this thesis.

R. K. Ahuja, K. Allukian, J. Böse, A. Breinbauer, A. Condotta, W. v. Corvin, O. Daske, K. F. Dörner, T. Farkas, M. Fliedner, S. Frühwirth-Schnatter, L. Gouveia, A. Hill, B. Hill, E. A. Hoshino, K. Hirche, K. C. Jha, U. Killisperger, P. Lenk, C. Otten, J. Pahl, P. Putz, M. Rondonell, K. Sörensen, M. v. Trappen, S. Voß, C. Wackerl, C. Wiesinger, C. Witting, W. Zeppelzauer.

A. Paper i

A GRASP Algorithm for the Connected Facility Location Problem

Alessandro Tomazic* Ivana Ljubić†
Faculty of Business, Economics, and Statistics
University of Vienna
Brünnerstr. 72, 1210 Vienna, Austria
{alessandro.tomazic, ivana.ljubic}@univie.ac.at

Abstract

We apply a Greedy Randomized Adaptive Search Procedure (GRASP) to solve the Connected Facility Location Problem heuristically. Diversification property is assured by applying a randomized greedy algorithm to construct feasible solutions in a multi-start fashion. Intensification elements are guaranteed due to two facility-based local search techniques. The computational study is conducted on a parameterized set of randomly generated benchmark instances. The obtained results reflect the quality of the proposed approach with respect to both, the quality of solutions and the computational effort, by comparison with lower bounds obtained from a Branch-and-Cut framework.

1 Introduction

Given an undirected connected graph G with vertices V and the set of edges E, a nontrivial partition $P = (F, C)$ of V identifying *facilities* and *customers*, *edge costs* $c : E \mapsto \mathbb{Q}^+$ and a function $p : F \mapsto \mathbb{Q}^+$ that defines the *opening costs* for the facilities, the *Connected Facility Location Problem* is defined as finding a connected subtree $T = (V[T], E[T])$, such that $C \subset V[T]$ and $G[F \cap V[T]]$ is connected, that minimizes the objective function

$$\sum_{v \in V[T] \cap N_T(C)} p(v) + \sum_{e \in E[T]} c(e) \ .$$

With $N_T(C)$, we denote the neighboring nodes of C in T. In a solution T with an edge connecting a facility v with customer u, we say that v is *open* and u is *supplied* by v. *Potential suppliers* F_p are those nodes from F that have at least one neighbor in C, i.e. $F_p = N_G(C) \cap F$.

It follows immediately from the definition that in every feasible solution customer nodes must be leaves. Therefore, given a graph G, edges between two customers can be deleted, without loss of generality.

In the design of telecommunication networks the following problem can be modelled as ConFL: build a last-mile network by replacing outdated copper- by fiber-optic-connections, thereby placing multiplexers to switch between them. Connect then multiplexers to each other, so that connection- and installation-costs are minimized.

The ConFL Problem has been introduced in [4], and the best-known approximation ratio of 4.23 has been obtained recently by Eisenbrand et al. [2]. Ljubić [5] concentrated on the rooted version of the problem, in which a facility r is open at no costs.

In the next Section we show how to integrate a greedy heuristic and two local-search methods in a GRASP framework. Section 3 describes how the problem can be transformed into the Minimum Steiner Arborescence Problem (SA) in graphs. We also propose to solve the SA using a branch-and-cut algorithm (B&C) whose bounds are then used in in section 4 to measure the quality of our heuristic approach. Section 5 gives some conclusions and ideas for improving the algorithm.

2 The GRASP Algorithm

A GRASP [3] is a multi-start iterative approach for combinatorial optimization problems where each iteration consists of two phases: greedy construction and local improvement. The best overall solution is reported as the final one.

In every iteration of the GRASP algorithm, we use a randomized *Greedy* procedure to construct a feasible solution. As a search intensification mechanism, we iteratively apply *open-* and *close-facility* moves, followed by a shortest path Steiner tree heuristic to find locally optimal solutions. The whole process is iterated in a multi-start fashion until a prespecified number of iterations is reached.

Diversification property is assured by dynamically increasing the number of potential candidates to be inserted within the randomized greedy procedure.

*Supported by the Austrian Research Promotion Agency, FFG, within Bridge 2 programme (812973)

†Supported by the Hertha-Firnberg Fellowship of the Austrian Science Foundation (FWF)

978-0-7695-3297-4/08 $25.00 © 2008 IEEE
DOI 10.1109/SAINT.2008.64

2.1. The randomized Greedy construction

For the construction phase a deterministic *Greedy* heuristic presented below is modified. The algorithm incrementally opens potential suppliers and connects them into a tree T, until all customers are supplied.

Greedy Algorithm The potential supplying facilities F_p are sorted in increasing order according to the following criterion:

$$\frac{p(v)\,dist(v,T)}{deg_{C'}(v)+1}, \quad v \in F_p \tag{1}$$

where $dist(v,T)$ is the length of a shortest path from v to a vertex in T with respect to c. If $T = \emptyset$ or $v \in V[T]$, $dist(v,T) = \min_{u,v \in F} c(u,v)$. $deg_{C'}(v)$ is the number of unsupplied customers that can be served by v. Hereby, a higher priority is given to the vertices close to the partially constructed subtree T, incident to many unsupplied customers or with low opening costs.

Among all closed potential suppliers, let v be the best one according to (1). We open v and connect all the neighboring customers to it if not already supplied cheaper. Then we extend T by adding the shortest path from v to T. Afterwards facilities are reassigned where appropriate, and if some of the previously opened facilities gets closed, the facility network is redesigned by application of the *Shortest Path Heuristic*[7] described below.

The algorithm iteratively opens facilities in this manner until all the customers are supplied.

Shortest Path Steiner Tree Heuristic (SPH) For a set of open facilities we use a heuristic to redesign the facility network. The basic subproblem at this stage is the Steiner Tree Problem with the set of open facilities chosen as terminals and the remaining facilities as Steiner nodes. Therefore we apply the well known Shortest Path Heuristic that efficiently returns an approximation of the optimal connection between open facilities. Hereby the tree is iteratively extended by adding the shortest path to the nearest unconnected terminal node starting with an arbitrary one.

Denote with $G[F]$ the subgraph induced by F, and with $E[F]$ the corresponding set of edges. The computational complexity of the SPH is $O(|F|(|E[F]| \log |F|)$ (see [1] for the explanation of an efficient implementation using Dijkstra's shortest path algorithm and binary heaps).

Randomized Greedy In a multistart version with starting alternatives, we add a tolerance to the selection of facilities to open. Our candidate list of length k will simply consist of the first k vertices in F_p - sorted with respect to (1) - that are not opened yet. So in each step of the method we randomly choose one among those k facilities. Setting the parameter k to 1 corresponds to the non-randomized version of the algorithm. In our implementation the number of candidates is dynamically set within the GRASP procedure and applies for a complete *Greedy* run. We begin with $k = 1$ and linearly (rounded to integers) increase it up to $3/20$ of the number of vertices available after each *Greedy* insertion.

Since in every iteration the Dijkstra algorithm and the SPH are called, the total runtime complexity of *Greedy* is $O(|F|(|F||E[F]| \log |F| + |C|))$.

2.2. Local Search Techniques

To intensify the search in the local regions of a such constructed greedy solution, we explore *open-* and *close-facility* neighborhoods in two phases: we first search for the locally optimal solution with respect to the open-facility neighborhood, in the second phase we search for the improvement by sequentially applying *close-facility* moves. In the case of facility closure the *Shortest Path Heuristic* is reapplied.

Our computational studies have shown that further repetitions of the two phases would be rather time consuming than beneficial. Therefore we decided to apply them only once per GRASP iteration.

A set of references related to the neighborhood search techniques for the Steiner Tree Problem can be found in [1], for example.

Open-facility Moves A closed potential supplier $v \in F_p$ is opened to check, whether supplying customers results in cost reduction. Thereby we consider the opening costs of v, the costs for connecting customers and the savings of previously paid connection costs for each reconnected customer. If this reduces the objective value, we perform the operations. When reconnecting customers we may end up in finding an open facility u not supplying any customers. In this case u can be closed and its opening costs are taken into account when evaluating the opening of v. Computational complexity of a single move is $O(|F||E[F]| \log |F| + |C|)$. For the total exploration of the neighborhood we need $O(|F|(|F||E[F]| \log |F| + |C|))$ time.

Close-facility Moves Closing a facility v can only result in a feasible solution when all the customers can be supplied by other open facilities. Here we focus on the open ones and try to reconnect its customers such that it costs us less than we gain by closing v. The closure of a facility may also have impact on the structure of our facility network. Since v isn't indispensable because of its supplying function anymore, solving the corresponding Steiner Tree Problem would return an optimal facility network for the current supply situation. This is approximated by running the Shortest Path Steiner Tree Heuristic described previously. A single move in that case has a computational complexity of $O(|F|(|E[F]| \log |F| + |C|))$. For the total exploration of the neighborhood we need $O(|F|^2(|E[F]| \log |F| + |C|))$ time. By using priority queues we might improve this computational complexity. When exploring the neighborhoods, we consider *first improvement* strategy where facilities to be opened/closed are always processed in a randomized order to prevent the algorithm from getting stuck at the same local optimum.

2.3. Integration in the GRASP-Framework

The *Greedy*-heuristic together with the local search techniques are embedded in a GRASP framework as described in Algorithm 1.

Algorithm 1 The GRASP-Framework

Input: $(G = (V, E), c, p)$
Output: A feasible ConFL solution (T, A)
for $iter = 0$ to $iter = |F_p|/5$ **do**
  $(T, A) = Greedy(G, c, p)$
  $OpenFacilityLocalSearch(G, c, p, T, A)$
  $CloseFacilityLocalSearch(G, c, p, T, A)$
end for

3 Relation to the Minimum Steiner Arborescene Problem

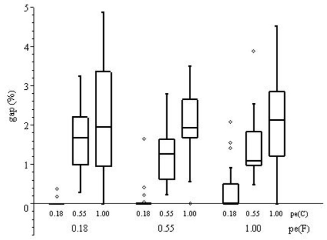
Given a directed connected graph $G_A = (V_A, A)$ with costs function on the arcs $w : A \mapsto \mathbb{Q}^+$, with a root $r \in V_A$ and a set of *terminals* $T_A \subset V_A$, the *Minimum Steiner Arborescence Problem* searches for a rooted subtree of G_A of minimum costs, such that there is a directed path from r to every $v \in T_A$.

Every ConFL problem on a graph $G = (V, E)$ can be transformed into a SA problem in the following way:

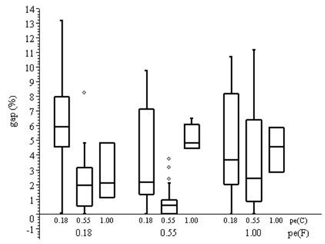
1. Introduce an artificial root r and for every potential supplier $v \in F_p$ its counterpart v'. Set $V_A = V \cup \{r\} \cup \{v' \mid v \in F_p\}$.
2. Connect all potential suppliers to r, i.e. set $A = \{(r, v) \mid v \in F_p\}$ and $w(r, v) = 0 \forall v \in F_p$.
3. Connect all facilities with each other: $A = A \cup \{(u, v), (v, u) \mid u, v \in F\}$ and $w(u, v) = w(v, u) = c(u, v), u, v \in F$.
4. Split potential suppliers: $A = A \cup \{(v, v') \mid v \in F_p\}$. Set $w(v, v') = p(v), v \in F_p$.
5. Connect facilities and clients only in one direction: $A = A \cup \{(v, v_c) \mid v \in F_p, v_c \in C\}$ and $w(v, v_c) = c(v, v_c)$.

In such obtained instance there are obviously no outgoing arcs from any customer node $v_c \in C$, and there are only outgoing arcs from the root r. To assure feasibility of a ConFL solution that is obtained by removing r and its adjacent arcs, and contracting splitted nodes again, we request that the outgoing degree of r must be equal to one.

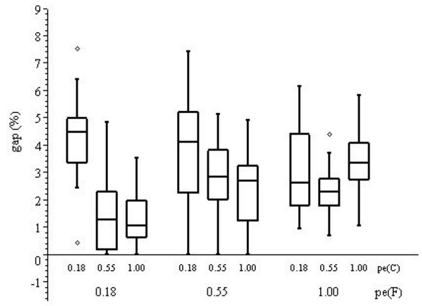
To solve the minimum SA problem on G_A to optimality, we use an adaptation of the branch-and-cut algorithm proposed in [6]. The bounds obtained using this algorithm are presented in the next Section.



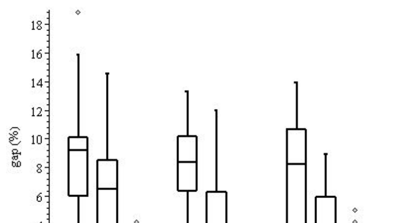
(a) SET1: $|F| = 20$, $|C| = 100$, $t_{avg} \approx 3s$



(b) SET1: $|F| = 100$, $|C| = 20$, $t_{avg} \approx 44s$



(c) SET1: $|F| = 50$, $|C| = 50$, $t_{avg} \approx 15s$



(d) SET3: $|F| = 50$, $|C| = 50$, $t_{avg} \approx 15s$

Figure 1. Percentage gaps between GRASP solutions and optimal values.

4 Computational Results

The procedure was tested on a set of randomly generated graphs with random integer weights[1]. Edges of the network

[1] All instances are available at homepage.univie.ac.at/alessandro.tomazic.

$G[F]$ are generated with probability $pe(F) \in \{0.1, 0.5, 1\}$, whereas connections between facilities and customers are established with probability $pe(C) \in \{0.18, 0.55, 1\}$. Lower and upper bounds for the edge weights were set to 50 and 100. We generated two sets, **SET1** ($|F| \times |C| \in \{100 \times 20, 20 \times 100, 50 \times 50\}$) and **SET2** ($100 \times 100$) of 81 and 27 instances, respectively, with facility opening costs randomly assigned to values between 150 and 200. For 27 additional 50×50 instances (**SET3**) we randomly assigned the facility opening costs to values between 800 and 1600.

After letting MAPLE generate a random facility network $G[F]$, we randomly link customers to the existing vertices using the parameters given above. This resulted in instances where almost always every facility was a potential supplier.

Our C++ implementation of the GRASP was tested on an Intel Core 2 Duo E4300 with 1.8 GHz, 3.25 GB RAM.

SET1 and SET3: Each single box in the diagrams of Figure 1 reflects the results of 10 GRASP runs on 3 random instances with the same construction properties (30 results). The computation time that our method needed did not exceed two minutes. The exact method completed the computations in 5 minutes on average for the instances of SET1 and SET3, but needed almost 30 minutes for three of them. Figure 1 depicts percentage gaps between GRASP solutions and optimal solutions (OPT) calculated as: $gap = (\text{GRASP} - \text{OPT})/\text{OPT}[\%]$.

We observe that GRASP easily finds optimal solutions if $|F|$ is small when compared to $|C|$ and the connections between F and C are sparse (Figure 1(a)). With increasing density $pe(C)$, the performance gets worse, but median gaps are still within 2% of optimum.

For instances whose number of facilities is large when compared to the number of customers (Figure 1(c)) we observe that GRASP has difficulties to deal with, by providing solutions whose median gaps are between 1% and 6% of optimum.

Given the same graph topology, we also tested the influence of the cost structure to the GRASP performance, by comparing 50×50 instances of the SET1 and the SET3 (Figures 1(c), 1(d)). There is obviously no direct dependency between the quality of obtained solutions and the parameters $pe(F)$ and $pe(C)$ of the SET1 group. However, when average facility opening costs are by an order of magnitude higher than the average connection costs (SET3), GRASP solves graphs with complete bipartite structure between F and C ($pe(C) = 1$) very efficiently (median gap less than 1%), and had difficulties with sparse structures ($pe(C) \in 0.18, 0.55$).

SET2: For 27 instances of size 100×100, after running B&C for one hour, only one instance was solved to optimality, and for three of them not even a feasible solution was found. On the contrary, the GRASP results were all obtained within less than 5 minutes, and in less than 2 minutes on average.

Group		LB-gap		UB-gap		GRASP	B&C-gap
$pe(F)$	$pe(C)$	avg	med	avg	med	$t[s]$	avg
0.1	0.18	8.6	8.8	–	–	41.5	–
0.1	0.55	5.5	5.8	3.2	1.9	123.9	9.1
0.1	1.0	5.7	5.5	-2.2	-2.0	69.7	3.4
0.5	0.18	6.1	6.3	-0.6	-0.5	106.0	5.5
0.5	0.55	5.8	6.1	-0.4	0.2	27.9	5.4
0.5	1.0	5.8	5.9	0.0	-0.2	185.5	5.8
1.0	0.18	4.7	4.4	-1.7	-1.7	275.1	3.0
1.0	0.55	4.9	4.7	-1.1	-0.7	46.8	3.8
1.0	1.0	3.4	3.2	-2.0	-2.0	15.6	1.4

Table 1. GRASP vs. branch-and-cut results for SET2 instances (100×100).

In Table 4, for three instances within a group and for 10 runs per instance, we report the following values: LB-$gap = (\text{GRASP} - \text{LB})/\text{LB}$, the average and median gaps between the GRASP solution and the B&C lower bound (LB); UB-$gap = (\text{UB} - \text{GRASP})/\text{UB}$, the average and median gaps between the GRASP solution and the B&C upper bound (UB); average GRASP running time in seconds ($t[s]$) and the optimality gap of the B&C obtained as $(\text{UB} - \text{LB})/\text{LB}$.

The most difficult instances for B&C are the sparse graphs, where GRASP even outperforms the upper bounds found by B&C (positive UB-gap values) after one hour.

We conclude that, on three sets of randomly generated instances with uniform topology and different cost structures, GRASP performs fast and provides stable results whose average gap to optimum varies between 0% and 10%. For both approaches, B&C and GRASP, the most difficult (easiest) instances appear to be those with sparse (dense) customer-facility topologies when $|F| = |C|$.

References

[1] M. P. de Aragão, C. C. Ribeiro, E. Uchoa, and R. Werneck. Hybrid local search for the Steiner problem in graphs. In *Extended Abstracts of MIC 2001*, pages 429–433, Porto, Portugal, 2001.

[2] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *Proceedings of SODA*, pages 1174–1183, 2008.

[3] T. Feo and M. Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[4] D. R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *FOCS*, pages 613–623, 2000.

[5] I. Ljubić. A hybrid VNS for connected facility location. In *Hybrid Metaheuristics 2007*, pages 157–169, 2007.

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

[7] H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Japonica*, 6:573–577, 1990.

B. Paper ii

ISBN 978-83-60810-48-4

Novel Presolving Techniques for the Connected Facility Location Problem

Alessandro Tomazic
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Vienna, Austria
alessandro.tomazic@wu.ac.at

***Abstract*—We consider the connected facility location problem (ConFLP), a useful model in telecommunication network design. First we introduce the extended connected facility location problem which generalizes the ConFLP by allowing pre-opened and pre-fixed facilities. This new concept is advantageous for applying complex sequences of reduction tests. By such an analysis of the solution space we anticipate solution dependencies in favor of following optimization methods. Besides transferring existing techniques designed for the facility location problem, the Steiner tree problem and the group Steiner tree problem, specific new reduction methods are introduced. The presented concepts based on graph theoretic formulations are also of theoretical interest. Additionally, we propose an efficient self-adaptive presolving strategy based on test dependencies and test impacts respectively. A computational study shows that the number of edges could be reduced up to 85% and the number of nodes up to 36% respectively on instances from the literature.**

***Keywords*-connected facility location; presolving; network design; Steiner tree;**

I. Introduction

A. Motivation

The *connected facility location problem* (ConFLP) is a highly useful model for the application to problems arising in the design of telecommunication networks. For instance the (partial) replacement of existing out-of-date copper based networks by modern fiber-optic cables can be handled as a ConFLP. We are given customers, that need to be connected to a central distributor by a tree-shaped network. In commonly used *Fiber-To-The-Curb* (FTTC) architectures, potential switching locations are given to which the customers may by connected by an existing copper infrastructure. Any choice of switch installations results in a set of terminals that have to be connected to the distributor using new fiber-optic technology. The practical objective is to minimize the overall installation costs for cables and switching devices.

The ConFLP is an NP-hard [1] optimization problem and therefore especially challenging in real world applications involving large instances. To ease the computational burden for algorithms to compute an optimal or heuristic solution, the application of problem presolving procedures is not just effective, but also unavoidable in many cases. The term *presolving* is used in turn which emphasizes the solution-oriented character of the applied techniques, compared to simple preprocessing methods. Such an analysis of the solution space may result in a remarkable reduction of the problem size. For certain instances no further methods need to be applied since the techniques used for presolving completely reduced them to trivial ones and therefore solves it to optimality. Exact and heuristical methods take advantage of the anticipation of preprocessable dependencies. Certainly a reduced number of provided variables is likely to accelerate the enumeration in exact branch & bound algorithms.

Partially supported by the Austrian Research Promotion Agency, FFG, within Bridge 2 programme (812973)

B. Contribution

We developed several presolving techniques for the ConFLP, transferred existing literature ideas for related problems and embedded these methods into a self-adaptive overall strategy. This algorithmical framework is based on problem and test specific reduction dependencies that we also present in this paper. In our studies we did not limit ourselves to problem reductions that necessarily result into a ConFLP instance again. Instead we generalized the ConFLP by the introduction of the extended connected facility location problem (EConFLP). This model is very convenient for transferring presolving information during the reduction process and allows the flexible integration of practical side constraints at the same time. The ideas are also of theoretical interest and variations may be considered for related problems. Computational results show that these methods can be effective.

C. Related work

The ConFLP model combines two classical problems in combinatorial optimization. On the one hand the *Steiner tree problem* (STP) asks for a tree of minimal edge costs that connects given *terminals* by optional use of *Steiner* nodes for the interlinkage [2]. On the other hand the ConFLP generalizes the well known *(Uncapacitated) facility location problem* (FLP). To solve the FLP selected facilities are installed at potential sites and each given customer is assigned to exactly one of them. The objective is to minimize the sum of the installation costs and the assignment costs. The ConFLP has been introduced by Karger and Minkoff [1] who gave the first approximation algorithm of constant factor. Solution approaches for the ConFLP in the literature include exact MIP-based methods [3], greedy random adaptive search [4] and a dual based heuristic [5]. The design of effective reduction methods was already carried out for related network design problems. Introductory work on preprocessing techniques can ba found in [6]. The STP was considered by Duin et al. [7] and Polzin et al. ([8], [9]). Ferreira et al. [10] developed sophisticated tests for the reduction of instances of the *group Steiner tree problems* (GSTP). Ideas from these works serve as a starting point for some of our elaborations presented in this paper. More recently, the extensive application of presolving techniques by Letchford et al. [11] enabled the solution of peviously unsolved FLP instances to optimality. The authors combined complex lower and upper bounding procedures to an effective *aggressive preprocessing* scheme, that reduces instances sufficiently for MIP solvers.

D. Problem definition

Given an undirected connected graph $G = (V, E)$, a nontrivial partition (F, C) of V identifying *facilities* and *customers*, nonnegative *edge costs* c and nonnegative *opening costs* for the facilities, the *connected facility location problem* consists of finding a connected subgraph $G' = (V', E')$ of G, such that

i.) each customer is adjacent to exactly one facility in G',
ii.) the subgraph of G' induced by the set of facilities in V' is connected, and
iii.) the total cost, defined as the sum of the edge costs and the costs for opening facilities,

$$\sum_{v \in N_{V'}(C)} p_v + \sum_{e \in E'} c_e \quad ,$$

is minimized.

For any node $u \in V$, by $N_{V'}(u)$ we denote the set of its neighboring nodes in V', and for any subset $X \subseteq V$, we set $N_{V'}(X) = \cup_{u \in X} N_{V'}(u)$. Due to the problem definition above and the non-negativity of costs c and p there exists an optimal solution such that G' is a tree. Some facilities in F may be used as pure Steiner nodes, in which case no opening costs need to be paid for them. If in a solution a facility v is adjacent to a customer u, we call v an *open facility* and we say that u *is supplied* by v, or that u is *assigned to* v. The set of *potential facilities* $F_p = N_F(C)$ contains the nodes in F that allow facility installations.

II. The Extended ConFLP (EConFLP)

In our presolving studies we did not limit ourselves to problem reductions that necessarily result into a ConFLP instance. Instead we generalize the understanding of such an instance by the following properties:

- A facility might be labeled as *open facility*, i.e. it has to be open in a solution.
- A facility might be labeled as *network facility*, i.e. it has to be part of a solution, either open or not. We do not allow a facility to be open and a network facility at the same time.
- An edge connecting two facilities might be labeled as *network edge*, i.e. it has to be part of the facility network in a solution. This implies each of the two ends to be network facilities if not opened yet.
- Facilities may belong to *groups*. A group specifies facility sets in which at least one node belongs to an optimal solution.

In order to solve the resulting EConFLP, algorithms might need to be modified with respect to the additional restrictions. Alternatively a ConFLP instance could easily be obtained from an EConFLP instance. The corresponding transformation into the ConFLP looks as follows:

i.) For each open facility v, introduce an artificial customer $\tilde{v}$ and connect it to v with $c_{\tilde{v}v} = 0$. Herewith in any ConFLP solution the facility v will be opened.
ii.) For each network facility v introduce an auxiliary facility v' and connect it to v by an edge of zero cost where $p_{v'} = 0$. Then proceed with v' as for an open facility described above. Note that v' needs to be introduced since we do not necessarily open v.
iii.) For each group g, introduce an artificial customer v_g and connect it by zero cost edges to the nodes in g.

III. Presolving Techniques

Our presolving methodology tries to reduce the initial problem stepwise. To refer to the current reduced structures we use tilde, (e.g. $\tilde{F}_p$). Furthermore we will use the *distance function* d_{uv} which returns the length of a shortest path between two nodes u to v in the current graph with respect to c. To restrict the function to a subgraph induced by the nodes $X \subseteq V$ we write d^X_{uv}. For the set of open and network facilities, we also say *solution* or *fixed facilities*. Given two disjoint node sets $S_1, S_2 \subseteq V[G]$, we define a *minimal* $\{S_1, S_2\}$-*cut* as a partition $\{R_{S_1}, R_{S_2}\}$ of $V[G]$ such that $S_1 \subseteq R_{S_1}$, $S_2 \subseteq R_{S_2}$ and the number of edges connecting R_{S_1} and R_{S_2} is minimized. After computing a solution for the reduced problem, a corresponding solution for the original problem needs to be constructed. This is achieved by the successive reversion of the modifications in the current problem and the corresponding solution adaptations in the opposite order of reduction. Corresponding restoration rules can easily be derived from the reduction steps so that we do not elaborate them here.

A. Presolving the facility subgraph

As we already observed, once the set of open facilities is known, the problem reduces to the Steiner tree problem on the facility subgraph G_F. Therefore, the traditional reduction procedures for the STP, most of them originally proposed by Duin and Volgenant [7], can easily be extended to the ConFLP. We now show how to generalize these tests for the ConFLP: we apply them on the subgraph $\tilde{G}_{\tilde{F}}$, with network facilities and open facilities treated as terminal nodes.

1) Degree 1 and 2 facilities: Every facility $v \in \tilde{F}$ with $deg_{\tilde{V}}(v) = 1$ that is not a potential facility can be deleted. If v is an open or network facility, we additionally fix its neighbor (if not an open facility yet) as a network facility.

For a facility $v \in \tilde{F}$ with $deg_{\tilde{V}}(v) = 2$ that is not a potential facility, an open facility or network facility, we can apply the following: delete v and insert an edge connecting its two former neighbors. Set the edge cost to the sum of the costs of the removed edges. If this edge already exists, then set its weight to the minimum of the new cost and its original value. This is possible because either none of the two edges incident with v is part of an optimal solution or both.

2) Shortest paths: An edge $e = uv \in \tilde{E}_{\tilde{F}}$ that is not a network edge is dispensable if $c_e \geq d^{\tilde{F}}_{uv}$, because e can always be replaced by a shortest path connecting u and v without increasing the objective value.

3) Network edges: For a network edge $e = uv \in \tilde{E}_{\tilde{F}}$ with not both ends being potential suppliers, i.e. $\{u, v\} \not\subseteq \tilde{F}_p$, u and v can be contracted, say to z. Then z becomes a solution facility if neither u nor v were fixed in the solution before. If one of the ends was an open facility, then we assign its opening costs to z and open it. If u or v was a network facility, then z is added to the network facilities.

4) Nearest node: Consider a solution facility v and let $x = \text{argmin}_{u \in N_{\tilde{F}}(v)} c_{vu}$ and $z = \text{argmin}_{u \in N_{\tilde{F}}(v) \setminus \{x\}} c_{vu}$. We add vx to our solution as a network edge if a solution facility v' exists such that

$$c_{vx} + d^{\tilde{F}}_{xv'} \leq c_{vz} \quad .$$

Note that the nearest facility x may be a solution facility and in this case the test corresponds to the *adjacent solution facilities test* for x and v.

5) Node nearer to solution facility: An edge $uv \in \tilde{E}_{\tilde{F}}$ will not appear in an optimal solution if a solution facility $x \notin \{u, v\}$ exists such that

$$max(d^{\tilde{F}}_{xu}, d^{\tilde{F}}_{xv}) \leq c_{uv} \quad .$$

This is possible, because instead of using the edge uv in the network we could always connect the two ends to a solution facility without paying additional costs. Note that in the case of one end being a terminal, this test corresponds to the *nearest node test*.

6) Bottleneck degree m: We consider a facility v that is not a potential facility with $m = deg_{\tilde{F}}(v) \geq 3$. Such a facility will have either degree two or will not belong to an optimal solution if the following property holds:

$$\sum_{u \in N_{\tilde{F}}(v)} c_{vu} \geq ST(K, \tilde{G}_{\tilde{F}} \backslash \{v\}), \ \forall K \subseteq N_{\tilde{F}}(v), \ |K| \geq 3,$$

where $ST(K, H)$ denotes the cost of an optimal Steiner tree connecting subset K of terminals on the graph H. Since solving the Steiner Tree subproblem would be way to expensive, we just apply a heuristic, namely the well known *Shortest Path Heuristic* for the STP (see [2] or [4]). In order to do so, for each pair of neighbors of $x, z \in N_{\tilde{F}}(v)$, we either insert a new edge $e = xz$ (if it does not exist) and set its cost to $s = c_{xv} + c_{vz}$ or we update the current edge weight to $c_{xz} = \min(c_{xz}, s)$. In the worst case we pay the price of adding $\binom{m}{2} - m$ edges to the problem for a single facility deletion which explains why this test is just applicable for small values of m. In a dense facility network we conversely may not need to add any edges but we might want to change the cost structure, besides the facility deletion.

7) Adjacent solution facilities: Let two adjacent facilities u and v be part of a solution, either as network or open facilities. If $c_{vu} \leq c_{vx}$ $\forall x \in F$ then the edge e_{vu} can be added to our solution as a network edge. To prove this, assume that the condition holds for v but an optimal solution S exists that does not contain vu. For connectivity reasons S contains a path in $\tilde{F}$ using an edge vx $(x \neq u)$ from v to u. So S could be improved by using vu instead of vx what contradicts with the optimality of S.

8) Facility cuts: In [9], Polzin and Daneshmand present a decomposition concept for the STP based on the detection of *node separator* subsets of low cardinality, i.e. subsets of nodes whose removal separates the terminals of the graph $\tilde{G}_{\tilde{F}}$. We extend this concept to the set of *edge separators*, by introducing additional presolving steps for the newly detected groups induced by these cuts. For solution facilities t_1 and t_2 we compute a minimal t_1-t_2-cut (S, T) in $\tilde{G}_{\tilde{F}}$. If there is just a single edge e connecting S and T we can label e as a network edge, since it will belong to any feasible solution. Additionally we consider the induced node separator sets $Q_S = \{v \in S : N_{\tilde{F}}(v) \cap T \neq \emptyset\}$ and $Q_T = \{v \in T : N_{\tilde{F}}(v) \cap S \neq \emptyset\}$. We add Q_S and Q_T to the set of groups, but control these additions by a parameter that limits the size of added groups. Some of the presolving techniques for the group Steiner tree problem can be transferred and applied to the concept of *EConFLP* as defined above. Recall that, given a graph $\tilde{G}_{\tilde{F}}$ with non-negative edge costs, and a collection $\mathcal{R}$ of subsets of $\tilde{F}$, called *groups*, the GSTP is to find a minimum-cost subtree of $\tilde{G}_{\tilde{F}}$ that contains at least one node from each group $R \in \mathcal{R}$. We consider the groups $R \subset \tilde{F}$, that arise from different tests introduced within this paper. Apart from such groups, we can initially add a group R_u for each customer u consisting of its potential suppliers $N_{\tilde{F}}(u)$.

9) Node nearer to group: This test is a generalization of the *node nearer to terminal test* for the GSTP. An edge $e = uv \in \tilde{E}_{\tilde{F}}$ will not appear in an optimal solution if a group $R \not\supseteq \{u, v\}$ exists such that

$$max(d^{\tilde{F}}_{ru}, d^{\tilde{F}}_{rv}) \leq c_e \quad \forall r \in R \quad .$$

This is possible, because instead of using the edge e in the network we could always connect the two ends with any node in R without additional costs. The special case of considering groups of cardinality one leads to the *node nearer to terminal test*.

10) Group cuts: For two disjoint groups R_1 and R_2, we compute a minimal (R_1, R_2)-cut in $G_{\tilde{F}}$, say (S, T). If there is just a single edge connecting S and T we can label it as a network edge, since it will belong to any solution. Additionally we consider the induced node sparator sets $R_S = \{v \in S :$

$N(v) \cap T \neq \emptyset\}$ and $R_T = \{v \in T : N(v) \cap T \neq \emptyset\}$. We add R_S and R_T to the set of groups, but limit the total number of added groups. This generalizes the *facility cut test* since facilities fixed in a solution are just groups of cardinality one. Here we also mix the two types by allowing singleton groups.

B. Presolving the facility-customer subgraph

Adapting presolving tests for the facility location problem we get the following applying to the ConFLP.

1) Degree 1 customers: For a customer $u \in \tilde{C}$ with $deg_{\tilde{G}}(u) = 1$ we remove u and force the facility v to be part of the facility network as an open facility.

2) Customer domination: For two customers u and v with $N_{\tilde{V}}(u) \subset N_{\tilde{V}}(v)$ we can delete an edge $e = vx$ with $x \notin N_{\tilde{V}}(u)$ if $c_e \geq c_{vz}\ \forall z \in N_{\tilde{V}}(u)$. The reason for this is that at least one of the potential facilities of u will be opened and therefore even if facility x was already open it would be cheaper to let v be supplied by the facility supplying u.

3) Network facility: We consider the network facilities that are potential suppliers in order to open them or exclude alternative potential suppliers. Consider such a facility v and a potential customer u, i.e. $v \in N_{\tilde{V}}(u)$. We may delete an edge ux $(x \neq v)$ if $c_{xu} \geq c_{uv} + p_v$. So opening v and supplying u by v would be cheaper than supplying u by x, even if x was open.

4) Open facility: In the case that we have fixed a facility v to be supplying in our solution we should check whether we can exclude other facilities from being potential suppliers for its potential customers. So for a customer u adjacent to v an edge $e = ux$ $(x \neq v)$ can be deleted if $c_{vu} \leq c_e$.

C. Presolving the whole graph

Finally, in this section we propose tests that apply to the EConFLP concept, involving the whole graph $\tilde{G}$, considering solution facilities, network edges and groups as well.

1) Facility-customer distance: For a potential facility v we can delete a supply edge $e = vu$ $(u \in \tilde{C})$ if a path P_{vu} in $\tilde{G}_{\tilde{F}\cup\{u\}}$ from v to u not containing e exists such that $d'_{vu} \leq c_{vu}$, where d'_{vu} denotes the length of the path P_{vu} plus the opening costs of the potential facility z on that path if not opened yet. Thereby, the weights of network edges are discarded:

$$d'_{vu} = \begin{cases} \sum_{e \in P_{vu}} c_e + p_z & \text{z closed} \\ \sum_{e \in P_{vu}} c_e & \text{z open} \end{cases} \quad (zu \in P_{vu}) \quad .$$

This test checks if it would be cheaper to add the whole path to the solution and possibly open the facility z than using the edge vu.

2) Solution-facility-customer distance: In this test we consider a potential facility v and one of its supply edges $e = vu$ $(u \in \tilde{C})$. Let F_S be the set of current solution facilities. We delete e if for a $x \in F_S$ a path P_{ux} in $\tilde{G}_{\tilde{F}\cup\{u\}}$ from u to x not containing e exists such that $d'_{u,x} \leq c_{vu}$. Here d' is the function used in the *facility-customer distance test*. In contrast to the *facility-customer distance test* we just try to supply and connect the customer u to any facility in the existing facility subnetwork without exceeding certain supply edge costs. In the case that v is already open or a network facility, this test covers the *facility-customer distance test*.

3) Group-customer distance: We extend the *terminal-customer distance test* to groups. The additional requirement for a supply edge deletion is the existence of the mentioned path for all the nodes of at least one group.

4) Potential facility leaves: Let a potential facility v have $deg_{\tilde{F}}(v) = 1$. If v is part of an optimal solution, then - provided it is not the only facility to do so - surely its incident edge $vx \in \tilde{E}_{\tilde{F}}$ in the facility network will be used in this solution. In the case that x is not a potential facility itself, we can contract x and v and set the opening costs of the resulting potential facility as $p_v + c_e$. Otherwise if x is a potential facility and there exists a group not containing v or one other solution facility, then the opening cost of v can be increased by c_e to provoke the success of the *network facility test*.

5) Groups: We remove groups that contain solution facilities, since this is redundant information. Additionally we transform groups of cardinality 1 to network facilities. Multiple and empty groups are dynamically removed when removing a facility.

D. The overall presolving strategy

The proposed techniques are all of polynomial time complexity. Although the STP is well known to be in the class of NP-hard optimization problems we use a polynomial method to solve the problem heuristically. Therefore the overall procedure based on reduction success is also polynomial. The correctness follows from the validity of the single reductions and the transformation of the EConFLP into the ConFLP. However the effort for carrying out test on a problem instance varies. For instance a degree-testing can be done in significantly less time than running numerous minimal cut computations. Therefore we tried to minimize the number of tests that we identified as computationally more expensive in our experiments. The latter ones are mostly the tests that require solving a non-trivial subproblem, e.g. finding minimal cuts or multiple shortest paths. So we first apply the less elaborate tests as long as this results in a problem reduction. Afterwards we once run the more time consuming procedures and perform a restart if a problem modification was detected. Since we still observed many redundant test runs we focus on concrete test interactions to minimize unnecessary iterations. One can observe some complex relationships between different reduction types. Figure 1 describes the potential changes of the problem structure with respect to the tests performed. An arc depicts the potentially successful presolving tests after a certain problem modification. Figure 2 shows the impacts of the modifications of the input graph on specific presolving steps. An arc describes a problem modification consequence of the corresponding reduction type. Our idea was to make the testing scheme adaptive. Therefore a prioirity is assigned to each reduction method. Initially these values are chosen to set up a basic ordering. We simply enumerate the tests from based on our computational study from easy to hard. So the tests are called according to their priority. Whenever a test did alter the current problem, we increment the priority of all the other tests that may depend on the performed problem modification. To keep the computational effort low, we still apply the 2-phase division that first works all the easy tests.

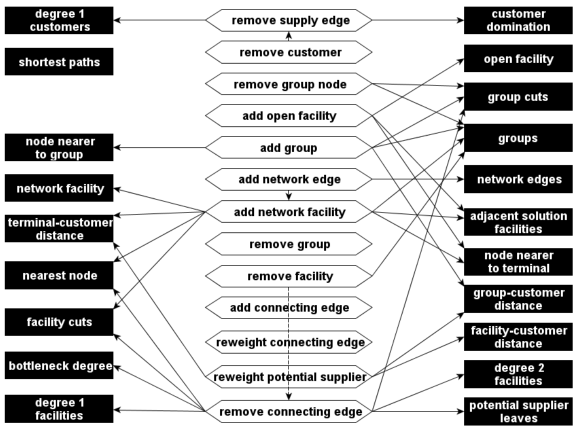


Fig. 1: Arc AM indicates that test A may result in a problem modification of type M.

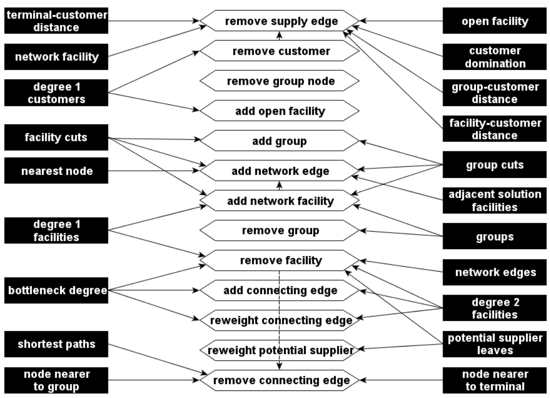


Fig. 2: Arc MB indicates that a modification of type M may have an impact on the success of test B.

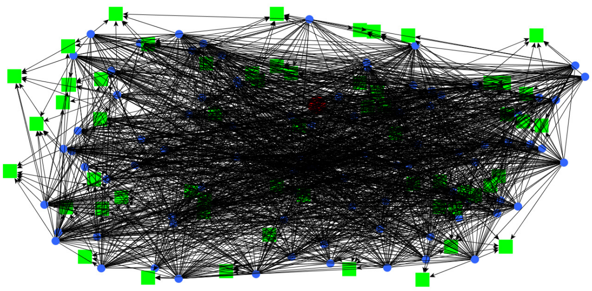
IV. Computational Results

We ran our scheme on the ConFLP instances used in [4]. The problems consist of a random facility network with randomly added customer assignments . The following parameters were adjusted to create the problem classes: the probability of creating an edge in the facility network, $G[F]$ (p_eF), the probability of creating a supply edge (p_eC) and the probability of defining a facility node as a potential supplier (pF_p). Edge weights are randomly assigned ranging from 50 to 100 and opening costs from 150 to 200 respectively. Our C++ implementation of the presolving algorithm was tested on an Intel Core 2 Duo E4300 machine with 1.8 GHz, 3.25 GB RAM. We used the following default parameter setting: in the *bottleneck degree test* we set $m = 3$; the maximal size of the added groups was set to 2 and number of groups was limited to 8. The simple categorization of tests into two complexity classes already speeds up the overall testing procedure. The number of easy test loops is about twice the number of hard test repetitions. Moreover it saves computational effort to exclude hard tests from a test loop, if no need can be detected by the dependencies illustrated in the previous section. The results shown in Table I are average values of 3 random instances per group. The computation times did not exeed 5 minutes per instance. The graphs having a sparse facility

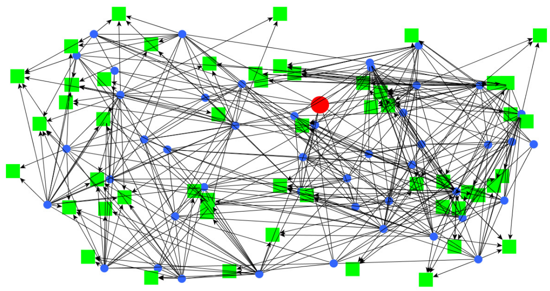
TABLE I: Average presolving effectiveness on 39 instances ($|F| = 100$,$|C| = 100$) with the relative reductions r_V and r_E on V end E. Each instance class consists of 3 random instances.

