Approximate Robust Optimization for the Connected Facility Location Problem

M. Gisela Bardossy
Merrick School of Business
University of Baltimore
1420 N. Charles Street
Baltimore, MD 21201
mbardossy@ubalt.edu

S. Raghavan
Smith School of Business and
Institute for Systems Research
University of Maryland
College Park, MD 20742
raghavan@umd.edu

Abstract

In this paper we consider the Robust Connected Facility Location (ConFL) problem within the robust discrete optimization framework introduced by Bertsimas and Sim [7]. We propose an Approximate Robust Optimization (ARO) method that uses a heuristic and a lower bounding mechanism to rapidly find high-quality solutions. The use of a heuristic and a lower bounding mechanism—as opposed to solving the robust optimization (RO) problem exactly—within this ARO approach significantly decreases its computational time. This enables one to obtain high quality solutions to large-scale robust optimization problems and thus broadens the scope and applicability of robust optimization (from a computational perspective) to other NP-hard problems. Our computational results attest to the efficacy of the approach.

Keywords: network design, robust optimization, connected facility location problem, heuristics, approximation

1 Introduction

The Connected Facility Location (ConFL) problem models a variety of problems that are significant in the telecommunications and data management literature (see [14, 16]), as well as the emergency management literature (see [26]). By design the ConFL problem is deterministic; however, the practical settings that motivate it are characterized by significant uncertainty. In this paper we make the first attempt to address this uncertainty in the ConFL problem using robust optimization (RO) (within the RO framework introduced by [7] for discrete optimization problems) and provide a heuristic approach that yields high-quality solutions.

The ConFL problem encompasses a large family of network design problems [3], where at minimum cost a set of facilities must be opened, customers must be assigned to open facilities, and lastly, open facilities must be connected through a Steiner tree. In this setting, the main source of uncertainty lies in the assignment costs as the location and demand of customers are often unknown. Furthermore, there might be limited information about the distribution of those costs making it difficult to follow a traditional 2-stage stochastic programming approach to address uncertainty in the problem. However, if one can estimate a range for the costs (i.e., a lower bound and an upper bound), one can search for a *robust solution*.

One approach in robust optimization is to protect against the worst-case scenario. In other words the decision maker wants to find a solution to the problem that minimizes the overall cost for the *worst case* scenario. However, since this may be viewed as an ultraconservative approach (which can lead to an expensive solution), Bertsimas and Sim [7] (BS) propose an alternative RO approach that allows the decision maker to adjust the level of conservatism for the solution. Roughly speaking, in the BS approach instead of optimizing for the worst-case scenario, the goal is to find a solution that protects the decision-maker against all cases in which up to Γ parameters, instead of all of the uncertain parameters, take their worst value (the remaining parameters take their best case values). When Γ is equal to the number of uncertain parameters in the solution it corresponds to the worst-case scenario solution.

The BS approach thus deals with discrete optimization problems with interval uncertainty in the objective function coefficients. It requires a large but polynomial number of *nominal problems* to be solved. The nominal problems are (suitably modified) deterministic variations of the original problem. Consequently, when the nominal problem is polynomially solvable the robust problem is also polynomially solvable (which is a particularly nice feature of the approach). On the flip side, if the original problem is NP-hard, the nominal problems are also NP-hard and computationally expensive to solve to optimality. Thus, there are some significant computational challenges in applying the BS approach to NP-hard robust optimization problems. Alternatively, in this paper, we demonstrate how to use a heuristic and lower bound mechanism to solve the nominal problems approximately and yet obtain high-quality solutions for the robust problem. This is a significant computational advantage, as it makes it practical to apply the BS robust optimization paradigm to a large class of NP-hard optimization problems.