Orig. ConFLP instance						Presolved EConFLP instance				
p_eF	p_eC	pF_p	$\|E\|$	$\|E_F\|$	$\|E_C\|$	$\|E\|$	$\|F\|$	$\|C\|$	r_V	r_E
0.18	0.18	0.3	1131	590	540	435	33	93	36.5	61.5
0.18	0.18	1.0	2406	594	1811	1811	100	100	0.0	24.7
0.18	0.55	0.3	2215	587	1628	1702	49	100	25.3	23.2
0.18	0.55	1.0	6062	575	5486	5486	100	100	0.0	9.5
0.18	1.00	0.3	3575	575	3000	3103	57	100	21.2	13.2
0.18	1.00	1.0	10572	572	10000	10029	100	100	0.0	5.1
0.55	0.18	0.3	3070	2538	531	647	87	94	9.0	78.9
0.55	0.18	1.0	4347	2538	1808	1808	100	100	0.0	58.4
0.55	0.55	0.3	4179	2524	1654	1815	72	100	13.8	56.6
0.55	0.55	1.0	8023	2528	5495	5495	100	100	0.0	31.5
0.55	1.00	0.3	5518	2518	3000	3154	66	100	17.0	42.8
1.00	0.18	0.3	5506	4950	556	828	96	98	3.0	85.0
1.00	0.55	0.3	6569	4950	1619	1869	80	100	9.75	71.6

network ($p_eF = 0.18$) can be preprocessed with less effort than others. The benefit of the applied methods obviously depends on the density of the customer-facility network (p_eC). If the bipartite subgraph is complete, the algorithm stops after less than 2 seconds without any reduction. On the other hand if both networks are sparse ($p_eF = p_eC = 0.18$) and not all facilities are potential suppliers ($pF_p = 0.3$), we are able to significantly reduce the size of the inputs. We obtain graphs whose numbers of nodes and edges are reduced by 36% and 61% respectively on average. We can also observe that the sparsity of the graph G is not a sufficient condition for a successful reduction. If $F_p = F$, i.e. $pF_p = 1$, the presolving is not able to remove nodes, even for a very sparse graph $G[F]$. The worst results are obtained for graphs having a complete bipartite structure and $F_p = F$. This can be explained by the fact that many tests originally designed for the STP are not directly applicable to potential suppliers. Since our benchmark instances obey a uniform structure, in which every node has almost the same degree, simple degree tests that lead facility removals have no effect at all. In extreme cases, when the degree of every customer equals $|F_p|$, the number of edges could be reduced by only 5.1%. Finally, the most remarkable reductions concerning the number of edges (between 71.6% and 85%) are obtained for graphs having complete facility networks.



(a)



(b)

Fig. 3: Presolving effect on a random instance.

V. Conclusions

In this paper we provide techniques for presolving instances of the ConFLP embedded in an algorithmic framework. We extend the concept of a the ConFLP by allowing terminal facility nodes (being open or not) and groups of facilities among which at least one needs to be included in the solution. Afterwards we also describe how such an extended ConFLP instance can be reversed into a ConFLP. The new extended ConFLP concept enables the transfer of several known tests for (group) Steiner tree problems and the facility location problem. Furthermore, we propose a bunch of new presolving ideas for the ConFLP structure itself and test all of them computationally. The overall methodology is based on identified problem modification dependencies. Our algorithmic framework is tested on a set of benchmark instances from the literature showing that the proposed presolving is especially beneficial for graphs obeying a sparse edge structure with respect to the edges connecting only facilities, facilities and customers, or both. We observe that increasing the number of potential suppliers decreases the effectivity of the presolving procedures. maximal degree m for the bottleneck test, are determined manually, by running only a small number of sample instances. One possible way to improve these features is to use intelligent learning techniques to train these so-called *control values*.

References

[1] D. Karger and M. Minkoff, "Building Steiner Trees with Incomplete Global Knowledge," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, 2000, pp. 613 –623.

[2] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, ser. ADM. Amsterdam, Netherlands: North-Holland, 1992, vol. 53.

[3] S. Gollowitzer and I. Ljubic, "MIP Models for Connected Facility Location: A Theoretical and Aomputational Study," *Computers & OR*, vol. 38, no. 2, pp. 435–449, 2011.

[4] A. Tomazic and I. Ljubic, "A GRASP Algorithm for the Connected Facility Location Problem," in *SAINT*. IEEE Computer Society, 2008, pp. 257–260.

[5] M. G. Bardossy and S. Raghavan, "Dual-based local search for the connected facility location and related problems," *INFORMS Journal on Computing*, vol. 22, no. 4, pp. 584–602, 2010.

[6] D.-Z. Du and P. M. Pardalos, *Handbook of Combinatorial Optimization*. Springer London, Limited, 2005.

[7] C. W. Duin and A. Volgenant, "Reduction Tests for the Steiner Problem in Graphs," *Networks*, vol. 19, pp. 549–567, 1989.

[8] T. Polzin and S. V. Daneshmand, "Extending Reduction Techniques for the Steiner Tree Problem," in *Lecture notes in computer science*. Springer Berlin/Heidelberg, 2001.

[9] T. Polzin and S. Daneshmand, "Practical Partitioning-Based Methods for the Steiner Problem," in *WEA*, ser. Lecture Notes in Computer Science, C. Àlvarez and M. J. Serna, Eds., vol. 4007. Springer, 2006, pp. 241–252.

[10] C. E. Ferreira and F. M. de Oliveira Filho, "New Reduction Tests for the Group Steiner Tree Problem," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 1176–1188, 2006.

[11] A. Letchford, "An Aggressive Reduction Scheme for the Simple Plant Location Problem," *submitted*, 2011.

C. Paper iii

Modeling techniques in tree and ring structure based locational network design

Alessandro Tomazic
Institute for Statistics and Mathematics
Vienna University of Economics and Business
Vienna, Austria
alessandro.tomazic@wu.ac.at

***Abstract*—Solving real world problems in network design by means of combinatorial optimization usually begins with finding or developing appropriate models. The application of models for which problem specific efficient heuristic or exact methods exist is favourable. Another desirable key feature of such a model is its flexibility to be adapted to represent the practical requirements. We consider two capacitated problems in locational network design: the capacitated connected facility location problem and the multi-depot ring star problem, the latter ensuring reliability. First we present several relations to other known optimization problems that are generalized by these models. Then we show how to integrate multiple highly relevant side constraints. Besides prize collecting, customer coverage and multiple distributor variants we elaborate problem specific features. The introduced modeling techniques allow the usage of these optimization models with their various existing solution approaches from the literature in a wider context and help to distinguish between related models.**

***Keywords*-multi-depot ring star problem;capacitated connected facility location problem;modeling;reliable network design;**

I. Introduction

Networks are used as foundations for numerous practical and theoretical applications. In the context of telecommunication we naturally deal with networks representing the underlying structures for data transfer based on different technologies. Commonly, information is sent from distributing devices to customers and vice versa. *Locational network design* extends the structural network design by the need of taking decisions on the installation of devices at potential sites. Typical scenarios are the need of signal repeaters or switching units. Depending on the field of application the design of networks from scratch or the extension of existing structures are tasks of high importance. Modern supply networks are required to be designed or refined following strategical business requirements. Certainly, a key aspect here is cost saving planning. For instance, the replacement of outdated copper-based infrastructure by powerful fiber-optic cables is expensive concerning laying as well as costs of material. To minimize the overall installation expenses the application of suitable optimization methods has become quite popular in the last decade. Such algorithms are usually developed for specific optimization models representing practical problems. Since the network specifications vary depending on the business requirements, we might be lacking a suitable model for which solution approaches already exist. Therefore, it is highly useful to have modeling techniques at hand that allow the embedding of a given problem into a known related model structure.

In this paper we consider two powerful network design models from the literature, the *capacitated connected facility problem* and the *multi-depot ring star problem*. Both models employ customer supply capacities and are based on different structures to link customers to distributors. The first one provides a *tree-based* supply structure for the customers. Conversely, the second model propagates *ring star* structures that yield reliability of service. The latter can be informally described as a set of rings that interlink customers, to which selected coustomers may also be assigned. We illustrate both problems and give an overview of several problems variants and related models from the literature. Our main contribution is the elaboration of various modeling techniques for each problem that enable us to integrate additional requirements of high practical relevance. For instance, we show how to model side constraints such as *customer coverage rates*, *customer price collection* and *multiple distributors*.

Our paper is structured as follows. In the following section II we consider the capacitated connected facility location problem. After defining the problem formally we explain relations to literature models and introduce our modeling techniques for the integration of additional requirements. Section III covers the elaborations for the multi-depot ring star problem. Our conclusions are summarized in the final section IV.

II. Capacitated connected facility location problems

The *connected facility location problem* (ConFLP) has been introduced by Karger and Minkoff [6]. Connected facility location problems are suitable models for well known Fiber-to-the-Curb (FTTC) strategies. Here we are given an existing copper structure connecting customers to a distributor. The task is to design a cost efficient fiber *core* network that replaces the outdated structure partially. From selected

This work was supported in part by the Austrian Research Promotion Agency, FFG, within Bridge 2 programme (812973).

transition points to the customers, the so-called *last mile*, the rather short existing copper cables are retained to minimize the overall replacement costs. Various solution approaches have been proposed for this uncapacitated problem in the literature. In [14] we proposed a heuristic GRASP method and elaborated effective reduction methods in [13]. The more general model considered in this paper is the capacitated connected facility problem (CapConFLP) introducing supply capacities and customer demands. Relating to a ftth renewal scenario the capacities correspond to technical limits of multiplexers which are needed for changing over from digital to analog transmission. In [9] a problem variant including a customer coverage is solved heuristically. Our modeling techniques will include the reduction of this derivative to the CapConFLP and show the flexibitity and capability of this model in general.

We first define our basic problem in section II-A. Then a bunch of related problems from the literature are presented in II-B. Section II-C contains our modeling techniques for the CapConFLP.

A. The capacitated connected facility location problem

The following definition is based on a rooted problem formulation which is equivalent to an unrooted counterpart from a complexity point of view. A mutual transformation is described among our modeling techniques in section II-C. We are given an undirected connected simple graph $G = (F \dot{\cup} U, E)$ representing the potential network resources. A node of G may either be part of the set of *customers* U or the set of *facilities* F. No edges between customers are allowed here. Let $r \in F$ be the *root* node and consider the following given nonnegative values.

- Demands: $d_u \quad \forall u \in U$
- Capacities: $k_v \quad \forall v \in F_p$
- Opening costs: $c_v \quad \forall v \in F_p$
- Supply costs: $c_e \quad \forall e \in E_{F \cup U}$
- Connection costs: $c_e \quad \forall e \in E_F$

Throughout the article the set of edges E restricted to edges connecting nodes in X is denoted by E_X. Facilities that are incident to a customer in G are called the *potential suppliers* and denoted by F_p. A solution for the CapConFLP is a subgraph T of G such that the facility network T_F is connected, $r \in T$, each customer is connected to exactly one potential supplier and the supply capacities are respected:

$$\sum_{u \in N_U(v)} d_u \leq k_v \quad \forall v \in F_p \quad ,$$

where the neighbors of a node v restricted to a node subset X of the current graph are denoted by $N_X(v)$. The CapConFLP asks for a solution $T = (V_T, E_T)$ of minimal total cost $z(T)$, where

$$z(T) = \sum_{v \in N_F(U)} c_v + \sum_{e \in E_T} c_e \quad .$$

An example of a solution network for the CapConFLP is given in figure 1. The requirement of a unique supplier for each customer makes the CapConFLP a *single-source* supply model.

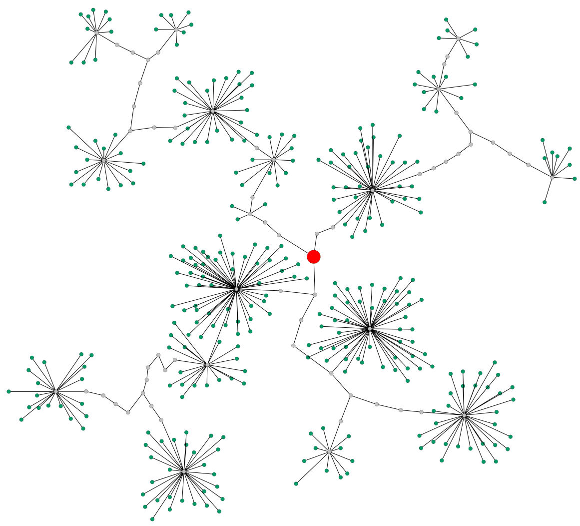


Figure 1: An example solution for the capacitated connected facility location problem with 332 customers.

B. Related models

1) The connected facility location problem: As mentioned in the introduction, the CapConFLP generalizes the connected facility location problem (ConFLP). By simply setting the customer demands to zero we are able to model each ConFLP instance as a CapConFLP instance.

2) The group Steiner tree problem: In the case of uniform assignment costs $c_{uv} \ \forall v \in N(v)$ for each customer u and zero opening costs, we can transform the ConFLP into the *group Steiner tree problem* (GSTP). The GSTP is a generalization of the well known Steiner tree problem (STP) and asks for a tree of minimal cost connecting given *groups* of terminal nodes. The GSTP is NP-hard, since it obviously generalizes the STP. It was introduced by Reich and Widmayer in [11]. The transformation is done in the following way:

i. Initialize the GSTP with the graph G_F.
ii. Define the set of groups R as all the potential supplier sets for the customers and a root group: $R = \{N(v) | v \in U\} \cup \{\{r\}\}$.

Since we do not pay for opening facilities and each customers assignment costs are constant, we can assign the customer to corresponding potential suppliers that are part of the Steiner tree. In general the resulting groups are not disjoint. To reconstruct a solution for the ConFLP from the solved GSTP just consider the group tree T and connect the customers arbitrarily to their original potential suppliers that are contained in T.

3) The facility location problem: Given a bipartite graph with partition sets F_p (facilities) and U (customers), with non-negative edge costs and non-negative opening costs assigned to facilities, the *uncapacitated facility location problem* (UFLP) consists of finding a subset of facilities to open, so that each customer is assigned to exactly one open facility and the sum of facility opening costs and customer assignment costs is minimized. Observe that having zero connection costs in the ConFLP, the connectivity requirement (connecting the open facilities by a Steiner tree) is superfluous and the ConFLP reduces to the UFLP.
The *capacitated facility location problem* (CapFLP) additionally incorporates given facility capacities and customer demands that have to be respected. This model is also well known as the *warehouse location problem* and is analogously generalized by the CapConFLP.

4) The Steiner tree problem: Recall that in the *Steiner tree problem* (STP) we search for the subgraph of a given egde-weighted graph G_F which connects given terminals at minimum costs. Assume that we are given a feasible CapConFLP. In the case of a unique potential supplier for each customer, the CapConFLP is equivalent to the STP. Simply solve the STP on G with the customers as terminals. Due to this reduction to the STP, the CapConFLP even remains NP-hard if the opening costs of facilities are zero.

5) The Steiner arborescence problem: As we already observed in [14], the ConFLP can be transformed into the *Steiner arborescence problem* (SAP). The SAP asks for an arborescence of minimal total arc weight that spans a set of terminals under the optional usage of Steiner nodes in a directed graph. This transformation replaces the facility opening costs by artificial arcs using a simple node splitting technique that was proved to be useful in different models on graph structures.

C. Modeling

The CapConFLP can model several side constraints of high practical relevance. To describe some of them, we begin with the transformation to force a facility to be part of the solution network.

1) Facility fixing: Suppose that a facility v is required to be part of a solution. First introduce an artificial customer u. If v is a potential supplier, add an artificial supplier v' and edges uv' and $v'v$. If v is no potential supplier, v' is not needed and just the edge uv of zero weight is inserted.

2) Unrooted CapConFLP: We can omit the root node requirement in a CapConFLP and obtain the *unrooted* CapConFLP when applying a facility fixing transformation on r. Then again we can turn the latter into a (*rooted*) CapConFLP by the following efficient instance expansion. Add an artificial root node and connect it by zero cost edges to each facility node.

3) Multi-rooted CapConFLP: Our model allows, as the ConFLP, efficient multiple root modeling. For a given set of roots $R \subset F$, the task is to supply the customers by a forest of minimal cost, such that each tree contains at least one root in R. The transformation is achieved by adding an artificial master root node r and connecting it to the existing roots by edges $rr_i \ \forall r_i \in R$ of zero cost. In similar network design problems as the ring star problems considered later in this paper the introduction of additional root locations boosts the problem complexity drastically.

4) Customer prize collection: Customers may not be supplied in the solution network under payment of individual *prizes*. This co-called *customer prize collecting* can be integrated by exploiting the capacitated problem structure. We add an artificial potential supplier t of zero opening cost, capacity big M and artificial supply edges tu for each customer u. Set the supply edge costs equal to the prizes for of the corresponding customers. The facility network connectivity is guaranteed after the insertion of the edge rt of weight zero.

5) Facility prize collection: We can model the case that we have to pay a prize p_v for a facility $v \in F \setminus F_p$ not being part of the solution. An artificial customer u of zero demand is added with the zero cost edge vu so that v becomes a potential supplier with zero opening cost and unlimited capacity. We choose another customer $x \in U$ of minimal degree and insert edges $zu \ \forall z \in N(u)$ of cost p_v. In the CapConFLP, the node weights are also respected in the case of a facility opening. The exclusive case can be set up by reducing the corresponding opening cost by the node weight.

6) Customer weights: Weights for customer nodes can easily be integrated into their supply costs since the CapConFLP is a single-source supply model.

7) Demand coverage: We might not need to supply all the customers. A target coverage value D for total customer demand satisfaction can be modeled. We introduce an artificial potential supplier t, set its capacity to the sum of the customer demands minus the required value, $k_t = \sum_{u \in U} d_u - D$, and connect it to the root node r by a new zero cost edge. If r is a potential supplier, we need an additional artificial supplier as in the facility fixing transformation. For $D = 0$ we face the special case treated in [9]. Note that in the more general case of having a target covering value depending on other values than the customer demands, we might still apply this technique as long as the value relation is linear.

8) Intersecting facilities and customers: Given that the customer and facility locations are not disjoint, we can still obtain an equivalent CapConFLP. This modification needs the replacement of each node $v \in F \cap U$, with a pair of nodes, $v_1 \in F$ and $v_2 \in U$, and the connection of all nodes $u \in N_U(v)$ to v_1, and all $u \in N_F(v)$ to v_2, without changing the edge costs. Finally, if v is a actually a potential supplier, we also need to connect v_1 and v_2 by an edge of zero weight and set the opening cost $f_{v_1} = f_v$.

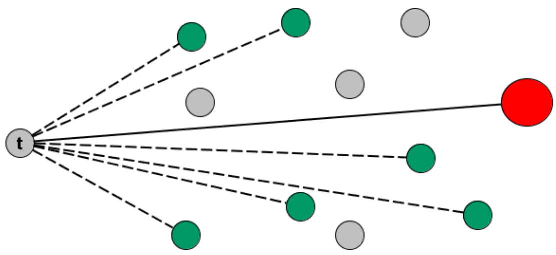


Figure 2: Modeling of the demand coverage constraint.

9) Unconnected facility network: Especially instances based on data coming from real world applications may not satisfy the requirement of the facility graph G_F to be connected. In this case we performed the following validation procedure, which was part of our preprocessing if the data did not correspond to an infeasible problem in our context. Let $F_1, \ldots, F_k$ be the connected components of G_F. If there exists a customer u that is not adjacent to a node in a component F_i, then we can discard F_i and delete all its nodes in G because not all customers can be supplied. For every remaining component F_j we solve the corresponding ConFLP on the graph G_{F_j} and keep the best solution we find. In the case that no component is left, we got infeasible data for our model. In the rooted case only the component containing the root node comes into consideration.

III. Multi-depot ring star problems

Models based on ring structures assure reliability in the sense of one-link failure insensitivity. Although this structural idea finds numerous applications in network design it has been considered extensively for routing in transportation networks. Therefore, distributors are consequently referred to as depots in this article to match with the literature notation. In the work of Fink et al. [5] a generalized model is proposed that extends several single depot ring-based models from the literature. The authors point out the different relationships and present a heuristical solution approach. However, the model considered in our paper concentrates on an even more general case.

The *ring star problem* was introduced by Labbé et al. [7] and simply asks for a single cost-minimal *ring star* for a depot connecting all the customers. In a ring star customers may either be part of a cycle containing the depot or they may be assigned to a cycle node. By restricting the allowed customer assignments in advance, we are able to divide the customers into two classes. On the one hand we have ring customers that require a reliable network access. On the other hand selected customers may just be connected to the rings by simple links. In [7] this problem is studied from a polyhedral point of view and an exact branch & cut algorithm is proposed. Later Baldacci et al. [3] introduced the capacitated version that, additionally, allows multiple ring stars for the depot and the optionel usage of Steiner nodes: the *capacitated m-ring star problem* (CmRSP). The authors developed exact solution procedures based on mixed integer programming (MIP) formulations. Currently, the efficient heuristic by Naji et al. [10] yields the best results for mid-sized instances of the CmRSP. We consider an even more general model that was applied to telecommunication network design problems in the literature recently: *the multi-depot ring star problem* (MDRSP). Three heuristic methods are proposed by Baldacci et al. [2] in their introductory work. The results are outerperformed by a hybrid heuristic of Tomazic [12] based on sophisticated MIP-based refinement and contraction techniques. After giving a formal definition of the MDRSP in III we list selected literature problems that are generalized by the MDRSP in III-B. Our main modeling ideas are presented in section III-C.

A. The multi-depot ring star problem

We are given an undirected complete graph $G = (V, E)$ that contains all the potential connections for building rings. The node set V is the disjoint union of depots D, customers U and optional Steiner nodes W. A set of selected potential assignments A contains arcs from customers to any other non-depot nodes. Each edge $e \in E$ and each assignment $a \in A$ is associated with a nonnegative cost c_e and c_a, respectively. A subcycle C in G containing exactly one depot is a ring. Such a ring paired with assignments of customers to its ring nodes yields a ring star. For each depot $d \in D$ a ring star limit m_d and a customer limit q_d per ring star is given. A solution to the multi-depot ring star problem (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 d is contained in at most m_d ring stars, and
- the number of customers in a ring star does not exceed the limit q_d for its depot d.

The MDRSP asks for a solution of minimal total cost, i.e. minimized sum of ring edge costs and assignment arc costs. Note that in our definition of the MDRSP we introduce heterogeneous capacities m_d and q_d depending on the considered depot d. The latter property increases flexibility for the application to real world problems as explained later. Figure 3 illustrates an example solution for the MDRSP.

B. Related models

1) The prize collecting traveling salesman problem: As a generalization of the classical *traveling salesman problem* (TSP) the *prize collecting traveling salesman problem* (PCTSP) allows omitting customers in the tour by paying corresponding *penalty prizes*. It was first considered by Balas [1] and also discussed in [5]. By not allowing any assignments and setting the ring limit to one for the unique depot, the MDRSP can model the PCTSP. The detailed

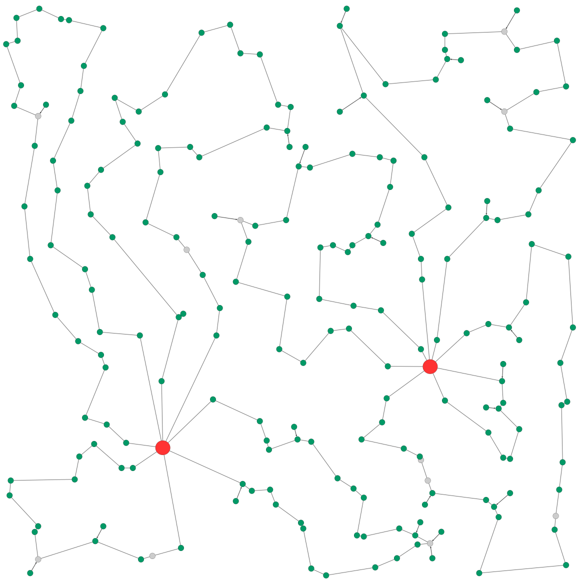


Figure 3: An example solution for the multi-depot ring star problem using 2 depots to serve 200 customers, where the ring star limit per depot is 4 and the ring star capacity is 30.

modeling idea will result from our general prize collection integration that is elaborated in section III-C1.

2) The Steiner ring star problem: The *Steiner ring star problem* (SRSP) was introduced by Lee et al. [8]. This single depot problem asks for a unique ring star of minimal overall edge and assignment costs. Additionally, customers are forbidden on the ring and therefore forced to be assigned to Steiner ring vertices. The authors propose an exact branch & cut method for this problem arising in the design of digital data networks. The SRSP can be represented straightforward by the MDRSP through a ring star limit of one, the optional assignment of all the customers and an infinite limit on the number of ring customers. To exclude the customers from being on the ring we set the edge costs involving customer nodes to some big M value.

3) The vehicle routing allocation problem: A related single-depot problem combining routing and allocation is the *vehicle routing allocation problem* (VRAP) introduced by Beasley and Nascimento [4]. The VRAP allows the disregard of customers under given penalty costs similar as in the PCTSP but considers ring star structures. We can model the VRAP and its multiple depot version as a MDRSP using the procedure for general prize collection integration given in section III-C1. The advantage over the incorporation of additional binary service variables in a MIP model, as proposed in [3], is the applicability of arbitrary solution methods designed for the MDRSP without any algorithmic customization.

C. Modeling

1) Prize collection: In many real world scenarios the connection of customers to the network may be neglected because of their remote location or their generally inconvenient location. Since such a decision usually means a loss of profit, penalty costs should be imposed upon such an exclusion. We can integrate this flexibility into the MDRSP by the following transformations. Introduce an artificial depot d' with customer per ring capacity $q_{d'} = |U|$ and ring star capacity $m_{d'} = 1$. Additionally, insert two Steiner nodes x and y. To y the customers may be assigned by arcs of their penalty costs, x is just needed to constitute a ring. The ring edge costs $c_{d'v}$, c_{xv} and c_{yv} are set to big M for all original network nodes $v \in V$ and $c_{d'x} = c_{xy} = c_{d'y} = 0$. In a solution of the resulting modified problem the *slack ring star* containing the ring nodes d',x and y will include customer assignments if paying the corresponding penalty prize is globally cost efficient. For the modeling of this feature we observe the moderate problem size increase of one depot, two Steiner nodes and $|U|$ assignments.

2) Customer coverage: Based on strategic planning just a subset of customers might have to be included in the solution network. Due to business requirements this number of neglectable customers is usually limited. Let the *coverage parameter* r describe the minimal number of connected customers. If $r = |U|$ this generalizing model reduces to the MDRSP again. Otherwise we can integrate this flexibility to partially supply the customers into the MDRSP by allowing a slack ring star as for the prize collection concept. However, the slack ring star capacity $q_{d'}$ has to be set to $|U| - r$ and the customer assignment costs to y will be zero. Certainly, also a combination of prize collection and customer coverage can be integrated into the MDRSP.

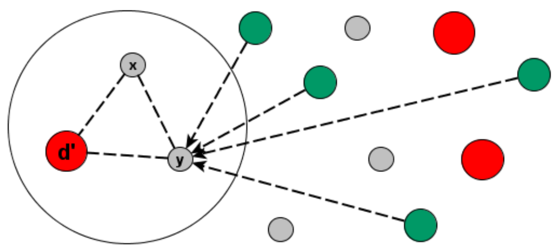


Figure 4: Modeling prize collecting or coverage in the MDRSP. Customers may be assigned to the Steiner node y which is one of two artificial ring Steiner nodes constituting the *slack ring star* served by the auxiliary depot d'.

3) Heterogeneous ring capacities: The MDRSP, as defined above, allows depot-dependent limits for the number of customers per ring star. Since the capacity of a ring star usually depends on the used technology for its physical installation, we might want to consider distinct *types* of ring stars. The MDRSP is capable to model this practical requirement by taking advantage of the individual ring star limits m_d in the following transformation. Assume that for a depot d we are given ring star types $t_1, ..., t_r$ (not necessarily different) with customer capacities $q_1, ..., q_r$. We substitute d by artificial depots $d_1, ..., d_r$, where each of them corresponds to a ring star type and the customer per ring star limits are set to the associated values. The number of allowed rings for a depot d_i corresponds to the number of installable rings of type t_i. Edge costs involving new depots equal the costs for d. Note that this transformation is just valid if $m_d = r$, otherwise we are not able to assure the total ring star limit m_d for d.

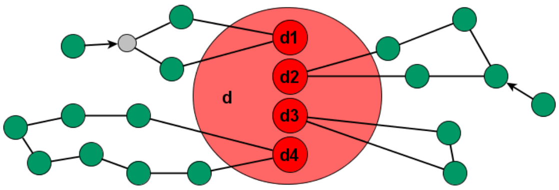


Figure 5: Example for a solution for the MDRSP when modeling heterogeneous customer per ring star capacities for a specific depot. The depot d is replaced by depots d_1, d_2, d_3, d_4.

4) Customer-to-depot assignments: In the definition of the MDRSP we do not allow an assigment of a customer u to a depot d. However, we can model this using a slack ring star R for d as in III-C1. It is of importance to force the slack ring to be built when allowed at non-artificial depots. Since we need to increment the ring star limit of d, omitting R would allow an additional ring star for d which may lead to an infeasible solution. To enforce the installation of R we assume that $q_d > 0$ and force an artificial customer to be assigned to R. Finally, one of the artificial Steiner nodes is set as a potential supplier of u instead of the depot d. Herewith, an obtained solution containing the assignment of u in R should be interpreted as an assignment of u to d.

5) Ring star set up costs: An incremented number of installed ring stars may effect the need for a distributing device with increased capacity. Likewise, ring stars might evoke one-time establishment costs. For a depot d we can integrate a constant ring star set up cost f_d by augmenting the costs for each edge incident to d by $f_d/2$.

IV. Conclusions

In this paper we considered two mathematical models in telecommunication network design and presented highly relevant modeling techniques for their practical application. On the one hand we show the flexibility of the capacitated connected facility location problem as a tree structure based model. On the other hand the multi-depot ring star problem is considered as a representative of reliable network design models guaranteeing one-link failure insensitivity. For both problems we give literature references and relationships to other combinatorial optimization models from the literature. Our main contribution is the elaboration of problem specific modeling techniques for both cases. These are of theoretical and practical interest. Since the modeled side constraints represent requirements of major practical relevance the application of existing solution approaches can be extended to a larger class of problems.

References

[1] E. Balas, "The prize collecting traveling salesman problem," *Networks*, vol. 19, no. 6, pp. 621–636, 1989.

[2] R. Baldacci and M. Dell'Amico, "Heuristic algorithms for the multi-depot ring-star problem," *Eur. J. Oper. Res.*, vol. 203, no. 1, pp. 270–281, 2010.

[3] R. Baldacci, M. Dell'Amico, and J. J. S. González, "The capacitated m-ring-star problem," *Oper. Res.*, vol. 55, no. 6, pp. 1147–1162, 2007.

[4] J. Beasley and E. Nascimento, "The vehicle routing-allocation problem: A unifying framework," *TOP: An Off. J. of the Span Soc. of Stat. and Oper. Res.*, vol. 4, no. 1, pp. 65–86, 1996.

[5] A. Fink, G. Schneidereit, and S. Voss, "Solving general ring network design problems by meta-heuristics," *M. Laguna and J.L.G. Velarde (eds.) Computing Tools for Modeling, Optimization and Simulation*, pp. 91–114, 2000.

[6] D. Karger and M. Minkoff, "Building Steiner trees with incomplete global knowledge," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, 2000, pp. 613 –623.

[7] M. Labbé, G. Laporte, I. R. Martín, and J. J. S. González, "The ring star problem: Polyhedral analysis and exact algorithm," *Networks*, vol. 43, no. 3, pp. 177–189, 2004.

[8] Y. Lee, S. Y. Chiu, and L. Sanchez, "A branch and cut algorithm for a steiner ring star problem," *Internat. J. Management Sci.*, vol. 4, pp. 21–34, 1998.

[9] M. Leitner and G. Raidl, "A lagrangian decomposition based heuristic for capacitated connected facility location," in *Proceedings of the 8th Metaheuristic International Conference (MIC 2009)*, 2009.

[10] Z. Naji-Azimi, M. Salari, and P. Toth, "A heuristic procedure for the capacitated m-ring-star problem," *Eur. J. Oper. Res.*, vol. 207, no. 3, pp. 1227–1234, 2010.

[11] G. Reich and P. Widmayer, "Beyond Steiner's problem: a VLSI oriented generalization," in *WG '89: Proceedings of the fifteenth international workshop on Graph-theoretic concepts in computer science*. New York, NY, USA: Springer-Verlag New York, Inc., 1990, pp. 196–210.

[12] A. Tomazic, "A branch & bound based hybrid heuristic for the multi-depot ring star problem," *submitted*, 2012.

[13] ——, "Novel preprocessing techniques for the connected facility location problem," *submitted*, 2012.

[14] A. Tomazic and I. Ljubic, "A GRASP algorithm for the connected facility location problem," in *SAINT*. IEEE Computer Society, 2008, pp. 257–260.

D. Paper iv

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

Alessandro Hill and Stefan Voß

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 dependent 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 we define an *equi-model matheuristic*. That is a refinement method in which the subproblems are modeled as smaller instances of the global problem. Hence 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.

A. Hill, Institute of Maritime Logistics, Hamburg University of Technology, Schwarzenbergstrasse 95 D, 21073 Hamburg, Germany, E-mail: alessandro.hill@tuhh.de
S. Voß, Institute of Information Systems (IWI), University of Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany, E-mail: stefan.voss@uni-hamburg.de · Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Chile

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. Urban areas need an almost complete coverage with improved service and rural areas at least partially. The arising costs, essentially for excavation and installation, and the corresponding major 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. The CmRSP is closely related to the *capacitated ring tree problem* [9] and is a special case of the recent *ring tree facility location problem* [1]. 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. As a generalization of the classical *traveling salesman problem* (TSP) the MDRSP is NP-hard.

Real-world instances of the MDRSP and related problems commonly involve several hundred customers that have to be supplied. As for the CmRSP, customer nodes represent houses when using a fiber-to-the-building model (FTTB) and represent even more single households when modeling a fiber-to-the-home scenario (FTTH). Potential network branches such as crossings result in Steiner nodes which leads to large instances even for middle-sized cities. For further details on modeling and examples we refer to [3]. 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 most 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.

A solution for a MDRSP is depicted in Figure 1. 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 homogeneous, 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 [12] and [5]. The *ring star problem* was introduced by [11] 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 [4] 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 [10]. The heuristic of [15] 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 [14]. The two-index cut set formulation of [4] was extended for the MDRSP by [3] 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

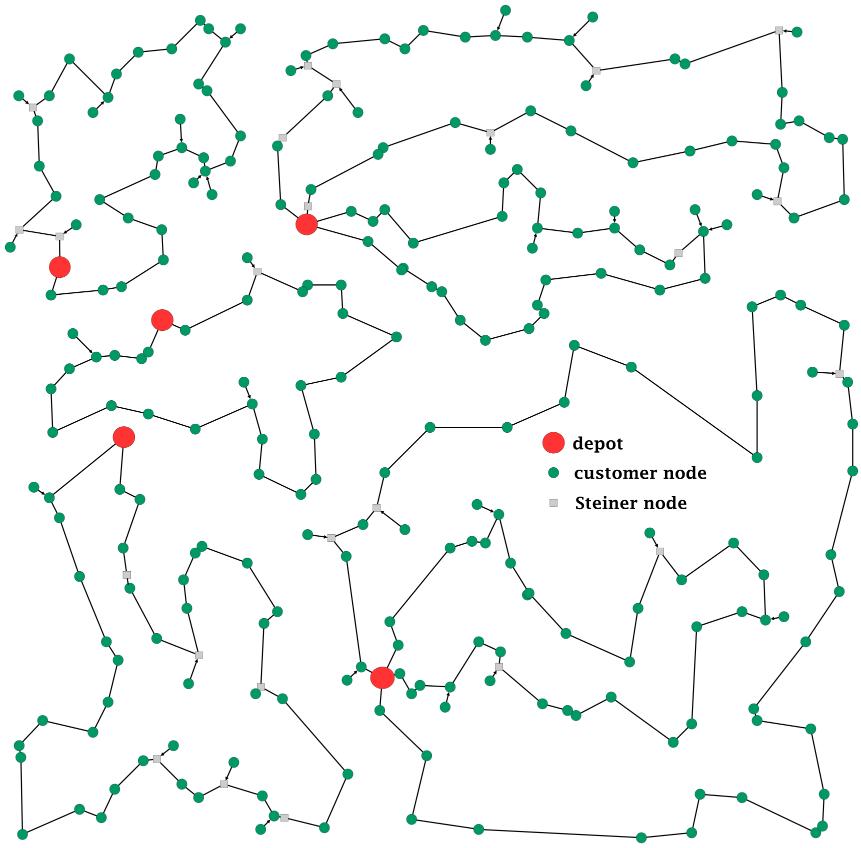


Fig. 1 MDRSP solution using five depots (instance B-100, $|U| = 250$, $m = 2$, $q = 40$)

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 [17] 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 [7]. They extract partial routes from an existing solution and reinsert them according to the exact solution of an ILP-based reallocation problem. [19] carried out this technique for the vehicle routing problem. In the work of [16] 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. Another hybrid refinement method by [2] based on ILP subproblems for an *inventory routing problem* was able to improve existing heuristic results. A framework for variable fixing-based refinements that incorporates local branching is presented in [8].
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 (hence the name *equi-model matheuristic*). 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. The corridor method is a related matheuristic approach that aims at the exploration of generic neighborhoods obtained by the imposition of exogenous constraints (or corridors) on the decision space of the target problem [18]. For some references regarding matheuristics the reader is referred to [13].

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. [4] 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 [3]. For a node set $S \subseteq V$ we denote the set of incident edges in G by $\delta(S)$. For a single node set $\{v\} \subset V$ we may write $\delta(v)$ for $\delta(\{v\})$.

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

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

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

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

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

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

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

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

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

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

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 [4].

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

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

The separation of these integrality cuts for the single depot case was carried out in [4]. 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 [4], out of

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

Ring multi-star cuts The ring multi-star cuts were introduced by [4] 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

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

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

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

We also add the following known valid inequalities as done before for related problems [11, 4].

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

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

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).

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

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:

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

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 Section 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 Section 4.3 since it is based on the latter ideas. The overall process is illustrated in Section 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 Section 4.1.1 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.

4.1.1 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.

(I) 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 4 illustrates such a selection.

(II) 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 4.

(III) Path star exchange: Following the path star idea we build two close clusters on distinct rings and join them. See Figure 5 for an example of such a clustering.

(IV) 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 $(U \cup W)\{v\}$ 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 3.

(V) 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 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 2 shows an example for such a cluster.

To enable the following problem construction, some cleaning steps for the

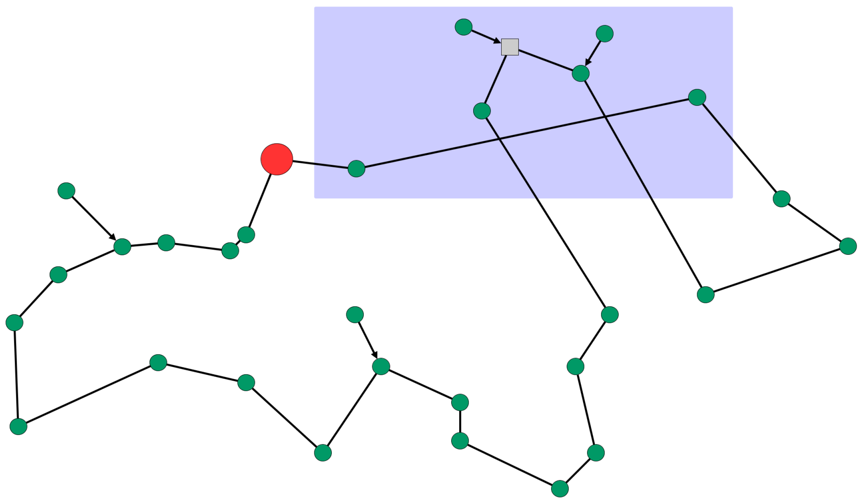


Fig. 2 Ring edge cluster (V) in a single ring star

cluster nodes are needed. Let S denote a current solution and C a selected cluster:

- 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.

4.1.2 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. We note that the solutions obtained after embedding the local solutions into the complementary structure of the incumbent solution form the considered neighborhood. An edge or a node might need to be forced to be part of the solution in the modeling process.

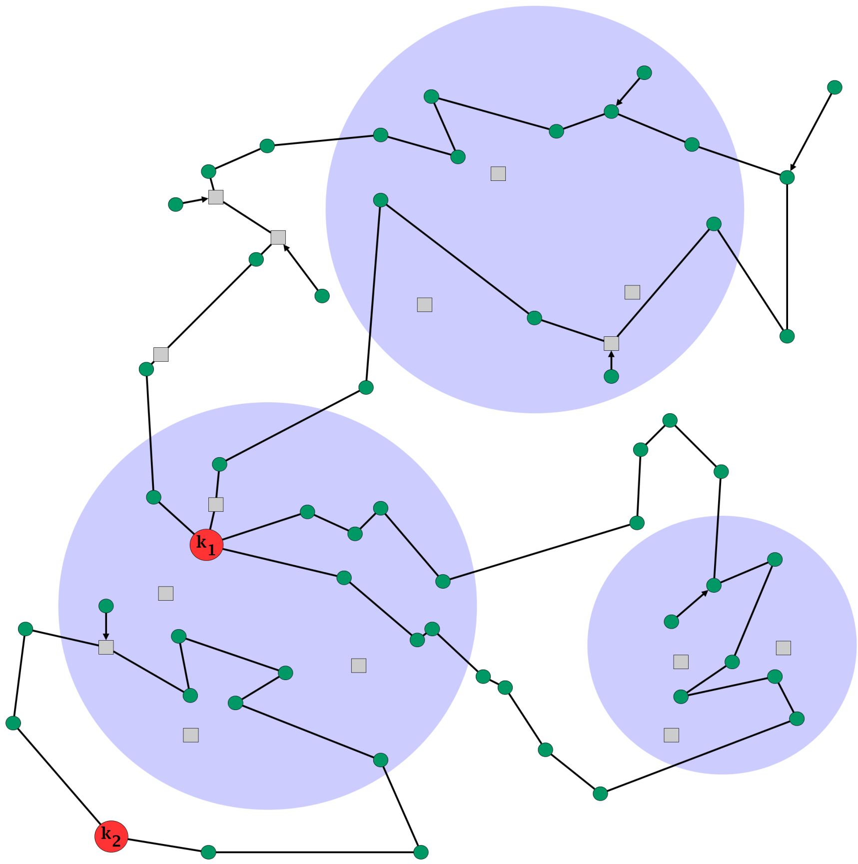


Fig. 3 Example for ball clustering (IV) in a network with 3 ring stars using the depots k_1 and k_2

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 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 6 illustrates such a

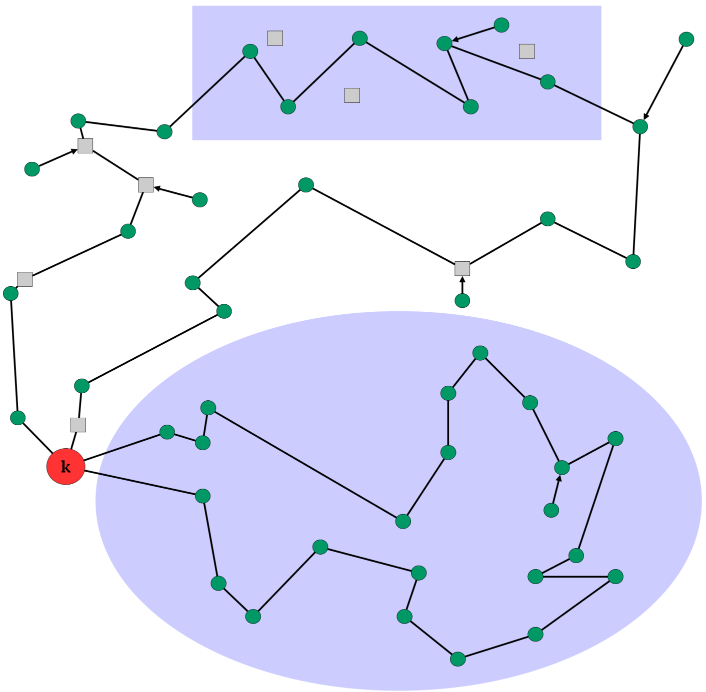


Fig. 4 Example for path star/ring star clustering (II)/(I) in a network with 2 ring stars connected to the depot k

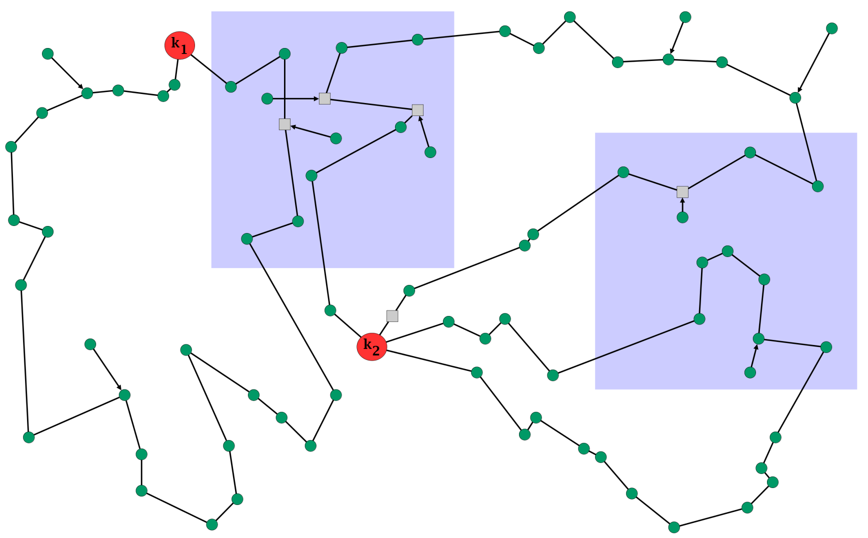


Fig. 5 Path star exchange clusters (III) in a network with 3 ring stars connected to depots k_1 and k_2

construction. If a path star T' that is substituted by a leg does not contain

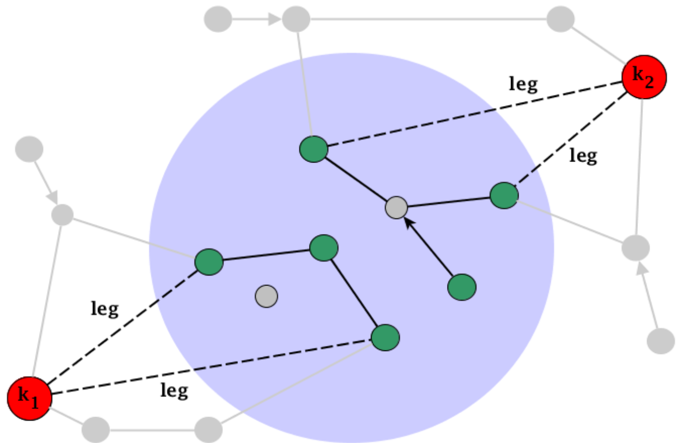


Fig. 6 Ball cluster leading to a ring star problem with two depots k_1 and k_2

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 7). 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 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.

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

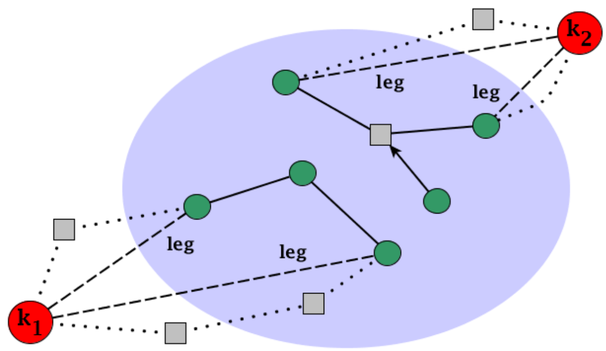


Fig. 7 Cluster with reconnectable zero-customer legs and their modeling from depots k_1 and k_2

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 h 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 [4] 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 kh to be present in the solution. Set zero cost for the edge hu and big M cost for other edges incident to h. For any other cluster node v let $c_{uv} = c_{hv}$. The integration of this aspect is significant when considering clusters containing ring star fragments that are close to depots. Figure 8 illustrates such a transformation.

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 allow connections to all the cluster nodes by edges of original routing cost. Its

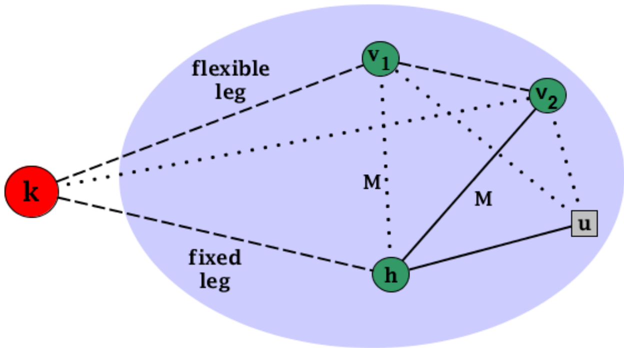


Fig. 8 Enabling one leg (represented by edge $\{k, v_1\}$) to be flexible for the depot k in the subproblem using big M edge costs while the other leg (represented by edge $\{k, h\}$) is fixed

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 9 illustrates this model modification.

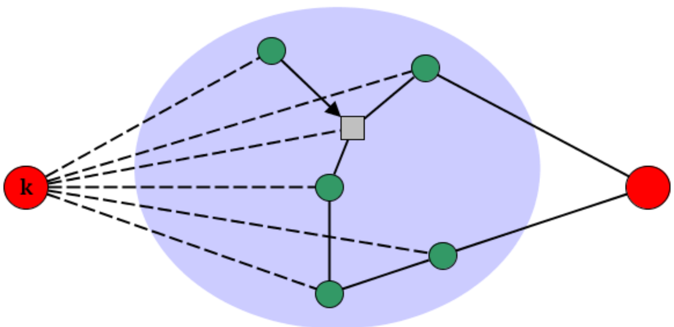


Fig. 9 Enabling ring stars from a depot k that is currently not connected to cluster nodes

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 [3] are largely 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.

4.1.3 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. In such a *phase* we perform multiple locally exact searches for a cluster type successively differing in their parameterization (e.g. different center center nodes). 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 a cover 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 well-distributed centers heuristically first, maximizing their mutual distance sum.
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 subproblems of a phase would be the cost of such a look ahead strategy.

4.2 Contraction-based 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 ($\text{CMDVRP}_{\text{het}}$) and homogeneous demands ($\text{CMDVRP}_{\text{hom}}$) with an underlying homogeneous vehicle fleet. As for the extraction based local searches in Section 4.1 a customer node limit (max_U) is used to keep the computational effort manageable.
Given a MDRSP solution S, we apply this technique locally according to the following subnetworks.

(VI) 2-Ring star: Consider the network restricted to two ring stars.

(VII) Depot: Consider the ring stars connected to a depot in S

(VIII) Global: Apply the contractions to the entire network S

We perform each variant using the following two types of contractions (Figure 10).

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 $\text{CMDVRP}_{\text{hom}}$ 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 $\text{CMDVRP}_{\text{het}}$ 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.

Furthermore, we consider recombinations that are not based on the underlying current solution as follows. Again, we build optimization subproblems as above, but do not cluster based on node distances in the incumbent ring star network. These clusters are built according to the node distances in the

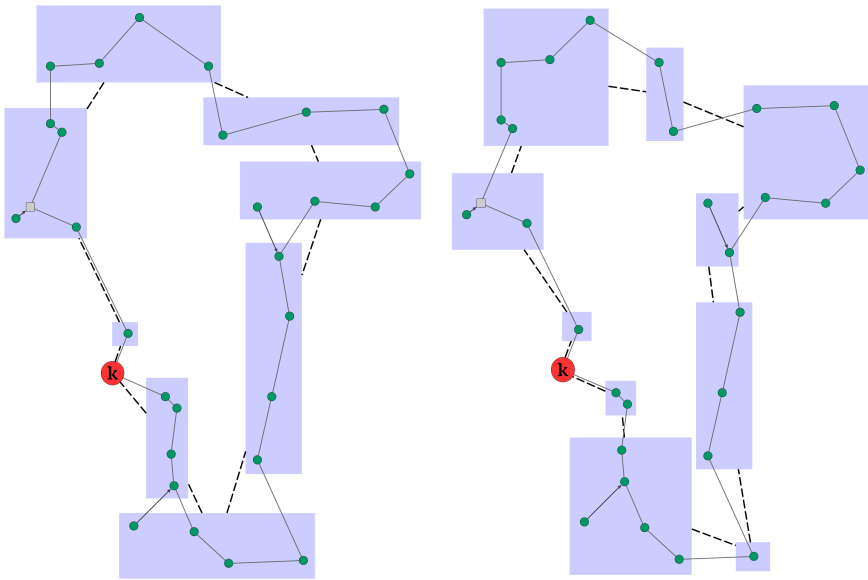


Fig. 10 Examples for the contraction ideas (VI)-(VIII) on given solutions and the induced current routing structure: fixed cluster size 4 (left) and flexible cluster size (right)

original graph G of the problem instead. The resulting optimization problems do not differ from the ones above. Given a cluster size, we greedily select $\lceil |U_R|/max_U \rceil$ distributed customers of $|U_R|$ by repeatedly adding the farthest customer. Starting with theses 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 ignored since we focus on the assignment of customers to the depots. However, the obtained networks undergo a subsequent local refinement phase before being compared to the incumbent. The created problem might turn out to be infeasible due to capacity restrictions and we proceed to the next step. 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 within the clusters are not reusable in general. Therefore, we solve a TSP on each ring using the nearest neighbor heuristic, followed by 2-opt and 3-opt searches and the exact local improvement procedures from Section 4.1 (ring star, path star, ball, path star exchange). 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 contraction-based 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 multi-start 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 contracted 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 Section 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

After having explained the specific elements of our approach, we summarize the overall strategy combining the various concepts described above. We distinguish between techniques changing the structure of a single ring star [(I),(II),(III)] and those allowing ring star interactions [(IV),(V),(VI),(VII),(VIII)]. 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 11.

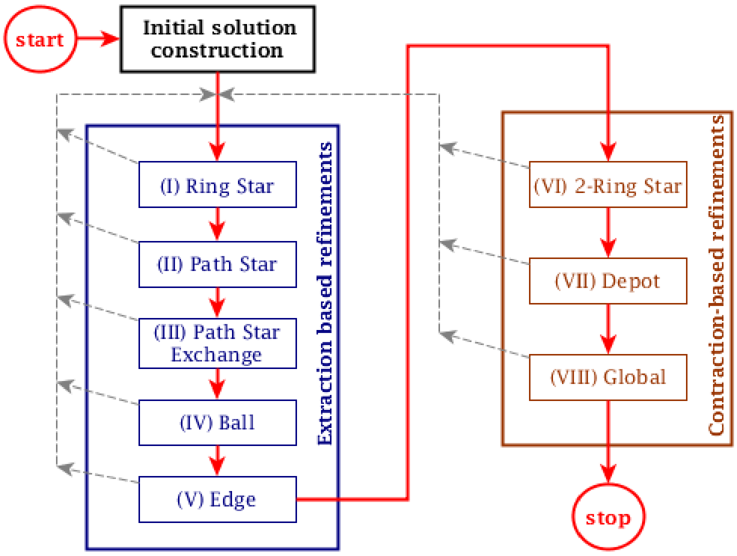


Fig. 11 Flow chart showing the overall optimization strategy

5 Computational study

Our algorithm is implemented in C++ using CPLEX 12.2 as ILP solver carrying out the branch & cut framework. Computations are performed on an Intel i5 U470 1.33GHz processor unit with 4 GB working memory available. CPLEX is set to run in the single thread mode. The memory usage for computations of our implementation does not exceed 100 MB in the performed tests. The ILP solver and computing power dependent parameter calibration is described in Section 5.1. For each ILP a time limit of 90 seconds is applied to avoid a runtime explosion due to solver-hard subproblems. This is rarely encountered, mostly when solving the routing subproblems for the contraction 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 $\text{CMDVRP}_{\text{het}}$ are solved by a branch & cut algorithm. Clustering is done using the minimum distance criterion for adding customers. The number of overall iterations is limited to 25, which is not reached in our experiments.
It should be noted that the results of our method are certainly machine and solver dependent. However, we are convinced that the integration of a state-of-the art MIP solver always comes with the drawback of uncertainty regarding replication. As pointed out in [6], several minor modeling and implementation details may have a major effect on the performance of a MIP solver based algorithm. As a consequence, any deviating intermediate result provided by the MIP solver is likely to guide our search into an alternate local optimum.

	Clustering	Max. number of cluster nodes
(I)	Ring Star	55
(II)	Path Star	20
(III)	Path Star Exch.	17
(IV)	Ball	22
(V)	Edge	22

	Contraction	Max. number of final customers (max_U) (flexible/fixed size)
(VI)	2-Ring-Star	14/22
(VII)	Depot	10/15
(VIII)	Global	10/16

Table 1 The parameters used within the local refinement techniques (I)-(VIII)

5.1 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 type of subproblem exceeds the runtime of two seconds in an instance we fix the corresponding parameter. Since cluster size limit variations mutually impact the subproblem behaviour, we rather use this approach as a rough solver capacity estimation. Table 5.1 shows the parameters that we used for testing the algorithm. We list both the bounds for the number of nodes in a cluster for techniques (I)-(V) and the target number of nodes for reoptimizing after the application of the contraction techniques (VI)-(VIII).

5.2 Results on literature instances

To assess the quality of our approach we use random instances that were generated by [3] to test their heuristics. Given uniformly distributed node coordinates in the plane, the distances are set to be Euclidean. These 276 medium sized problems contain up to 300 customers, 750 Steiner nodes and 5 depots. Various combinations of q, m and $|D|$ are considered. They are divided into classes A and B where q is 30 and 40, respectively. 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 the three heuristics in [3].
Tables 2 and 3 show the objective function values of our solutions (obj) for several instances listed in [3]. Additionally, the relative improvements in percent compared to the best results obtained by [3] are given in column Δ. Column Ω gives the number of solved ILPs, column σ the number of improvements and column t(s) the algorithm runtime. 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 three 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 still observe 8%. On average we improve the current

Table 2 Results for instance class A ($q = 30$) and relative improvement (Δ) compared to [3]

P	$\lvert U \rvert$	$\lvert W \rvert$	$\lvert D \rvert$	m	obj	Δ	Ω	σ	t(s)
A-1	100	200	2	2	47051	1.9	801	31	1204
A-2					42185	4.2	620	24	509
A-3					40115	0.9	447	18	149
A-4					42544	1	615	27	258
A-5					41440	1	449	19	277
A-6					41725	3	578	22	623
A-7		250			42983	1.5	697	23	242
A-8					41793	0.8	435	17	253
A-9					43946	0.4	425	15	340
A-10					41797	0.6	925	19	326
A-11					43800	0.6	1053	16	373
A-12					42120	1.4	436	15	182
A-13	150	300	3		51442	1.5	1051	30	427
A-14					54746	-0.6	432	25	689
A-15					53660	1.6	513	21	532
A-16			2	3	61482	-0.1	789	33	1056
A-17					53296	0.3	1029	29	469
A-18					52727	1.1	325	16	157
A-19			4	2	50252	6.2	249	15	207
A-20					52661	-0.7	395	24	590
A-21					51357	1.1	993	33	510
A-22			2	3	57432	0.4	1759	47	1182
A-23					59326	-0.6	669	22	884
A-24					55874	3.1	728	29	603
A-25				4	52421	2.6	628	25	463
A-26					55209	2.3	1224	33	1124
A-27					51799	4.7	1462	36	658
A-28					54302	-0.2	816	25	928
A-29					54540	1.4	486	24	753
A-30					53513	1.7	1157	40	705
A-31		375	3	2	49336	2.7	1075	27	628
A-32					49262	2.9	647	18	378
A-33					52387	2	508	25	926
A-34			2	3	51619	1.4	966	30	560
A-35					55750	-0.2	585	20	413
A-36					55769	3.5	449	19	472
A-37			4	2	48963	8	1738	37	618
A-38					48518	7.9	1620	34	501
A-39					47878	5.4	932	36	462
A-40			2	3	52078	0.7	1065	26	684
A-41					55837	0.5	369	28	339
A-42					55396	1.6	1302	44	898
A-43				4	54934	2.8	954	45	532
A-44					53335	1.9	762	37	637
A-45					52717	0.7	608	38	604
A-46					50563	5.2	563	28	214
A-47					55497	3.1	364	26	406
A-48					56089	1.7	761	38	649
A-49	200	400	4	2	60133	1.6	822	42	474
A-50					62916	2.3	1076	42	1084
A-51					60101	2	1300	41	1061
A-52			3	3	56942	2.9	926	33	773
A-53					59388	2.6	2599	47	1109
A-54					58993	2.2	1611	36	1208
A-55			2	4	66569	1.9	1529	47	1187
A-56					63215	2.6	1940	53	1164
A-57					65426	-0.9	1757	45	1103
A-58			5	2	62155	4.6	838	29	547
A-59					62839	6.1	1325	34	876
A-60					60863	4.4	2002	37	1193
A-61			3	3	57631	1.8	1033	40	559
A-62					68473	1	527	33	520
A-63					59143	-0.7	2089	39	1454
A-64			2	4	70434	0.5	1320	53	1093
A-65					64362	1.8	1220	40	710
A-66					64558	0.2	741	40	1019
A-67		500	4	2	62417	2.7	1350	55	1274
A-68					58520	3.1	1348	53	957
A-69					67930	0.4	1106	45	678
A-70			3	3	60328	2	888	41	921
A-71					60856	4.5	1286	49	993
A-72					64508	1	545	35	448
A-73			2	4	69625	-0.7	1981	45	1506
A-74					64391	0.3	1722	45	1065
A-75					64371	1.9	641	41	646
A-76			5	2	58317	0.6	1441	44	943
A-77					57648	6	801	34	709
A-78					59692	3.1	1324	41	645
A-79			3	3	64374	0	1049	33	1204
A-80					59534	-0.7	1785	42	852
A-81					60452	2.9	428	26	359
A-82			2	4	69623	0.9	1635	51	1476
A-83					68559	-0.5	1472	28	892
A-84					64874	-0.1	1239	32	1021
A-85	250		5	2	67004	2.2	2101	63	1244
A-86					76053	-1.2	1030	41	1278
A-87					66997	3.6	1038	40	997
A-88			3	3	74177	-1.1	1351	49	1341
A-89					75457	0.8	1245	72	1384
A-90					75866	-1.5	1525	31	1109
A-91			4		68793	1	1135	50	1230
A-92					66586	4.7	1069	59	1071
A-93					70588	5.1	1709	58	1465
A-94			3	4	69220	4.6	1461	63	1509
A-95					81000	1.1	2503	80	1199
A-96					69019	3	1536	59	1362
A-97		625	5	2	67496	5.9	2876	48	1157
A-98					66055	-1.1	1993	52	1148
A-99					64307	6.2	1154	38	673
A-100			3	3	81620	-1.1	586	53	942
A-101					82710	0.2	1886	42	979
A-102					69917	2.1	1410	42	1044
A-103			4		68738	0.9	1586	44	1039
A-104					63510	7.5	2083	51	1084
A-105					64872	5.8	1502	52	1235
A-106			3	4	76479	2.2	1814	50	1146
A-107					72660	0.4	983	70	757
A-108					70744	1.6	2309	65	1331
A-109	300	600	4	3	70081	1.9	1962	56	1206
A-110					73804	7.6	1727	58	1404
A-111					80235	1.1	1455	61	1252
A-112			3	4	81094	2.7	2229	66	1201
A-113					85892	-0.8	1399	59	1208
A-114					76871	3.1	3531	99	1457
A-115			5	3	78559	2.8	1327	71	1476
A-116					72859	2.3	2523	67	1198
A-117					76313	1.4	1091	52	1222
A-118			3	4	89131	-2.6	1946	71	1202
A-119					80751	4.1	1002	63	1424
A-120					82723	1.3	1890	64	1369
A-121		750	4	3	77565	0.9	1111	55	1151
A-122					80552	4.3	1377	46	1242
A-123					73077	3.3	1927	56	1104
A-124			3	4	78750	4	1847	74	1169
A-125					83593	0.9	2402	53	1449
A-126					83225	2.1	1712	60	1376
A-127			5	3	75033	3.5	4386	94	1215
A-128					76194	4.9	1728	62	1279
A-129					72859	3.5	1814	60	1341
A-130			3	4	77074	0.8	1436	63	1313
A-131					86321	-0.1	2635	64	1333
A-132					79753	-0.8	1470	59	1537

Table 3 Results for instance class B ($q = 40$) and relative improvement (Δ) compared to [3]

P	$\|U\|$	$\|W\|$	$\|D\|$	m	obj	Δ	Ω	σ	$t(s)$
B-1	100	200	2	2	43298	0.4	267	14	103
B-2					39318	2.8	861	26	176
B-3					38229	4.2	393	13	61
B-4					41042	4.5	937	16	221
B-5					40676	1.2	1147	37	240
B-6					40513	2	606	13	116
B-7					40917	2.1	894	14	402
B-8					42305	3.1	580	13	150
B-9					41064	2.5	749	16	270
B-10		250			40519	1.9	433	12	119
B-11					40507	2	587	14	214
B-12					41429	0.8	739	11	271
B-13					40220	2	770	16	178
B-14					42190	0.6	960	21	296
B-15					39901	0.8	824	20	252
B-16					40417	7	1536	24	311
B-17					41698	4.8	853	32	474
B-18					40627	4.2	530	21	147
B-19	150	300			51453	0.2	1840	40	1088
B-20					50845	2	994	21	432
B-21					52625	0.8	1544	35	1502
B-22			3		48846	5	792	19	214
B-23					49228	3.2	1764	35	710
B-24					51095	3.2	1314	32	834
B-25			2	3	52083	0.7	597	21	319
B-26					53578	1.8	751	24	446
B-27					52106	1.6	910	22	359
B-28					49833	2	779	28	336
B-29					48374	3.2	653	23	269
B-30					48895	3.7	2274	45	1450
B-31		375		2	50168	1.1	885	25	1177
B-32					49458	5.4	1202	26	942
B-33					48185	2.5	831	29	314
B-34			3		50124	1.9	1314	31	497
B-35					49131	3.5	883	19	219
B-36					48738	3.5	1173	27	272
B-37			2	3	48733	4.5	1067	21	383
B-38					51181	2.6	1240	32	447
B-39					52646	0.8	2803	52	1391
B-40					52470	2.4	1165	37	438
B-41					49745	5.3	719	36	300
B-42					50990	3.4	1368	40	360
B-43	200	400	3	2	56109	2.6	4133	36	1302
B-44					55796	5.7	1133	37	525
B-45					58907	4.4	891	24	413
B-46			2	3	58660	3.8	1325	42	798
B-47					61006	1.7	1007	34	505
B-48					57799	0	2791	46	1129
B-49			4	2	57728	2.7	2015	38	1350
B-50					57703	6.3	2684	48	1150
B-51					59576	3.4	841	36	404
B-52			2	3	59726	0.9	794	28	654
B-53					61752	0.9	1413	42	665
B-54					61653	1.1	1581	49	1402
B-55				4	63847	4.2	795	25	629
B-56					59539	6.4	1879	44	1151
B-57					58908	4.8	1020	23	754
B-58					60225	3.1	978	26	634
B-59					57860	6	1254	35	842
B-60					57932	6.6	2094	25	1340
B-61		500	3	2	59721	0.5	1617	33	983
B-62					56335	2.6	2075	31	1090
B-63					60389	1	1876	42	1352
B-64			2	3	62023	1.6	961	32	759
B-65					60120	1.9	3093	46	1474
B-66					60686	1.6	851	24	559
B-67			4	2	55483	7.1	1838	24	1062
B-68					55090	7.3	1839	36	576
B-69					57629	2.4	1474	35	862
B-70			2	3	59370	0.5	1061	35	741
B-71					58569	2.4	788	25	554
B-72					58789	-0.6	1231	42	752

P	$\|U\|$	$\|W\|$	$\|D\|$	m	obj	Δ	Ω	σ	t(s)
B-73				4	63328	0.5	1504	44	1312
B-74					63818	0.1	2994	55	1039
B-75					61253	-0.8	3340	42	1222
B-76					61615	2.6	903	27	359
B-77					66336	0.8	1558	37	1111
B-78					60349	2.4	2615	41	1109
B-79	250		4	2	63161	3.1	2231	50	1463
B-80					64416	1.6	2952	53	1403
B-81					63312	1.9	2212	43	1017
B-82			2	4	70942	3.2	1853	56	1315
B-83					67619	3.2	1727	43	1035
B-84					63433	3.8	2648	60	1403
B-85			5	2	62920	4.6	2675	46	1021
B-86					63439	8.7	3159	47	1469
B-87					64691	3.2	1984	41	1146
B-88			3	3	64853	4	1562	39	574
B-89					66508	4.9	2676	50	1237
B-90					64646	5.5	3123	50	1361
B-91			2	4	65663	4.4	1686	56	1018
B-92					70955	2.5	3377	55	1494
B-93					65817	3.4	970	46	871
B-94		625	4	2	62130	6.2	2494	44	1460
B-95					59974	2.8	2818	41	864
B-96					63252	4	2969	56	1235
B-97			2	4	70723	3.6	1521	51	1122
B-98					66049	2.1	2688	67	1320
B-99					69543	-0.5	2219	77	1465
B-100			5	2	63763	6.9	3022	82	1418
B-101					62377	6.1	2456	45	1272
B-102					61573	5.4	1680	31	1173
B-103			3	3	67023	2.8	3860	70	1345
B-104					64406	2.5	2223	46	957
B-105					65471	3.1	3083	56	1195
B-106			2	4	69046	1.3	1001	39	1470
B-107					66262	1.7	3981	64	1549
B-108					76071	0.9	4646	51	1125
B-109	300	600	5	2	69244	6	3247	72	1374
B-110					68938	4	1816	57	1237
B-111					69300	4.7	2223	45	1062
B-112			3	3	74306	1.4	3497	81	1192
B-113					68388	4.2	3946	69	1336
B-114					77137	1.2	1786	52	1506
B-115			2	4	80458	0.6	1453	25	1044
B-116					79425	2.6	2633	44	1123
B-117					73814	1.2	2368	41	1430
B-118			4	3	69512	4	2553	70	1335
B-119					68566	9.1	1954	64	1452
B-120					72801	4.4	2528	47	1384
B-121			3	4	74264	4.7	2921	56	1270
B-122					72250	9.8	5969	81	1093
B-123					73991	4.7	2355	48	1711
B-124					73431	4.5	2505	65	1480
B-125					68686	4	3627	59	1214
B-126					71475	4.4	2619	47	1309
B-127		750	5	2	67685	6.2	2480	73	1223
B-128					73016	3.2	2310	65	1513
B-129					70948	4.6	3508	58	1415
B-130			3	3	75423	0.8	1895	60	1214
B-131					78466	0.5	4121	77	1503
B-132					71846	5.2	2705	74	1479
B-133			2	4	71343	2.6	2408	75	1308
B-134					72656	2.1	1889	61	1296
B-135					82240	0.1	1261	62	1236
B-136			4	3	68501	4.5	2584	53	1495
B-137					73401	5	4487	48	1147
B-138					71444	4.5	1946	50	1236
B-139			3	4	71648	4.8	3393	45	1203
B-140					75802	4.7	2397	58	1482
B-141					71824	4.4	4607	83	1469
B-142					69879	13.4	1871	53	1345
B-143					71394	4.7	2751	77	1212
B-144					69514	6.9	3272	71	1344

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 complex multiple ring star exchange movements which pays off on those instances. The runtime spent on the instances reported in [3] is about 38% less than the one of our method on average. Since cluster selection heuristics are at most of quadratic complexity by far the most of the computational time is used for the solution of the ILPs. Our algorithm involves the solution of up to 6645 ILPs per instance (1900 on average) and identifies up to 99 local improvements for an instance. Figure 12 shows the effectiveness of the different search techniques by the number of their improvements per instances. The total numbers of refinements were: (I):3913; (II):297; (III):4923; (IV):5687; (V):580; (VI):575; (VII):68; (VIII):14.

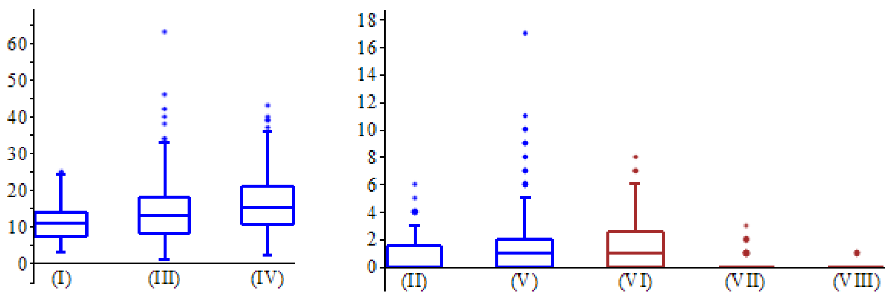


Fig. 12 Numbers of refinements per search type per instance

5.3 Performance analysis

From their results, [3] conclude that class A instances are harder to solve due to the lower ring star customer capacity (30). We agree on this point, arguing that it is harder to improve their results by our approach on average. However, it is possible 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 contraction 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 addressed 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 was 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 itself. Furthermore, to enable global restructuring of the network we developed contraction based techniques. Again, the corresponding optimization was done by the application of exact methods. Therefore, we solved vehicle routing problem variants to optimality. The latter techniques were combined with 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. We suppose that our 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.

Acknowledgements We are indebted to Roberto Baldacci for providing the test instances, detailed results and the correspondence. Furthermore, we thank Kenneth Sörensen for his valuable comments.

References

1. 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. LAGOS'15 - VIII Latin-American Algorithms, Graphs and Optimization Symposium.
2. 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.
3. 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.
4. R. Baldacci, M. Dell'Amico, and J. J. Salazar González. The capacitated m-ring-star problem. *Operations Research*, 55(6):1147–1162, 2007.
5. 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.
6. M. Fischetti and M. Monaci. Exploiting erraticism in search. *Operations Research*, 62(1):114–122, 2014.
7. 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.
8. A. Hill and S. Voß. Generalized local branching heuristics and the capacitated ring tree problem. Research Paper 2014020, University of Antwerp, Faculty of Applied Economics, September 2014.

9. A. Hill and S. Voß. Optimal capacitated ring trees. *EURO Journal on Computational Optimization*, online available, DOI: 10.1007/s13675-015-0035-x, 2015.
10. E. A. Hoshino and C. C. de Souza. A branch-and-cut-and-price approach for the capacitated m-ringstar problem. *Discrete Applied Mathematics*, 160(18):2728 – 2741, 2012.
11. 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.
12. G. Laporte and I. R. Martín. Locating a cycle in a transportation or a telecommunications network. *Networks*, 50(1):92–108, August 2007.
13. V. Maniezzo, T. Stützle, and S. Voß, editors. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming.* Springer, Berlin, 2009.
14. 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.
15. 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.
16. 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.
17. 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.
18. M. Sniedovich and S. Voß. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics*, 35(3):551–578, 2006.
19. 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.

E. Paper v

EURO J Comput Optim
DOI 10.1007/s13675-015-0035-x

ORIGINAL PAPER

Optimal capacitated ring trees

Alessandro Hill · Stefan Voß

Received: 9 July 2014 / Accepted: 11 February 2015

Abstract We study a new network design model combining ring and tree structures under capacity constraints. The solution topology of this capacitated ring tree problem (CRTP) is based on ring trees which are the union of trees and 1-trees. The objective is the minimization of edge costs but could also incorporate other types of measures. This overall problem generalizes prominent capacitated vehicle routing and Steiner tree problem variants. Two customer types have to be connected to a distributor ensuring single and double node connectivity, respectively, while installing optional Steiner nodes. The number of ring trees and the number of customers supplied by such a single structure are bounded. After embedding this combinatorial optimization model in existing network design concepts, we develop a mathematical formulation and introduce several valid inequalities for the CRTP that are separated in our exact algorithm. For a set of literature-derived instances we consider various reliability scenarios and present computational results.

Keywords Capacitated ring tree problem · Steiner tree · Ring tree · Vehicle routing · Survivable network design · Integer programming

Mathematics Subject Classification 90C11 · 90C27 · 90C90

A. Hill (✉)
Department of Engineering Management, University of Antwerp, Prinsstraat 13, 2018 Antwerp, Belgium
e-mail: alessandro.hill@uantwerpen.be

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

Published online: 06 March 2015

Introduction

In supply network design as well as telecommunications, the graph class of trees is widely used as a base structure to model optimization problems. Typically, a set of specified customer nodes has to be connected to a central distributor by a selection of supply edges for which individual installation costs apply. In a natural way there exists a tree that minimizes the overall connection costs when considering this basic setting. The determination of such a tree is well-known as the spanning tree problem (SPTP). However, many real-world networks would allow the establishment of links that do not necessarily connect two customers directly, but utilize optional intermediate nodes. The usage of these Steiner nodes might be either essential for the network connectivity, result in an overall cost reduction or be non-advantageous. Although providing a broader applicability, the resulting well-known steiner tree problem (STP) (Voß 2006) is more challenging as its complexity is known to be NP-hard.

A crucial requirement for the design of networks in various applications is the ability to provide reliable service to the customers. Even after a link failure due to technological or environmental reasons the customer connectivity might be highly desirable in the remaining network. Since trees can be characterized as graphs in which two nodes are connected by a unique path, any missing link disconnects the network. To overcome this weakness, the ring structure has proven to be a suitable option because of its 2-connectivity property: after the removal of any single edge the graph is still connected.

The model that we introduce in this work fills a gap in the existing literature. We bring together the tree structure and the ring structure under additional capacity constraints. In this capacitated ring tree problem (CRTP) we are given two categories of customers that have to be connected to a central distributor by using optional Steiner nodes. We say customers are of type 2 if they require a link-failure reliability with respect to the distributor, sometimes also called 1 + 1 protection. The remaining customers are labeled as of type 1 and need simple connectivity. Albeit, the latter might be equipped with additional reliability if this is favorable in terms of the overall network cost. We want to find a set of rings that intersect in the distributor node and contain all type 2 customers. At the same time the remaining type 1 customers have to be connected to these ring structures by forming trees or be ring nodes themselves. Such an individual structure that is connected to the distributor, either a pure tree or a ring with its attached trees, is called a ring tree. We impose two capacity limits on the resulting network: the number of these ring trees as well as the number of customers in such a ring tree are bounded. We allow these ring trees to be pure trees that are directly attached to the distributor but count each incident non-ring edge as one ring tree. The objective is to minimize the overall costs for the installed edges. Figure 1 illustrates a solution for the CRTP implementing four ring trees.

To the best of our knowledge the modeling of this advanced ring extension structure under capacity constraints has not been explicitly considered in the literature so far. By allowing the assignment of trees to rings in the CRTP we present the first research approach into this direction. A major strategical planning feature of these models is the anterior two-type categorization of the customers. Ring tree problems find applications in the design of telecommunication networks. The CRTP can be used in local access

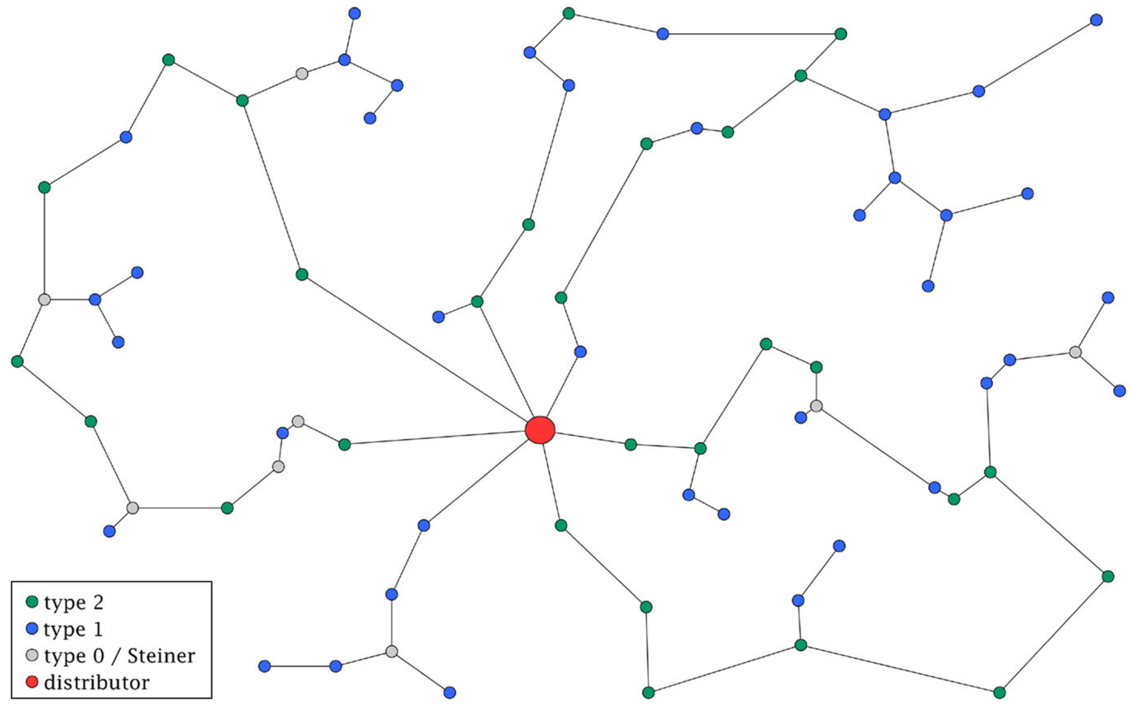


Fig. 1 A solution for a capacitated ring tree problem based on 4 ring trees (3 of which are 1-trees and one is a tree)

network design, for modeling backbone networks or even combining these levels. It can be used to integrate ring-based reliability in recent real world applications which are based on steiner trees (Grötschel et al. 2013). In transportation network planning, we can represent ship routes by rings and simultaneously model the inter-modal freight distribution networks from their ports of call. Here, we see another particular strength of the model in the ability of linking two strategical levels.

Given the above exposition, the idea of this work is to provide a new type of problem, the CRTP, which allows for generic treatment and extended understanding of developing new algorithmic approaches for the CRTP as well as some of the arising subproblems. This paper is structured as follows. A formal definition of the CRTP is presented in Sect. 2 with the notation used throughout this work. In Sect. 3 we relate our new model to existing concepts in the greater network design literature. In Sect. 4 we present our mathematical formulation on which the exact algorithm is based. Along the description of our exact algorithm in Sect. 5 we develop valid inequalities and show in detail how these can be separated. In our computational study we apply our algorithms on literature-derived test instances for different reliability scenarios. The results and an analysis of the impact of reliability variation are provided in Sect. 6. We close with our conclusions in Sect. 7.

The capacitated ring tree problem

Before giving a formal definition of the capacitated ring tree problem we introduce the base topology of the CRTP in a graph theoretic manner. Throughout this work we denote the node set of a graph G as $V[G]$, its set of edges by $E[G]$ and the arc set by $A[G]$ if G is directed. Recall that a 1-tree can be characterized as a connected undirected graph containing a unique cycle.

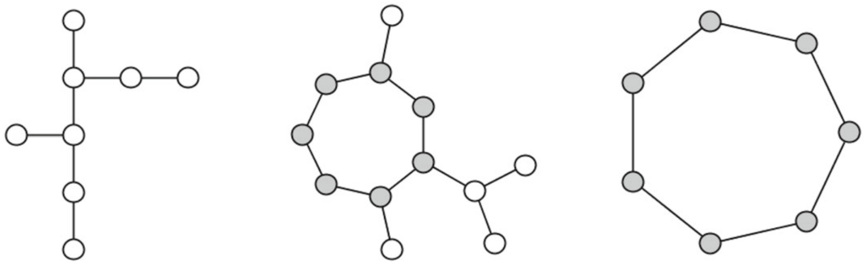


Fig. 2 Some ring trees and their fundamental subcycles

Definition 1 A ring tree is a connected graph containing at most one cycle.

In other words, a ring tree is a connected graph Q with at most $|V[Q]|$ edges. Therefore, the graph class of ring trees is the disjoint union of trees and 1-trees. Since the class of cycle graphs is included in 1-trees, ring trees generalize both, rings and trees. We recall that 1-trees have been proven useful for deriving lower bounds and solution techniques for the classical TSP (Held and Karp 1971).

Given a tree T we can create $(|V[T]|^2 - |V[T]|)/2 - |E[T]|$ distinct subcycles in T by the insertion of single chords which are called fundamental cycles. Figure 2 depicts examples for the ring tree structure and fundamental cycles. Similarly, we can define a directed ring tree as a directed graph that is either an arborescence or the union of a directed (fundamental) cycle C and arborescences rooted in $V[C]$.

Definition 2 We are given an undirected complete simple graph G. Its node set is the disjoint union of type 1 customers U_1, type 2 customers U_2 and Steiner nodes W, complemented by a distributor node d: $V[G] = U_1 \dot{\cup} U_2 \dot{\cup} W \dot{\cup} \{d\}$. Each edge $e \in E[G]$ is associated with a non-negative weight c_e. Let a ring tree limit m and a customer per ring tree limit q be given. For a set of ring trees $S = \{Q_1 \subseteq G, \ldots, Q_k \subseteq G\}$ we denote the network graph by $N_S = (\bigcup_{Q \in S} V[Q], \bigcup_{Q \in S} E[Q])$. S represents a solution for the CRTP if

- each type 1 customer is contained in exactly one ring tree,
- each type 2 customer is contained in exactly one ring tree's fundamental cycle,
- each steiner node is contained in at most one ring tree,
- the number of ring trees k is at most m,
- the number of customers in a ring tree does not exceed q, and
- for each ring tree, d is either a degree-two cycle node or a leaf if no fundamental cycle is present.

The CRTP asks for a solution of minimal total cost, i.e. minimized sum of edge costs $\sum_{e \in E[N_S]} c_e$.

Note that following our definition of the CRTP we allow the direct assignment of trees to the distributor. It is easy to see that requiring every distributor-outbound structure to link back to it would favor solutions containing Steiner rings which we want to avoid here. We assume that the distributor has the same capacity consumption through a tree serving a certain number of customers as it has by serving a ring (tree) with equally many customers. When applying the customer limit we consider each tree induced by an edge incident to d individually. Some ring-based models require m to be met exactly (e.g. Baldacci et al. 2007), which we relax here for the sake of overall cost efficiency. We define U to be the set of customers $U_1 \cup U_2$ that require to be contained in a solution and assume that $mq \geq |U|$ since the CRTP instance is obviously infeasible. The NP-hardness of the CRTP follows from its reducibility to the travelling salesman problem (see Sect. 3), for instance. Figure 1 above illustrates a solution for the CRTP.

Related models

In this section, we show the originality of the CRTP by summarizing relationships to existing related network design models. We focus on the models with an overall edge cost minimization objective function and do not address various extensions such as price-collecting problems or revenue maximization. Figure 6 illustrates the relationships between the models mentioned in the following. In addition to the 1-connectivity required for type 1 nodes and the 2-connectivity for the type 2 nodes, we denote the optional Steiner node usage as a 0-connectivity requirement.

Ring models

The ring component of the CRTP is used in classical capacitated vehicle routing problems (VRPs) to represent vehicle routes. The CRTP reduces to a unit-demand VRP with a homogeneous vehicle fleet when all nodes ($\neq d$) are of type 2. As a consequence, when $m = 1$ and $q \geq |V[G]|$ the CRTP generalizes the prominent travelling salesman problem (TSP), asking for a Hamiltonian cycle of minimal total edge costs. The steiner travelling salesman problem (STSP) (Letchford et al. 2013) asks for a cost minimal tour in which an edge may be traversed multiple times as illustrated in Fig. 3. Moreover, we pay the edge cost for each of the edges in a solution network, which is generally not a simple graph. Since the CRTP does not admit edges to be used multiple times we cannot relate the STSP to it in a straightforward way. For a set of predefined clusters, the generalized travelling salesman problem (GTSP)

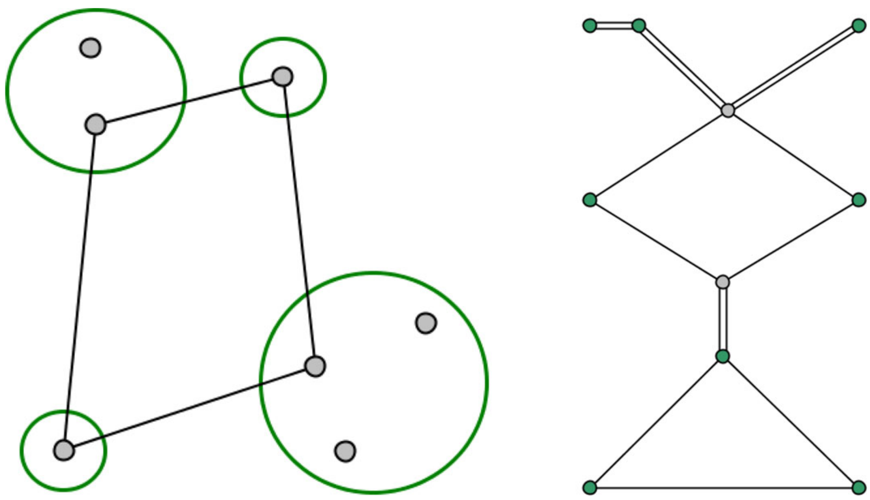


Fig. 3 A generalized travelling salesman tour (*left*) and a solution for the steiner travelling salesman problem (*right*)

(Fischetti et al. 1997) asks for a ring that just includes one node of each cluster rather than all of them. Figure 3 illustrates such a non-spanning tour. Obviously, the GTSP is a TSP if all the clusters are of order one it can only be modeled by the CRTP in this special case. In contrast to most routing models we allow Steiner nodes when designing ring trees in the CRTP. A collection of related vehicle routing models and existing exact algorithms can be found in Baldacci et al. (2010).