Bertsimas and Sim [7] discuss the application of approximation algorithms in the context of robust optimization. They show that an α-approximation algorithm for the nominal problem results in an α-approximation algorithm for the RO problem. Our work expands the scope of their result in a *computational practical manner*. Approximation algorithm ratios are worst-case results and they do not (generally) provide tailored lower bounds for specific problem instances. Further, many NP-hard problems are max-SNP hard (or inapproximable unless P=NP) and thus constant approximation ratios are not known for them. In a practical setting then the approximation algorithm may be somewhat dissatisfying when trying to obtain near-optimal solutions to the problem at hand. In the realm where one has access to a (good) heuristic and a (good) lower bounding mechanism for individual problem instances (which is often the case for hard combinatorial optimization problems), we show how to expand upon Bertsimas and Sim [7] to develop an approximate robust optimization procedure that provides both a heuristic solution and an overall lower bound for the RO problem. We illustrate this approach on the robust ConFL problem.

The rest of this paper is organized as follows. In §2 we introduce the robust ConFL problem and related literature. In §3 we review Bertsimas and Sim [7]'s robust optimization approach for discrete optimization problems with interval uncertainty in the objective function coefficients, and then present our Approximate Robust Optimization (ARO) method. In §4 we apply the ARO method to the robust ConFL problem. We also illustrate the ARO method on a small robust ConFL problem. We also consider a special case of the robust ConFL problem with disk uncertainty areas. In §5 we illustrate the effectiveness of the ARO method with extensive computational experiments on the robust ConFL problem. §6 provides concluding remarks.

2 Problem Definition and Related Literature

We start by defining the ConFL problem, and later, expanding its definition to describe the robust ConFL problem considered in this paper.

2.1 Connected Facility Location Problem

The ConFL problem can be stated as follows. We are given a graph $G = (V, E)$, and three disjoint sets: $D \subseteq V$, set of demand nodes (or customers); $F \subseteq V$, set of potential facility nodes; and $S \subseteq V$, set of potential Steiner nodes, with $D \cup F \cup S = V$. We seek to find a minimum cost network where every demand node is assigned to an open facility and open facilities are connected through a Steiner tree T constructed on the subgraph of G on the nodes $F \cup S$ (i.e., $G(F \cup S) = (F \cup S, E(F \cup S))$). There are facility opening costs, $f_i \geq 0$, incurred for each facility; assignment costs, $a_{ij} \geq 0$, for assigning a customer $j \in D$ to a facility $i \in F$; and edge costs, $b_{ij} \geq 0$, for an edge $\{i, j\} \in E(F \cup S)$ if it is used on the Steiner tree T. The nodes in

S may be viewed as pure Steiner nodes and can only be used in the tree T as Steiner nodes, while the nodes in F may be used as Steiner nodes on the tree T incurring a facility opening cost even when no customers are assigned to them.[1] The final network cost is given by $\sum_{i \in Z} f_i + \sum_{\{i,j\} \in E(T)} b_{ij} + \sum_{j \in D} a_{i(j)j}$, where $i(j)$ is the facility serving demand node j, Z is the set of open facilities, and T is a Steiner tree connecting the open facilities. We first describe a formulation for the ConFL problem.

$$\text{Minimize} \sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} a_{ij} x_{ij} \tag{1a}$$

subject to

$$\sum_{\{i,j\} \in E(R)} y_{ij} \leq \sum_{l \in R \setminus k} z_l, \ \forall R \subset (S \cup F), |R| \geq 3, \forall k \in R \tag{1b}$$

$$y_{ij} \leq z_i, y_{ij} \leq z_j, \ \forall \{i,j\} \in E(S \cup F) \tag{1c}$$

$$\sum_{\{i,j\} \in E(S \cup F)} y_{ij} = \sum_{l \in (S \cup F)} z_l - 1 \tag{1d}$$

$$\sum_{i \in F} x_{ij} = 1, \ \forall j \in D \tag{1e}$$

$$x_{ij} \leq z_i, \ \forall i \in F, \forall j \in D \tag{1f}$$

$$x_{ij} \in \{0,1\}, \ \forall i \in F, \forall j \in D \tag{1g}$$

$$y_{ij} \in \{0,1\}, \ \forall \{i,j\} \in E(S \cup F) \tag{1h}$$

$$z_l \in \{0,1\}, \ \forall l \in S \cup F \tag{1i}$$

The formulation uses three sets of binary variables. The x_{ij} variables represent whether (or not) demand node j is connected to facility location i. The y_{ij} variables represent whether (or not) edge $\{i,j\}$ is in the Steiner tree connecting open facilities. The z_l variables represent whether (or not) node l is in the Steiner tree connecting open facilities. The objective function adds up the facility opening cost, the core network