Tree models

The CRTP generalizes the (rooted) capacitated minimum spanning tree problem (CSPTP) with unit node demands. The CSPTP asks for a minimum spanning tree in which the sum of given node demands in each subtree induced by an edge incident to the distributor is bounded by σ. A CSPTP can be formulated as a CRTP with $m = \infty$, $q = \sigma$, $U_2 = W = \emptyset$ and U_1 containing all the non-distributor nodes. A survey on heuristics for related problems can be found in Amberg et al. (1996). The minimum capacitated Steiner tree problem (CSTP) shares the cardinality constraints but allows the usage of Steiner nodes in the network. We note in passing that an explicit consideration of the CSTP is somewhat lacking in literature. When even relaxing the ring tree capacity constraints, this problem is equivalent to the STP.

Ring star models

A ring that is extended by single node assignments is known to follow the ring star pattern (Labbé et al. 2004) as illustrated in Fig. 4. Each node either belongs to a ring or is a leaf node of degree 1. An efficient layout then usually means the interlinkage of customers to a central distributor by (disjoint) ring stars such that the overall edge

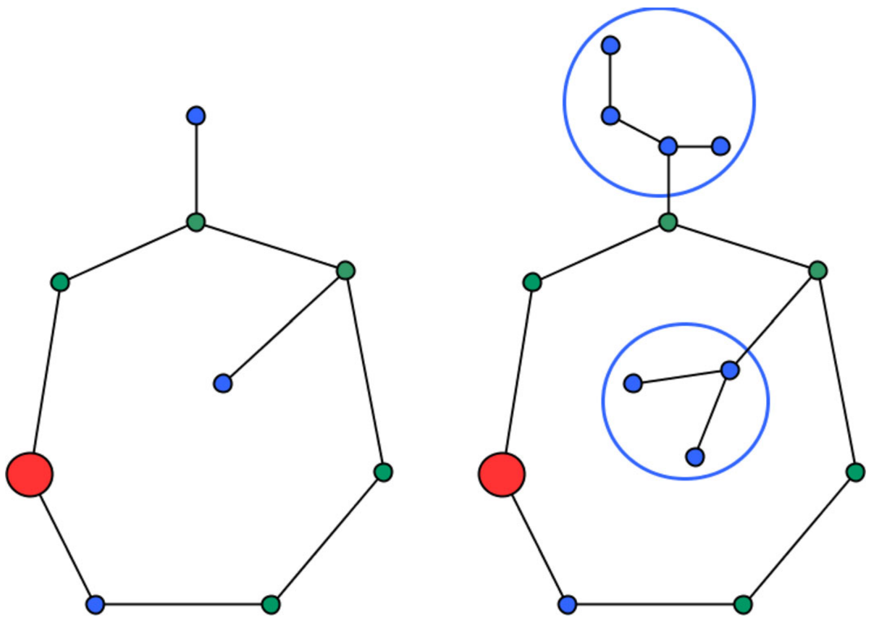


Fig. 4 A CRTP approximating ring star network (*left*) and its realization using the ring tree structure (*right*)

costs are minimized. Due to practical requirements capacity limits may apply to the number of customers per ring star or the number of installed ring stars (Baldacci et al. 2007). In this capacitated ring star problem (CRSP) the customers that are allowed to be assigned to rings are given in advance. The CRTP goes beyond this idea by replacing single customer assignments by assignments of trees but does not generalize the CRSP. In ring star problems the allowed assignments of customers to the rings are commonly the result of a previous optimization-based modeling step. Once a solution is at hand, the actual assignment is realized by the installation of a shortest path from the assigned type 1 customer to its chosen ring supplier. Multiple such paths are possibly implemented by a combining tree structure as illustrated in Fig. 4. Hence, the optimization potential is fully utilized in the CRTP by the integration of the design of the type 1 customer assignment structures into the overall model. With an increasing rate of the latter customers we magnify the overall cost-saving potential compared to the described two-step approach.

In the travelling purchaser problem (TPP) (Ramesh 1981), a cost-efficient tour has to be designed to purchase required products at selected markets. These products can be obtained from various markets at different prices. A decision to purchase a certain product at a market on the route can be interpreted as a product assignment to a route that includes this market, resulting in a ring star structure. In Gouveia et al. (2011), an extension is considered in which the tour length as well as the number of assignments per market are restricted. However, in the TPP the assignable products cannot be tour nodes whereas a type 1 customer can be a ring node in the CRTP and the CRSP, respectively.

Survivable network design

Requiring a certain degree of connectivity between network nodes is the basic concept in survivable network design problems (SNDPs). The survivability of a node is either measured by the number of edge-disjoint paths to the remaining network or the stronger node-disjoint paths. In the CRTP, these underlying connectivity requirements with respect to the distributor are of order 0, 1 and 2, depending on the node type. They are typical for low-connectivity-constrained survivable network design problems (Stoer 1992; Fortz et al. 2000). However, the CRTP enforces a ring tree topology whereas SNDP models do not restrict the obtained network structure as long as the connectivity requirements are fulfilled. Figure 5 gives examples for optimal SNDP topologies that result from the given survivability requirements and the edge cost structure. Related models, polyhedral results and solution methods can be found in Stoer (1992) and Kerivin and Mahjoub (2005). Due to its rather generic survivability requirement, special cases including regular survivability and bounded survivability got particular attention. Some results with a special focus on low redundancy are summarized in Fortz (2000). The numerous suitable applications for SNDP-based models motivated their extensions to design networks that satisfy various supplementary requirements. These additional restrictions are largely of capacity-bounding type which reflect technological or business limitations. Well-established representatives are node degree constraints, hop constraints, diameter constraints, node/edge supply

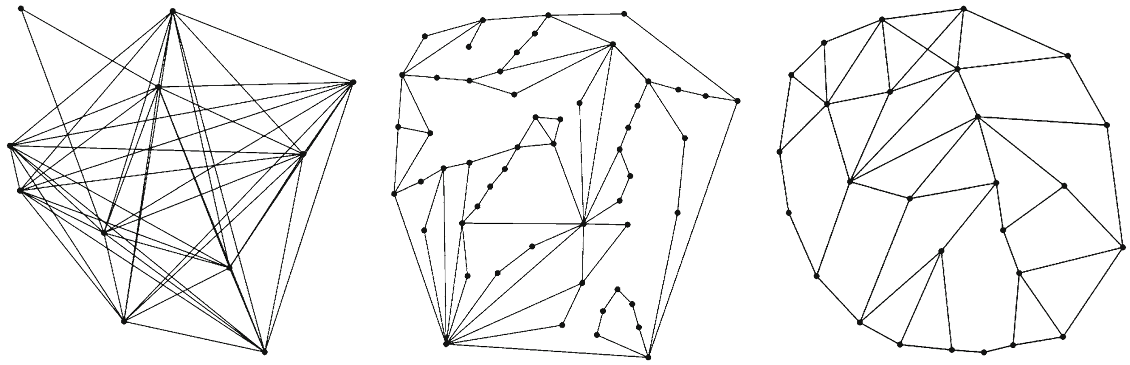


Fig. 5 SNDP solution topologies for SNDlib instances ZIB54, DFN-GWIN and SUN Orlowski et al. (2010)

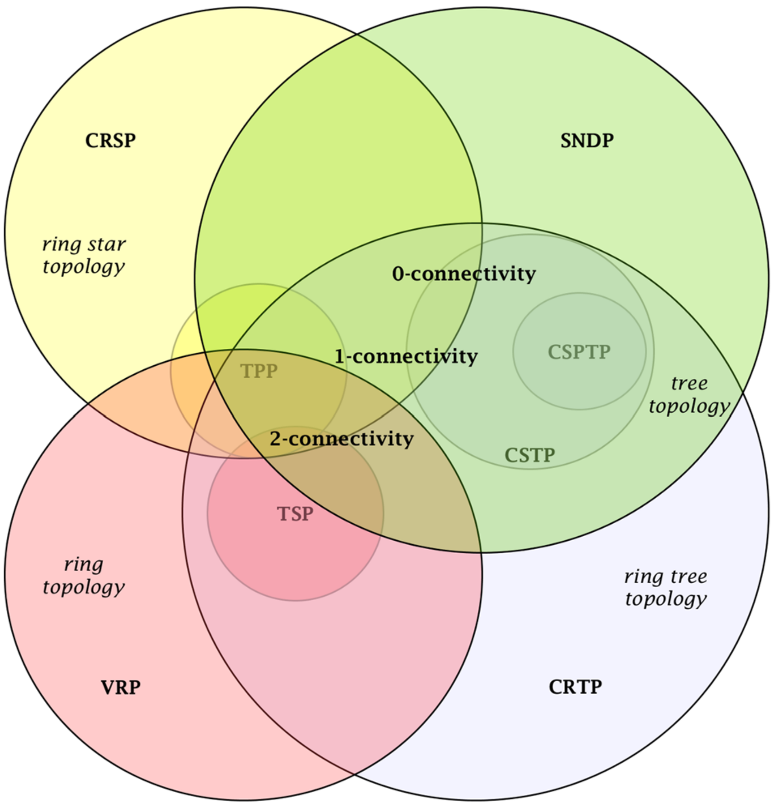


Fig. 6 The capacitated ring tree problem and related network design models

capacity constraints, cardinality constraints, mesh constraints and their combinations. In Fortz et al. (2000), the authors introduce a capacity constraint on the number of customers on the rings in a two-connected network to bound the rerouting distances in the case of a link failure. Several network design models can be considered as SNDPs with imposed capacity constraints. Figure 6 summarizes the major problems and problem classes discussed in this section. It also puts the CRTP into perspective.

Mathematical formulation

We present a mathematical model for the CRTP that is based on a directed network representation. Since non-compact formulations were shown to be computationally more efficient than flow-based formulations in many cases (e.g. Baldacci et al. 2007) we propose a 2-index cut set formulation. Advanced branch and cut techniques for an efficient algorithm are developed in the next section. As concluded in Magnanti and Wolsey (1994), the LP lower bounds obtained by a directed formulation of the steiner tree problem are at least as good as their counterparts from the undirected case. Similarly, this holds for directed formulations of vehicle routing problems. Therefore, we formulate the CRTP based on the complete orientation of G, denoted by H. The resulting forward and backward arcs are assigned the cost of the corresponding edge in $E[G]$. We search for a solution based on directed ring trees which can be transformed into a solution of the CRTP by definition. A binary variable x_a indicates whether an arc a is used in such a directed representation. The installation of a forces a corresponding binary edge variable y_e to take value 1. A continuous circulation flow variable $f_a \in [0, 1]$ takes value 1 if the arc a is part of a directed ring and 0 otherwise. Our directed formulations might also be used for an asymmetric capacitated ring tree problem (ACRTP) that we will not further investigate in this paper. The CRTP can be formulated as a steiner arborescence problem with additional side constraints. To achieve this, artificial sink nodes have to be introduced that represent terminals for the arborescence rooted in the distributor whenever a directed path is closed to a ring. However, we decided to develop a separate model for the CRTP without such a reformulation to underline its importance in its own right.

In our mathematical formulation we occasionally use ij to denote an arc (i, j) for the sake of simplified notation. For two disjoint node sets $X, Y \subset V[H]$ in a directed graph H, we define $\delta_Y^+(X) = \{(i, j) \in A[H] : i \in X, j \in Y\}$ and $\delta_Y^-(X) = \{(i, j) \in A[H] : i \in Y, j \in X\}$. If clear from context we may omit to mention Y in the case that $V[H] \backslash X \subseteq Y$. For $X = \{i\}, i \in V[H]$, we may use $\delta_Y^-(i)$ and $\delta_Y^+(i)$, respectively. We also use $X(Y) = X \cap Y$ for denoting intersecting sets, as for example the customers $U(S)$ in a node set $S \subseteq V[G]$.

$$\text{minimize} \quad \sum_{e \in E[G]} c_e y_e \tag{1}$$

$$\text{subject to} \quad \sum_{a \in \delta^-(S)} x_a \geq \frac{|U(S)|}{q} \quad \forall\, S \subset V[H] \backslash d, \tag{2}$$

$$\sum_{a \in \delta^-(i)} x_a = 1 \quad \forall i \in U, \tag{3}$$

$$\sum_{a \in \delta^-(i)} x_a \leq 1 \quad \forall i \in W, \tag{4}$$

$$\sum_{a \in \delta^+(d)} x_a \leq m, \tag{5}$$

$$x_{ij} + x_{ji} = y_{ij} \quad \forall \{i, j\} \in E[G], \tag{6}$$

$$\sum_{a \in \delta^-(i)} f_a = \sum_{a \in \delta^+(i)} f_a \quad \forall i \in V[H], \tag{7}$$

$$\sum_{a \in \delta^-(i)} f_a = 1 \quad \forall i \in U_2, \tag{8}$$

$$0 \leq f_a \leq x_a \quad \forall a \in A[H], \tag{9}$$

$$x_a \in \{0, 1\} \quad \forall a \in A[H], \tag{10}$$

$$y_e \in \{0, 1\} \quad \forall e \in E[G]. \tag{11}$$

Our cut set formulation is based on binary arc variables x_a for the arcs in $A[H]$. The assignment constraints (3) ensure an in-degree equal to one for each customer, whereas the capacity constraints (4) limit the inbound arcs to one for each Steiner node. The capacitated connectivity constraints (2) bound the number of customers per ring tree to q. We model underlying circulation structures by arc flow variables f_a and (in)equalities (7), (8) and (9). Since we consider directed ring trees, inequality (5) is sufficient to limit the number of ring trees to m. To obtain a simple undirected solution network and identify its edges we implement the variable linking equalities (6). When $q = |U|$ and $U_2 = \emptyset$ the right hand side of (2) is bounded by 1, leading to a well-known cut set formulation for the STP. If $q = |U|$ and $U_1 = W = \emptyset$ then we obtain a corresponding model for the TSP. Although we are just dealing with a total of $3|E[G]|$ variables we are faced with an exponential number of constraints of type (2). The objective (1) measures the network cost by summing up the costs of installed edges.

We note that each edge variable y_{ij} could be eliminated by adding the constraint $x_{ij} + x_{ji} \leq 1$, dropping inequalities (6) and using the objective function $\sum_{a \in A[H]} c_{ij} x_{ij}$. However, computational tests showed that the y variables had a positive effect on the number of explored nodes as well as the overall computation time.

A CRTP variant that considers a different cost function for edges on fundamental cycles than for edges of attached trees can be modeled by modifying the objective. Let c_e^r be the cost of a ring edge $e \in E[G]$ and $c_{e'}^t$ the cost of a non-ring edge. Then the total cost of a ring tree design can be measured by replacing (1) by the following objective function.

$$\sum_{e=\{i,j\} \in E[G]} \left[c_e^r (f_{ij} + f_{ji}) + c_e^t (y_e - f_{ij} - f_{ji}) \right] \tag{12}$$

Exact solution techniques

In this section, we develop an efficient branch & bound algorithm based on our non-compact mathematical formulation in Sect. 4. An emphasis is put on bound-tightening, which we achieve by CRTP specific cutting techniques and solution polishing. These two matters are crucial for the efficiency of a mathematical programming-based approach as extensively discussed in the literature (e.g. Mitchell 2009). For various hard combinatorial optimization problems the most competitive algorithms rely on the application of sophisticated cutting planes combined with efficient primal heuristics.

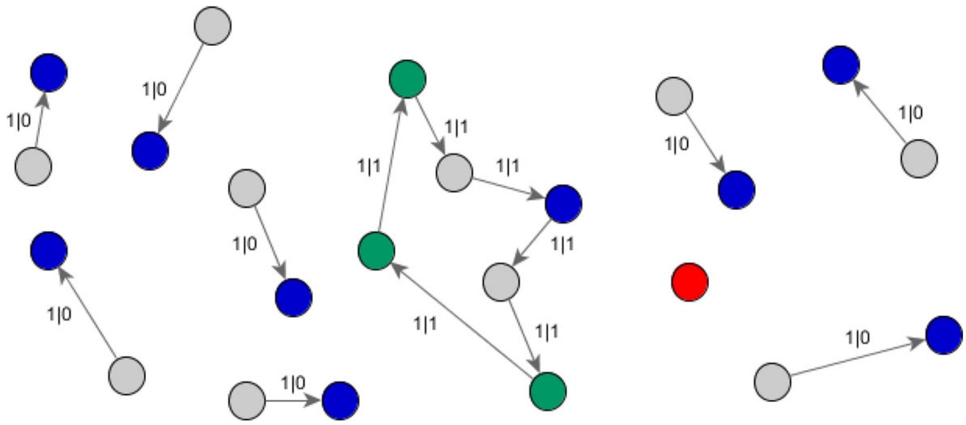


Fig. 7 A typical solution of a LP-relaxed CRTP in the directed formulation before the enforcement of the capacitated connectivity constraints (2)

Strengthening the lower bounds

In the following, we present valid inequalities and corresponding separation techniques to improve the lower bounds during the branch and cut algorithm. Due to the specific CRTP topology we combine cutting planes based on ideas from network design models for trees and vehicle routing. In the special cases that $U_2 = \emptyset$ or $U_1 = \emptyset$ some of our valid inequalities collapse to equivalent ones for the STP or the VRP, respectively. Let LP denote the linear program obtained after relaxing the integrality of variables x and y in our formulation. We consider an optimal fractional arc solution for the LP-relaxed subproblem in the branch and bound tree as the assignment of values $x^* : a \in A[H] \rightarrow [0, 1]$ and f^* for the circulation flow, respectively. Such a typical solution combines characteristics from the Steiner tree problem with VRP typical subtours. In Fig. 7 a solution of a LP relaxation is depicted before the separation of inequalities (2). For a more convenient formulation of the inequalities we introduce a continuous auxiliary ring node variable z_i for each node $i \in V[H]\backslash\{d\}$ that identifies i as a fundamental cycle node. These variables are linked to the node's total inbound circulation flow as follows.

$$z_i = \sum_{a \in \delta^-(i)} f_a \quad \forall\, i \in V[H]\backslash\{d\} \tag{13}$$

Inasmuch as $z_i = 1$ holds $\forall\ i \in U_2$ we are more interested in the connectivity of type 1 customers and steiner nodes with respect to d. Optimal ring node values complementing x^* and f^* are denoted by z^*. The different node types in the CRTP give rise to various cut arc configurations for a given node subset. We refer to the illustration in Fig. 8 along our descriptions.

Circulation inequalities The inequalities (9) that link the circulations to the ring arc variables can be further tightened for mandatory cycle nodes through (14).

$$f_a = x_a \quad \forall\, a \in \delta^-(i), \quad i \in U_2 \cup \{d\} \tag{14}$$

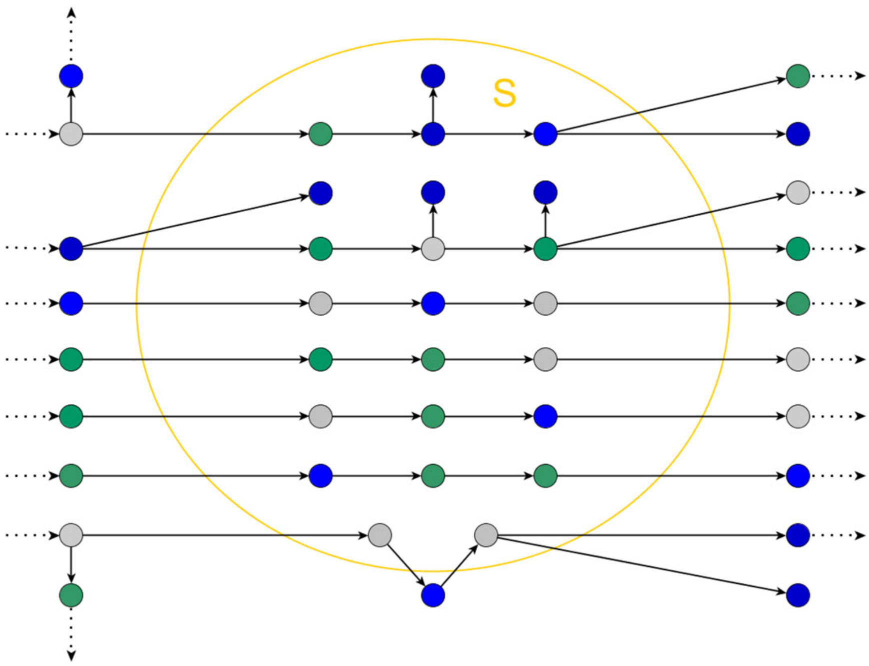


Fig. 8 A CRTP cut set and examples for the various types of intersections with ring tree structures considered by cutting planes

In the CRTP, we even allow non-type-2 nodes to obtain double connectivity by being a ring node. In terms of our formulation such a node $i \in U \cup W$ is equipped with reliability if there is circulation flow entering i, i.e. $z_i > 0$. If this is the case, then the cycle structure requires a unique subsequent ring node on i's ring. Thus, there is at most one natural ring node (type 2 customer or distributor) connected by an arc from i.

$$\sum_{a \in \delta^+_{U_2 \cup \{d\}}(i)} x_a \leq z_i \quad \forall i \in V[H] \backslash d \tag{15}$$

To avoid reverse circulation flow we can require the outbound circulation flow from j to nodes in $V[H] \backslash i$ to be at least the circulation flow f_{ij} on each arc $(i, j) \in A[H]$.

$$f_{ij} \leq \sum_{a \in \delta^+_{V[H] \backslash \{i\}}(j)} f_a \quad \forall (i, j) \in A[H] \tag{16}$$

Since there are $|A[H]|$ such inequalities (16) we separate them dynamically by a straightforward arc search.

Connectivity inequalities The following inequalities are also well-known as subtour elimination constraints and impose a unitary lower bound on the right-hand side of (2).

$$\sum_{a \in \delta^-(S)} x_a \geq 1 \quad \forall S \subset V[H] \backslash d : i \in S, \forall i \in U \tag{17}$$

To separate (17) for a customer i we compute a directed $d - i$ cut (D, S) of minimal weight w in H with respect to arc weights x^*. If $w < 1$ then we add an inequality (17) for the cut set S.

Capacitated connectivity inequalities Inequalities (2) dominate (17) if $U_2(S) > q$. The separation of (2) requires the computation of a minimal $d - s$ cut (R, S) in a directed auxiliary graph $H^{-}_{1,2}$ with node set $V[H^{-}_{1,2}] = V[H] \cup \{s\}$ and arc set $A[H^{-}_{1,2}] = A[H] \cup \{(i, s) : i \in U\}$. An arc (i, s) has weight $1/q$ $\forall i \in U$ and the remaining arcs have weight x^*_{ij}. S is a violating cut set if the obtained cut weight is less than $|U|/q$.

In fact, we can even assume that the sum of the inbound arc variable values in (2) is integer to get stronger rounded versions.

$$\sum_{a \in \delta^-(S)} x_a \geq \left\lceil \frac{|U(S)|}{q} \right\rceil \quad \forall S \subset V[H] \backslash d \tag{18}$$

If S violates (2) it necessarily violates (18) and then we add this stronger inequality instead.

Capacitated ring tree multi-star inequalities Furthermore, we introduce several capacitated ring tree multi-star inequalities for the CRTP which generalize (2). For a set of nodes S not including d, we additionally estimate the number of distinct customers in $U \backslash S$ that are connected to a node in S to ensure a sufficient number of arcs entering S. Due to the (ring) tree topology such a customer can be incident to multiple arcs in $\delta^-_{U \backslash S}(S)$. Hence counting all these inbound customer arcs generally results in an overestimation of the number of inbound customers. Nevertheless, we can give a lower bound on the number of inbound customers for a given subset X of S by calculating the inbound customer circulation flow $\sum_{a \in \delta^-(X)} f_a$. Moreover, for each type 2 customer $i \in X$ we can replace an in-flow variable f_{ji} by the in-arc variable x_{ji}. At the same time we obtain a lower approximation using the fact that the out-degree of a customer node is at most q. Thus, we can sum over all the inbound customer arc variables while correcting by dividing through the decremented ring tree customer capacity $q - 1$. These two arguments are combined in the following inequalities on $X \subseteq S$ and its complementary set $S \backslash X$ in S.

$$\sum_{a \in \delta^-(S)} x_a \geq \frac{1}{q} \left(|U(S)| + \sum_{a \in \delta^-_{U \backslash S}(U_2(X))} x_a + \sum_{a \in \delta^-_{U \backslash S}(X \backslash U_2)} f_a + \frac{1}{q} \sum_{a \in \delta^-_{U \backslash S}(S \backslash X)} x_a \right) \quad \forall X \subseteq S,\ S \subset V[H] \backslash d \tag{19}$$

These inequalities are similar to partial multi-star inequalities known for the VRP. We are able to efficiently separate these CRTP specific inequalities for a fixed set $X \subset V[H] \backslash d$. The separation of inequalities (19) is based on the minimal cut computation for (2) with modified arc costs in $H^{-}_{1,2}$. We set the weight for an arc $a \in U \times U_2(X)$ to $(1 - 1/q)x^*_a$ and for $a \in U \times (X \backslash U_2)$ to $x^*_a - f^*_a/q$ using the fact that $f^*_a \leq x^*_a$. The arc weight for $a \in U \times (V[H] \backslash X)$ is $(1 - 1/q^2)x^*_a$. The sets we selected are inspired by the cut arc node type combinations illustrated in Fig. 8. More precisely, we enforce (19) for $X \subseteq \{\bigcup_{L \in P} L : P \in \mathcal{P}(\{W, U_1, U_2\})\}$ resulting in at most eight different types of inequalities. When adding such a cut we can replace the maximal out-degree q by $\min\{q, |S|\}$ and if $S \cap W = \emptyset$ by $\min\{q-1, |S|\}$.

An alternative way to strengthen (2) is to count arcs leaving S towards customers not in S since they consume capacity. Actually, even a non-ring arc (i, j) in $\delta^+_W(S)$ implies at least one more customer since there exists an optimal solution without Steiner leave nodes. However, this customer might be already incorporated as a node in S. Such potential ears with respect to S are the reason that we cannot relate inbound and outbound arcs at the same time. In contrast to (19), every customer that is reached from S can be counted without approximation as follows.

$$\sum_{a \in \delta^-(S)} x_a \geq \frac{1}{q} \left(|U(S)| + \sum_{a \in \delta^+_{U \setminus S}(S)} x_a \right) \quad \forall S \subset V[H] \setminus d \tag{20}$$

Note that the separation of inequalities (20) is NP-hard since it is equivalent to finding a directed cut of maximal weight.

Rounded ring tree multi-star inequalities Although rounding the right hand side of (20) results in further dominating valid inequalities, the constraint linearity would be violated. So we use the techniques from Baldacci et al. (2007) to derive linear inequalities through an estimate as follows. Lemma 1 of Baldacci et al. (2007) states that for integers $(\alpha, \beta, \gamma) \in \mathbb{N}^3$ with $\alpha > \gamma > 0$ and $\alpha \mod \gamma \neq 0$ the inequality $\lceil \frac{\alpha-\beta}{\gamma} \rceil \geq \lceil \frac{\alpha}{\gamma} \rceil - \frac{\beta}{\alpha \mod \gamma}$ holds. We use this after rewriting the summation terms for the case that $|U| > q$ and $|U| \mod q \neq 0$. Note that if $|U| \leq q$ then we deal with an instance that is effectively uncapacitated.

$$\begin{aligned}
\sum_{a \in \delta^-(S)} x_a &\geq \left\lceil \frac{1}{q} \left(|U(S)| + \sum_{a \in \delta^+_{U \setminus S}(S)} x_a \right) \right\rceil \\
&\geq \left\lceil \frac{1}{q} \left(|U(S)| + \left\lceil |U \setminus S| - \sum_{i \in U \setminus S} \sum_{a \in \delta^-_{V[H] \setminus S}(i)} x_a \right\rceil \right) \right\rceil \\
&\geq \left\lceil \frac{1}{q} \left(|U| - \sum_{i \in U \setminus S} \sum_{a \in \delta^-_{V[H] \setminus S}(i)} x_a \right) \right\rceil \\
&\geq \left\lceil \frac{|U|}{q} \right\rceil - \frac{1}{|U| \mod q} \sum_{i \in U \setminus S} \sum_{a \in \delta^-_{V[H] \setminus S}(i)} x_a \quad \forall S \subset V[H] \setminus d : S \neq \emptyset
\end{aligned} \tag{21}$$

The partial multi-star inequalities (20) and (21) cannot be separated polynomially (Letchford et al. 2002). Therefore, we check whether any of these is violated by any cut set identified in a previous separation procedure and eventually add it.

Ring closure inequalities Compared to (2) the following inequalities ensure type 2 customer connectivity towards d in our directed formulation.

$$\sum_{a \in \delta^+(S)} f_a \geq 1 \quad \forall S \subset V[H] \backslash d : i \in S, \forall i \in U_2 \tag{22}$$

Inequalities (22) can be adapted to be applicable to nodes of type 0 and 1. Since such a node i is not necessarily a ring node, we express the constraint based on the optional circulation flow z_i through i.

$$\sum_{a \in \delta^+(S)} f_a \geq z_i \quad \forall S \subset V[H] \backslash d : i \in S, \forall i \in U_1 \cup W \tag{23}$$

The separation of (22) and (23) is done by minimal $i - d$ cut computations in H using arc weights f^*. The violation of the first inequality is detected as for (17) and we add (23) if the obtained cut weight is lower than z_i^*.

Capacitated ring closure inequalities The connectivity requirement in inequalities (22) can be extended to capacitated ring closure inequalities that take into account the ring tree capacity q when imposing necessary rings.

$$\sum_{a \in \delta^+(S)} f_a \geq \frac{|U_2(S)|}{q} \quad \forall S \subset V[H] \backslash d \tag{24}$$

After rounding the constant term as in (18) we obtain rounded capacitated ring closure inequalities.

$$\sum_{a \in \delta^+(S)} f_a \geq \left\lceil \frac{|U_2(S)|}{q} \right\rceil \quad \forall S \subset V[H] \backslash d \tag{25}$$

Inequality (24) is separated by the computation of a minimal $s - d$ cut S, D on the directed auxiliary graph H_2^+ with $V[H_2^+] := V[H] \cup \{s\}$ and additional arcs from s to all the type 2 customers: $A[H_2^+] := A[H] \cup \{(s, i) : i \in U_2\}$. The weight of an arc $(i, j) \in A[H_2^+]$ is $1/q$ if $i = s$ and else f_{ij}^*. The cut set S violates (24) if the cut weight is less than $|U_2|/q$. Furthermore, we can take into account type 1 ring customers in S since they consume ring tree capacity, too. They can be identified by the conveyed circulation flow z_i $\forall i \in U_1$. Therefore, inequalities (24) are generalized by stronger inequalities (26) that count the number of type 1 ring nodes based on the circulations.

$$\sum_{a \in \delta^+(S)} f_a \geq \frac{1}{q} \left(|U_2(S)| + \sum_{i \in U_1(S)} z_i \right) \quad \forall S \subset V[H] \backslash d \tag{26}$$

We separate them on the graph $H_{1,2}^+$ which is obtained from H_2^+ by extending the arc set to $A[H_{1,2}^+] = A[H_2^+] \cup \{(s, i) : i \in U_1\}$ with arc weights z_j^*/q $\forall (s, j) \in s \times U_1$. A $s - d$ cut weight less than $(|U_2| + \sum_{j \in U_1} z_j^*)/q$ indicates that S is a cut set that violates the inequality the most. We note that an alternative separation technique can be derived by expressing z_i as $\sum_{a\in\delta^+(i)} f_a$ using (7) and modifying the corresponding arc costs in H_2^+.

Capacitated ring closure multi-star inequalities To ensure ring-node-to-distributor connectivity we take into account a unitary capacity consumption for each arc from a ring node in S to a customer in $V[H]\backslash S$. This does not hold for an arbitrary node in S since the connected customer outside of S is not necessarily part of a ring that intersects with S. However, we can tighten the capacitated ring closure inequalities (26) by a similar counting argument. We utilize the circulation information to count the number of customers outside of S that are connected from ring nodes in S as follows.

$$\sum_{a\in\delta^+(S)} f_a \geq \frac{1}{q}\left(|U_2(S)| + \sum_{i\in U_1(S)} z_i + \sum_{a\in\delta^+_{U\backslash S}(S)} f_a\right) \quad \forall\, S \subset V[H]\backslash d \tag{27}$$

To derive an even tighter version of (27) we first rewrite the introduced outbound customer circulation flow-term as

$$\sum_{a\in\delta^+_{U_1\backslash S}(S\backslash U_2)} f_a + \sum_{a\in\delta^+_{U\backslash S}(U_2(S))} f_a + \sum_{a\in\delta^+_{U_2\backslash S}(S\backslash U_2)} f_a \tag{28}$$

We observe that values of the circulation flow variables in the last summation term will be equal to the corresponding arc variable values by (14). The second term counts the customers in $U\backslash S$ that are connected from type 2 customers in S by ring arcs. In fact, every customer that is connected from a type 2 node consumes capacity of a ring tree that requires a fundamental cycle. Thus, we can exchange the summation flow variables f to arc variables x for this term as well. Unfortunately, this argument can just be applied conditionally to the first sum. More precisely, we cannot count an outbound arc (i, j) since we do not know whether the originating node $i \in S$ is connected to a ring. This lifting procedure on the right-hand side of (27) yields the following right-hand side.

$$\frac{1}{q}\left(|U_2(S)| + \sum_{a\in\delta^+_{U_1\backslash S}(S\backslash U_2)} f_a + \sum_{a\in\delta^+_{U\backslash S}(U_2(S))} x_a + \sum_{a\in\delta^+_{U_2\backslash S}(S\backslash U_2)} x_a\right) \tag{29}$$

The separation procedure for inequalities (27) can be deduced from (22) and (26). An arc $a \in A[H^+_{1,2}]$ has weight $(1-1/q)f^*_a$ if $a \in (U \cup W) \times U$ and f^*_a otherwise. We extend this by include an approximating component to reflect (29). Thereby, the weight of an arc $a \in U_2 \times U$ of $H^+_{1,2}$ is set to $\max\{0, f^*_a - x^*_a/q\}$ based on the suggested variable exchange. So far, we tried to enforce connectivity from ring nodes to the depot. Conversely, we are able to formulate capacitated inequalities that ensure sufficient inbound circulation flow for a cut set as follows.

$$\sum_{a\in\delta^-(S)} f_a \quad \geq \quad \frac{1}{q}\left(|U_2(S)| + \sum_{i\in U_1(S)} z_i + \sum_{a\in\delta^-_{U\setminus S}(S\setminus U_2)} f_a + \sum_{a\in\delta^-_{U\setminus S}(U_2(S))} x_a\right) \quad \forall\, S \subset V[H]\setminus d \tag{30}$$

The separation procedure can be adapted from (27) including (29) on $H^-_{1,2}$ which we will not elaborate here. The obtained types of ring closure inequalities all together ensure a certain outbound connectivity of S whereas the various ring tree inequalities target sufficient inbound connectivity in a similar way. However, we remind that due to the ring tree structure we will not be able to match the number of arcs entering such a cut set S with the order of the leaving arcs in general.

Strengthening the upper bounds

In a branch and bound algorithm, it is crucial to generate tight upper bounds that are used for pruning. During the initial branching process integer feasible CRTP solutions are found scarcely and are at best of moderate quality. Therefore, we compute an integer-feasible start solution in our algorithm using a multi-start local search heuristic as described in Hill (2015). Based on several construction strategies various single and multi ring tree exchange neighborhoods are explored to identify potential improvements. Ties can be broken by reusing these techniques for a CRTP specific solution polishing to optimize solutions found during the exact method. Consequently, each time an integer feasible solution is found we perform local search and if this results in an improved solution we replace the incumbent.

Cut management

In our algorithm, we add (14) and (15), separate (2), (16), (17), (19), (15), (22), (23) and (30). Inequalities incorporating (29) are separated heuristically as explained above whereas (20), (18), (25) and (21) are added if violated for any of the obtained cut sets. In addition, to these inequalities we include constraints in our initial model which do not improve the theoretical lower bounds computed by solving the LP but in practice speed up the overall solution process.

Since customers of type 2 are required to be ring nodes and Steiner leave nodes cannot improve a solution, we add (31). These inequalities are implied by (22) and (3).

$$\sum_{a\in\delta^+(i)} x_a \geq \sum_{a\in\delta^-(i)} x_a \quad \forall\, i \in U_2 \cup W \tag{31}$$

We additionally add inequality (27) for $S = U_2$ to the initial model. Furthermore, we add inequalities (18) for $S = V[G]\setminus\{d\}$ and $S = \{v\}\ \forall\, v \in V(G)\setminus d$.

Besides our own CRTP-specific cutting techniques we activated the solver's internal cutting routines that implement common cuts. Various experiments with the different branching strategies using different prioritizations of arc and edge variables have shown that the pseudo-cost branching is most effective for our instances.

We let the CPLEX-internal cut management decide whether to purge added cuts if convenient. However, integrality enforcing cuts of type (18) and capacitated ring closure cuts (27) are forced to stay in the model permanently.

Computational study

In our computational study, we follow two objectives. On the one hand, we give results of our exact branch and cut algorithm and compare them to the results of our heuristic solution approach from Hill (2015). On the other hand, we consider various reliability scenarios and draw some conclusions about the cost of reliability in terms of overall costs and computational effort.

Implementation details

The algorithms was implemented in C++ using the CPLEX 12.6 branch and cut framework. Computations were done on an Intel i7-3667U 2.00 GHz processor unit. CPLEX was set to run in the single thread mode. We searched for an optimal LP-feasible solution at the root node and generated inequalities for all violated cuts. In our experiments it turned out that our algorithm performed better when additionally utilizing the solver cutting techniques. Among the various branching strategies suggested in the literature, we decided to use a strategy based on pseudo-costs which is implemented in the solver.

Scenarios

Our 675 CRTP instances[1] are derived from the 45 class A random instances generated for the CRSP in Baldacci et al. (2007). These TSPLib-based CRSP instances with $12 \leq |U| \leq 100$, $3 \leq q \leq 38$ and $m \in \{3, 4, 5\}$ already served for computational studies in Hoshino and de Souza (2009) and Naji-Azimi et al. (2010). They were derived from the three TSPLib instances $eil51$, $eil76$ and $eil101$ by declaring the first input node as the depot, the following $\alpha\%$ ($\alpha \in \{25, 50, 75, 100\}$) nodes as customers and the remaining nodes as Steiner nodes. Capacities were set up based on $m \in \{3, 4, 5\}$ such that the ring star utilization is about 90 %. The edge costs correspond to the Euclidean distances.

During our adaptation process, we assigned customers to be of type 1 using the following strategies. We prioritized according to their closeness to d (DC), remoteness to d (DF), closeness to a random customer (RC), remoteness to a random customer (RF) or performed a random assignment (R). For each class of obtained instances we use five different type 1 customer rates: $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$. Note that $U_1 = \emptyset$ and $U_1 = U$ result in a VRP variant and a CSTP, respectively. The used random seed depends on the CRSP instance and is constant for its derived CRTP instances. For

[1] The instances can be obtained from the corresponding author.

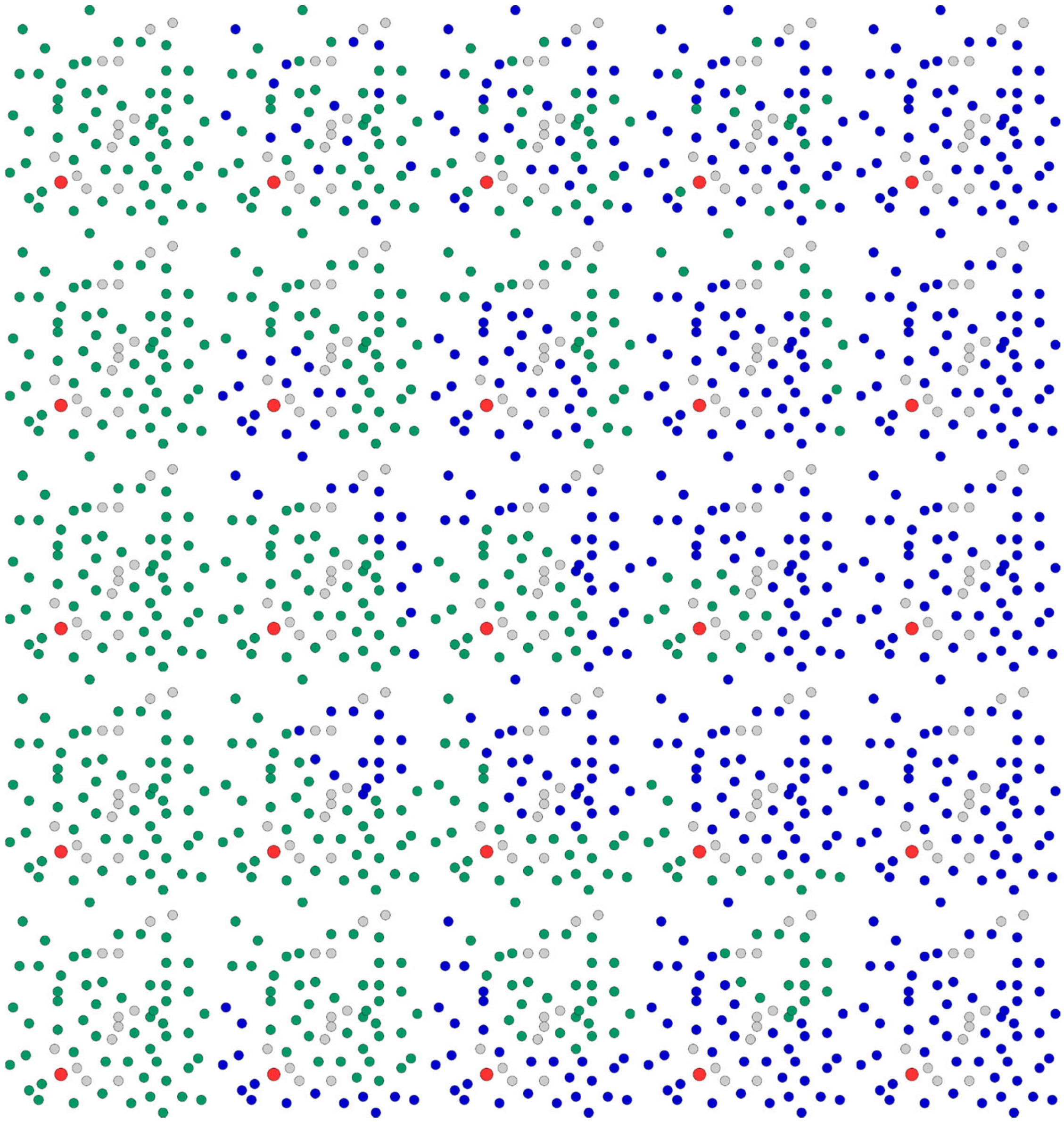


Fig. 9 CRTP random instances (*left to right*: 0, 0.25, 0.5, 0.75, 1—type 1 customer rate; *top to bottom*: R, DC, DF, RC, RF—type 1 customer assignment strategy)

two instances I and I' that are constructed based on the same strategy with $r_1 < r_1'$ we have $U_1' \subset U_1$. Hence the optimal values z and z' obey $z \geq z'$. The scenarios are illustrated in Fig. 9 for the instance Q-30.