[1] Our definition for the ConFL problem follows Bardossy and Raghavan [3]. They show this general definition of the ConFL captures other variants where the sets D, F, S overlap; or where facilities incur a cost only when customers are served from that facility. Consequently, this general definition includes all known variants of the ConFL problem as well as the the Steiner tree star problem [17], and the rent or buy problem [12].

cost (i.e., the Steiner tree defined by the y_{ij} variables), and the cost of serving each demand node by an open facility. The first set of constraints (1b), (1c) and (1d) ensure that open facilities are connected by a Steiner tree (constraints (1b) are usually referred to as generalized subtour elimination constraints (GSECs) in the literature). Constraints (1e) ensure that each demand node is served by one facility and constraints (1f) ensure that demand nodes are only served by open facilities.

Related Literature on Connected Facility Location: The first description of the ConFL problem in the literature is due to Karger and Minkoff [14], where in attempting to solve a network design problem with incomplete information they discuss the ConFL problem as a solution strategy. Gupta et al. [11] coined the terminology ConFL while considering a virtual private network design with demand uncertainty. They then gave a 10.66 approximation algorithm for the ConFL problem by adapting a rounding technique. Swamy and Kumar [23] described a primal-dual approximation algorithm for the ConFL problem. Their algorithm works in two phases and has an approximation ratio of 8.55. Jung et al. [13] improved this primal-dual algorithm and lowered the approximation ratio to 6.55. Eisenbrand et al. [9] presented a randomized algorithm that improves the approximation ratio for the ConFL problem to 4 that slightly degrades to 4.23 when the algorithm is derandomized.

With a focus on computationally solving the problem Ljubić [20] introduced a variable neighborhood search heuristic that is combined with reactive tabu search. She also proposed a branch-and-cut approach for solving the ConFL problem to optimality. Tomazic and Ljubić [24] proposed a greedy randomized adaptive search procedure for the ConFL problem that produced solutions that were on average as large as 10% from the optimal in their test instances. Bardossy and Raghavan [3] proposed a dual-based local search (DLS) heuristic that yields high-quality solutions rapidly. The approach first applies a dual-ascent heuristic that provides both a high-quality lower bound and a starting solution for the local search heuristic. They report, over the 2,748 problems they tested, the DLS heuristic found solutions that were on average at most 1.07% from optimality. Gollowitzer and Ljubic [10] propose several mathematical formulations for the ConFL problem based on direct graphs and compare their linear-programming relaxations. Leitner et al. [19] present a new formulation based on a mixed graph, investigate the associated polytope and share their computational experience with a branch-and-cut approach based on this new formulation. In a subsequent paper [18] they adapt this formulation to an asymmetric variant of the ConFL problem. While there has been a significant amount of research focused on the ConFL problem, none of these works consider uncertainty in the assignment costs (which actually is the case in the motivating examples of [11, 14]). This motivates our study of the robust ConFL problem (with interval uncertainty on the assignment costs).

2.2 Robust ConFL

The robust ConFL problem considers the problem where each assignment cost $\tilde{a}_{ij}$, $\{i, j\} \in \delta(D)$ ($\delta(D)$ denotes the edges between D and F), takes values in the interval $[a_{ij}, a_{ij} + d_{ij}]$, $d_{ij} \geq 0$. Motivated by the practical applications of the ConFL, our model considers two sources of uncertainty: customer location and demand uncertainty. In the first setting customer location is unknown on a plane but limited to a defined area. Then, the assignment costs range between the best-case scenario: the closest Euclidean distance between the facility node and a customer location within its uncertainty region, a_{ij} (if the facility node falls within the uncertainty region, $a_{ij} = 0$), and the worst-case scenario: the farthest possible distance between them, $a_{ij} + d_{ij}$. In other words the uncertain assignment cost, $\tilde{a}_{ij} \in [a_{ij}, a_{ij} + d_{ij}]$. On the other hand, when demand quantities are uncertain we assume that they also range within a predetermined range. Let $\tilde{q}_j \in [q_j, q_j + \Delta_j]$ be the uncertain demand quantity for customer j, and ρ_{ij} be the per unit demand assignment cost for customer j to facility i. Then, the uncertain assignment cost is given by $\tilde{a}_{ij} \in [\rho_{ij}q_j, \rho_{ij}q_j + \rho_{ij}\Delta_j] = [a_{ij}, a_{ij} + d_{ij}]$, where $a_{ij} = \rho_{ij}q_j$ and $d_{ij} = \rho_{ij}\Delta_j$. In both cases uncertainty translates into interval uncertainty in the objective function coefficients, as unknown assignment costs vary between a minimum and a maximum value without reference to a probability distribution.

As discussed earlier, we follow the Bertsimas and Sim [7] robust optimization approach for discrete optimization problems with interval uncertainty in the objective function coefficients. Thus, given a budget Γ, we wish to find a solution that minimizes the maximum value of the solution when up to Γ assignment costs are at their worst-case value ($a_{ij} + d_{ij}$) and the remaining values are at their best case value (a_{ij}).

Related Literature on Robust Optimization: Robust optimization has received significant attention in recent years. The first introduction of robust optimization is due to Soyster [22], who considered "columnwise" uncertainty in a linear problem, and consequently, proposed a linear optimization model that is feasible for all data. This approach is ultraconservative as it only allows decision variables that permit feasible solution under all possible scenarios. This ultraconservative strategy has motivated the search for alternative concepts to model the uncertainty. Since one does not expect all data to take their worst case values researchers have proposed the concept of uncertainty sets that constrain the possible perturbed data values of the problem in some fashion.

The pioneering work by Ben-Tal and Nemirovski [5, 6] proposes somewhat less conservative models by considering ellipsoidal uncertainty sets. Roughly speaking this approach bounds the perturbations in the data values to lie in an ellipsoid. Controlling the size of these ellipsoidal sets has the interpretation of a budget of uncertainty that the decision-maker selects in order to easily trade off robustness and performance. One nice feature of this approach is the robust counterpart takes the form of conic quadratic problems. Bertsimas and Sim [8] use polyhedral uncertainty sets (this bounds the uncertain perturbations to lie in a polyhedral set and includes interval uncertainty) that encode a budget of uncertainty in terms of cardinality constraints, as opposed to the size of an ellipsoid. The uncertainty sets they consider control the number of parameters of the problem that are allowed to vary from their nominal values, providing a different trade-off between the optimality of the solution, and its robustness to parameter perturbation. Bertsimas and Sim [7] apply this concept of robustness (with interval uncertainty in data values) in the context of discrete optimization. This is the approach that we take within this paper.

Snyder [21] provides a comprehensive review of facility location problems under uncertainty. Much of the previous facility location work focuses on the concept of minmax regret that was introduced by Kouvelis and Yu [15]. Here a solution is evaluated based on the difference in cost against the optimal solution for a given realization of data (this is referred to as regret). Instead of finding a solution that minimizes the worst case cost, the goal is to find a solution that minimizes the worst case regret. More recently, Baron et al. [4] addressed the facility location problem in a network facing uncertain demand over multiple periods. The decision maker has to select the set of facilities to open, their capacity and allocate demand to open facilities on each period. In this setting, they compare the solution obtained by two approaches with respect to demand uncertainty: boxed and ellipsoidal. They show that the alternate models of uncertainty lead to very different solution network topologies, with the model with box uncertainty set opening fewer, larger facilities. They also show that both the box and ellipsoidal uncertainty cases can provide small but significant improvements over the solution to the problem when demand is deterministic and set at its nominal value. The robust spanning tree problem with interval data was introduced by Kouvelis and Yu [15]. As the ConFL problem, its motivation also comes from the telecommunications industry when connection costs are uncertain due to congestion rates. Yaman et al. [25] show the worst-case version of this problem (i.e., where all edges can take their worst-case values) is polynomially solvable, while Aron and Van Hentenryck [2] shows that minmax regret variant of the problem is NP-complete.