Results

Tables 1, 2 and 3 show the computational results for the instances of type R. We limit ourselves here to the latter since they turned out to be computationally most challenging for our algorithm. The first 8 columns indicate the CRTP base instance name (P), the type 1 customer rate (r_1), node set cardinalities ($|V|$,$|U_2|$,$|U_1|$,$|W|$) and the capacity bounds (m,q). Each base instance is derived from a CRSP instance of class A from Baldacci et al. (2007), in the given order. Lower and upper bounds obtained

Table 1 Results for CRTP instances for reliability expansion scenario (R) and type 1 customer rates $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$

P	r_1	$\|V\|$	$\|U_2\|$	$\|U_1\|$	$\|W\|$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	Nodes
Q-1	1	26	0	12	13	3	5	141	157	157	**157**	**157**	0	2	1
	0.75		3	9				190	207	210	**210**	215	0	4	4
	0.5		6	6				193	221	227	**227**	**227**	0	8	45
	0.25		9	3				214	236	236	**236**	**236**	0	3	5
	0		12	0				238	241	242	**242**	**242**	0	1	1
Q-2	1	26	0	12	13	4	4	145	163	163	**163**	164	0	2	1
	0.75		3	9				181	207	207	**207**	**207**	0	2	1
	0.5		6	6				214	233	240	**240**	**240**	0	9	118
	0.25		9	3				238	247	249	**249**	**249**	0	3	2
	0		12	0				248	251	251	**251**	**251**	0	1	1
Q-3	1	26	0	12	13	5	3	147	170	170	**170**	173	0	1	1
	0.75		3	9				201	235	242	**242**	244	0	11	81
	0.5		6	6				225	245	251	**251**	**251**	0	4	2
	0.25		9	3				258	278	279	**279**	**279**	0	3	1
	0		12	0				274	279	279	**279**	**279**	0	1	1
Q-4	1	26	0	18	7	3	7	194	207	207	**207**	**207**	0	1	1
	0.75		4	14				230	249	256	**256**	**256**	0	10	97
	0.5		9	9				263	267	274	**274**	**274**	0	6	27
	0.25		13	5				272	284	292	**292**	**292**	0	12	161
	0		18	0				292	292	301	**301**	305	0	5	20
Q-5	1	26	0	18	7	4	5	206	217	217	**217**	220	0	1	1
	0.75		4	14				244	277	285	**285**	**285**	0	17	116
	0.5		9	9				282	304	313	**313**	318	0	27	128
	0.25		13	5				301	317	334	**334**	**334**	0	102	889
	0		18	0				334	334	339	**339**	**339**	0	5	26
Q-6	1	26	0	18	7	5	4	213	227	227	**227**	231	0	1	1
	0.75		4	14				246	276	278	**278**	**278**	0	5	22
	0.5		9	9				307	320	336	**336**	**336**	0	67	433
	0.25		13	5				338	353	361	**361**	**361**	0	13	73
	0		18	0				365	374	375	**375**	**375**	0	2	2
Q-7	1	26	0	25	0	3	10	245	245	245	**245**	248	0	0	1
	0.75		6	19				277	283	294	**294**	**294**	0	11	135
	0.5		13	12				293	296	313	**313**	**313**	0	55	1414
	0.25		18	7				312	312	327	**327**	**327**	0	20	501
	0		25	0				324	326	328	**328**	**328**	0	1	4
Q-8	1	26	0	25	0	4	7	252	252	252	**252**	267	0	0	1
	0.75		6	19				293	300	311	**311**	315	0	12	132
	0.5		13	12				308	319	345	**345**	**345**	0	769	5530
	0.25		18	7				332	342	357	**357**	**357**	0	60	754
	0		25	0				358	358	362	**362**	**362**	0	1	8

Table 1 continued

P	r_1	$\|V\|$	$\|U_2\|$	$\|U_1\|$	$\|W\|$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	Nodes
Q-9	1	26	0	25	0	5	6	254	254	254	**254**	262	0	0	1
	0.75		6	19				293	307	319	**319**	322	0	17	150
	0.5		13	12				331	352	369	**369**	372	0	326	2882
	0.25		18	7				351	369	378	**378**	379	0	20	296
	0		25	0				392	394	396	**396**	397	0	2	11
Q-10	1	51	0	12	38	3	5	109	156	156	**156**	**156**	0	16	1
	0.75		3	9				163	181	190	**190**	196	0	234	32
	0.5		6	6				185	203	213	**213**	215	0	340	365
	0.25		9	3				204	220	222	**222**	**222**	0	9	2
	0		12	0				234	238	242	**242**	**242**	0	7	1
Q-11	1	51	0	12	38	4	4	143	159	159	**159**	163	0	16	1
	0.75		3	9				184	199	209	**209**	**209**	0	89	54
	0.5		6	6				208	226	230	**230**	**230**	0	54	34
	0.25		9	3				230	238	238	**238**	**238**	0	7	1
	0		12	0				240	250	251	**251**	**251**	0	10	1
Q-12	1	51	0	12	38	5	3	154	170	170	**170**	172	0	15	1
	0.75		3	9				182	203	203	**203**	**203**	0	20	1
	0.5		6	6				218	240	251	**251**	**251**	0	508	116
	0.25		9	3				248	271	278	**278**	**278**	0	77	46
	0		12	0				269	279	279	**279**	**279**	0	11	1
Q-13	1	51	0	25	25	3	10	227	244	245	**245**	248	0	28	2
	0.75		6	19				254	279	293	302	305	3.1	3600	2686
	0.5		12	13				281	295	311	**311**	312	0	2760	2482
	0.25		18	7				292	310	322	**322**	**322**	0	858	796
	0		25	0				314	323	328	**328**	**328**	0	32	30
Q-14	1	51	0	25	25	4	7	226	250	252	**252**	267	0	17	3
	0.75		6	19				271	296	304	**304**	321	0	583	301
	0.5		12	13				305	327	341	352	352	3.1	3600	2050
	0.25		18	7				332	344	357	**357**	**357**	0	1795	1145
	0		25	0				350	355	362	**362**	**362**	0	55	29
Q-15	1	51	0	25	25	5	6	234	254	254	**254**	262	0	14	2
	0.75		6	19				293	320	331	335	339	1.1	3600	3035
	0.5		12	13				319	348	359	370	372	3	3600	1440
	0.25		18	7				331	360	372	387	387	3.9	3600	1012
	0		25	0				344	382	390	**390**	397	0	13	9

by our exact method using a one hour time limit can be found in columns lb and ub, respectively. The root node relaxation objective value is given in lb_0. To show the effectiveness of our cutting techniques we provide the lower bounds in the root node obtained by the pure model (1)–(11) in column $\underline{lb}_0$. The primal bound resulting from

Table 2 Results for CRTP instances for reliability expansion scenario (R) and type 1 customer rates $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$

P	r_1	$\|V\|$	$\|U_2\|$	$\|U_1\|$	$\|W\|$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	Nodes
Q-16	1	51	0	37	13	3	14	285	303	304	**304**	**304**	0	7	1
	0.75		9	28				329	346	350	375	375	6.6	3600	3745
	0.5		18	19				345	356	364	376	378	3.2	3600	2878
	0.25		27	10				353	366	379	**379**	380	0	1428	4254
	0		37	0				371	376	380	**380**	381	0	32	21
Q-17	1	51	0	37	13	4	11	278	308	308	**308**	309	0	13	2
	0.75		9	28				329	351	363	**363**	369	0	3472	2867
	0.5		18	19				364	376	384	399	399	3.8	3600	2361
	0.25		27	10				375	384	396	404	404	1.9	3600	3852
	0		37	0				395	404	410	**410**	418	0	358	200
Q-18	1	51	0	37	13	5	9	295	311	314	**314**	**314**	0	11	24
	0.75		9	28				346	372	374	408	408	8.2	3600	1239
	0.5		18	19				370	397	401	431	431	7	3600	1700
	0.25		27	10				386	411	417	436	436	4.5	3600	1880
	0		37	0				429	435	446	**446**	452	0	359	1316
Q-19	1	51	0	50	0	3	19	376	376	376	**376**	377	0	1	1
	0.75		12	38				402	407	418	427	436	2.1	3600	5032
	0.5		25	25				425	429	435	445	447	2.3	3600	6217
	0.25		37	13				433	441	451	**451**	454	0	1953	2396
	0		50	0				451	454	462	**462**	473	0	1311	1068
Q-20	1	51	0	50	0	4	14	382	384	384	**384**	386	0	4	56
	0.75		12	38				406	418	423	458	458	7.7	3600	2236
	0.5		25	25				430	444	448	493	493	9.1	3600	2700
	0.25		37	13				456	464	471	502	502	6.2	3600	4800
	0		50	0				476	480	493	**493**	513	0	799	2042
Q-21	1	51	0	50	0	5	12	387	390	390	**390**	392	0	6	80
	0.75		12	38				420	439	447	491	501	9	3600	1474
	0.5		25	25				460	471	478	526	526	9.1	3600	2132
	0.25		37	13				484	489	497	525	525	5.3	3600	3233
	0		50	0				502	506	522	526	541	0.8	3600	5792
Q-22	1	76	0	18	57	3	7	180	213	213	**213**	214	0	77	2
	0.75		4	14				227	271	272	**272**	**272**	0	624	28
	0.5		9	9				250	282	288	318	318	9.6	3600	268
	0.25		13	5				267	294	303	318	318	4.8	3600	414
	0		18	0				301	320	331	**331**	332	0	1020	953
Q-23	1	76	0	18	57	4	5	205	229	232	**232**	235	0	97	139
	0.75		4	14				255	300	302	309	312	2.1	3600	396
	0.5		9	9				288	333	336	**336**	**336**	0	1869	144
	0.25		13	5				315	352	359	369	369	2.8	3600	0
	0		18	0				348	383	386	**386**	390	0	1710	339

Table 2 continued

P	r_1	$\|V\|$	$\|U_2\|$	$\|U_1\|$	$\|W\|$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	Nodes
Q-24	1	76	0	18	57	5	4	231	249	257	**257**	259	0	223	369
	0.75		4	14				269	314	325	**325**	**325**	0	1829	161
	0.5		9	9				319	357	368	379	379	2.9	3600	330
	0.25		13	5				356	395	397	**397**	**397**	0	345	3
	0		18	0				398	448	448	**448**	451	0	2663	370
Q-25	1	76	0	37	38	3	14	252	275	320	**320**	**320**	0	856	9
	0.75		9	28				322	360	363	390	390	6.8	3600	502
	0.5		18	19				353	369	372	402	402	7.4	3600	622
	0.25		27	10				373	384	390	403	403	3.3	3600	607
	0		37	0				390	409	409	**409**	413	0	2586	1556
Q-26	1	76	0	37	38	4	11	283	326	326	**326**	336	0	231	123
	0.75		9	28				340	378	382	402	402	5	3600	372
	0.5		18	19				374	408	410	455	455	9.8	3600	312
	0.25		27	10				394	415	418	460	460	9.2	3600	361
	0		37	0				423	434	446	458	458	2.6	3600	1176
Q-27	1	76	0	37	38	5	9	306	333	340	**340**	343	0	539	1379
	0.75		9	28				355	405	407	446	446	8.7	3600	240
	0.5		18	19				394	422	426	473	473	9.9	3600	149
	0.25		27	10				420	443	443	497	497	10.9	3600	223
	0		37	0				458	472	477	506	506	5.6	3600	1110
Q-28	1	76	0	56	19	3	21	374	382	383	**383**	395	0	21	7
	0.75		14	42				407	426	427	462	462	7.6	3600	869
	0.5		28	28				430	436	438	477	477	8.1	3600	659
	0.25		42	14				444	451	461	465	472	1	3600	4168
	0		56	0				462	467	476	**476**	495	0	3600	2353
Q-29	1	76	0	56	19	4	16	382	388	389	**389**	402	0	28	14
	0.75		14	42				421	437	441	488	488	9.7	3600	396
	0.5		28	28				448	462	466	520	520	10.4	3600	316
	0.25		42	14				475	487	492	532	532	7.4	3600	610
	0		56	0				492	500	514	535	543	4	3600	1725
Q-30	1	76	0	56	19	5	13	391	396	399	**399**	414	0	38	148
	0.75		14	42				438	468	469	533	533	11.9	3600	253
	0.5		28	28				468	492	493	554	554	11	3600	234
	0.25		42	14				495	509	512	558	558	8.2	3600	545
	0		56	0				527	534	546	557	561	1.9	3600	1411

the heuristic from Hill (2015) can be found in ub_0. Bold objective values are optimal. The corresponding computation time (in seconds) and the number of explored nodes in the branch and bound tree can be found in columns $t(s)$ and *nodes*, whereas the

Table 3 Results for CRTP instances for reliability expansion scenario (R) and type 1 customer rates $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$

P	r_1	$\|V\|$	$\|U_2\|$	$\|U_1\|$	$\|W\|$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	$nodes$
Q-31	1	76	0	75	0	3	28	473	473	473	**473**	478	0	2	1
	0.75		18	57				503	515	516	551	551	6.4	3600	1676
	0.5		37	38				528	532	537	564	564	4.9	3600	1399
	0.25		56	19				541	547	554	564	573	1.8	3600	2800
	0		75	0				567	567	572	**572**	584	0	230	463
Q-32	1	76	0	75	0	4	21	478	478	482	**482**	494	0	8	35
	0.75		18	57				506	530	531	573	573	7.4	3600	539
	0.5		37	38				537	550	552	612	612	9.8	3600	954
	0.25		56	19				573	581	586	618	618	5.2	3600	1640
	0		75	0				593	597	603	626	626	3.7	3600	3890
Q-33	1	76	0	75	0	5	17	482	482	488	**488**	495	0	88	456
	0.75		18	57				528	546	552	623	623	11.3	3600	178
	0.5		37	38				562	576	585	623	623	6.1	3600	343
	0.25		56	19				584	598	608	656	656	7.4	3600	522
	0		75	0				617	623	641	674	674	4.9	3600	2358
Q-34	1	101	0	25	75	3	10	162	274	274	**274**	282	0	450	20
	0.75		6	19				256	308	314	**314**	327	0	1760	114
	0.5		12	13				299	332	337	353	353	4.6	3600	323
	0.25		18	7				324	351	356	363	363	2	3600	180
	0		25	0				353	365	366	**366**	**366**	0	121	1
Q-35	1	101	0	25	75	4	7	238	288	289	**289**	293	0	333	24
	0.75		19	6				289	344	344	367	367	6.2	3600	34
	0.5		12	13				327	367	367	405	405	9.3	3600	60
	0.25		18	7				361	385	385	416	416	7.5	3600	27
	0		25	0				392	407	409	425	425	3.8	3600	362
Q-36	1	101	0	25	75	5	6	251	295	299	**299**	**299**	0	330	47
	0.75		19	6				296	362	361	393	393	8.1	3600	10
	0.5		12	13				326	377	378	403	403	6.2	3600	15
	0.25		18	7				371	406	407	429	429	5.1	3600	17
	0		25	0				422	435	440	452	452	2.7	3600	48
Q-37	1	101	0	50	50	3	19	346	409	411	**411**	**411**	0	410	10
	0.75		12	38				406	457	457	492	492	7.1	3600	9
	0.5		25	25				445	472	473	499	499	5.3	3600	70
	0.25		37	13				465	482	483	503	503	3.9	3600	45
	0		50	0				486	492	493	508	523	2.9	3600	645
Q-38	1	101	0	50	50	4	14	356	415	415	**415**	420	0	380	1
	0.75		12	38				416	460	460	480	480	4.1	3600	117
	0.5		25	25				451	484	484	517	517	6.5	3600	43
	0.25		37	13				484	501	501	531	531	5.7	3600	76
	0		50	0				514	521	525	537	537	2.3	3600	223

Table 3 continued

P	r_1	$\lvert V\rvert$	$\lvert U_2\rvert$	$\lvert U_1\rvert$	$\lvert W\rvert$	m	q	$\underline{lb}_0$	lb_0	lb	ub	ub_0	Δ	$t(s)$	$nodes$
Q-39	1	101	0	50	50	5	12	368	422	426	**426**	443	0	790	67
	0.75		12	38				424	479	481	505	505	4.8	3600	128
	0.5		25	25				470	569	495	527	527	6.1	3600	65
	0.25		37	13				506	523	523	564	564	7.3	3600	49
	0		50	0				542	551	553	574	574	3.6	3600	126
Q-40	1	101	0	75	25	3	28	462	498	511	**511**	516	0	840	168
	0.75		18	57				519	554	555	594	594	6.6	3600	223
	0.5		37	38				543	569	570	592	592	3.8	3600	159
	0.25		56	19				573	586	588	612	612	4	3600	220
	0		75	0				596	600	606	**606**	622	0	2098	916
Q-41	1	101	0	75	25	4	21	475	500	516	**516**	519	0	780	112
	0.75		18	57				533	559	559	595	595	6	3600	40
	0.5		37	38				551	582	582	607	607	4.1	3600	141
	0.25		56	19				591	600	603	619	619	2.6	3600	177
	0		75	0				616	623	624	639	642	2.3	3600	532
Q-42	1	101	0	75	25	5	17	483	521	522	**522**	529	0	93	85
	0.75		18	57				546	584	584	653	653	10.6	3600	20
	0.5		37	38				571	597	598	645	645	7.3	3600	189
	0.25		56	19				607	623	622	670	670	7.1	3600	123
	0		75	0				642	648	649	689	689	5.8	3600	223
Q-43	1	101	0	100	0	3	38	554	554	555	**555**	**555**	0	1	1
	0.75		25	75				600	612	611	652	652	6.2	3600	260
	0.5		50	50				620	623	624	657	660	5	3600	532
	0.25		75	25				634	639	644	648	656	0.7	3600	2170
	0		100	0				658	660	663	**663**	683	0	292	578
Q-44	1	101	0	100	0	4	28	561	561	564	**564**	568	0	2	50
	0.75		25	75				608	624	624	663	663	5.9	3600	207
	0.5		50	50				632	642	644	690	690	6.7	3600	455
	0.25		75	25				655	661	665	683	691	2.7	3600	1523
	0		100	0				677	681	684	700	700	2.3	3600	993
Q-45	1	101	0	100	0	5	23	567	570	570	**570**	576	0	2	1
	0.75		25	75				612	625	629	695	695	9.5	3600	100
	0.5		50	50				657	670	674	717	717	6	3600	203
	0.25		75	25				683	687	689	730	730	5.6	3600	206
	0		100	0				705	708	709	743	743	4.6	3600	952

optimality gap $(ub - lb)/lb$ can be found in column Δ. The run time of the primal heuristic never exceeded 25 s during our tests.

As expected, the pure tree or ring structured problems can usually be solved more efficiently in terms of optimality gap and number of explored nodes. We observed the instances with balanced customer reliability requirements as the most challenging. Even though we could solve 64 % of the purely ring-based instances and all the purely tree-based instances to optimality, we proved optimality for just 31 % of the problems with $r_1 = 0.5$. Our heuristic algorithm from Hill (2015) produced solutions that were optimal for 27 % of the instances. Additionally, the local search techniques polished integer-feasible solutions during the branch and cut procedure in many cases for the remaining instances. For the entire test set we obtained an average optimality gap of 2.6 %.

The presented cutting techniques turned out to have a significant effect on the quality of the computed lower bounds. Regarding the lower bounds obtained for the root node we observe an augmentation of 6.3 % compared to the values stemming from the pure model ($\underline{lb}_0$). This corresponds to a root optimality gap reduction of about 50 % on average.

Since we let CPLEX manage the cuts, we do not have information about which of the generated cuts are active in the LPs. Our extensive separation strategies produce large numbers of cuts that are passed to the solver. For the different instance sizes ($|V| \in \{26, 51, 76, 101\}$) we separate 857, 4742, 7546 and 11,270 cuts per instance on average.

The cost of reliability

We are particularly interested in the effect of increased reliability requirements on the overall costs. Certainly, different cost functions as well as parameters such as the capacity limits m, q and the reliability distribution have a strong impact on solutions for CRTP instances. Nevertheless, we give some consequences of reliability parameterization in our different scenarios based on our solution approaches. As the ring tree structure suggests, the CRTP solutions can be quite different, from pure tree or ring-based ones.

The series of optimal solutions for the different type 1 customer scenarios in Fig. 10 give an impression of the topological spectrum covered by the CRTP. With an increasing type 1 customer rate, we expect a smaller number of fundamental cycles as it is the case for the exemplary evolution for instance Q-1 in Fig. 11.

For the increasing type 1 customer rates, we extended the type 1 customer set incrementally in our scenarios. Therefore, we can assume that the function of optimal network costs is monotonically increasing for a decreasing type 1 customer rate. Depending on the distribution of the reliability requirements among the customers this results in different correlations between the optimal network cost and the type 1 customer rate, as shown in Fig. 12. The curves show the relative cost increase with a decreasing type 1 customer rate averaged over all the instances for our scenarios. We observe that providing additional reliability to all the customers increased the overall network costs by 35–65 % for our instances. More precisely, installing initial reliability is costly, whereas it gets less expensive the more reliability is already implemented. This is intuitive due to the fact that the rerouting of an existing ring is more efficient than the implementation of a ring structure on a widely tree-spanned customer domain. Therefore, the reliability cost function tends to be concave. In the scenario that assigns

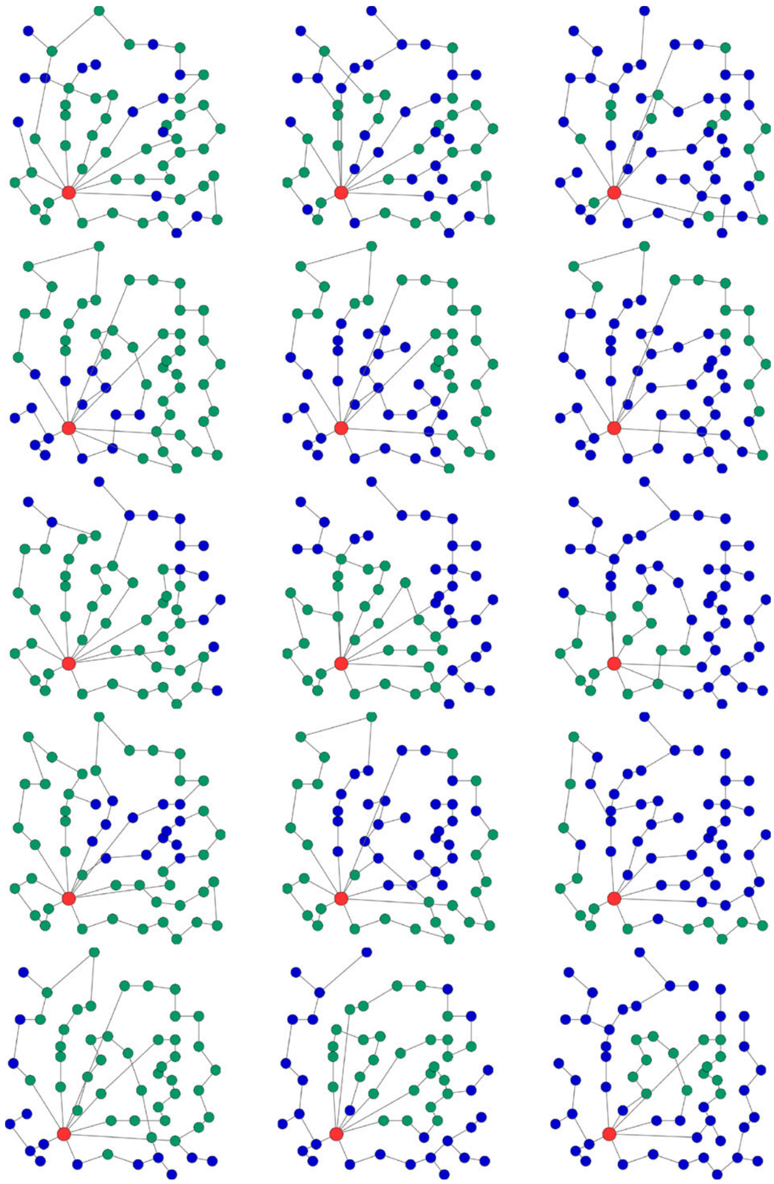


Fig. 10 Solutions for CRTP random instances (Q-30; *left to right*: 0.25, 0.5, 0.75—type 1 customer rate; *top to bottom*: R, DC, DF, RC, RF—type 1 customer assignment strategy)

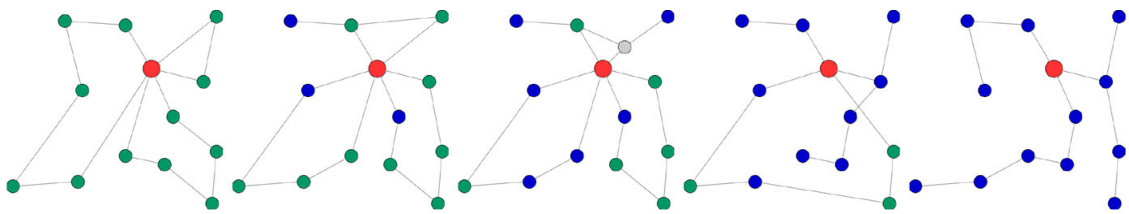


Fig. 11 Optimal solutions for the CRTP base instance Q-1 ($m = 3$, $q = 5$) with type 1 customer rates 0.00, 0.25, 0.50, 0.75, and 1.00 using random type 1 customer assignments (R)

reliability to customers closer to the distributor first (DF), we see this function to be less curved on average than when randomly turning customers into type 2 (R). In turn, providing reliability in remote areas (DC) requires to close rings towards a distant d which is more elaborate when the ring tree capacity limits are tight. We expect this

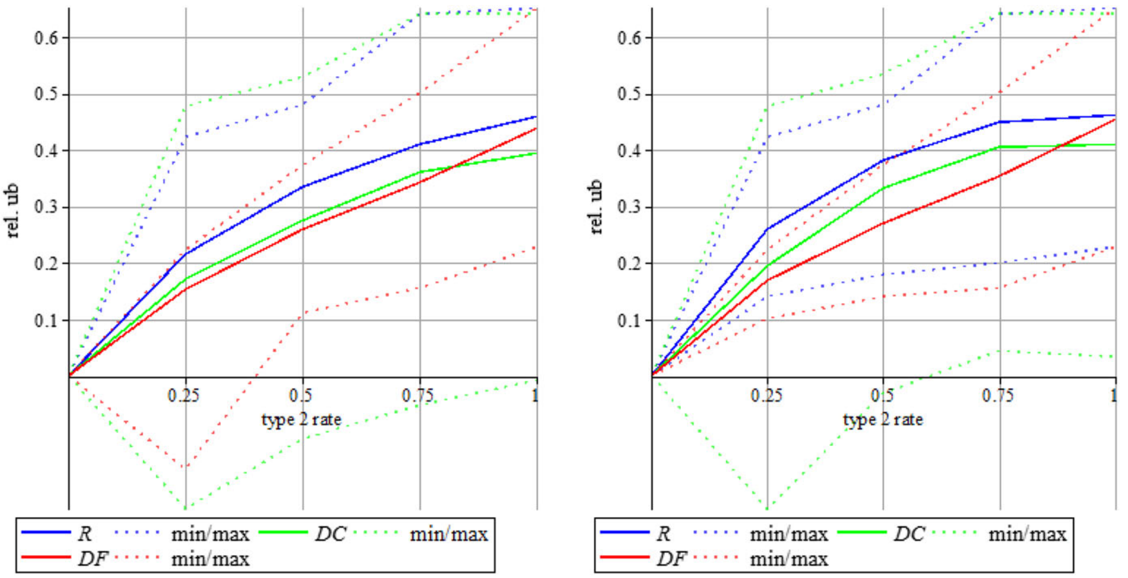
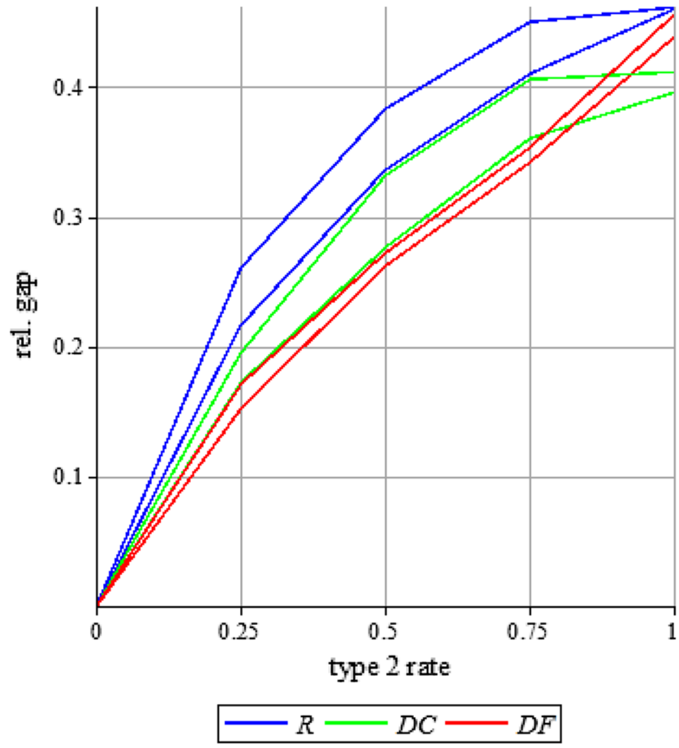


Fig. 12 The relative cost of reliability for different reliability expansion scenarios based on the best upper bounds for the network costs. Upper bounds on the *left* and lower bounds on the *right*

Fig. 13 The average relative optimality gaps for different reliability scenarios



effect to become even stronger when reducing the ring tree customer limit since the number of required ring trees increases.

In Fig. 13 we show the average relative optimality gaps for different reliability scenarios. It can be seen that our algorithm achieves tighter results for instances of type DF compared to DC. However, type R instances are even harder to solve.

Conclusions

We presented a novel model for designing cost-optimized capacitated networks. This capacitated ring tree problem (CRTP) combines ring and tree structures that are common models in telecommunication applications and in logistics. Our approach generalizes existing optimization models and allows a broader use due to its capability of embracing problems that were previously modeled independently. We related the resulting ring tree topology to tree-based, ring-based, ring-star-based and survivable network design concepts previously studied in the literature. The presented mathematical formulation for the problem was used to elaborate an efficient branch and cut algorithm based on mathematical programming. Therefore, we developed cutting techniques tailored for the capacitated ring tree structure. We showed how to separate valid inequalities exactly and explained our heuristic addition of violated inequalities with hard separation problems. A local search-based heuristic was used to produce starting solutions that support the solver's search and to polish integer-feasible solutions during the branch and bound method. For a set of small- and medium-sized capacity-tight literature derived instances we gave computational results for our algorithms. Using different reliability scenarios we observed that a balanced types 1 and 2 reliability ratio yields the most difficult instances for our methods. After studying different reliability distributions we obtained an indication that instances with uniformly distributed customers with an additional reliability tend to be of increased difficulty. Nevertheless, we were able solve instances with up to 50 nodes to optimality. When considering existing scenarios that imply a pure tree or ring structure we could even solve instances up to 100 nodes.

We suggest further research on the CRTP in terms of heuristics and model extensions. It seems to be a fruitful model for the application of efficient metaheuristics or matheuristics that take advantage of the specific solution network structure. Corresponding efficient solution techniques could either be integrated in our exact methods or could be used to tackle bigger problem sizes. We are also aware that a column generation-based algorithm could improve the presented results, especially in the case of an increasing number of ring trees. A model extension that could be of practical use concerns the integration of lower bounds on the number of customers served by a ring tree. Another balancing measure could be the introduction of separate lower and upper bounds q_1, q_2, q_r for type 1 customers, type 2 customers or ring customers, respectively.

Acknowledgments We would like to thank the anonymous reviewers for their detailed comments which led to an improvement of this work. Moreover, we thank Edna Hoshino for discussion.

References

Ahuja RK, Orlin JB, Sharma D (2000) Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. Math Program 91:71–97

Amberg A, Domschke W, Voß S (1996) Capacitated minimum spanning trees: algorithms using intelligent search. Comb Optim Theory Pract 1:9–39

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

Baldacci R, Toth P, Vigo D (2010) Exact algorithms for routing problems under vehicle capacity constraints. Ann Oper Res 175:213–245
Chopra S, Rao MR (1994) The Steiner tree problem i: formulations, compositions and extension of facets. Math Program 64(1–3):209–229
Fischetti M, Salazar JJ, Toth P (1997) A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper Res 45(3):378–394
Fortz B (2000) Design of survivable networks with bounded rings. Springer-Verlag, Berlin
Fortz B, Labbé M, Maffioli F (2000) Solving the two-connected network with bounded meshes problem. Oper Res 48(6):866–877
Goemans M (1994) The steiner tree polytope and related polyhedra. Math Program 63(1–3):157–182
Golden B, Raghavan S, Wasil EA (2008) The vehicle routing problem : latest advances and new challenges. Operations Research/Computer Science Interfaces Series, vol 43. Springer, New York
Gouveia L, Paias A, Voß S (2011) Models for a traveling purchaser problem with additional side-constraints. Comput Oper Res 38(2):550–558
Grötschel M, Raack C, Werner A (2013) Towards optimizing the deployment of optical access networks. EURO J Comput Optim. doi:10.1007/s13675-013-0016-x
Held M, Karp RM (1971) The traveling-salesman problem and minimum spanning trees: part ii. Math Program 1:6–25
Hill A (2015) Multi-exchange neighborhoods for the capacitated ring tree problem. In: Ivan D, Stefka F, Ivan L (eds) Numerical methods and applications, lecture notes in computer science. Springer International Publishing, New York, pp 85–94
Hoshino EA, de Souza CC (2009) A branch-and-cut-and-price approach for the capacitated m-ring-star problem. Electron Notes Discrete Math 35:103–108 (LAGOS'09-V Latin-American Algorithms, Graphs and Optimization Symposium)
Hwang FK, Richards DS (1992) Steiner tree problems. Networks 22(1):55–89
Kerivin H, Mahjoub AR (2005) Design of survivable networks: a survey. Networks 46(1):1–21
Koch T, Martin A (1998) Solving steiner tree problems in graphs to optimality. Networks 32:207–232
Labbé M, Laporte G, Martín IR, Salazar JJ (2004) The ring star problem: polyhedral analysis and exact algorithm. Networks 43(3):177–189
Letchford AN, Eglese RW, Lysgaard J (2002) Multistars, partial multistars and the capacitated vehicle routing problem. Math Program 94(1):21–40
Letchford AN, Nasiri SD, Theis DO (2013) Compact formulations of the steiner traveling salesman problem and related problems. Eur J Oper Res 228(1):83–92
Magnanti T, Wolsey L (1994) Optimal trees. CORE Discussion Papers 1994026, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE)
Mitchell JE (2009) Integer programming: branch and cut algorithms. In: Floudas CA, Pardalos PM (eds) Encyclopedia of optimization. Springer, New York, pp 1643–1650
Naddef D, Rinaldi G (2001) Branch-and-cut algorithms for the capacitated VRP. In: Toth P, Vigo D (eds) The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp 53–84
Naji-Azimi Z, Salari M, Toth P (2010) A heuristic procedure for the capacitated m-ring-star problem. Eur J Oper Res 207(3):1227–1234
Narula SC, Ho CA (1980) Degree-constrained minimum spanning tree. Comput Oper Res 7(4):239–249
Orlowski S, Wessäly R, Pióro M, Tomaszewski A (2010) SNDlib 1.0-survivable network design library. Networks 55(3):276–286
Ramesh T (1981) Traveling purchaser problem. Opsearch 18(1–3):78–91
Ribeiro CC, Souza MC, Souza C (2001) Variable neighborhood search for the degree-constrained minimum spanning tree problem. Discrete Appl Math 118:43–54
Robbins HE (1939) A theorem on graphs, with an application to a problem of traffic control. Am Math Monthly 46(5):281–283
Stoer M (1992) Design of survivable networks, vol 1531 of Lecture Notes in Mathematics. Springer-Verlag, Berlin
Voß S (2006) Steiner tree problems in telecommunications. In: Resende MGC, Pardalos PM (eds) Handbook of optimization in telecommunications. Springer, New York, pp 459–492

F. Paper vi

Multi-exchange Neighborhoods for the Capacitated Ring Tree Problem

Alessandro Hill(✉)

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

Abstract. A *ring tree* is a tree graph with an optional additional edge that closes a unique cycle. Such a cycle is called a *ring* and the nodes on it are called *ring nodes*. The *capacitated ring tree problem* (CRTP) asks for a network of minimal overall edge cost that connects given customers to a depot by ring trees. Ring trees are required to intersect in the depot which has to be either a ring node of degree two in a ring tree or a node of degree one if the ring tree does not contain a ring. Customers are predefined as of *type 1* or *type 2*. The type 2 customers have to be ring nodes, whereas type 1 customers can be either ring nodes or nodes in tree sub-structures. Additionally, optional Steiner nodes are given which can be used as intermediate network nodes if advantageous. Capacity constraints bound both the number of the ring trees as well as the number of customers allowed in each ring tree. In this paper we present first advanced neighborhood structures for the CRTP. Some of them generalize existing concepts for the TSP and the Steiner tree problem, others are CRTP-specific. We also describe models to explore these multi-node and multi-edge exchange neighborhoods in one or more ring trees efficiently. Moreover, we embed these techniques in a heuristic multi-start framework and show that it produces high quality results for small and medium size literature instances.

Keywords: Capacitated ring tree problem · Network design · Local search

1 Introduction

The design of cost efficient networks under capacity constraints is of undoubted importance for applications in various industries. Especially in the field of transportation and telecommunication significant cost savings were achieved through the application of appropriate optimization models in the last decades. Topologically, many networks are based on fundamental structures such as trees or rings. The extensively studied *minimum weight spanning trees* (MSTs) assure connectivity such that a unique path between any two nodes in the network exists,

This research was partially supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX project).

I. Dimov et al. (Eds.): NMA 2014, LNCS 8962, pp. 85–94, 2015.
DOI: 10.1007/978-3-319-15585-2_10

whereas the *capacitated minimum spanning tree problem* (CMSTP) [2] asks for such a tree of minimal total edge costs while limiting the number of nodes of sub-trees connected to a depot by a single edge. In practice, the integration of optional intermediate *Steiner nodes* is highly relevant and is facilitated by the well-known *Steiner tree problem* (STP) [7]. On the contrary, a prominent ring based optimization problem is the *travelling salesman problem* (TSP), asking for a travel cost minimizing sequence in which each customer of a given set should be visited before returning to a depot. Such a *tour* is required for each vehicle starting from the depot in the *vehicle routing problem* (VRP) [3]. The need for multiple vehicles arises from the commonly limited transport capacity to deliver or pick up goods from or to the customers. Beyond these concepts, the recent *capacitated ring tree problem* (CRTP) [5] integrates the ring structure and the tree structure into an optimization model under consideration of capacities and the useful Steiner nodes. The implemented *ring tree* structure is defined to be either a tree, a ring or a ring with additional disjoint trees attached to some of its nodes. Moreover, certain customers are prespecified to be of type 2 and thus required to be contained in sub-rings in ring trees. The remaining type 1 customers can be such *ring nodes* or nodes in sub-trees. Additional capacity constraints bound the total number of customers on each ring tree as well as the number of ring trees originating from the depot. Figure 1 shows a feasible network that satisfies these requirements and minimizes the overall edge costs, i.e. the objective function. The CRTP is NP-hard as are its special cases, the STP and the TSP, but computationally even more challenging [5]. For most real world applications heuristic solution approaches are indispensable due to the size limits for efficient exact algorithms. Therefore, in this paper we generalize known neighborhood structures for the purely tree [1] and purely ring based [6] special cases by treating the ring tree case. Furthermore, the CRTP gives rise to interesting structured neighborhoods on its own that we introduce and show how to efficiently explore. We embed these techniques in a multi-start heuristic framework and show its efficiency on a set of literature instances.

After a formal definition of the CRTP in Sect. 2 we introduce the novel neighborhoods and corresponding exploration techniques in Sect. 3. The embedding of these ideas in a multi-start heuristic is described in Sect. 4 before we close with our conclusion in Sect. 5.

2 The Capacitated Ring Tree Problem

In the following we give a formal definition of the CRTP using basic graph theoretic notation. We consider a network $\mathcal{N}$ synonymous with an undirected simple graph with node set $V[\mathcal{N}]$ and edge set $E[\mathcal{N}]$. The graph obtained after the removal of a node $v \in V[\mathcal{N}]$ is denoted by $\mathcal{N} \backslash v$.

Definition. *We are given a set of nodes* $V = U_2 \dot\cup U_1 \dot\cup W \dot\cup \{d\}$ *where the nodes in* U_t *correspond to type* t *customers, nodes in* W *are Steiner nodes and* d *represents a central depot. The cost of connecting two nodes* $u \neq v$ *in* V *by an edge* $e = \{u, v\}$ *is* $c_e > 0$. *A solution for the CRTP is a network* $\mathcal{N}$ *obtained from*

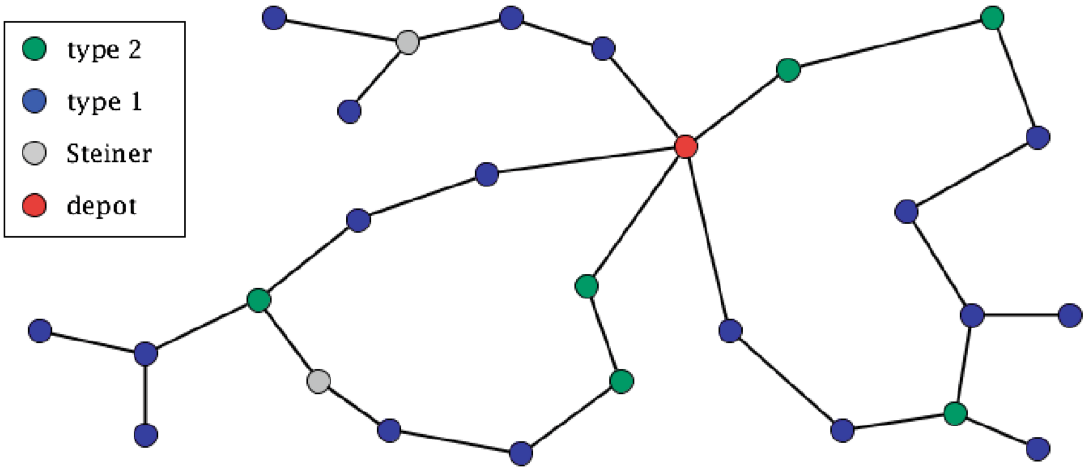


Fig. 1. A CRTP solution with 24 customers in 3 ring trees.

the union of a set of rings $\mathcal{R} = \{R_1, ..., R_k\}$ *and a set of trees* $\mathcal{T} = \{T_1, ..., T_l\}$ *on* V *such that*

- *each type 2 customer is contained in exactly one ring,*
- *each type 1 customer is contained in exactly one ring or tree,*
- *each Steiner node is contained in at most one ring or tree,*
- *each ring contains the depot* d*,*
- *each tree contains either the depot* d *or a node of a ring,*

and $\mathcal{N}$ *is capacity feasible, i.e.*

- *the number of connected components in* $\mathcal{N}\backslash d$ *is at most* m *and*
- *the number of type 1 and type 2 customers in each connected component of* $\mathcal{N}\backslash d$ *does not exceed* q*.*

The CRTP asks for such a network of minimal total edge cost $c(\mathcal{N}) = \sum_{e \in E[\mathcal{N}]} c_e$*.*

From each connected component of $\mathcal{N}\backslash d$ we obtain a *ring tree* Q by adding the depot d and the edges connecting d and Q in $\mathcal{N}$. Such a ring tree forms either a tree or a ring with disjoint trees attached to it. Figure 1 illustrates a solution network based on 2 rings and 4 trees according to our definition of the CRTP.

3 Neighborhood Structures

In the following we elaborate several structured neighborhoods for the CRTP and explain how to efficiently explore them. They partially generalize existing concepts for the TSP, VRP, STP and CMSTP but we also introduce CRTP-specific neighborhoods that do not have non-trivial counterparts in these specializations. For the sake of simplified descriptions we introduce some notation which refers to a CRTP solution network $\mathcal{N}$ unless explicitly stated differently. Let $U = U_1 \cup U_2$ be the set of all customers. For a ring tree $\mathcal{Q} \subseteq \mathcal{N}$ we denote the set of neighbors of a node $v \in V[\mathcal{Q}]$ in Q as $N_{\mathcal{Q}}[v]$. Let $P_{\mathcal{Q}}[u, v]$ be the set of paths that connect two distinct nodes $u, v \in V[\mathcal{Q}]$. We recall that if $\mathcal{Q}$ contains a ring then $|P_{\mathcal{Q}}[u,v]| \leq 2$, otherwise $\mathcal{Q}$ is a tree and thus $|P_{\mathcal{Q}}[u,v]| = 1$. Then we define $T_{\mathcal{Q}}[u,v]$ as the set of *path trees* of $\mathcal{Q}$ obtained from extending each path $\mathcal{P} \in P_{\mathcal{Q}}[u,v]$ by the non-ring structures in $\mathcal{Q}$ attached to the nodes of $\mathcal{P}$. Finally, for a node set $X \subset V$ we define $\Delta_{\mathcal{Q}}[X]$ as the set of edges with one end in X and the other end in $V[\mathcal{Q}]\backslash X$.

1-edge-opt. In contrast to purely ring-based models, a 1-edge-opt neighborhood can be defined for the CRTP by considering the feasible removal of an edge $e \in E[\mathcal{Q}]$ followed by the insertion of an edge $e' \notin E[\mathcal{Q}]$ for each ring tree $\mathcal{Q} \subseteq \mathcal{N}$. We first observe that given a ring without type 2 customers, the edge with the highest cost can be deleted and $\mathcal{N}$ is still feasible. Therefore, we assume that each ring in $\mathcal{N}$ contains a type 2 customer. In the case that e is a ring edge e' is required to *repair* the destroyed ring if possible. The ring-tree-opt neighborhood below will cover this case. Thus let $e = \{u, v\}$ be a non-ring edge of $\mathcal{Q}$ and let u be the node on each path from v to d. Then the deletion of e creates two connected components of $\mathcal{Q}$, one containing d and another one that contains v, more precisely a tree $\mathcal{T}_v$. To establish a valid solution we consider the insertion of each re-connecting edge $e' \in \Delta_{\mathcal{Q}}[V[\mathcal{T}_v]]$ subject to adherence to the capacity constraints. In particular, we may create a new (ring)tree by allowing e' to be incident to d.