3 Approximate Robust Optimization (ARO) Method

Before we introduce the Approximate Robust Optimization (ARO) Method, we briefly review the BS approach. In general terms, given the following combinatorial optimization problem $\min_{\mathbf{v}\in W} \tilde{\mathbf{c}}^T\mathbf{v}$ where $W \subseteq \{0,1\}^p$ represents the set of feasible solutions, [7] define the robust counterpart, where each entry $\tilde{c}_j, j \in P = \{1, 2, ..., p\}$ takes values in $[c_j, c_j + d_j]$, $d_j \geq 0$, $j \in P$ as follows

$$\mathcal{Z}^* = \underset{\mathbf{v}\in W}{\text{minimize}} \quad \mathbf{c}^T\mathbf{v} + \max_{\{U|U\subseteq P, |U|\leq\Gamma\}} \sum_{j\in U} d_j v_j \tag{2}$$

The interpretation of this formulation is that at most Γ of the uncertain coefficients will take their highest value. Consequently, the decision maker minimizes the maximum cost of a solution with at most Γ coefficients at their highest value. The deviation term (or penalty term), $\max_{\{U|U\subseteq P,|U|\leq\Gamma\}} \sum_{j\in U} d_j v_j$, represents the sum of the maximum deviation of a specified number, Γ, of uncertain coefficients in the solution. Notice, the value $\Gamma = 0$ corresponds to the best-case scenario problem (i.e., it ignores the deviation term and assumes the minimum cost coefficient for each decision variable), and the value of $\Gamma = p$ yields the worst-case scenario solution (i.e., each cost coefficient is included in the deviation term).

Furthermore, Bertsimas and Sim [7] propose an algorithm to find a solution to (2). They show that one can find the optimal solution to problem (2) by solving at most $p + 1$ (deterministic) nominal problems. The nominal problems are deterministic instances of the original problem with some of the coefficients at their lowest value and others modified by an added term, plus a constant. To apply the method one must first identify the values of the deviation coefficients, d_j, and label them in decreasing order such that $d_1 \geq d_2 \geq ... \geq d_p \geq d_{p+1} = 0$. Then for each deviation coefficient, d_l, one defines a nominal problem,

G_l, given by

$$G_l = \Gamma d_l + \underset{\mathbf{v}\in W}{\text{minimize}} \quad \mathbf{c}^T\mathbf{v} + \sum_{j=1}^{l}(d_j - d_l)v_j \tag{3}$$

Then, the optimal function value to problem (2) is given by $\mathcal{Z}^* = \min_{l=1,\ldots,p+1} G_l$ and the optimal solution, $\mathbf{v}^* = \text{argmin}_{l=1,\ldots,p+1} G_l$. The number of nominal problems is at most $p+1$ because for equal d_l values one must only solve one nominal problem.[2]

The BS approach requires that several nominal problems are solved to optimality in order to solve the actual robust problem. However, when the nominal problem is an NP-hard combinatorial optimization problem it can be computationally expensive to solve it to optimality. Bertsimas and Sim [7] show that an α-approximation algorithm for the nominal problem results in an α-approximation algorithm for the robust optimization problem. In the situation where one has access to a (good) heuristic and a (good) lower bounding mechanism for individual problem instances, instead of an approximation algorithm, we now show how to expand the scope of the BS approach to develop an approximate robust optimization procedure that provides both a *heuristic solution* and an *overall lower bound* for the robust optimization problem.

Figure 1 describes the Approximate Robust Optimization (ARO) algorithm. Within the algorithm, we can use any (good) heuristic with a (good) lower bounding mechanism to approximately solve the nominal problems, $\Omega_l = G_l - \Gamma d_l = \min_{\mathbf{v}\in W} \mathbf{c}^T\mathbf{v} + \sum_{j=1}^{l}(d_j - d_l)v_j$. Let $\mathbf{v}_l^H$ denote the heuristic solution to the nominal problem Ω_l, Ω_l^H its objective value, and Ω_l^{LB} its lower bound. Given a heuristic solution $\mathbf{v}_l^H$ to the nominal problem (Ω_l) we can use it as a heuristic solution (in fact any feasible solution to the combinatorial optimization problem is a potential heuristic solution) to the robust optimization problem. Let $\mathcal{Z}_l^H$ denote the objective function value of heuristic solution $\mathbf{v}_l^H$, evaluated for the robust optimization problem (2). In other words

$$\mathcal{Z}_l^H = \mathbf{c}^T\mathbf{v}_l^H + \max_{\{U|U\subseteq P,|U|\leq\Gamma\}} \sum_{j\in U} d_j(\mathbf{v}_l^H)_j.$$

Here $(\mathbf{v}_l^H)_j$ denotes the jth component of vector $\mathbf{v}_l^H$. The heuristic solution $\mathbf{v}_l^H$ is easily evaluated by adding the Γ largest deviation terms in the given solution to $\mathbf{c}^T\mathbf{v}_l^H$. Let $l^* = \arg\min_{l=1,\ldots,p+1} \mathcal{Z}_l^H$, and let $t^* = \arg\min_{l=1,\ldots,p+1} \Gamma d_l + \Omega_l^{LB}$. Then the heuristic solution to nominal problem Ω_{l^*} is selected as the

[2] Álvarez-Miranda et al. [1] show that for a given value of Γ the number of nominal problems required can be decreased to $p - \Gamma + 2$. Specifically, $\mathcal{Z}^* = \min_{l=\Gamma,\ldots,p+1} G_l$ and the optimal solution, $\mathbf{v}^* = \text{argmin}_{l=\Gamma,\ldots,p+1} G_l$.

Input: Problem instance and Γ.
Output: Solution, $\mathbf{v}^H$, solution value, $\mathcal{Z}^H$, and a solution quality assessment, β
foreach d_l **do**
  Find a heuristic solution $\mathbf{v}_l^H$ and lower bound Ω_l^{LB} for the nominal problem:
  $\Omega_l = G_l - \Gamma d_l = \min_{\mathbf{v}\in W} \mathbf{c}^T\mathbf{v} + \sum_{j=1}^{l}(d_j - d_l)v_j$.
  Let $\mathcal{Z}_l^H = \mathbf{c}^T\mathbf{v}_l^H + \max_{\{U|U\subseteq P,|U|\leq\Gamma\}}\sum_{j\in U} d_j(\mathbf{v}_l^H)_j$.
end
Let $l^* = \arg\min_{l=1,\ldots,p+1} \mathcal{Z}_l^H$.
Let $t^* = \arg\min_{l=1,\ldots,p+1} \Gamma d_l + \Omega_l^{LB}$.
$\mathbf{v}^H = \mathbf{v}_{l^*}^H$; $\mathcal{Z}^H = \mathcal{Z}_{l^*}^H$; $\mathcal{Z}^{LB} = \Gamma d_{t^*} + \Omega_{t^*}^{LB}$
$\beta = \frac{\mathcal{Z}^H - \mathcal{Z}^{LB}}{\mathcal{Z}^{LB}}$.