2-edge-opt. The prominent TSP-tailored edge swaps can be applied to each ring in $\mathcal{N}$. In a similar manner ties can be broken by facilitating capacity-feasible re-combinations of two distinct ring trees $\mathcal{Q}_1$ and $\mathcal{Q}_2$ as known for the VRP. More specifically, for two ring edges $e = \{u, v\} \in E[\mathcal{Q}_1]$ and $e' = \{w, x\} \in E[\mathcal{Q}_2]$ we consider their replacement by $\{u, w\}$ and $\{v, x\}$ or $\{u, x\}$ and $\{v, w\}$. Figure 2 illustrates such an improvement move. If both edges are incident to d the neighborhood is empty. By allowing $\mathcal{Q}_1 = \mathcal{Q}_2$ and avoiding sub-tours we obtain the mentioned 2-opt for the TSP.

Moreover, we consider the deletion of two non-ring edges followed by the reconnection of the cut-off sub-trees $\mathcal{T}_1 \subseteq \mathcal{Q}_1$ and $\mathcal{T}_2 \subseteq \mathcal{Q}_2$ to other ring trees as depicted in Fig. 3. We hereby partially generalize the 1-edge-opt neighborhood. Since we regard the capacity constraints such a move can have an ejecting effect with respect to attached sub-trees when for instance reconnecting $\mathcal{T}_1$ to $\mathcal{Q}_2$. Finally, taking into account the removal of an edge e in a ring $\mathcal{R} \subseteq \mathcal{Q}_1$ and a non-ring edge $e' \in E[\mathcal{Q}_2]$ yields the remainder of this neighborhood. Let $\mathcal{T}_2$

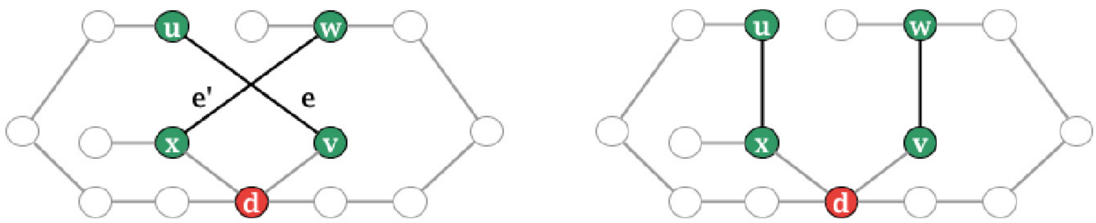


Fig. 2. A 2-edge-opt improvement based on the ring edges $\{u, v\}$ and $\{w, x\}$.

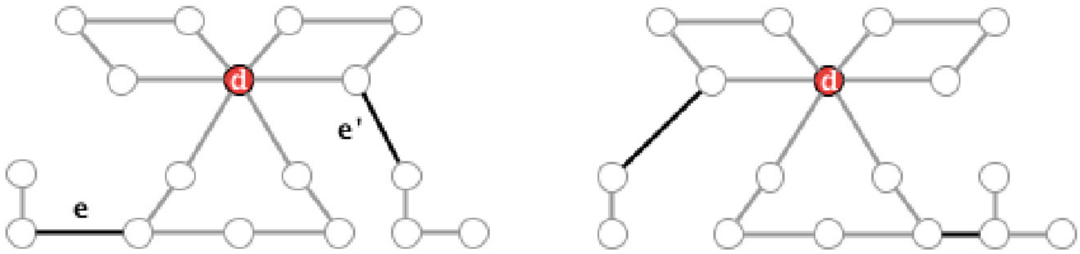


Fig. 3. A 2-edge-opt improvement based on the non-ring edges e and e'.

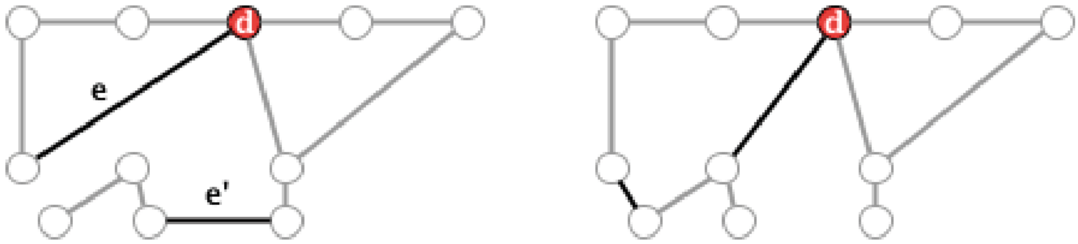


Fig. 4. A 2-edge-opt improvement based on a ring edge e and a non-ring edge e'.

be the sub-tree of $\mathcal{Q}_2$ induced by e' as in the 1-edge-opt neighborhood. The corresponding modification of $\mathcal{Q}_1$ in $\mathcal{N}$ corresponds to the replacement of a e by a path tree obtained from $\mathcal{T}_2$, whereas $\mathcal{Q}_2$ is reduced by $\mathcal{T}_2$. Figure 4 shows such a transformation.

1-node-opt. We consider moving a single customer node u from its current ring tree $\mathcal{Q}_1$ to a ring tree $\mathcal{Q}_2$. Obviously, the capacity of $\mathcal{Q}_2$ needs to be sufficient when performing such an operation. We ensure the preservation of the ring tree structure after the extraction of u from $\mathcal{Q}_1$ by the incorporation of a MST on the neighbors $N_{\mathcal{Q}_1}[u]$. Note that the degree of the depot has to be limited by m minus the number of current ring trees beside $\mathcal{Q}_1$ to satisfy the ring tree capacity m. Although the *degree constrained minimum spanning tree problem* (DCMSTP) is known to be NP-hard in general this special case can be solved polynomially using a Prim's algorithm in a slightly modified version starting from d. If u is of type 1 it may be inserted into $\mathcal{Q}_2$ either as a leaf or as an intermediate node that splits an edge $\{v, w\}$ into edges $\{v, u\}$ and $\{u, w\}$. Type 2 customers may only be inserted in this edge replacing manner into a ring instead.

2-node-opt. We consider swapping two customers that are not necessarily in distinct ring trees. This neighborhood can be constructed by intersecting two 1-node-opt spaces.

Steiner-node-opt. This neighborhood is inspired by known STP improvement moves and consists of all the feasible solutions obtained after deleting or inserting a single Steiner node. Certainly, a Steiner leaf node can simply be removed, whereas a node with degree 2 can be replaced by an edge connecting both neighbors if this results in an overall cost reduction. For an arbitrary Steiner node $x \in V[\mathcal{Q}]$, the re-connection can be accomplished by a MST on $N_{\mathcal{Q}}[x]$ as for the

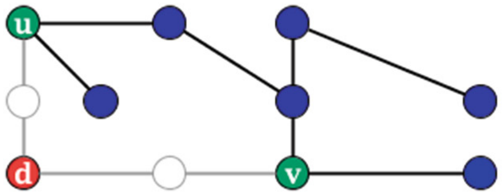


Fig. 5. A minimum spanning tree based improvement in a ring-tree-opt.

1-node-opt neighborhood. Conversely, we also consider the insertion of a Steiner node $x \notin V[\mathcal{N}]$ into $\mathcal{N}$. We take into account the *splitting* of an existing edge $\{u, v\}$ into $\{u, x\}$ and $\{x, v\}$. Moreover, two incident edges $\{u, v\}$ and $\{u, w\}$ with $u \neq d$ can be replaced by the *star configuration* $\{x, u\}$, $\{x, v\}$ and $\{x, w\}$.

Ring-tree-opt. This advanced neighborhood contains the solutions obtained by the rearrangement of the tree structure induced by two specifically situated mandatory ring nodes. Let $\mathcal{T} \in T_{\mathcal{Q}}(u, v)$ be a path tree in a ring tree $\mathcal{Q} \in \mathcal{N}$ for $\{u, v\} \subseteq U_2 \cup \{d\}$ such that $V[\mathcal{T}]\backslash\{u, v\}$ does not contain type 2 customers or the distributor. Then we can build a DCMSTP on the nodes of $\mathcal{T}$. As in previous neighborhoods a single degree constraint applies when $d \in \{u, v\}$ to avoid the installation of more additional ring trees than allowed. An improving solution in this neighborhood connects u and v by a path tree of less cost as illustrated in Fig. 5. This neighborhood is also valid for nodes u and v such that $V[T_{\mathcal{Q}}(u, v)] \cap U_2 = \emptyset$ and therefore, in particular applicable when $\mathcal{Q}$ is a tree.

Ring-tree-split-opt. This neighborhood contains solutions that can be obtained by *splitting* a ring tree $\mathcal{Q} \subseteq \mathcal{N}$ into two separate ring trees. This presumes enough capacity in $\mathcal{N}$ to install an additional ring tree. Basically, we try to repair a single ring edge removal by the feasible insertion of two new ring-closing edges. As in the ring-tree-opt search let $\mathcal{T}$ be a path tree for two distinct nodes u and v in $V[\mathcal{Q}] \cap (U_2 \cup \{d\})$ with $V[\mathcal{T}]\backslash\{u, v\} \cap \{d\} \cup U_2 = \emptyset$. Then we consider the removal of each ring edge $e \in E[\mathcal{T}]$ followed by the insertion of two edges $\{d, w\}$ and $\{d, x\}$ for $\{w, x\} \subseteq V[\mathcal{T}]$ as shown in Fig. 6. If $u = d$ then $\mathcal{Q}$ splits into a tree and a ring tree, whereas the splitting of a pure tree $\mathcal{Q}$ is contained in the 1-edge-opt neighborhood.

Ejection-chain-opt. Extracting a customer node u_1 from a ring tree $\mathcal{Q}_1$ and inserting it into a ring tree $\mathcal{Q}_2$ might be cost saving but not feasible because $\mathcal{Q}_2$ is capacity tight, i.e. $\mathcal{Q}_2$ contains q customers. However, the ejection of a customer u_2 in $\mathcal{Q}_2$ and its insertion into a ring tree $\mathcal{Q}_3$ can facilitate the move.

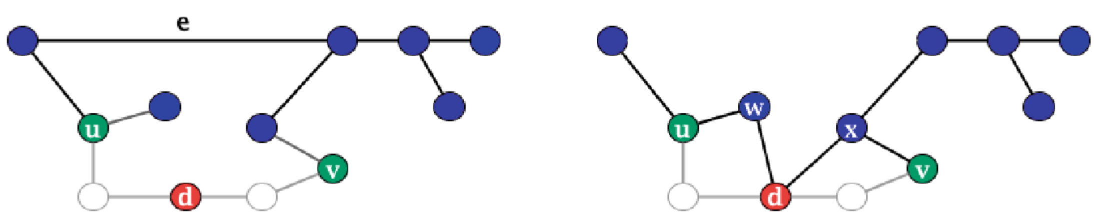


Fig. 6. An improving solution in the ring-tree-split neighborhood.

In this ejection-chain-opt neighborhood we consider all these double node moves for distinct ring trees $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$. Note that if $\mathcal{Q}_3 = \mathcal{Q}_1$ then it corresponds to the 2-nodes-opt neighborhood.

4 A Multi-start Local Search Heuristic

Our heuristic is based on the iterated exploration of the introduced CRTP neighborhoods. We apply the corresponding local searches (LQSs) in a multi-start fashion on a set of start solutions obtained from different initial constructions. For a CRTP instance P, let $\Sigma(P)$ be the procedure that returns a solution pool based on the strategies that we briefly summarize in the following. On the one hand we apply *cluster first*, *route second* techniques as in [4] to solve the VRP obtained after temporarily declaring all customers type 2. Different cluster distance metrics (e.g. min/max/avg cluster node distance) give rise to multiple solutions that are added to the pool. Then we conversely focus on the design of (partial) rings or (partial) trees based on the computation of MSTs and the construction of *nearest first* TSP routes. We combine these partial networks on the different sets of customers and turn them into a feasible solution by a correction mechanism that repeatedly applies moves similar to the ones described in our local search neighborhoods. Our overall algorithm applies the local searches on each of the solutions in $\Sigma(P)$ in a best-fit fashion until no enhancement can be found. The order in which the different neighborhoods are explored corresponds to the increasing potential structural impact. The resulting *multi-start CRTP heuristic* can be described as follows.

Input: CRTP P
;
foreach $N' \in \Sigma(P)$ **do**
    $z \leftarrow \infty$;
    while $c(N') < z$ **do**
        $z \leftarrow c(N')$;
        LQS(N', Ring-tree-opt);
        LQS(N', 1-edge-opt);
        LQS(N', 2-edge-opt);
        LQS(N', 1-node-opt);
        LQS(N', 2-node-opt);
        LQS(N', Steiner-node-opt);
        LQS(N', Ring-tree-split-opt);
        LQS(N', Ring-tree-join-opt);
        LQS(N', Ejection-chain-opt);
    end
    if $c(N') < c(N)$ **then** $N \leftarrow N'$;
end
return N;

We implemented the algorithm in c++ and ran tests on an Intel i7-3667U 2.00 GHz processor unit for the 225 small to medium size instances[1] used in [5]. The type 1 customers in these TSPLIB based instances with $|V| \in \{26, 51, 76, 101\}$ are randomly assigned according to a rate $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$. Various combinations of m and q with an average customer capacity slack $(mq-|U|)/mq$ of 0.14 make them capacity tight. The computational results are given in Appendix 1. The run time of the heuristic procedure never exceeded 25 s. Table 1 contains the computational results with the first 4 columns indicating the CRTP instance, the type 1 customer rate r_1, the number of nodes $|V|$ and customers $|U|$. The network cost $c(\mathcal{N})$ is then given along with the relative gaps $\Delta_{lb} = [c_{lb}(\mathcal{N}) - c(\mathcal{N})]/c(\mathcal{N})$ and $\Delta_{ub} = [c(\mathcal{N}) - c_{ub}(\mathcal{N})]/c(\mathcal{N})$ to the lower bound $c_{lb}(\mathcal{N})$ and the upper bound $c_{ub}(\mathcal{N})$ obtained by the exact method in [5]. We do not intend to compete with the branch & cut algorithm but rather give an idea of the solution quality obtained by the heuristic. Since we initialized the exact method with the heuristic solution and use the local search techniques along the branch & bound $\Delta_{ub} \geq 0$ holds.

5 Conclusions

We introduced advanced multi-edge and multi-node exchange neighborhood structures for the CRTP. They partially generalize existing concepts for prominent tree and ring based combinatorial optimization problems. We presented suitable models to explore these neighborhoods efficiently and a heuristic framework to turn these techniques into an efficient heuristic. Using this diversifying multi-start algorithm we are able to obtain optimal results in many cases for a set of small and medium sized literature instances. The average gap to known lower bounds is 3.8 %. We suggest this first heuristic approach for the CRTP as a reference for related models and further algorithms.

[1] The instances can be obtained from the author.

Appendix 1

Table 1. Heuristic results for CRTP instances from [5] with type 1 customer rates $r_1 \in \{0, 0.25, 0.5, 0.75, 1\}$ compared to bounds obtained by a branch & cut algorithm.

P	r_1	$\|V\|$	$\|U\|$	Δ_{lb}	$c(\mathcal{N})$	Δ_{ub}
1	1	26	12	**0**	**157**	**0**
	0.75			**-2.3**	215	**2.3**
	0.5			**0**	**227**	**0**
	0.25			**0**	**236**	**0**
	0			**0**	**242**	**0**
2	1			**-0.6**	164	**0.6**
	0.75			**0**	**207**	**0**
	0.5			**0**	**240**	**0**
	0.25			**0**	**249**	**0**
	0			**0**	**251**	**0**
3	1			**-1.7**	173	**1.7**
	0.75			**-0.8**	244	**0.8**
	0.5			**0**	**251**	**0**
	0.25			**0**	**279**	**0**
	0			**0**	**279**	**0**
4	1		18	**0**	**207**	**0**
	0.75			**0**	**256**	**0**
	0.5			**0**	**274**	**0**
	0.25			**0**	**292**	**0**
	0			**-1.3**	305	**1.3**
5	1			**-1.4**	220	**1.4**
	0.75			**0**	**285**	**0**
	0.5			**-1.6**	318	**1.6**
	0.25			**0**	**334**	**0**
	0			**0**	**339**	**0**
6	1			**-1.7**	231	**1.7**
	0.75			**0**	**278**	**0**
	0.5			**0**	**336**	**0**
	0.25			**0**	**361**	**0**
	0			**0**	**375**	**0**
7	1		25	**-1.2**	248	**1.2**
	0.75			**0**	**294**	**0**
	0.5			**0**	**313**	**0**
	0.25			**0**	**327**	**0**
	0			**0**	**328**	**0**
8	1			**-5.6**	267	**5.6**
	0.75			**-1.3**	315	**1.3**
	0.5			**0**	**345**	**0**
	0.25			**0**	**357**	**0**
	0			**0**	**362**	**0**
9	1			**-3.1**	262	**3.1**
	0.75			**-0.9**	322	**0.9**
	0.5			**-0.8**	372	**0.8**
	0.25			**-0.3**	379	**0.3**
	0			**-0.3**	397	**0.3**
10	1	51	12	**0**	**156**	**0**
	0.75			**-2**	196	**2**
	0.5			**0**	**215**	**0**
	0.25			**0**	**222**	**0**
	0			**0**	**242**	**0**
11	1			**-2.5**	163	**2.5**
	0.75			**0**	**209**	**0**
	0.5			**0**	**230**	**0**
	0.25			**0**	**238**	**0**
	0			**0**	**251**	**0**
12	1			**-1.2**	172	**1.2**
	0.75			**0**	**203**	**0**
	0.5			**0**	**251**	**0**
	0.25			**0**	**278**	**0**
	0			**0**	**279**	**0**
13	1		25	**-1.2**	248	**1.2**
	0.75			-4	305	1
	0.5			**0**	**312**	**0**
	0.25			**0**	**322**	**0**
	0			**0**	**328**	**0**
14	1			**-5.6**	267	**5.6**
	0.75			**-5.3**	321	**5.3**
	0.5			-3.1	352	0
	0.25			**0**	**357**	**0**
	0			**0**	**362**	**0**
15	1			**-3.1**	262	**3.1**
	0.75			-2.2	339	1.2
	0.5			-3.5	372	0.5
	0.25			-3.9	387	0
	0			**-1.8**	397	**1.8**

P	r_1	$\|V\|$	$\|U\|$	Δ_{lb}	$c(\mathcal{N})$	Δ_{ub}
16	1		37	**0**	**304**	**0**
	0.75			-6.6	375	0
	0.5			-3.7	378	0.5
	0.25			**-0.3**	380	**0.3**
	0			**-0.3**	381	**0.3**
17	1			**-0.3**	309	**0.3**
	0.75			**-1.6**	369	**1.6**
	0.5			-3.8	399	0
	0.25			-1.9	404	0
	0			**-1.9**	418	**1.9**
18	1			**0**	**314**	**0**
	0.75			-8.2	408	0
	0.5			-7	431	0
	0.25			-4.5	436	0
	0			**-1.3**	452	**1.3**
19	1		50	**-0.3**	377	**0.3**
	0.75			-4.1	436	2.1
	0.5			-2.7	447	0.4
	0.25			**-0.7**	454	**0.7**
	0			**-2.3**	473	**2.3**
20	1			**-0.5**	386	**0.5**
	0.75			-7.7	458	0
	0.5			-9.1	493	0
	0.25			-6.2	502	0
	0			**-3.9**	513	**3.9**
21	1			**-0.5**	392	**0.5**
	0.75			-10.7	501	2
	0.5			-9.1	526	0
	0.25			-5.3	525	0
	0			-3.5	541	2.8
22	1	76	18	**0**	**214**	**0**
	0.75			**0**	**272**	**0**
	0.5			-9.6	318	0
	0.25			-4.8	318	0
	0			**0**	**332**	**0**
23	1			**-0.9**	235	**0.9**
	0.75			-3.1	312	1
	0.5			**0**	**336**	**0**
	0.25			-2.8	369	0
	0			**-1**	390	**1**
24	1			**0**	**259**	**0**
	0.75			**0**	**325**	**0**
	0.5			-2.9	379	0
	0.25			**0**	**397**	**0**
	0			**-0.7**	451	**0.7**
25	1		37	**0**	**320**	**0**
	0.75			-6.8	390	0
	0.5			-7.4	402	0
	0.25			-3.3	403	0
	0			**-1**	413	**1**
26	1			**-3**	336	**3**
	0.75			-5	402	0
	0.5			-9.8	455	0
	0.25			-9.2	460	0
	0			-2.6	458	0
27	1			**-0.9**	343	**0.9**
	0.75			-8.7	446	0
	0.5			-9.9	473	0
	0.25			-10.9	497	0
	0			-5.6	506	0
28	1		56	**-3**	395	**3**
	0.75			-7.6	462	0
	0.5			-8.1	477	0
	0.25			-2.4	472	1.5
	0			**-3.8**	495	**3.8**
29	1			**-3.2**	402	**3.2**
	0.75			-9.7	488	0
	0.5			-10.4	520	0
	0.25			-7.4	532	0
	0			-5.4	543	1.5
30	1			**-3.6**	414	**3.6**
	0.75			-11.9	533	0
	0.5			-11	554	0
	0.25			-8.2	558	0
	0			-2.6	561	0.7

P	r_1	$\|V\|$	$\|U\|$	Δ_{lb}	$c(\mathcal{N})$	Δ_{ub}
31	1		75	**-1**	478	**1**
	0.75			-6.4	551	0
	0.5			-4.9	564	0
	0.25			-3.4	573	1.6
	0			**-2.1**	584	**2.1**
32	1			**-2.4**	494	**2.4**
	0.75			-7.4	573	0
	0.5			-9.8	612	0
	0.25			-5.2	618	0
	0			-3.7	626	0
33	1			**-1.4**	495	**1.4**
	0.75			-11.3	623	0
	0.5			-6.1	623	0
	0.25			-7.4	656	0
	0			-4.9	674	0
34	1	101	25	**-1.8**	282	**1.8**
	0.75			**-4**	327	**4**
	0.5			-4.6	353	0
	0.25			-2	363	0
	0			**0**	**366**	**0**
35	1			**-1.4**	293	**1.4**
	0.75			-6.2	367	0
	0.5			-9.3	405	0
	0.25			-7.5	416	0
	0			-3.8	425	0
36	1			**0**	**299**	**0**
	0.75			-8.1	393	0
	0.5			-6.2	403	0
	0.25			-5.1	429	0
	0			-2.7	452	0
37	1		50	**0**	**411**	**0**
	0.75			-7.1	492	0
	0.5			-5.3	499	0
	0.25			-3.9	503	0
	0			-5.7	523	2.9
38	1			**-1.2**	420	**1.2**
	0.75			-4.1	480	0
	0.5			-6.5	517	0
	0.25			-5.7	531	0
	0			-2.3	537	0
39	1			**-3.8**	443	**3.8**
	0.75			-4.8	505	0
	0.5			-6.1	527	0
	0.25			-7.3	564	0
	0			-3.6	574	0
40	1		75	**-1**	516	**1**
	0.75			-6.6	594	0
	0.5			-3.8	592	0
	0.25			-4	612	0
	0			**-2.6**	622	**2.6**
41	1			**-0.6**	519	**0.6**
	0.75			-6.2	595	0
	0.5			-4.2	607	0
	0.25			-2.6	619	0
	0			-2.8	642	0.5
42	1			**-1.3**	529	**1.3**
	0.75			-10.6	653	0
	0.5			-7.3	645	0
	0.25			-7.1	670	0
	0			-5.8	689	0
43	1		100	**0**	**555**	**0**
	0.75			-6.2	652	0
	0.5			-5.5	660	0.5
	0.25			-1.9	656	1.2
	0			**-2.9**	683	**2.9**
44	1			**-0.7**	568	**0.7**
	0.75			-5.9	663	0
	0.5			-6.7	690	0
	0.25			-3.8	691	1.2
	0			-2.3	700	0
45	1			**-1**	576	**1**
	0.75			-9.5	695	0
	0.5			-6	717	0
	0.25			-5.6	730	0
	0			-4.6	743	0

References

1. Ahuja, R.K., Orlin, J.B., Sharma, D.: Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. Math. Program. **91**, 71–97 (2000)
2. Amberg, A., Domschke, W., Voß, S.: Capacitated minimum spanning trees: Algorithms using intelligent search. Comb. Optim. Theory Pract. **1**, 9–39 (1996)
3. Golden, B., Raghavan, S., Wasil, E.A.: The Vehicle Routing Problem: Latest Advances and New Challenges. Operations research/Computer Science Interfaces Series. Springer, New York (2008). http://www.worldcat.org/oclc/254177374
4. Hill, A., Voß, S.: An equi-model matheuristic for the multi-depot ring star problem. Research paper, University of Antwerp, Faculty of Applied Economics (2014). http://EconPapers.repec.org/RePEc:ant:wpaper:2014015
5. Hill, A., Voß, S.: Optimal capacitated ring trees. Research paper, University of Antwerp, Faculty of Applied Economics (2014). http://EconPapers.repec.org/RePEc:ant:wpaper:2014012
6. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. **21**(2), 498–516 (1973). http://pubsonline.informs.org/doi/abs/10.1287/opre.21.2.498
7. Voß, S.: Steiner tree problems in telecommunications. In: Resende, M.G.C., Pardalos, P.M. (eds.) Handbook of Optimization in Telecommunications, pp. 459–492. Springer, US (2006). http://dx.doi.org/10.1007/978-0-387-30165-5_18

G. Paper vii

Generalized local branching heuristics and the capacitated ring tree problem

Alessandro Hill
Institute of Maritime Logistics
Hamburg University of Technology
Schwarzenbergstrasse 95 D, 21073 Hamburg, Germany
Tel.: +49-40-42878-4451
Fax: +49-40-42878-4452
alessandro.hill@tuhh.de

Stefan Voß
Institute of Information Systems (IWI)
University of Hamburg
Von-Melle-Park 5, 20146 Hamburg, Germany
Tel.: +49-40-42838-3064
Fax: +49-40-42838-5535
stefan.voss@uni-hamburg.de
and
Escuela de Ingeniería Industrial
Pontificia Universidad Católica de Valparaíso, Chile

In this paper we present a heuristic framework that is based on mathematical programming to solve network design problems. Our techniques combine local branching with locally exact refinements. In an iterative strategy an existing solution is refined by solving restricted mixed integer programs (MIPs) to optimality. These are obtained from the master problem MIP by limiting the number of variable flips for structured subsets of the binary edge variables. We introduce generalized local branching cuts which enforce the latter using two parameters at the same time: the number of considered variables and the number of allowed variable flips.

Using this concept we develop an efficient algorithm for the capacitated ring tree problem (CRTP), a recent network design model for partially reliable capacitated networks that combines cycle and tree structures. Our implementation operates on top of an efficient branch and cut algorithm for the CRTP. The sets of refinement variables are deduced from single-ball network node clusters. We provide computational results and an extensive analysis of the algorithm for a set of literature instances. We show that the approach is capable of improving existing best results for the CRTP and outperforms the pure refinement or local branching approaches.

Key words: capacitated ring tree problem; local branching; mathematical programming; local search; network design; matheuristic

1. Motivation and contribution

Network design applications in telecommunications and transportation environments typically involve a large number of decision variables in suitable optimization models. Although exact algorithms are usually not applicable when it comes to medium and large size instances they have proven useful in heuristic frameworks, also referred to as *matheuristics* Maniezzo et al. (2010). This class of heuristics combines mathematical programming concepts and classical (meta-)heuristic paradigms.

Integer programming based *refinement algorithms* have been successfully applied to complex network optimization problems (e.g. Franceschi et al. (2006), Archetti et al. (2012), Cafieri et al. (2014), Hill and Voß (2014), Naji-Azimi et al. (2012)) as well as other classes of challenging combinatorial optimization problems (e.g. Maniezzo et al. (2010), Lalla-Ruiz and Voß (2014)). These methods typically incorporate an exact mathematical programming based approach that is applied to a local improvement model for an existing solution. They get more effective with increasing complexity of the underlying problem structure Archetti et al. (2012), Hill and Voß (2014). Due to the limited computational efficiency of the exact method that is used to carry out the refinements, the mentioned techniques are in fact only effective locally on small-sized substructures. Commonly, random, multi-start or contraction based perturbation mechanisms are added to (partially) overcome local optimality.

In this work we suggest an approach that aims at increasing the number of decisions considered for local refinement. This is achieved by bounding the scale of modification in terms of binary variable flips in return. The basic idea of considering neighboring solutions within a certain Hamming distance is known as *local branching* and has been introduced as a polishing procedure in general mixed integer programs (MIPs) in Fischetti and Lodi (2003). Several highly efficient heuristics for various combinatorial optimization problems successfully incorporate this concept (e.g. Rodríguez-Martín and Salazar (2010), Legato and Trunfio (2014), Smet et al. (2014)). However, we are not aware of other literature work that considers the combination of subsets of binary variables of different sizes with suitable Hamming bounds at the same time. Moreover, our approach provides a complete algorithmic framework, suitable for a variety of combinatorial optimization problems. We iteratively build refinement models by adding new *generalized local branching cuts* to the master program and solve these extended MIPs to optimality. Herewith, we are able to arbitrarily increase the local area that is considered for refinement by adequately limiting the allowed variable flips.

We show that the sketched ideas can be turned into an effective algorithm for capacitated network design. We devise an efficient heuristic for the capacitated ring tree problem (CRTP) which was introduced in Hill and Voß (2015) recently. The CRTP combines ring based models such as the classic traveling salesman problem (TSP) with tree based models such as the Steiner tree problem (STP) under capacity constraints. Heuristics and exact algorithms for the CRTP are discussed in Hill (2015), Hill and Voß (2015) and Hoshino and Hill (2014). Even though the CRTP can be broadly applied as it generalizes several prominent network design problems, our techniques can be transferred to related models with reasonable effort. More generally, we suggest that our main ideas can be used to solve a variety of discrete, or even (partially) continuous, optimization problems. The main contributions of this work are

- the development of a generic framework for heuristic network design based on a generalized local branching, combining local branching and integer programming based refinement techniques, and
- the design and analysis of an efficient heuristic algorithm for the CRTP incorporating these concepts which is able to find new best solutions for literature instances.

Section 2 contains a formal description of the CRTP along with the MIP formulation used in our algorithm. After the presentation of the generic local branching based refinement technique in Section 3 we develop the heuristic algorithm for the CRTP in Section 4. In Section 5 we provide the improved results for literature instances and computationally compare different configurations of our method. We close the paper with conclusions in Section 6.

2. The capacitated ring tree problem

The *capacitated ring tree problem* (CRTP) was introduced in Hill and Voß (2015) and generalizes several classical ring and tree based network design models. The base topology is the *ring tree* defined as a graph consisting of a cycle $\mathcal{C}$ and node disjoint trees $\mathcal{T}_1, ..., \mathcal{T}_k$, each of them intersecting with C in exactly one node. By allowing $\mathcal{C}$ to be a cycle of order one the ring tree graph class contains both, pure trees and cycle graphs. To simplify our description we say that a *ring tree star* of *order h centered in d* is a graph obtained by the union of ring trees $\mathcal{Q}_1, ..., \mathcal{Q}_h$ that intersect in the node d such that d is a leaf in $\mathcal{Q}_i$ if

$\mathcal{Q}_i$ is a tree and a cycle node of degree 2 otherwise, $\forall\ i \in \{1, ..., h\}$. Figure 1 depicts three characteristic ring tree stars.

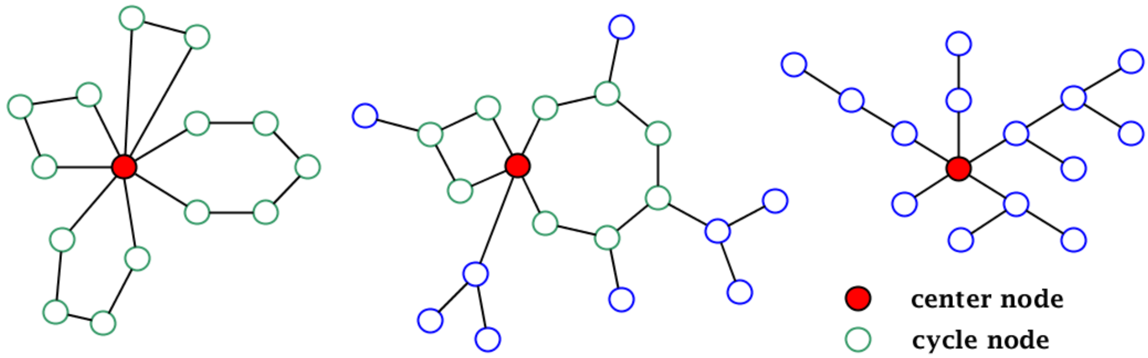


Figure 1 **Ring tree stars of order 4 (left), 3 (center) and 5 (right).**

For given capacity bounds $m,q \in \mathbb{N}$, a ring tree star $\mathcal{N}$ centered in d is a solution for the CRTP if it contains given customer nodes $U = U_1 \dot{\cup} U_2$ and a subset of given Steiner nodes W such that

(i) each customer node in U_2 is on a cycle in $\mathcal{N}$,

(ii) the order of $\mathcal{N}$ is at most m, and

(iii) each connected component in $\mathcal{N} \setminus d$ contains at most q customers.

Let c_e be the cost for the installation of an edge e in $\mathcal{N}$. Then the CRTP asks for a solution that minimizes the sum of the edge costs of the ring tree star, i.e. $\sum_{e \in E[\mathcal{N}]} c_e$. The CRTP is NP-hard as it generalizes the TSP, among others. We say that nodes in U_2, also called type 2 nodes, correspond to customers of type 2, whereas nodes in U_1 are of type 1 and correspond to type 1 customers. Type 1 nodes have to be connected to d by at least one path while type 2 nodes have to be connected to d by exactly two node disjoint paths. The cycles in $\mathcal{N}$ are also called *rings*. By requiring the type 2 nodes to be part of such rings we provide additional reliability to the corresponding customers: there are exactly two (node) disjoint paths from such a node to d. This double-connectivity is optional for the remaining type 1 nodes, and the Steiner nodes in W are not even required to be nodes in $\mathcal{N}$ unless beneficial regarding the overall network cost. A solution for the CRTP is illustrated by Figure 2. We denote the set of all available nodes $U \dot{\cup} W \dot{\cup} \{d\}$ as V and the set of potential network edges $\{e \subseteq V : |e| = 2\}$ as E. Moreover, we refer to the set of nodes of a graph $\mathcal{G}$ by $V[\mathcal{G}]$ and to its edges by $E[\mathcal{G}]$.

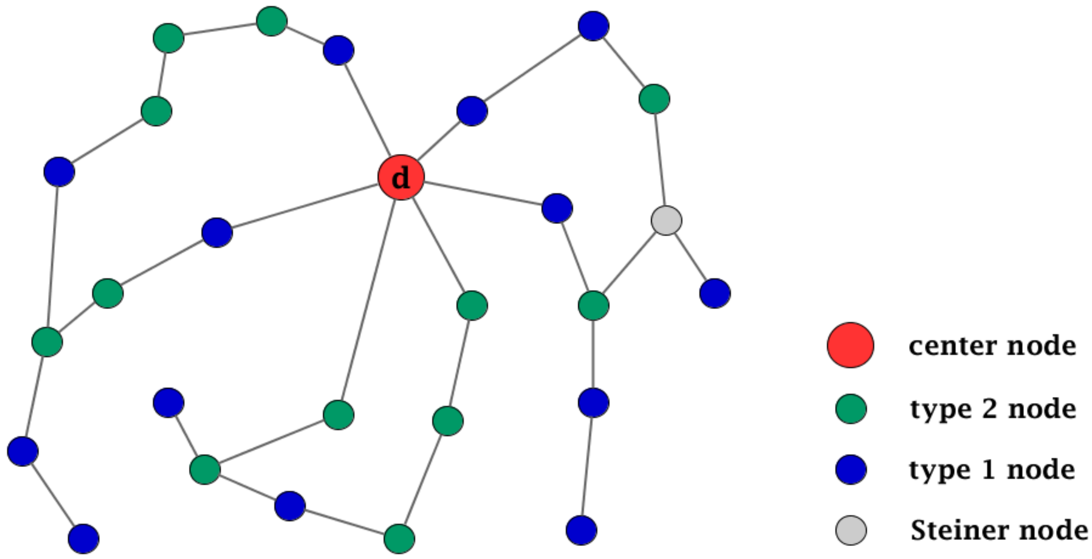


Figure 2 **An optimal solution for instance 13 ($q = 10$, $m = 3$, $|U_1| = 13$, $|U_2| = 12$) implementing a ring tree star of order 3.**

The following non-compact MIP formulation (**F**) was developed in Hill and Voß (2015). It is based on a directed Steiner tree problem formulation in which rings are enforced by circulations. We observe that removing one center-incident edge in each ring turns a ring tree star into a tree. This tree can be transformed into a directed tree by rooting it in the center d and replacing the edges by arcs such that each leaf node can be reached from d via a directed path. A directed cycle can be formed in the directed network by the insertion of an arc towards d. To ensure that each type 2 node is on such a directed cycle we require a circulation to pass through them and d for each ring. Such a directed network induces a solution for the CRTP obtained by replacing arcs by edges. We denote the set of potential arcs as A and a binary arc variable x_a is used to indicate whether arc a will be installed. The circulation on an arc is modeled by a continuous arc circulation flow variable f_a. Arcs leaving [entering] a node set S are denoted by $\delta^+(S)$ $[\delta^-(S)]$.

$$(\mathbf{F}) \quad \min \sum_{e \in E[G]} c_e y_e \tag{1}$$

$$s.\,t. \sum_{a \in \delta^-(S)} x_a \geq \frac{|U(S)|}{q} \quad \forall\, S \subset V \setminus d, \tag{2}$$

$$\sum_{a \in \delta^-(\{i\})} x_a = 1 \quad \forall\, i \in U, \tag{3}$$

$$\sum_{a \in \delta^-(\{i\})} x_a \;\leq\; 1 \quad \forall\, i \in W, \tag{4}$$

$$\sum_{a \in \delta^+(\{d\})} x_a \;\leq\; m, \tag{5}$$

$$x_{ij} + x_{ji} \;=\; y_{ij} \quad \forall\, \{i,j\} \in E, \tag{6}$$

$$\sum_{a \in \delta^-(\{i\})} f_a \;=\; \sum_{a \in \delta^+(i)} f_a \quad \forall\, i \in V, \tag{7}$$

$$\sum_{a \in \delta^-(\{i\})} f_a \;=\; 1 \quad \forall\, i \in U_2, \tag{8}$$

$$0 \;\leq\; f_a \;\leq\; x_a \quad \forall\, (i,j) \in A, \tag{9}$$

$$x_a \in \{0,1\} \quad \forall\, a \in A, \tag{10}$$

$$y_e \in \{0,1\} \quad \forall\, e \in E. \tag{11}$$

Assignment constraints (3) ensure an in-degree equal to one for each customer, whereas the capacity constraints (4) limit the inbound arcs to one for each Steiner node. The *capacitated connectivity constraints* (2) bound the number of customers per ring tree to q. These exponentially many constraints are separated dynamically during the branch and bound procedure presented in Hill and Voß (2015). We enforce the underlying circulation by (in)equalities (7), (8) and (9). Since we consider directed ring tree stars, inequality (5) is sufficient to limit the number of ring trees to m. To obtain a simple undirected solution network and identify its edges we implement the variable linking equalities (6). For a more detailed discussion of this formulation we refer to Hill and Voß (2015).

3. Generalized local branching

In this section we describe our generic framework. The specific application of our techniques to the CRTP follows in Section 4. Since we mainly use integer programming techniques we give descriptions using terminology and concepts from mathematical programming, in particular, a branch and bound framework. We assume that we have a MIP formulation at hand in which the network structure is encoded by binary edge variables. The presented techniques can be adapted to different mathematical programming approaches and, moreover, to related network design problems. Therefore, we consider a generic MIP

$$(\mathbf{P}) \quad min\ c^T y,\ Ay \leq b,\ y \in \{0,1\}^{|E|}, \tag{12}$$

with y_e being the variable indicating whether the edge $e \in E$ is installed in the solution network. Without loss of generality, we omit eventual continuous variables in $(\mathbf{P})$ for the sake of a simplified description. The constraints in (12) describe the integer feasible solutions as a subset of the fractional solutions contained in the polyhedron Γ described by $Ay' \leq b,\ y' \in \mathbb{R}^{|E|}$. A *cut* for $(\mathbf{P})$ is an inequality that describes a halfspace in $\mathbb{R}^{|E|}$ whose intersection with Γ is non-trivial. Moreover, we assume that we are given a feasible *reference solution* $\tilde{y}$ for $(\mathbf{P})$, as illustrated in Figure 3 (left), which represents a *solution network* $\mathcal{N}_{\tilde{y}}$. We reformulate the concepts of local branching and integer programming based refinements in Sections 3.1 and 3.2 before we bring them together in Section 3.3. Finally, we describe the introduced cuts on the unit cube in Section 3.4.

3.1. Local branching

Local branching (LB) was introduced by Fischetti and Lodi in Fischetti and Lodi (2003) as a polishing heuristic for a general purpose MIP solver. It is applied whenever an integer feasible solution $\tilde{y}$ is found in the branch and bound algorithm that replaces the current incumbent. To carry out the local search a restricted MIP is solved that is obtained by adding a *local branching cut* to the master problem. For $k \in \mathbb{N}_0$ such an inequality

$$\sum_{e \in E : \tilde{y}_e = 1} (1 - y_e) + \sum_{e \in E : \tilde{y}_e = 0} y_e \ \leq\ k \tag{13}$$

induces a *k-opt neighborhood* $N(\tilde{y}, k)$ of $\tilde{y}$. It contains each feasible solution y for $(\mathbf{P})$ within *Hamming distance* $\Delta(\tilde{y}, y) = |\{e \in E : \tilde{y}_e \neq y_e\}| \leq k$ from $\tilde{y}$. Figure 3 (center) illustrates $N(\tilde{y}, k)$ in a branching scheme for $k = 2$. This technique found its way into commercial solvers such as CPLEX. *Inverse local branching cuts* can be obtained by reversing the sense of (13) but turned out to be less effective in practice.

A related concept in the heuristic literature is *limited discrepancy search* (LDS) Harvey and Ginsberg (1995), in which the decision tree is also traversed respecting a bound on the deviation from a reference solution. Compared to LB, LDS does not necessarily take place in an exact mathematical programming environment and is rather constraint satisfaction oriented in its original version.

3.2. Refinement techniques

The exploration of neighborhoods of $\tilde{y}$ by exact methods is likewise the key ingredient for *MIP refinement techniques*. In contrast to the LB idea in Section 3.1 it focuses on the re-optimization of a subset of decision variables without bounding $\Delta(\tilde{y}, y)$ for a neighboring solution y. In terms of mathematical programming this idea can be translated to *variable fixing* of the remaining variables. However, variable fixing techniques typically use information from solutions of the linear relaxed problem to deduce integer feasible solutions (e.g. Danna et al. (2005), Berthold (2014)). Related approaches that round fractional solutions to integers are used to find an integer feasible solution at all (e.g. Fischetti et al. (2005)). Additionally, structural knowledge about the underlying optimization problem is usually exploited in problem specific branch and bound algorithms. For the network $\mathcal{N}_{\tilde{y}}$ we can fix the current state of edges $F \subseteq E$ by adding the following *variable fixing equalities* to $(\mathbf{P})$.

$$y_e = \begin{cases} 0 & if \quad e \notin E[\mathcal{N}_{\tilde{y}}] \\ 1 & if \quad e \in E[\mathcal{N}_{\tilde{y}}] \end{cases} \quad \forall\, e \in F \tag{14}$$

This defines the neighborhood $N(\tilde{y}, F)$ containing all the feasible solutions in $(\mathbf{P})$ such that a variable y_e that corresponds to an edge $e \in F$ is forced to 1 if e is installed in $\mathcal{N}$ and to 0 otherwise. The remaining edge variables (for edges in $E \setminus F$) are free to take any constraint-feasible value in $\{0, 1\}$. Figure 3 (right) depicts solutions in $N(\tilde{y}, F)$.

3.3. Combining local branching and refinements

In this section we generalize the concepts of local branching and MIP refinements to obtain the generic concept of *generalized local branching* (GLB). To develop the latter we first observe that the refinement techniques described above can be formulated as local branching on a 2-partition of the edge set: the fixed edges and the flexible ones. Again, let $F \subseteq E$ be the set of edges that should be fixed to their current values in $\tilde{y}$ and let $E \setminus F$ contain the remaining edges whose variables are considered for refinement. Then we can achieve this by the addition of two *partial* local branching cuts. The first one defines the trivial neighborhood $N(\tilde{y}^F, 0)$ restricted to the edges in F. The second one corresponds to all the feasible solutions $N(\tilde{y}^{E \setminus F}, |E|)$. More generally, let $\mathcal{F} = (F_1, ..., F_p)$ be a cover of the edge set E. Then we apply individual Hamming bounds $\mathcal{K} = (k_1, ..., k_p) \in \mathbb{N}_0^p$ to the corresponding variable subsets. Hereby we can incorporate pre-knowledge about how many

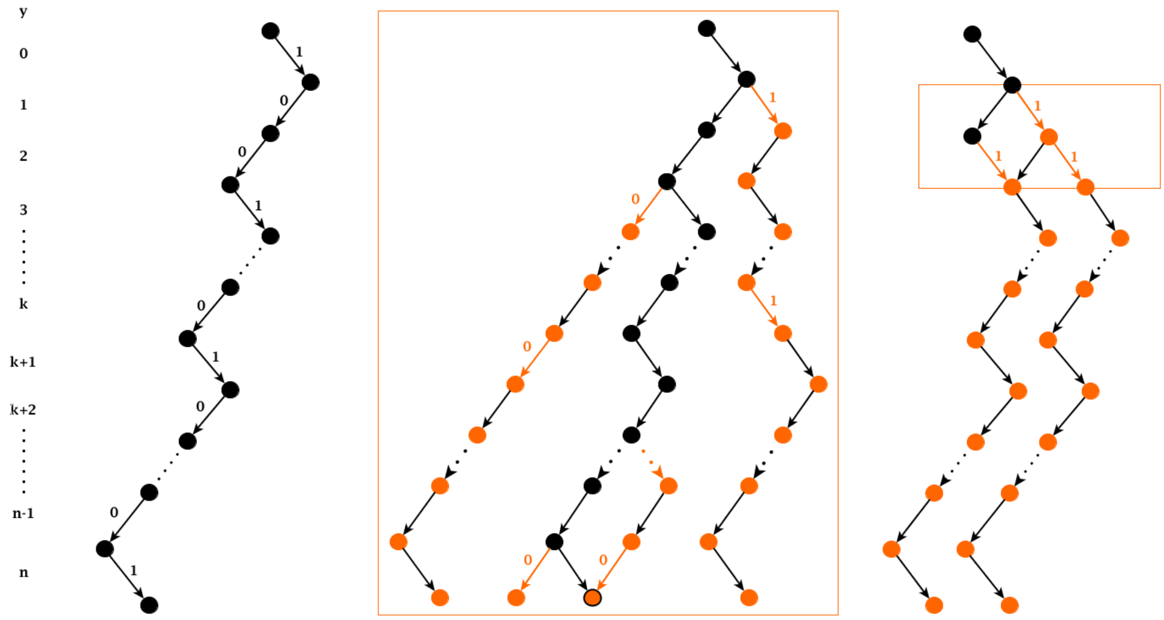


Figure 3 **A solution $\tilde{y}$ of (P) in a branch and bound tree (left), five solutions in $N(\tilde{y},2)$ (center) and three solutions in $N(\tilde{y},F)$ with $F=\{e_0,e_3,...,e_n\}$ (right).**

changes we expect after sub-optimizing each set F_i. We define *generalized local branching cuts* as a system of LB cuts

$$\sum_{e\in F_i:\tilde{y}_e=1}(1-y_e)+\sum_{e\in F_i:\tilde{y}_e=0}y_e \;\leq\; k_i \quad \forall\, i\in\{1,...,p\}. \tag{15}$$

For each edge set F_i we limit the number of variable flips among the corresponding edge variables by the constant k_i. In particular, if $k_i=0$ then the part of the current solution $\mathcal{N}_{\tilde{y}}$ represented by F_i is fixed. $k_i\geq|E|$ means that the partial solution is considered for full refinement. We denote the corresponding neighborhood by $N(\tilde{y},\mathcal{F},\mathcal{K})$. We refer to (**P**) extended by GLB cuts (15) as the *generalized local branching problem* (GLBP) corresponding to $\mathcal{F}$ and $\mathcal{K}$. Note that this concept is related to the concept of defining corridors within the corridor method Sniedovich and Voß (2006). However, the latter attempts to increase the set of decisions that is considered for refinement depending on the optimization method at hand, whereas our approach is designed to work on arbitrary subnetworks, in presence of the Hamming bounds, though. Following the approach in Fischetti and Lodi (2003) we can define *inverse generalized local branching cuts* for $\mathcal{F}$ and $\mathcal{K}$ as inequalities (15) with flipped signs which we will not further investigate in this work.

So far we did not address strategies to set up a suitable $\mathcal{F}$ and $\mathcal{K}$. In Section 4 we focus on the CRTP and present a practical implementation of these concepts. The described GLB cuts can then also be integrated in an exact algorithm as it was originally suggested for LB in Fischetti and Lodi (2003). As common for exact algorithms, the branch and cut method for the CRTP that we use in the next section already incorporates such a polishing, based on CRTP-specific local search Hill (2015). However, we consider the pure improvement heuristic in this work which is embeddable in arbitrary algorithmic frameworks.

3.4. GLB cuts on the unit cube

In the following we discuss the geometric interpretation of the GLB cuts introduced above using the unit cube. Let Φ be the convex hull of the integer points in Γ. We recall that cuts used to accelerate cutting plane algorithms (e.g. disjunctive, cover, Gomory mixed integer) tighten Γ to describe Φ. In contrast to these techniques, the GLB cuts aim at downsizing Φ regardless of the resulting linear programming relaxation. However, the mentioned cutting plane techniques, together with problem specific ones, play an important role when solving the reduced GLBPs within the overall algorithm in Section 4.

GLB cuts are in general not facet defining. To see this, consider an edge variable based formulation for the symmetric TSP on $n > 3$ nodes and $m = n(n-1)/2$ edges. Let $\mathcal{F}_0 = (\{y_1, y_3, y_{n+1}, ..., y_m\}, \{y_2, y_4, ..., y_n\})$ and $\mathcal{K}_0 = (3, 0)$ be GLB parameters and $y_1, ..., y_n$ the ordered variables that correspond to the edges in the current reference tour. Then the GLB cuts obtained through $\mathcal{F}_0$ and $\mathcal{K}_0$ allow up to three variable flips for the variables that correspond to the non-incident tour edges y_1 and y_3 as well as all unused edges. Certainly, no feasible solution can be obtained by removing either one of these tour edges while adding two unused edges. At the same time, the elimination of y_1 and y_3 cannot be repaired by a single edge insertion. Only the augmentation of the first Hamming bound to 4 would lead to facet-defining cuts and a non-trivial neighborhood.

In the following example in $\{0,1\}^3$ let $\tilde{y} = [0,0,0]$ be a reference solution. For the sake of simplicity we assume that $\Phi = \{0,1\}^3$. Although the purpose of this setting is to better understand the GLB cuts, it could be a model for a price-collecting spanning tree problem on three nodes. We limit ourselves to GLB cuts with $p = 2$ to avoid trivial cuts. For $F_1 = \{y_1\}$, $F_2 = \{y_2, y_3\}$ and $k_1 = 0$, $k_2 = 1$ let $\mathcal{F}_1 = (F_1, F_2)$ and $\mathcal{K}_1 = (k_1, k_2)$. Then we construct the following GLB cuts according to (15).

$$y_1 \leq 0 \quad , \quad y_2 + y_3 \leq 1 \tag{16}$$

We denote the corresponding hyperplanes by H_1 and H_2. These cuts reduce Φ to the convex hull of $N(\tilde{y}, \mathcal{F}_1, \mathcal{K}_1)$ projected onto H_1. As shown in Figure 4 (left), the resulting triangle-shaped linear search space is spanned by the integer-feasible solutions $\tilde{y}$, $[0,1,0]$ and $[0,0,1]$. To see the effect of the chosen Hamming bounds, let us first consider $k'_2 = 0$ and the corresponding hyperplane H'_2. Figure 4 (center) shows the resulting trivial neighborhood containing $\tilde{y}$ only. Conversely, if $k''_2 = 2$ then the resulting hyperplane H''_2 does not cut off any points in Φ and all the four integer solutions in $\Phi \cap H_1$ are considered. The latter corresponds to a MIP refinement search as described in Section 3.2.

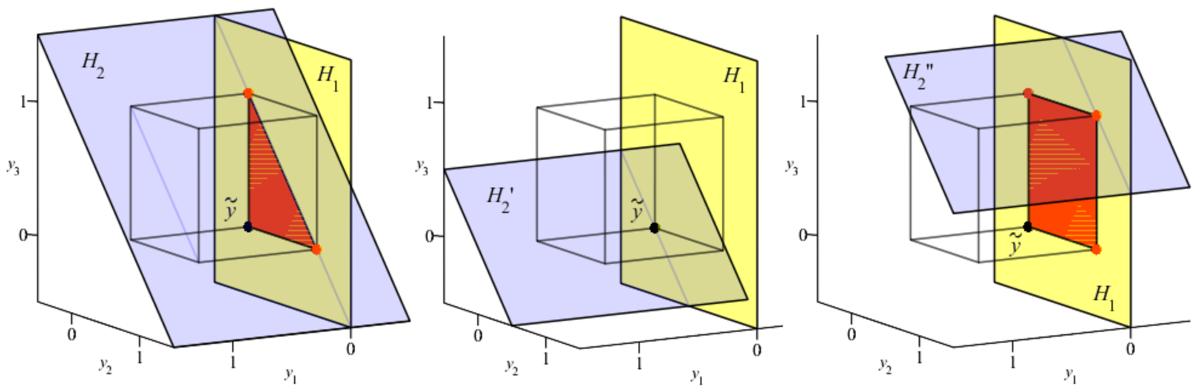


Figure 4 **Three GLB cuts with respect to reference solution** $\tilde{y} = [0,0,0]$.

Using k_2 we can explore all the points within Hamming distance one to $\tilde{y}$ by rearranging $\mathcal{F}_1$. This can be achieved by the variable partitions $\mathcal{F}_1$, $(\{y_3\}, \{y_1, y_2\})$ and $(\{y_2\}, \{y_1, y_3\})$. The resulting neighborhoods are illustrated in Figure 5. The exploration of these three search spaces is inefficient from a computational point of view due to notable redundancy regarding integer points. The three integer feasible solutions appear as potential improved solutions six times in total. Alternatively, Φ can be reduced to the corresponding simplex by the pure LB cut $y_1 + y_2 + y_3 \leq 1$ which can be expressed as a GLB cut with $\mathcal{F} = (\{y_1, y_2, y_3\})$ and $\mathcal{K} = (1)$.

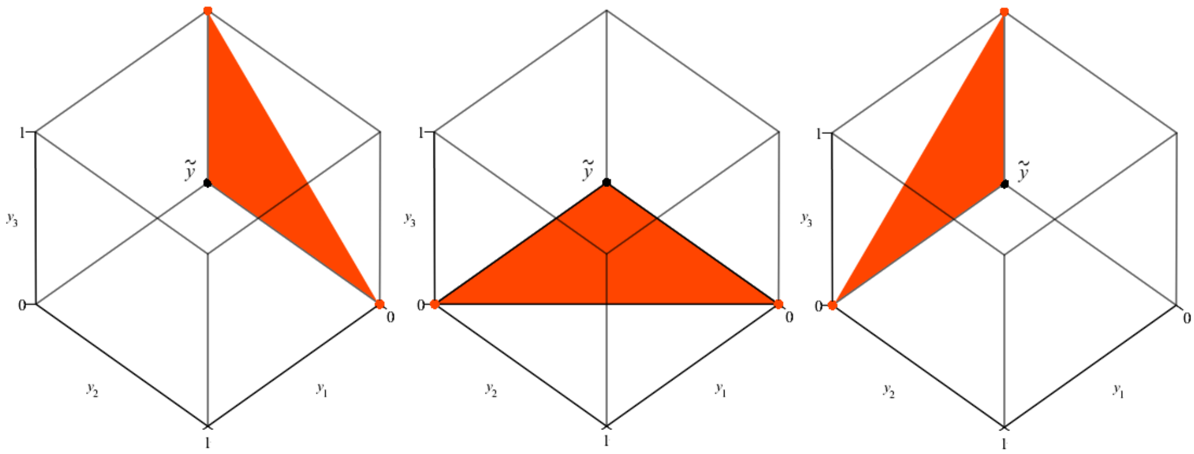


Figure 5 **Three GLB search spaces with respect to reference solution $\tilde{y}$.**

Furthermore, the union of the subspaces depicted in Figure 5 does not cover Φ. In order to take into account all integer solutions, let us now alter the Hamming bound for the first variable set F_1. Since $|F_1| = 1$ we only consider the incremented value, i.e. $k_1' = 1$. Figure 6 illustrates the effect on Φ using k_2 (left), k_2' (center) and k_2'' (right). It can be seen that the corresponding hyperplane H_1' does not cut off any integer points. Therefore, this parameterization corresponds to pure local branching as described in Section 3.1.

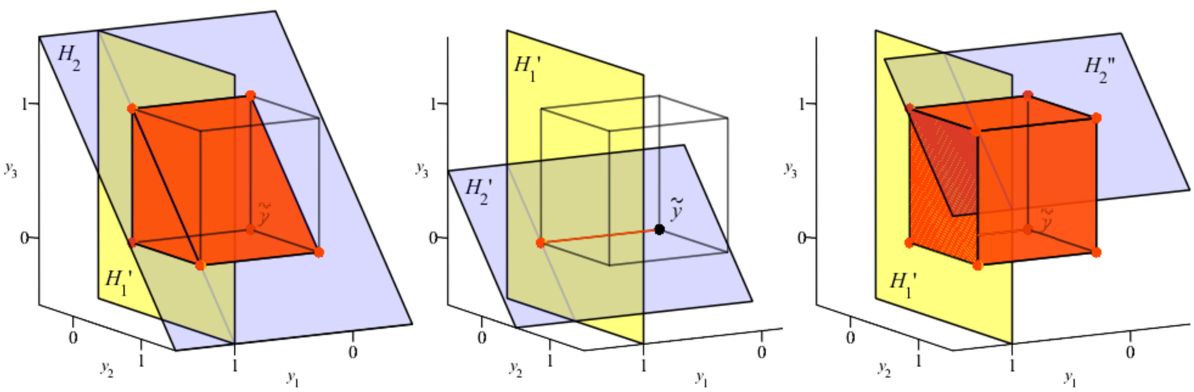


Figure 6 **Three GLB cuts with respect to reference solution $\hat{y}$.**

By definition, GLB cuts can be parameterized for a given variable partition $\mathcal{F}$ to induce all subspaces of Φ that contain integer points within Hamming distance k_i with respect to F_i for each $1 \leq i \leq p$; in particular, the reference solution $\tilde{y}$. For the example above, eight configurations exist for the 3-partition of the variable set ($k_1, k_2, k_3 \in \{0, 1\}$), eighteen for the three 2-partitions ($k_1 \in \{0, 1\}, k_2 \in \{0, 1, 2\}$ or $k_1 \in \{0, 1, 2\}, k_2 \in \{0, 1\}$) and four for the

1-partition ($k_1 \in \{0,1,2,3\}$). This results in eleven (=1+3+3+3+1) distinct neighborhoods for a fixed $\tilde{y}$. In Section 4 we propose a strategy to explore such neighborhoods effectively for the CRTP.

4. A generalized local branching algorithm for the CRTP

We first describe single-ball node clustering techniques that will serve to build effective GLB cuts in Section 4.1. The CRTP tailored GLB techniques are presented in Section 4.2. We explain characteristics of the underlying exact mathematical programming approach to be taken into account, such as cut management, in Section 4.3. This section is closed by the overall strategy in Section 4.4.

4.1. Single-ball clusters

In general local search algorithms the exploration of multiple suitable diverse neighborhoods is known to be effective, as shown for the CRTP in Hill (2015). To study the strength of our GLB cuts we will limit ourselves to simple but effective *ball* type neighborhoods of a reference network $\mathcal{N}$ in this work. In several variants, these were already successfully implemented in MIP refinement techniques in Hill and Voß (2014). Hence, we use the following node clustering strategy to generate structured 2-partitions of the edge variables needed to deduce GLB cuts.

For a *cluster center node* $v \in V[\mathcal{N}]$ we define a *(single-)ball cluster* $B_{\mathcal{N}}(v,r) \subseteq V[\mathcal{N}]$ as the set containing v and the $r-1$ $(0 \le r < |V[\mathcal{N}]|)$ closest nodes to v with respect to the distance function c. Two such balls are given in Figure 7.

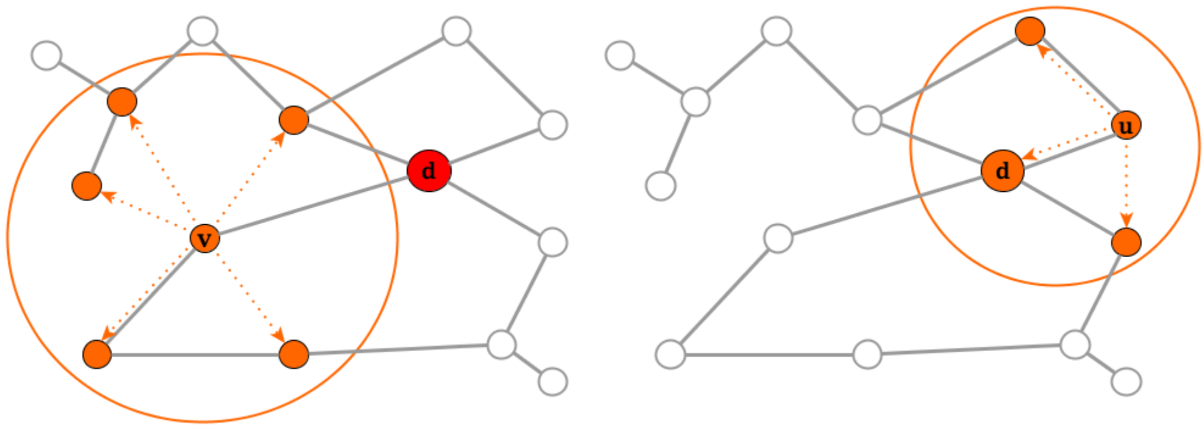


Figure 7 Two single-ball clusters $B_{\mathcal{N}}(v,6)$ and $B_{\mathcal{N}}(u,4)$.

For the overall effectiveness of the GLBPs the selection of the ball cluster centers is crucial. To locally optimize the current solution $\mathcal{N}$ evenly, we build ball node clusters for well distributed cluster centers in order to cover the whole network. Starting with a cluster center node of highest cumulative distance to the remaining nodes, we iteratively choose the next cluster center by adding the most remote node with respect to the previously selected ones. We build a cluster $B_{\mathcal{N}}(v,r)$ for each hereby obtained node v and a suitable r. The edges incident to the depot are of particular importance since they are closely related to the number of installed ring trees. Moreover, the opening and closure of ring trees is facilitated by these edges. It turned out to be beneficial to include d in each ball. For a very large relative number of Steiner nodes it could be convenient to limit their inclusion which was not necessary in our experiments.

4.2. Generalized local branching cuts for the CRTP

We now use the idea of GLB cuts from Section 3.3 to construct GLBPs based on the single-ball clusters described in Section 4.1. For such a cluster $B \subseteq V[\mathcal{N}]$ let $I_E[B]$ be the set of edges in E with both end nodes in B. Then we add GLB cuts (15) for the induced edge partition

$$\mathcal{F}_B = (I_E[B], E \setminus I_E[B])$$

and Hamming bounds

$$\mathcal{K} = (k, 0).$$

We parameterize the GLBPs by the cluster size and the Hamming bound k as follows. Assume that we know an estimate $\underline{r}$ for the largest computing machine dependent cluster size such that the pure refinement problem can be solved efficiently by our exact method. In practice, $\underline{r}$ can be determined by a calibration mechanism in which, for a sufficiently large $\overline{k}$, a cluster is incrementally increased as long as the corresponding GLBP can be solved within a reasonable time limit. Contrary, let $\overline{r}$ be the estimated largest cluster size such that the GLBP for a small non-trivial $\underline{k}$ can be solved efficiently. $\overline{r}$ can be obtained by a similar procedure as used for $\underline{r}$.

We say a GLB *scheme* of *order* m is a sequence of pairs $(r_1,k_1),...,(r_m,k_m)$ used to construct GLBPs with $|B| = r_i$ and Hamming bound k_i for $i \in \{1,...,m\}$. The scheme that we use in our algorithm arises from linear interpolation with respect to $(\underline{r},\overline{k})$ and $(\overline{r},\underline{k})$. More precisely, we use a step size ρ to define the cluster sizes $\underline{r}, \underline{r}+\rho, ..., \overline{r}$. The Hamming bounds are set to $\overline{k}, \overline{k}-\kappa, ..., \underline{k}$ where $\kappa = \rho(\overline{k}-\underline{k})/(\overline{r}-\underline{r})$. Here we presume that increasing the complexity of the GLBP by enlarging the set of cluster nodes B, and therewith the set of flexible variables $I_E[B]$, can be compensated by the reduction of the number of allowed variable flips k, which turned out to be suitable for our approach.

Certainly, $\underline{r}$ and $\overline{r}$ are sensitive to the hardness of the instance and the chosen balls. In particular, the capacity bounds m and q, the customer type ratio $|U_1 \cap B|/|B|$ and the Steiner node portion $|W \cap B|/|B|$ in B can have an impact on the performance of the exact method. In Figure 8 we illustrate two GLB cuts using a cylindrical representation of the GLBP solution spaces in terms of k and $r = |B|$.

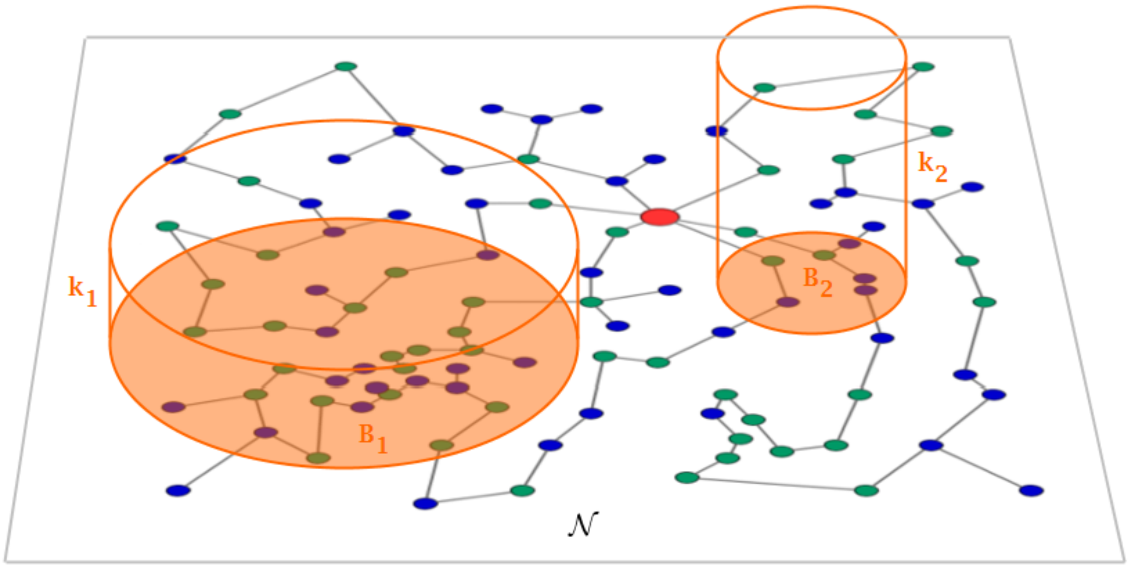


Figure 8 **Cylindrical illustration of two different single-ball GLB cuts with $r_1 = |B_1| = 34$ and $r_2 = |B_2| = 6$, and Hammig bounds k_1 and k_2, each of them (and both together) inducing a GLBP for the solution $\mathcal{N}$.**

4.3. The underlying branch and cut method

We use the branch and cut method presented in Hill and Voß (2015) as underlying mathematical programming based algorithm. When running the method including the GLB cuts some formulation-specific characteristics need to be considered. As typical for branch and cut algorithms the cut management plays an important role for the efficiency and the stability of the method. Model cuts as well as valid cuts added at the root node are indispensable to obtain strong lower bounds. However, their separation can be time consuming. To avoid the repeated generation of such master problem root cuts in each GLBP we pre-compute these in an initialization phase and add them to each model. Moreover, it turned out to be advantageous to continuously extend this set of inequalities by the cuts separated within the GLBPs. Unfortunately, the found solver-internal cuts cannot be accessed and eventually have to be re-calculated dynamically.

We provide the current incumbent solution to the branch and cut method each time it is called for solving a GLBP. The local search based polishing procedures to accelerate the branch and cut algorithm may return a solution that violates the current GLB cuts. Nonetheless, we update the current best solution in this case since they it is feasible for the original problem.

Assuming a metric edge cost function c, and therefore c satisfying the triangle inequality, the mathematical formulation for the CRTP does not need constraints enforcing a single directed path from each cycle node to the depot. However, these restrictions are necessary when performing GLB refinements. The partial fixing of a cycle structure in general forces the creation of non-ring-tree structures as shown in Figure 9.

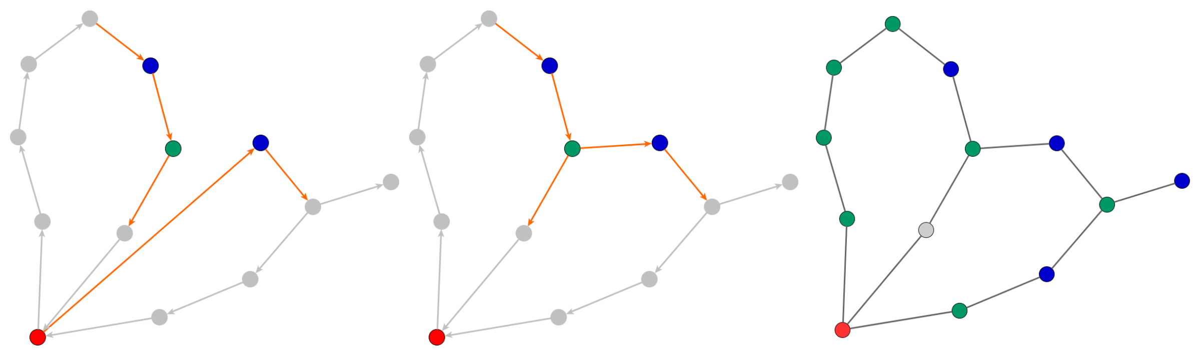


Figure 9 An infeasible structure due to missing ring enforcing circulation constraints.

4.4. Overall strategy

In this section we present the overall strategy used in the GLB heuristic. Basically, we refine an initial ring tree star $\mathcal{N}$ according to the GLB parameters (r, k) provided by a GLB scheme sorted by increasing cluster size $r = |B|$. Figure 10 illustrates the relation of k and $|B|$ in our algorithm.

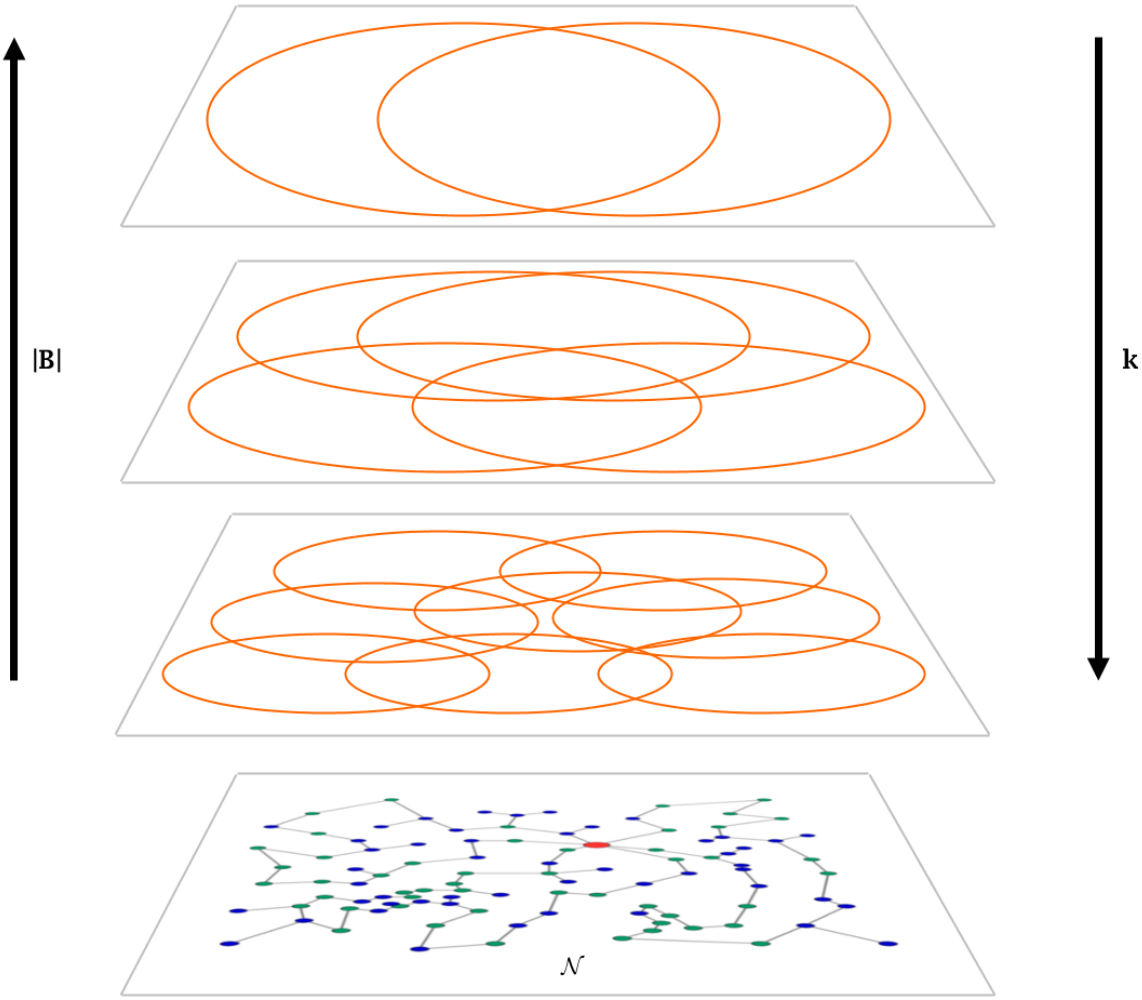


Figure 10 **With an increasing ball size $|B|$ we decrease k and the number of considered balls in our GLB scheme.**

For a cluster size r and a Hamming bound k the procedure 1 locally optimizes $\mathcal{N}$ using the ball clustering strategy presented in Section 4.1 until no improvement can be found. Improvements are immediately incorporated in $\mathcal{N}$. The function findSingleBallCenters($\mathcal{N}, P$) returns a set of customers used for the construction of ball clusters. The GLB cut for the refinement edge variables induced by a node set B and a Hamming bound k is returned by $\text{Cut}_{GLB}(I_E[B], k, \mathcal{N})$. $\text{Cut}_{GLB}(E \setminus I_E[B], 0, \mathcal{N})$ returns the GLB cut that fixes the remaining edge variables in the master MIP, denoted by $\mathbf{F}(P)$. We denote the MIP obtained by the addition of a cut R to $\mathbf{F}(P)$ by $\mathbf{F}(P) \oplus R$. A GLBP $\mathbf{F}$ is solved by the exact algorithm by calling the procedure solve($\mathbf{F}$) which either returns an improving solution or the incumbent.

Algorithm 1 REFINE()

Input: CRTP P, solution $\mathcal{N}$, cluster node limit r, Hamming bound k;
repeat
  $c_{old} \leftarrow c(\mathcal{N})$;
  $Z \leftarrow$ findSingleBallCenters$(\mathcal{N}, P)$;
  for all $v \in Z$ **do**
    $B \leftarrow B_{\mathcal{N}}(v, r)$;
    $\mathbf{F} \leftarrow \mathbf{F}(P) \oplus \text{Cut}_{GLB}(I_E[B], k, \mathcal{N})$;
    $\mathbf{F} \leftarrow \mathbf{F} \oplus \text{Cut}_{GLB}(E \setminus I_E[B], 0, \mathcal{N})$;
    $\mathcal{N} \leftarrow$ solve$(\mathbf{F})$;
  end for
until $c(\mathcal{N}) = c_{old}$
return $\mathcal{N}$;

Algorithm 2 describes the main procedure including the construction of a start solution by generateStartSolution(P) and the calculation of the GLB scheme by generateSortedSchemeGLB(P).

Algorithm 2 The GLB heuristic for the CRTP.

Input: CRTP P;
$\mathcal{N} \leftarrow$ generateStartSolution(P);
$\mathcal{H} \leftarrow$ generateSortedSchemeGLB(P);
while $|\mathcal{H}| > 0$ **do**
  $(r, k) \leftarrow$ pop$(\mathcal{H})$;
  $\mathcal{N} \leftarrow$ REFINE$(P, \mathcal{N}, r, k)$;
end while

5. Computational study

In this section we provide computational results for literature instances and give a comprehensive analysis of the algorithm's performance.

5.1. Framework, implementation and parameterization

We study our algorithm on 225 CRTP instances[1] that were suggested for the CRTP in Hill and Voß (2015). These contain up to 101 nodes and are deduced from TSPlib-based instances. For 102 instances no optimal solutions are known according to Hill and Voß (2015). The algorithm is implemented in C++ using the CPLEX 12.6 MIP solver. Computations are done on an Intel i7-4610M 3.00 GHz processor unit with 16 GB RAM.

To show the strength of our new strategy we consider the following four configurations.

(A) **Heuristically enhanced GLB:** the GLBPs are constructed for a pre-specified scheme; heuristic start solution Hill (2015); local search polishing of each improved solution found Hill (2015).

(B) **Self-contained GLB:** the GLBPs are constructed for a pre-specified scheme; random start solution; *no* local search.

(C) **Pure refinement:** the GLBPs are constructed for one "small" ball size only; k is set to the number of current ball edges; heuristic start solution Hill (2015).

(D) **Pure local branching:** the GLBPs are constructed for one "large" ball size only; k is set to a small value; heuristic start solution Hill (2015).

To solve the GLBPs we use the efficient branch and cut algorithm developed in Hill and Voß (2014) using a time limit of 120 seconds. This exact approach originally incorporates the heuristic techniques to construct start solutions and for solution polishing developed in Hill (2015). Note that we use an improved version of the heuristic presented in Hill (2015): each construction in the multi-start algorithm is repeated using a shortest path Steiner tree heuristic instead of the computation of the minimum spanning tree. To understand the impact of the latter, configuration (B) does not include these features. A random start solution is generated by randomly selecting type 2 customers to build rings at full capacity instead. The remaining type 1 customers are iteratively used to construct trees from ring nodes in a random fashion. This results in feasible networks of 1.2 to 6.5 times the cost of the best known solution (3.1 on average). Configurations (C) and (D) represent the pure

[1] The test set can be downloaded from `http://dimacs11.cs.princeton.edu/downloads.html`

refinement and LB introduced in Section 3.2 and Section 3.1.

In our experiments we set the cluster size step size ρ to 12. We determine a minimal cluster size $\underline{r} = 14$ and a maximal cluster size $\overline{r} = \min(62, |\tilde{\mathcal{N}}|)$ as suitable by a series of calibration runs. For the maximal and the minimal number of variable flips we use $\overline{k} = 14$ and $\underline{k} = 4$, respectively. The number of ball clusters constructed for a pair (r, k) is set to $\lfloor 2.9|V[\mathcal{N}]|/r \rfloor$. Figure 12 (left) shows the complete GLB scheme. To further strengthen the primal GLBP bounds we turn on the solver-internal polishing heuristics, including local branching.

5.2. Results

Tables 1, 2 and 3 contain the results obtained by our GLB method with configurations (A), (B), (C) and (D). The column information for each instance is explained as follows.

P : CRTP instance

|U| : overall number of customers

μ : rate of type 1 customers ($|U_1|/|U|$)

$c(\tilde{\mathcal{N}})$: start solution costs

$c(\mathcal{N})$: costs computed by the GLB configuration

Ω : number of considered GLBPs (MIPs)

\# : number of refinements

t : GLB runtime in seconds

γ : best achieved relative cost improvement in % with respect to the solution cost α computed by the heuristic in (Hill 2015): $(\alpha - c(\mathcal{N}))/\alpha$

θ : relative reduction of the optimality gap[2] $ub - lb$ in %; $100(ub - c(\mathcal{N}))/(ub - lb)$

Bold cost values indicate that no better solution was found by our heuristics. An $*$ is prefixed if the solution is proven optimal according to the results in Hill and Voß (2015). When $\Omega = 0$ then the instance is already solved to optimality during the initial cut generation phase.

Algorithm (A) clearly outperforms (B), (C) and (D) in terms of number of best solutions found. As shown in Figure 11, 98% of the best solutions are found by (A) compared to 37%, 66% and 73% for (B), (C) and (D). The run times for the whole instance set are about 68, 109, 14 and 26 hours, respectively. Even though starting with a random solution

[2] lb and ub are the bounds computed by the branch and cut algorithm in (Hill and Voß 2015)

Table 1 Results for the 225 TSPlib-based CRTP instances.

P	$\vert U\vert$	μ	(A) $c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	(B) $c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	(C) $c(\mathcal{N})$	Ω	#	t	(D) $c(\mathcal{N})$	Ω	#	t	Best γ	θ
1	12	1	**∗157**	**∗157**	0	0	1	265	**∗157**	0	0	1	**∗157**	0	0	1	**∗157**	0	0	2	0	0
2		0.75	215	**∗210**	9	1	17	348	**∗210**	9	1	9	**∗210**	9	1	14	**∗210**	9	1	16	2.4	0
3		0.5	**∗227**	**∗227**	3	0	8	340	232	9	1	11	**∗227**	3	0	7	**∗227**	3	0	8	0	0
4		0.25	**∗236**	**∗236**	3	0	6	413	**∗236**	9	2	7	**∗236**	3	0	5	**∗236**	3	0	5	0	0
5		0	**∗242**	**∗242**	3	0	5	364	**∗242**	9	1	4	**∗242**	3	0	5	**∗242**	3	0	4	0	0
6	12	1	**∗163**	**∗163**	0	0	0	271	**∗163**	0	0	1	**∗163**	0	0	0	**∗163**	0	0	0	0.6	0
7		0.75	**∗207**	**∗207**	0	0	2	366	**∗207**	0	0	2	**∗207**	0	0	1	**∗207**	0	0	2	0	0
8		0.5	**∗240**	**∗240**	3	0	7	412	**∗240**	3	0	5	**∗240**	3	0	7	**∗240**	3	0	7	0	0
9		0.25	**∗249**	**∗249**	3	0	4	346	**∗249**	9	1	6	**∗249**	3	0	5	**∗249**	3	0	5	0	0
10		0	**∗251**	**∗251**	0	0	1	348	**∗251**	0	0	1	**∗251**	0	0	1	**∗251**	0	0	2	0	0
11	12	1	173	**∗170**	9	1	2	241	**∗170**	9	2	2	**∗170**	9	1	2	**∗170**	9	1	2	1.8	0
12		0.75	**∗242**	**∗242**	3	0	9	363	**∗242**	9	3	14	**∗242**	3	0	8	**∗242**	3	0	8	0.8	0
13		0.5	**∗251**	**∗251**	3	0	7	397	**∗251**	9	1	11	**∗251**	3	0	8	**∗251**	3	0	13	0	0
14		0.25	**∗279**	**∗279**	0	0	1	398	**∗279**	6	0	4	**∗279**	3	0	3	**∗279**	3	0	2	0	0
15		0	**∗279**	**∗279**	0	0	1	348	**∗279**	0	0	1	**∗279**	0	0	1	**∗279**	0	0	0	0	0
16	18	1	**∗207**	**∗207**	0	0	1	400	**∗207**	0	0	1	**∗207**	0	0	1	**∗207**	0	0	1	0	0
17		0.75	**∗256**	**∗256**	7	0	15	525	**∗256**	7	0	12	**∗256**	4	0	8	**∗256**	3	0	13	0	0
18		0.5	**∗274**	**∗274**	7	0	11	625	**∗274**	18	1	12	**∗274**	4	0	8	**∗274**	3	0	8	0	0
19		0.25	**∗292**	**∗292**	7	0	8	564	**∗292**	21	4	15	**∗292**	4	0	4	**∗292**	3	0	5	0	0
20		0	**303**	**303**	7	0	7	498	312	20	4	14	**303**	4	0	6	**303**	3	0	5	0.7	0
21	18	1	**∗217**	**∗217**	0	0	1	417	**∗217**	0	0	1	**∗217**	0	0	1	**∗217**	0	0	0	1.4	0
22		0.75	**∗285**	**∗285**	8	0	23	547	**∗285**	25	6	33	**∗285**	5	0	11	**∗285**	3	0	17	0	0
23		0.5	**∗313**	**∗313**	7	0	17	560	**∗313**	25	8	51	**∗313**	4	0	10	**∗313**	3	0	12	1.6	0
24		0.25	**∗334**	**∗334**	7	0	21	643	**∗334**	24	7	90	**∗334**	4	0	7	**∗334**	3	0	13	0	0
25		0	**∗339**	**∗339**	7	0	5	535	347	25	6	11	**∗339**	4	0	4	**∗339**	3	0	4	0	0
26	18	1	**∗227**	**∗227**	0	0	1	391	**∗227**	0	0	1	**∗227**	0	0	1	**∗227**	0	0	1	1.8	0
27		0.75	**∗278**	**∗278**	7	0	15	548	283	7	0	20	**∗278**	4	0	10	**∗278**	3	0	12	0	0
28		0.5	**∗336**	**∗336**	7	0	23	510	**∗336**	25	8	54	**∗336**	4	0	11	**∗336**	3	0	18	0	0
29		0.25	**∗361**	**∗361**	7	0	16	632	**∗361**	25	8	36	**∗361**	4	0	9	**∗361**	3	0	11	0	0
30		0	**∗375**	**∗375**	7	0	5	541	**∗375**	7	0	3	**∗375**	4	0	4	**∗375**	3	0	3	0	0
31	25	1	**∗245**	**∗245**	0	0	0	769	**∗245**	0	0	0	**∗245**	0	0	0	**∗245**	0	0	0	1.2	0
32		0.75	**∗294**	**∗294**	9	0	22	798	**∗294**	9	0	19	**∗294**	6	0	8	**∗294**	3	0	14	0	0
33		0.5	**∗313**	**∗313**	9	0	38	765	**∗313**	27	2	122	**∗313**	6	0	8	**∗313**	3	0	36	0	0
34		0.25	**∗327**	**∗327**	9	0	14	806	328	27	2	65	**∗327**	6	0	4	**∗327**	3	0	10	0	0
35		0	**∗328**	**∗328**	9	0	4	873	333	33	9	17	**∗328**	6	0	2	**∗328**	3	0	2	0	0
36	25	1	**∗252**	**∗252**	0	0	0	704	**∗252**	0	0	0	**∗252**	0	0	0	**∗252**	0	0	0	6	0
37		0.75	**∗311**	**∗311**	9	0	23	749	**∗311**	33	9	60	**∗311**	6	0	10	**∗311**	3	0	23	1.3	0
38		0.5	**∗345**	**∗345**	9	0	53	892	348	24	2	376	**∗345**	6	0	10	**∗345**	3	0	64	0	0
39		0.25	**∗357**	**∗357**	9	0	19	839	**∗357**	39	13	160	**∗357**	6	0	7	**∗357**	3	0	12	0	0
40		0	**∗362**	**∗362**	9	0	4	814	365	36	16	19	**∗362**	6	0	3	**∗362**	3	0	3	0	0
41	25	1	**∗254**	**∗254**	9	0	2	721	**∗254**	9	0	1	**∗254**	6	0	1	**∗254**	3	0	1	3.1	0
42		0.75	**∗319**	**∗319**	9	0	16	827	**∗319**	39	16	48	**∗319**	6	0	7	**∗319**	3	0	11	0.9	0
43		0.5	**371**	**371**	9	0	65	693	**371**	30	6	120	**371**	6	0	12	**371**	3	0	42	0.3	0
44		0.25	**∗378**	**∗378**	9	0	17	772	**∗378**	24	3	33	**∗378**	6	0	6	**∗378**	3	0	10	0.3	0
45		0	397	**∗396**	24	1	6	898	**∗396**	24	1	6	**∗396**	18	1	4	**∗396**	9	1	4	0.3	0
46	12	1	**∗156**	**∗156**	3	0	4	265	157	9	2	3	**∗156**	3	0	3	**∗156**	3	0	3	0	0
47		0.75	**194**	**194**	4	0	47	354	197	9	2	44	**194**	3	0	33	**194**	1	0	24	1	0
48		0.5	**215**	**215**	3	0	19	334	216	9	2	20	**215**	3	0	19	**215**	3	0	14	0	0
49		0.25	**∗222**	**∗222**	3	0	25	370	**∗222**	9	2	26	**∗222**	3	0	27	**∗222**	3	0	18	0	0
50		0	**∗242**	**∗242**	3	0	51	364	**∗242**	6	0	34	**∗242**	3	0	46	**∗242**	3	0	37	0	0
51	12	1	**∗159**	**∗159**	6	0	24	271	163	9	2	15	**∗159**	4	0	20	**∗159**	2	0	18	2.5	0
52		0.75	**∗209**	**∗209**	3	0	73	300	210	9	2	50	**∗209**	3	0	63	**∗209**	3	0	52	0	0
53		0.5	**∗230**	**∗230**	3	0	47	368	**∗230**	15	2	90	**∗230**	3	0	45	**∗230**	3	0	65	0	0
54		0.25	**∗238**	**∗238**	0	0	4	381	**∗238**	0	0	5	**∗238**	0	0	3	**∗238**	0	0	8	0	0
55		0	**∗251**	**∗251**	0	0	6	348	**∗251**	0	0	7	**∗251**	0	0	5	**∗251**	0	0	12	0	0
56	12	1	172	**∗170**	9	1	19	241	**∗170**	9	2	12	**∗170**	9	1	16	**∗170**	9	1	43	1.2	0
57		0.75	**∗203**	**∗203**	3	0	37	335	204	9	2	30	**∗203**	3	0	31	**∗203**	3	0	69	0	0
58		0.5	**∗251**	**∗251**	3	0	67	315	**∗251**	9	2	81	**∗251**	3	0	55	**∗251**	3	0	135	0	0
59		0.25	**∗278**	**∗278**	3	0	73	429	**∗278**	7	0	115	**∗278**	3	0	59	**∗278**	3	0	150	0	0
60		0	**∗279**	**∗279**	0	0	5	348	**∗279**	0	0	17	**∗279**	0	0	12	**∗279**	0	0	32	0	0
61	25	1	248	**∗245**	24	1	20	769	**∗245**	33	11	23	**∗245**	18	1	15	**∗245**	9	1	35	1.2	0
62		0.75	305	304	21	1	466	774	**302**	33	7	447	305	6	0	51	304	9	1	705	1	100
63		0.5	**∗311**	**∗311**	13	0	188	737	312	36	11	289	**∗311**	6	0	35	**∗311**	3	0	64	0.3	0
64		0.25	**∗322**	**∗322**	9	0	55	826	**∗322**	36	13	176	**∗322**	6	0	29	**∗322**	3	0	32	0	0
65		0	**∗328**	**∗328**	9	0	21	873	**∗328**	36	12	1381	**∗328**	6	0	15	**∗328**	3	0	14	0	0
66	25	1	**∗252**	**∗252**	9	0	14	704	**∗252**	33	10	23	**∗252**	6	0	9	**∗252**	3	0	9	6	0
67		0.75	309	**∗304**	32	1	144	789	307	36	10	203	**∗304**	18	1	58	**∗304**	9	1	66	5.3	0
68		0.5	**349**	**349**	13	0	218	757	352	33	10	401	**349**	6	0	96	**349**	3	0	114	0.9	72.2
69		0.25	**∗357**	**∗357**	13	0	100	819	358	36	15	213	**∗357**	6	0	32	**∗357**	3	0	43	0	0
70		0	**∗362**	**∗362**	9	0	34	814	365	36	16	68	**∗362**	6	0	24	**∗362**	3	0	26	0	0
71	25	1	257	**∗254**	24	1	11	721	**∗254**	33	8	15	**∗254**	18	1	7	**∗254**	9	1	7	3.1	0
72		0.75	341	**335**	21	1	318	795	340	54	14	962	341	6	0	45	**335**	9	2	208	1.2	100
73		0.5	**372**	**372**	9	0	424	694	377	39	12	814	**372**	6	0	60	**372**	3	0	272	0	118
74		0.25	387	**384**	24	1	252	879	**384**	42	14	564	**384**	18	1	49	**384**	9	1	238	0.8	80
75		0	397	**396**	24	1	50	898	**396**	45	17	85	**396**	18	1	33	**396**	9	1	32	0.3	0

Table 2 Results for the 225 TSPlib-based CRTP instances.

P	$\lvert U \rvert$	μ	(A) $c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	(B) $c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	(C) $c(\mathcal{N})$	Ω	#	t	(D) $c(\mathcal{N})$	Ω	#	t	Best γ	θ
76	37	1	∗**304**	∗**304**	16	0	9	1205	∗**304**	69	20	12	∗**304**	8	0	6	∗**304**	3	0	5	0	0
77		0.75	369	369	16	0	622	1135	**363**	64	17	331	369	8	0	58	369	3	0	286	3.3	51.8
78		0.5	**376**	**376**	16	0	434	1231	**376**	98	28	1588	**376**	8	0	56	**376**	3	0	365	0.5	100
79		0.25	380	∗**379**	40	1	211	1224	∗**379**	93	23	294	∗**379**	24	1	43	380	3	0	181	0.3	0
80		0	∗**380**	∗**380**	16	0	26	1208	∗**380**	45	2	28	∗**380**	8	0	17	∗**380**	3	0	46	0.3	0
81	37	1	309	309	16	0	13	1208	∗**308**	61	18	14	309	8	0	9	309	3	0	25	0.3	0
82		0.75	369	∗**363**	37	1	198	966	∗**363**	64	21	319	369	8	0	52	∗**363**	9	1	303	1.7	0
83		0.5	399	**398**	49	1	648	1341	410	64	21	1022	**398**	27	1	86	**398**	9	1	506	0.3	93.4
84		0.25	**404**	**404**	20	0	145	1328	422	77	25	355	**404**	9	0	34	**404**	3	0	130	0	100
85		0	**412**	**412**	16	0	38	1259	416	67	26	93	**412**	8	0	23	**412**	3	0	66	1.4	0
86	37	1	∗**314**	∗**314**	16	0	16	1107	∗**314**	66	23	31	∗**314**	8	0	10	∗**314**	3	0	32	0	0
87		0.75	406	**400**	45	3	1123	1152	434	66	24	1193	404	24	1	145	406	3	0	445	2	76.2
88		0.5	429	**423**	49	2	2168	1190	424	72	23	1673	**423**	27	2	161	**423**	9	1	947	1.9	73.5
89		0.25	447	441	37	1	1147	1264	**433**	72	26	907	447	8	0	58	447	3	0	380	0.7	84.6
90		0	448	∗**446**	40	1	73	1351	452	82	34	126	∗**446**	24	1	40	∗**446**	9	1	128	1.3	0
91	50	1	∗**376**	∗**376**	0	0	0	1442	∗**376**	0	0	0	∗**376**	0	0	0	∗**376**	0	0	1	0.3	0
92		0.75	427	427	27	0	398	1512	**426**	105	35	700	427	11	0	47	427	3	0	209	2.3	88.7
93		0.5	447	**444**	65	1	450	1681	448	162	57	2161	**444**	33	1	85	447	3	0	174	0.7	90
94		0.25	**454**	**454**	27	0	72	1693	**454**	194	59	640	**454**	11	0	21	**454**	3	0	45	0	0
95		0	**463**	**463**	27	0	33	1669	464	163	48	116	**463**	11	0	14	**463**	3	0	40	2.1	0
96	50	1	386	∗**384**	65	1	24	1388	∗**384**	105	27	36	∗**384**	33	1	10	∗**384**	9	1	28	0.5	0
97		0.75	**442**	**442**	27	0	626	1744	**442**	143	42	3319	**442**	11	0	81	**442**	3	0	348	3.6	54.6
98		0.5	475	**474**	60	1	2412	1647	484	178	49	3680	475	11	0	125	475	3	0	523	3.9	57.6
99		0.25	502	**498**	65	1	646	1674	500	182	66	4014	**498**	33	1	61	502	3	0	193	0.8	87.1
100		0	494	∗**493**	60	1	68	1681	∗**493**	192	51	383	494	11	0	18	494	3	0	50	4.1	0
101	50	1	392	391	65	1	41	1561	∗**390**	135	32	70	391	33	1	14	391	9	1	45	0.5	0
102		0.75	489	482	71	3	2492	1687	**480**	149	34	3886	485	33	2	139	489	3	0	400	4.4	75
103		0.5	515	**512**	66	2	1889	1726	**512**	170	57	2899	515	11	0	128	515	3	0	464	2.7	70.7
104		0.25	531	**525**	65	2	679	1871	**525**	208	70	2328	**525**	33	2	65	531	3	0	239	0	100
105		0	533	**526**	76	3	92	1762	529	195	62	513	**526**	44	3	35	533	3	0	69	2.8	100
106	18	1	**214**	**214**	8	0	9	569	217	25	6	13	**214**	5	0	6	**214**	3	0	17	0	0
107		0.75	∗**272**	∗**272**	8	0	460	549	273	21	7	309	∗**272**	5	0	278	∗**272**	3	0	451	0	0
108		0.5	**307**	**307**	8	0	595	639	315	21	6	584	**307**	5	0	193	**307**	3	0	427	3.5	63.9
109		0.25	**318**	**318**	8	0	529	553	**318**	39	7	1315	**318**	5	0	180	**318**	3	0	459	0	100
110		0	∗331	**330**	21	1	197	676	333	22	5	86	**330**	15	1	137	**330**	9	1	437	0.6	0
111	18	1	**235**	**235**	8	0	17	548	236	25	7	23	**235**	5	0	10	**235**	3	0	33	0	0
112		0.75	**306**	**306**	8	0	269	527	310	21	5	313	**306**	5	0	137	**306**	3	0	337	1.9	54.5
113		0.5	∗**336**	∗**336**	8	0	355	589	337	18	4	179	∗**336**	5	0	179	∗**336**	3	0	425	0	0
114		0.25	368	**367**	21	1	698	618	369	29	10	362	**367**	15	1	348	**367**	9	1	891	0.5	80.5
115		0	**390**	**390**	8	0	159	812	392	22	5	82	**390**	5	0	124	**390**	3	0	336	0	0
116	18	1	**259**	**259**	8	0	74	602	260	21	6	53	**259**	5	0	56	**259**	3	0	159	0	0
117		0.75	∗**325**	∗**325**	8	0	351	565	327	21	5	304	∗**325**	5	0	173	∗**325**	3	0	483	0	0
118		0.5	377	**376**	21	1	528	673	379	25	7	354	**376**	15	1	230	**376**	9	1	885	0.8	72.8
119		0.25	∗**397**	∗**397**	8	0	228	660	399	26	7	139	∗**397**	5	0	119	∗**397**	3	0	365	0	0
120		0	∗**448**	∗**448**	8	0	135	781	458	18	4	80	∗**448**	5	0	100	∗**448**	3	0	218	0.7	0
121	37	1	∗**320**	∗**320**	21	0	57	1071	325	69	22	53	∗**320**	9	0	25	∗**320**	3	0	72	0	0
122		0.75	387	**379**	51	2	1204	1171	396	98	25	2021	387	9	0	172	387	3	0	508	2.8	58.5
123		0.5	**393**	**393**	21	0	1227	1321	397	88	26	1637	**393**	9	0	183	**393**	3	0	471	2.2	69.8
124		0.25	403	**402**	47	1	665	1151	406	61	23	368	403	9	0	81	403	3	0	283	0.2	92.5
125		0	**410**	**410**	20	0	102	1226	411	61	25	124	**410**	9	0	48	**410**	3	0	132	0.7	0
126	37	1	336	∗**326**	51	3	121	1074	330	61	17	85	∗**326**	27	3	61	∗**326**	9	2	169	3	0
127		0.75	402	**401**	47	1	2799	1085	424	109	27	2254	402	9	0	238	402	3	0	504	0.2	95
128		0.5	435	**434**	51	1	1904	1325	455	74	31	1660	**434**	27	1	257	435	3	0	469	4.6	52.9
129		0.25	457	**449**	77	2	2810	1422	454	117	36	1432	454	27	2	254	457	3	0	501	2.4	74.1
130		0	**458**	**458**	21	0	195	1291	464	61	22	305	**458**	9	0	88	**458**	3	0	262	0	100
131	37	1	343	**342**	51	1	158	1085	**342**	61	17	125	**342**	27	1	60	**342**	9	1	169	0.3	0
132		0.75	446	**439**	55	2	3463	1078	462	100	30	3836	445	27	1	581	446	3	0	535	1.6	82
133		0.5	472	**463**	45	3	2777	1167	471	111	34	2659	472	9	0	205	468	9	1	1065	2.1	78.6
134		0.25	493	**478**	65	4	3168	1394	480	119	43	2517	484	36	2	534	493	3	0	790	3.8	64.9
135		0	500	**485**	56	4	448	1312	494	109	34	476	491	27	2	171	500	3	0	455	4.2	26.5
136	56	1	388	∗**383**	81	4	135	1862	∗**383**	126	33	119	387	36	1	41	387	9	1	123	3.1	0
137		0.75	**455**	**455**	32	0	2450	1903	476	180	49	5728	**455**	13	0	424	**455**	3	0	779	1.5	79.9
138		0.5	**460**	**460**	32	0	1883	1899	471	199	52	5006	**460**	13	0	352	**460**	3	0	704	3.6	56.2
139		0.25	**465**	**465**	32	0	477	2056	473	71	2	914	**465**	13	0	92	**465**	3	0	230	1.5	100
140		0	∗**476**	∗**476**	32	0	148	1761	477	196	54	705	∗**476**	13	0	71	∗**476**	3	0	158	3.8	0
141	56	1	∗**389**	∗**389**	32	0	121	1856	395	131	49	186	∗**389**	13	0	64	∗**389**	3	0	157	3.2	0
142		0.75	476	**472**	77	3	4513	1608	475	181	41	5097	**472**	39	3	503	474	9	1	1127	3.3	66.2
143		0.5	**507**	**507**	32	0	2024	1860	516	208	56	5848	**507**	13	0	248	**507**	3	0	637	2.5	76
144		0.25	**522**	**522**	32	0	1670	1908	530	171	68	3789	**522**	13	0	179	**522**	3	0	588	1.9	74.7
145		0	**528**	**528**	32	0	249	1793	531	198	54	672	**528**	13	0	90	**528**	3	0	233	2.8	67.1
146	56	1	414	∗**399**	84	4	227	1809	400	130	43	197	411	39	1	113	413	9	1	220	3.6	0
147		0.75	527	**522**	72	3	3021	1835	561	190	50	6871	524	39	1	668	527	3	0	660	2.1	82.7
148		0.5	542	**538**	77	2	4688	1669	546	171	50	5497	541	39	1	791	542	3	0	1038	2.9	73.9
149		0.25	558	553	77	1	1752	1820	**551**	193	56	3713	556	39	1	363	558	3	0	660	1.3	84.7
150		0	**557**	**557**	32	0	168	1803	559	200	58	738	**557**	13	0	52	**557**	3	0	119	0.7	100

Table 3 Results for the 225 TSPlib-based CRTP instances.

P			(A)					(B)					(C)				(D)				Best	
	$\|U\|$	μ	$c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	$c(\tilde{\mathcal{N}})$	$c(\mathcal{N})$	Ω	#	t	$c(\mathcal{N})$	Ω	#	t	$c(\mathcal{N})$	Ω	#	t	γ	θ
151	75	1	478	**∗473**	0	0	4	2616	**∗473**	0	0	3	**∗473**	0	0	4	**∗473**	0	0	9	1.1	0
152		0.75	546	**542**	101	3	3051	2430	554	209	73	6273	543	45	2	622	546	3	0	770	1.6	74.4
153		0.5	558	**551**	102	3	2631	2465	552	297	89	3334	558	15	0	405	558	3	0	984	2.3	52.7
154		0.25	573	573	43	0	556	2519	**561**	295	109	2489	573	16	0	92	573	3	0	282	2.1	71.2
155		0	576	**∗572**	108	2	224	2713	612	314	108	1584	**∗572**	45	1	75	576	3	0	178	2.1	0
156	75	1	494	486	123	5	213	2552	**∗482**	189	59	171	487	60	4	55	494	3	0	108	2.5	0
157		0.75	572	**556**	94	5	6275	2586	589	350	99	11264	563	45	2	929	572	3	0	1051	3	59.7
158		0.5	588	**584**	99	2	5506	2523	587	294	98	6717	588	15	0	515	588	3	0	1395	4.6	53.4
159		0.25	618	**613**	111	4	2407	2717	618	328	116	5689	615	48	3	192	617	9	1	691	0.8	84.4
160		0	620	**612**	102	4	314	2731	637	293	117	1293	620	15	0	76	620	3	0	195	2.2	39.7
161	75	1	495	**∗488**	98	2	211	2425	491	240	68	321	495	15	0	45	495	3	0	125	1.4	0
162		0.75	598	**575**	88	6	3071	2544	593	379	93	13122	598	15	0	520	598	3	0	1325	7.7	32.1
163		0.5	623	**617**	108	4	6060	2770	701	366	97	12065	622	45	1	758	623	3	0	1351	1	84.2
164		0.25	652	**641**	105	5	4282	2524	666	342	109	7123	646	48	2	277	652	3	0	529	2.3	68.9
165		0	**662**	**662**	42	0	393	2672	**662**	337	103	2435	**662**	15	0	75	**662**	3	0	251	1.8	63.9
166	25	1	**280**	**280**	13	0	91	836	282	33	10	61	**280**	6	0	62	**280**	3	0	154	0.7	0
167		0.75	317	**315**	32	1	1371	804	316	33	10	669	**315**	18	1	273	317	3	0	601	3.7	0
168		0.5	352	352	13	0	740	770	**349**	33	11	457	352	6	0	251	352	3	0	584	1.1	75.5
169		0.25	**363**	**363**	14	0	518	831	367	36	10	376	**363**	7	0	223	**363**	3	0	489	0	100
170		0	**∗366**	**∗366**	13	0	225	984	368	30	10	152	**∗366**	6	0	148	**∗366**	3	0	357	0	0
171	25	1	**293**	**293**	14	0	47	809	295	42	13	47	**293**	7	0	16	**293**	3	0	50	0	0
172		0.75	**367**	**367**	13	0	1408	904	373	48	10	1443	**367**	6	0	416	**367**	3	0	838	0	100
173		0.5	**400**	**400**	13	0	1279	786	404	33	9	1023	**400**	6	0	343	**400**	3	0	745	1.2	86.7
174		0.25	**416**	**416**	13	0	891	974	417	36	15	716	**416**	6	0	228	**416**	3	0	553	0	100
175		0	**424**	**424**	34	1	370	1049	426	30	9	164	**424**	20	1	204	**424**	9	1	522	0.2	93.8
176	25	1	**298**	**298**	35	1	77	918	302	42	15	45	**298**	21	1	20	**298**	9	1	51	0.3	0
177		0.75	385	**383**	35	1	2527	840	385	36	11	1212	**383**	21	1	850	385	3	0	634	2.5	68.6
178		0.5	**403**	**403**	13	0	1300	940	405	33	12	723	**403**	6	0	340	**403**	3	0	607	0	100
179		0.25	429	429	14	0	797	908	**427**	60	17	915	429	7	0	286	429	3	0	698	0.5	90.9
180		0	**452**	**452**	14	0	311	1007	468	36	14	214	**452**	7	0	180	**452**	3	0	453	0	100
181	50	1	**∗411**	**∗411**	28	0	56	1780	412	99	31	86	**∗411**	11	0	23	**∗411**	3	0	55	0	0
182		0.75	**481**	**481**	28	0	2398	1798	486	162	40	6113	**481**	11	0	442	**481**	3	0	688	2.2	68.5
183		0.5	**499**	**499**	28	0	1944	1736	501	164	50	4728	**499**	11	0	365	**499**	3	0	912	0	100
184		0.25	**501**	**501**	65	1	1234	1957	505	166	52	1271	**501**	33	1	303	**501**	9	2	703	0.4	89.8
185		0	**506**	**506**	27	0	287	2020	**506**	171	55	891	**506**	11	0	174	**506**	3	0	396	3.4	86.5
186	50	1	**420**	**420**	27	0	101	1925	**420**	151	37	283	**420**	11	0	51	**420**	3	0	113	0	0
187		0.75	477	**475**	78	2	2717	1703	489	107	34	2624	477	33	1	801	477	9	2	1169	1	74.7
188		0.5	**505**	**505**	29	0	936	1839	511	194	54	4523	**505**	11	0	310	**505**	3	0	681	2.3	64.1
189		0.25	**531**	**531**	28	0	1199	1865	532	115	43	1551	**531**	11	0	243	**531**	3	0	572	0	100
190		0	**534**	**534**	28	0	269	2016	535	192	59	679	**534**	11	0	120	**534**	3	0	269	0.6	75.4
191	50	1	436	**428**	116	5	307	1724	**428**	97	37	249	**428**	44	4	87	436	3	0	127	3.5	0
192		0.75	505	**501**	77	3	6116	1704	502	120	40	2617	504	33	1	667	505	3	0	875	0.8	83.5
193		0.5	527	**516**	96	6	1498	1588	517	197	48	1680	520	55	6	552	527	3	0	912	2.1	65.9
194		0.25	559	**551**	73	3	1951	1985	555	138	56	2395	559	33	1	434	559	9	1	1179	2.3	68.6
195		0	574	**568**	67	2	686	2051	575	179	60	899	**568**	33	2	336	574	3	0	727	1	71.3
196	75	1	516	**513**	93	2	269	2542	514	192	53	203	516	16	0	55	516	4	0	154	0.6	0
197		0.75	581	**580**	51	4	5632	2856	593	264	74	7507	**580**	51	4	989	581	16	3	2412	2.4	64.2
198		0.5	591	**587**	86	2	4213	2716	597	268	80	5653	590	48	1	967	591	4	0	1249	0.8	77.5
199		0.25	612	**602**	111	4	2278	2883	620	335	110	4384	604	64	3	724	612	3	0	970	1.6	59
200		0	**608**	**608**	44	0	255	2794	610	374	121	1159	**608**	16	0	122	**608**	4	0	324	2.3	0
201	75	1	**517**	**517**	44	0	182	2866	518	298	70	619	**517**	16	0	76	**517**	4	0	198	0.4	0
202		0.75	589	**584**	54	3	5952	2720	600	259	73	9043	587	48	1	784	589	3	0	1192	1.8	69.2
203		0.5	607	**603**	59	4	5665	2811	604	303	94	9184	605	48	2	1005	607	4	0	1844	0.7	83.9
204		0.25	**619**	**619**	44	0	1074	2651	621	354	109	4127	**619**	16	0	368	**619**	4	0	863	0	100
205		0	638	**637**	97	1	634	2774	638	318	108	1338	638	16	0	152	638	4	0	394	0.8	86.7
206	75	1	526	**525**	111	3	337	2673	526	265	68	410	527	48	2	85	526	9	2	206	0.8	0
207		0.75	642	630	48	3	6514	2569	**629**	341	77	11699	630	48	5	1044	642	3	0	1404	3.8	65.2
208		0.5	639	**634**	59	2	4999	2812	648	335	104	9711	637	48	1	784	639	3	0	1270	1.7	76.6
209		0.25	666	**663**	87	2	3145	2567	665	312	96	6494	**663**	64	3	708	666	3	0	729	1	85.4
210		0	674	673	102	1	627	2852	**669**	339	114	1871	673	48	1	236	674	3	0	435	3	49.8
211	100	1	**∗555**	**∗555**	56	0	218	3608	1161	334	67	331	**∗555**	21	0	115	**∗555**	4	0	297	0	0
212		0.75	642	**635**	71	5	10093	3318	880	408	119	11755	640	63	1	1922	642	4	0	2141	2.6	58.2
213		0.5	647	**641**	115	5	4712	3279	967	400	126	7458	642	63	3	1094	647	12	1	2671	2.9	51.6
214		0.25	656	**652**	133	2	2195	3445	1104	390	131	5099	**652**	63	2	567	656	4	0	673	0.6	193
215		0	674	**666**	133	2	567	3458	1163	429	147	1502	**666**	63	2	227	674	12	1	574	2.5	0
216	100	1	568	**∗564**	133	1	358	3195	971	228	67	271	**∗564**	63	1	118	568	4	0	236	0.7	0
217		0.75	661	**658**	70	2	9428	3288	1210	389	94	10952	**658**	63	2	1980	661	4	0	2419	0.8	87.1
218		0.5	679	**672**	89	3	7644	3578	1124	442	140	10838	677	63	1	2191	679	4	0	3119	2.6	60.8
219		0.25	690	**687**	145	3	2639	3466	883	420	137	6451	690	63	1	532	690	12	1	1256	0.6	121.7
220		0	700	**692**	166	4	698	3525	1181	508	153	2140	695	84	2	244	700	4	0	515	1.1	50.9
221	100	1	575	**574**	145	2	330	3297	1102	322	71	408	575	63	1	111	575	12	1	374	0.3	0
222		0.75	684	**670**	92	6	11271	3282	1008	585	124	19657	676	105	4	3129	684	4	0	2873	3.6	62.1
223		0.5	717	**708**	73	4	10367	3565	1032	488	141	13613	714	63	2	2807	717	4	0	4102	1.3	79.1
224		0.25	729	**717**	123	3	3674	3640	1152	452	134	6215	720	63	2	609	729	4	0	876	1.8	68.3
225		0	729	**720**	133	2	692	3530	1155	486	159	1785	725	63	1	266	729	4	0	495	3.1	33.1

in (B) helps finding better local optima than with the other strategies for 10 instances, the algorithm convergence is significantly slower. Moreover, it especially seems to suffer from the costly start solution for the larger instances. On average we observe 28 refinements and 84 GLBPs per instance for (B), compared to 1 and 38 for (A).

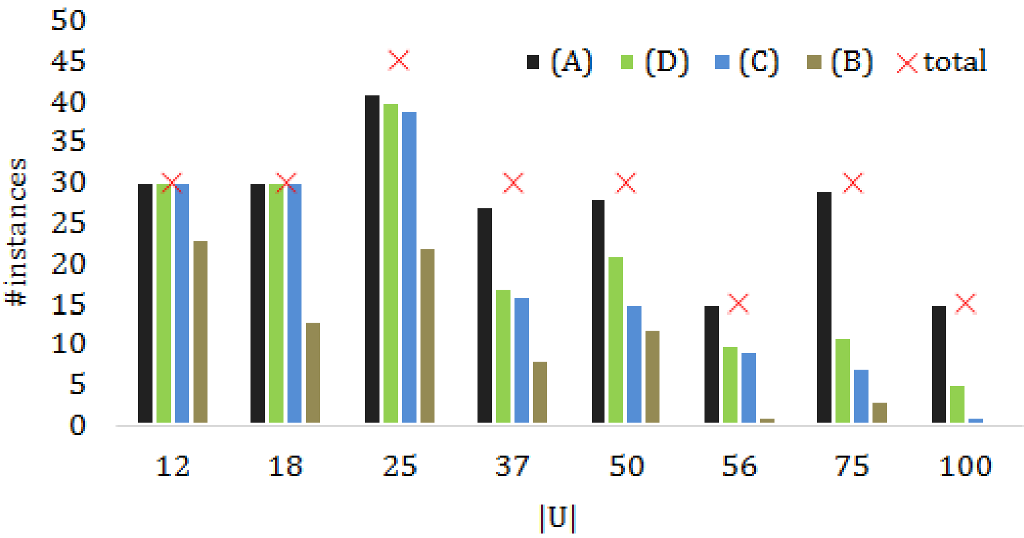


Figure 11 Total number of instances by number of customers and number of best solutions found by heuristics (A), (B), (C) and (D).

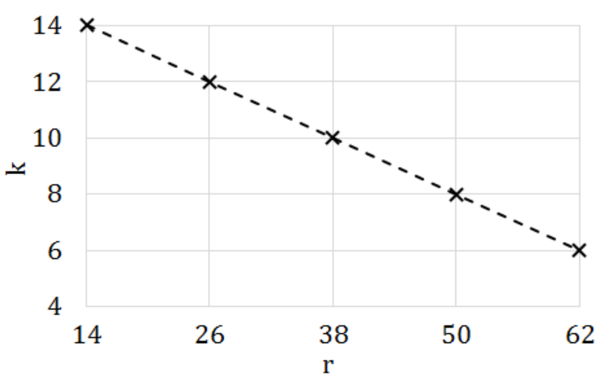
Let us sum up the performance of (A) compared to existing approaches. We improve 42% of the solutions found by the initial heuristic and 65% of the heuristic results in (Hill 2015). On average we achieve an improvement of 1.6% for the problems without known optimal solutions. We improve 36% of the best known results that were obtained by the branch and cut algorithm in Hill and Voß (2015), reducing the optimality gap by 10% on average for the corresponding instances. However, we are not able to compete with the pure exact method for 15% of the problems.

5.3. Algorithm analysis

To better understand the algorithm performance we provide further details about the computations with configuration (A) in this section. In the following we break down the computational details into the different numbers of instance customers for each GLBP parameterization over all the 225 instances. [$|U| \in \{12, 18, 25, 37, 50, 56, 75, 100\}$; #instances: $30, 30, 45, 30, 30, 15, 30, 15$ (see also Figure 11).] We recall that these pairs of k and $r = |B|$

in the GLB scheme are computed in advance based on an initial calibration. Depending on $|U|$ we utilize the first pairs $(k_1, r_1), ..., (k_h, r_h)$ such that $r_{h-1} < |U|$ and $r_h \geq |U|$. The used scheme is shown in Figure 12 (left).

The average number of edge variable flips of a refinement is illustrated in Figure 12 (right). Even though these average values range within $[0, k]$ as expected, the used local search procedures might lead to best (GLBP-infeasible) solutions beyond the allowed Hamming distance to the incumbent.



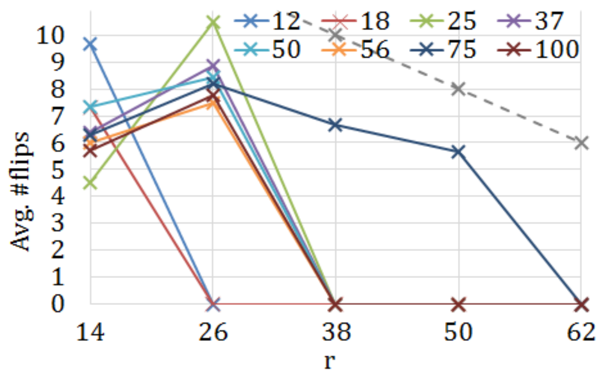
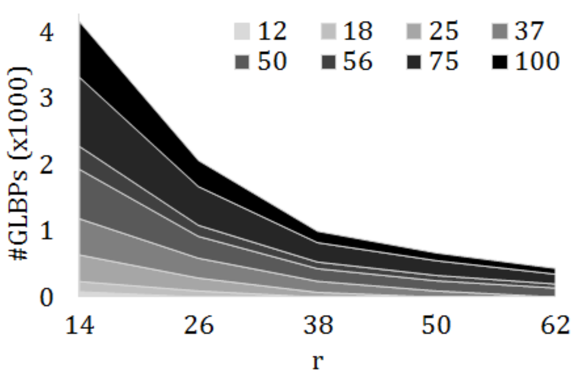


Figure 12 **GLB scheme (left) and average number of edge variable flips per improvement (right).**

Figure 13 shows the total number of GLBPs (left) and the number of improvements found (right) for each parameterization and number of customers. The majority of improvements, except for instances with $|U| = 75$, are carried out when considering balls of size 14 and 26. In total, solving 227 (2.7%) of 8514 GLBPs result in an improved solution.



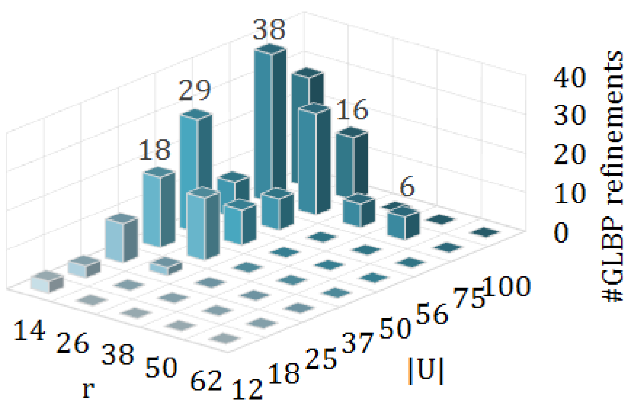


Figure 13 **Number of GLBPs (left) and number of improvements (right).**

The average GLBP improvement of the network costs is given in Figure 14 (left). These numbers support that $k \in \{14, 26\}$ does not only yield the most improvements, but also the highest cost reduction per improvement on average. Figure 14 (right) shows the average run times of the exact algorithm for the GLBPs using the time limit of 120 seconds. Even though some parameterizations and instance classes tended to yield more time consuming GLBPs for the solver, our GLB scheme seems to be well balanced within the used framework. Finally, 1.5% of the overall runtime is spent for GLBPs that lead to an improvement.

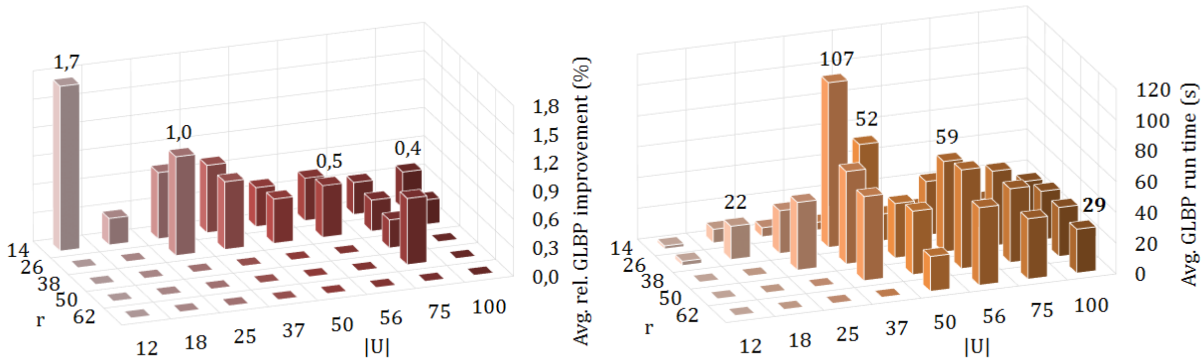


Figure 14 **Average relative objective improvement (left) and average runtime (right).**

Figure 15 (left) shows the average relative optimality gap at the root node. These results seem to be related with the average runtimes in Figure 14 (right). Therefore, the root node gap might be a good indicator for the hardness of a GLBP. The average number of explored nodes within the branch and cut method is given in Figure 15 (right).

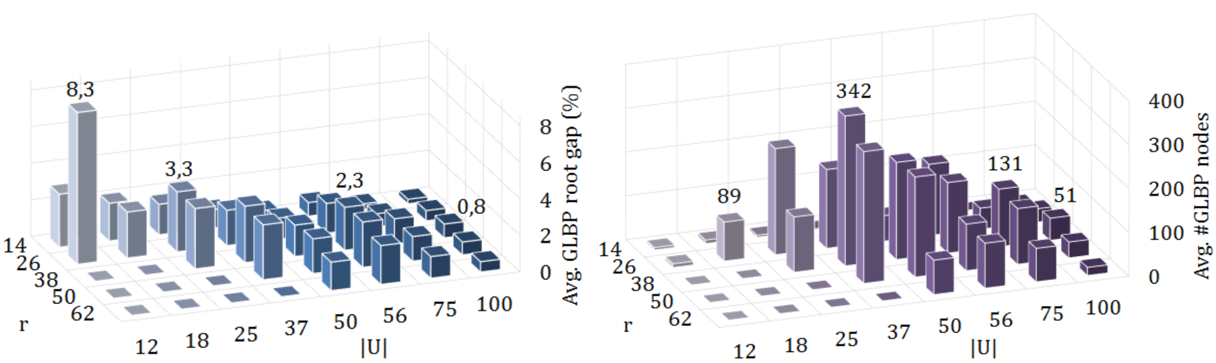


Figure 15 **Average GLBP root node gap (left) and average number of explored GLBP branching nodes (right).**

93.6% of the GLBPs could be solved within the time limit. The average optimality gap when the GLBP could not be solved and the time limit was reached is illustrated in Figure 16 (left). Figure 16 (right) shows the number of GLBPs that could not be solved. GLBPs with $r = k = 14$ were mostly solved to optimality since this was a finding of our initial GLB calibration, as Figure 14 (right) already suggests. For instances with $|U| \in \{37, 56, 75\}$ up to 45% of the GLBPs could not be solved. We recall that this does not necessarily mean that no improving solution was found by the exact method.

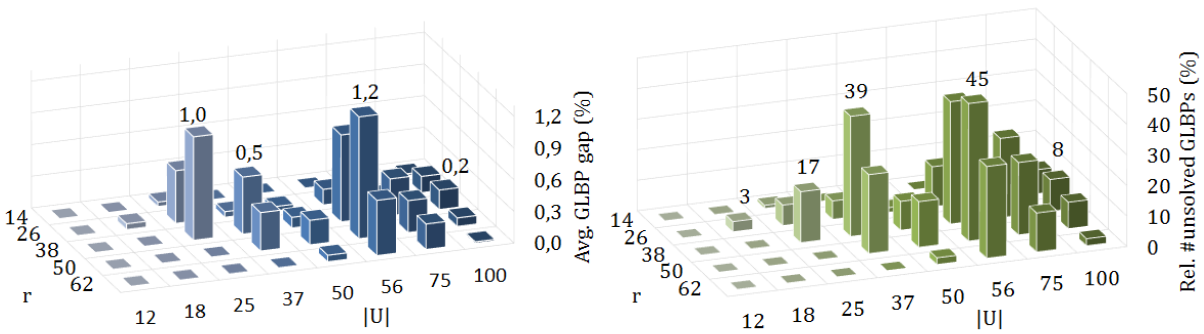


Figure 16 **Average relative optimality gap when time limit reached (left) and number of unsolved GLBPs (right).**

Finally, we would like to mention that we also experimented with a strategy (A′) which utilizes the reversed (A) GLB scheme. In other words, we considered balls of large size first and incrementally made them smaller while increasing k. We observed that for 8 instances better solutions than with (A) were found and for 2 instances (A′) could not compete with (A). Furthermore, the average runtime could be reduced by 0.5% and 5.0% in total.

6. Conclusions

We presented a novel heuristic framework to solve a broad class of network design problems. A key ingredient is GLB, a concept that generalizes the ideas of local branching and local refinement techniques based on mathematical programming. GLB refinement problems are created by the addition of GLB cuts to an ILP formulation of the overall problem. These are iteratively solved while increasing the number of involved decision variables and at the same time decreasing the number of variable flips. Hereby, we control the complexity of these subproblems in order to solve them to optimality.

Using this idea we designed a heuristic for the CRTP based on an exact branch and cut algorithm. This approach turned out to be powerful since we were able to obtain new best solutions for literature instances. Compared to the pure MIP refinements or local branching it produced significantly more best solutions. Furthermore, we could improve solutions obtained by the pure exact algorithm in most cases.

The proposed approach represents a promising strategy when no improving solution can be found by other algorithms at hand (in our case a multi-start local search heuristic and an exact algorithm). As typical for exact refinement methods, the ability to incrementally enlarge the refinement neighborhoods results in an adjustable algorithm performance.

Using the presented solutions as a starting point for an exact algorithm could significantly accelerate the solution process and improve the obtained bounds. Moreover, the techniques could be integrated into an exact method to effectively polish feasible solutions that are found along the search. The study of extended problem-specific solution refinement substructures, beside the used single-balls, could further improve the results. Furthermore, we suggest to transfer our techniques to related optimization models in network design to study their effectiveness.

Acknowledgements

We thank the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) for generous travel support and are indebted to Bernard Fortz and Malte Fliedner for their useful comments.

References

Archetti, C., L. Bertazzi, A. Hertz, M. G. Speranza. 2012. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing* **24** 101–116. doi:10.1287/ijoc.1100.0439.

Berthold, T. 2014. RENS. *Mathematical Programming Computation* **6** 33–54. doi:10.1007/s12532-013-0060-9. URL `http://dx.doi.org/10.1007/s12532-013-0060-9`.

Cafieri, S., P. Hansen, L. Liberti. 2014. Improving heuristics for network modularity maximization using an exact algorithm. *Discrete Applied Mathematics* **163** 65 – 72. doi:http://dx.doi.org/10.1016/j.dam.2012.03.030. URL `http://www.sciencedirect.com/science/article/pii/S0166218X12001345`.

Danna, E., E. Rothberg, C. L. Pape. 2005. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming A* **102** 71–90. doi:10.1007/s10107-004-0518-7. URL `http://dx.doi.org/10.1007/s10107-004-0518-7`.

Fischetti, M., F. Glover, A. Lodi. 2005. The feasibility pump. *Mathematical Programming B* **104** 91–104. doi:10.1007/s10107-004-0570-3. URL `http://dx.doi.org/10.1007/s10107-004-0570-3`.

Fischetti, M., A. Lodi. 2003. Local branching. *Mathematical Programming B* **98** 23–47. doi:10.1007/s10107-003-0395-5. URL `http://dx.doi.org/10.1007/s10107-003-0395-5`.

Franceschi, R. De, M. Fischetti, P. Toth. 2006. A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming B* **105** 471–499.

Harvey, W. D., M. L. Ginsberg. 1995. Limited discrepancy search. C. S. Mellish, ed., *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 1995*, vol. 1. Morgan Kaufmann, 607–615.

Hill, A. 2015. Multi-exchange neighborhoods for the capacitated ring tree problem. *Springer Lecture Notes in Computer Science, Volume 8962* 85–94doi:10.1007/978-3-319-15585-2_10. URL http://dx.doi.org/10.1007/978-3-319-15585-2_10.

Hill, A., S. Voß. 2014. An equi-model matheuristic for the multi-depot ring star problem. *Research paper, Department of Engineering Management, University of Antwerp* .

Hill, A., S. Voß. 2015. Optimal capacitated ring trees. *EURO Journal on Computational Optimization* **online available, DOI: 10.1007/s13675-015-0035-x**. doi:10.1007/s13675-015-0035-x. URL http://dx.doi.org/10.1007/s13675-015-0035-x.

Hoshino, E. A., A. Hill. 2014. A column generation approach for the capacitated ring tree problem. *Proceedings of XVIII ALIO/EURO Workshop on Applied Combinatorial Optimization* .

Lalla-Ruiz, E., S. Voß. 2014. POPMUSIC as a matheuristic for the berth allocation problem. *Annals of Mathematics and Artificial Intelligence* 1–17doi:10.1007/s10472-014-9444-4. URL http://dx.doi.org/10.1007/s10472-014-9444-4.

Legato, P., R. Trunfio. 2014. A local branching-based algorithm for the quay crane scheduling problem under unidirectional schedules. *4OR* **12** 123–156. doi:10.1007/s10288-013-0235-2. URL http://dx.doi.org/10.1007/s10288-013-0235-2.

Maniezzo, V., T. Stützle, S. Voß, eds. 2010. *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*, *Annals of Information Systems*, vol. 10. Springer.

Naji-Azimi, Z., M. Salari, P. Toth. 2012. An integer linear programming based heuristic for the capacitated m-ring-star problem. *European Journal of Operational Research* **217** 17 – 25. doi:http://dx.doi.org/10.1016/j.ejor.2011.08.026. URL http://www.sciencedirect.com/science/article/pii/S0377221711007673.

Rodríguez-Martín, I., J. J. Salazar. 2010. A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research* **37** 575 – 581. doi:http://dx.doi.org/10.1016/j.cor.2008.09.003. URL http://www.sciencedirect.com/science/article/pii/S0305054808001585. Hybrid Metaheuristics.

Smet, P., T. Wauters, M. Mihaylov, G. V. Berghe. 2014. The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega* **46** 64 – 73. doi:http://dx.doi.org/10.1016/j.omega.2014.02.003. URL http://www.sciencedirect.com/science/article/pii/S0305048314000176.

Sniedovich, M., S. Voß. 2006. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics* **35** 551–578.

Taillard, E., S. Voß. 2002. POPMUSIC - partial optimization metaheuristic under special intensification conditions. C.C. Ribeiro, P. Hansen, eds., *Essays and Surveys in Metaheuristics*. Kluwer, Boston, 613–629.

Voß, S. 2006. Steiner tree problems in telecommunications. M. G. C. Resende, P. M. Pardalos, eds., *Handbook of Optimization in Telecommunications*. Springer US, 459–492. doi:10.1007/978-0-387-30165-5_18. URL `http://dx.doi.org/10.1007/978-0-387-30165-5_18`.