Figure 1: ARO Method

heuristic solution to the robust optimization problem. That is, $\mathbf{v}^H = \mathbf{v}_{l^*}^H$. Its objective value $\mathcal{Z}^H = \mathcal{Z}_{l^*}^H$, and $\mathcal{Z}^{LB} = \Gamma d_{t^*} + \Omega_{t^*}^{LB}$ provides a lower bound to the robust objective. Thus $\mathbf{v}^H$ is the heuristic solution obtained by the ARO method to the robust optimization problem (2). We now show that $\mathcal{Z}^{LB}$ is a lower bound for the robust optimization problem. Consequently $\mathcal{Z}^H$ is at most $\beta = \frac{\mathcal{Z}^H - \mathcal{Z}^{LB}}{\mathcal{Z}^{LB}}\%$ greater than the optimal solution. In other words the solution $\mathbf{v}^H$ has an optimality gap of $\beta = \frac{\mathcal{Z}^H - \mathcal{Z}^{LB}}{\mathcal{Z}^{LB}}$.

Let α_l be the optimality gap of heuristic solution $\mathbf{v}_l^H$ to the nominal problem Ω_l (i.e., $\alpha_l = \frac{\Omega_l^H - \Omega_l^{LB}}{\Omega_l^{LB}}$) and let $\alpha = \max_{l=1,\ldots,p+1} \alpha_l$.

Theorem 3.1. $\mathbf{v}^H$ *is a* $(1+\alpha)$*-approximate solution to the robust optimization problem.*

Proof. Since $\alpha \geq \alpha_l$ for $l = 1,\ldots,p+1$, each heuristic solution $\mathbf{v}_{l^*}^H$ to the nominal problem Ω_l is a $(1+\alpha)$-approximate solution. It follows then from Theorem 4 in [7] that $\mathbf{v}^H$ is a $(1+\alpha)$-approximate solution to the robust optimization problem. □

In other words $\mathbf{v}^H$ has an optimality gap of at most α. As noted earlier the above result follows directly from Theorem 4 in [7]. We can strengthen this result, and show that the ARO Method has an even tighter optimality gap (it should be evident from the earlier discussion that the approximation ratio is equal to 1 plus the optimality gap). In particular we show that $\mathcal{Z}^{LB}$ is a valid lower bound, and that the optimality gap β is less than or equal to the optimality gap of the nominal problem that provides the lower bound for the robust problem (i.e., t^*). In other words $\beta \leq \alpha_{t^*} \leq \alpha$.

Theorem 3.2. $\beta \leq \alpha_{t^*}$.

In order to prove Theorem 3.2 we need the following two lemmas.

Lemma 3.3. *$\mathcal{Z}^{LB}$ is a lower bound to $\mathcal{Z}^*$.*

Proof. Let $\mathcal{Z}^* = G_{\bar{l}}$, where $\bar{l}$ is the nominal problem that solves (2) to optimality. Then $\Gamma d_{\bar{l}} + \Omega_{\bar{l}}^{LB} \leq G_{\bar{l}} = \mathcal{Z}^*$. On the other hand, $\mathcal{Z}^{LB} = \min_{l=1,\ldots,p+1} \Gamma d_l + \Omega_l^{LB} \leq \Gamma d_{\bar{l}} + \Omega_{\bar{l}}^{LB}$. □

Lemma 3.4. *$\mathcal{Z}^H \leq \Gamma d_l + \Omega_l^H$ for all $l \in P$.*

Proof.

$$\begin{aligned} \mathcal{Z}^H &\leq \mathcal{Z}_l^H \text{ for all } l \in P \\ &= \mathbf{c}^T \mathbf{v}_l^H + \max_{\{U | U \subseteq P, |U| \leq \Gamma\}} \sum_{j \in U} d_j (\mathbf{v}_l^H)_j. \end{aligned} \tag{4}$$

From equation (19) in [7], the inner maximization can be replaced as follows.

$$\max_{\{U | U \subseteq P, |U| \leq \Gamma\}} \sum_{j \in U} d_j (\mathbf{v}_l^H)_j = \min_{\theta \geq 0} \{ \sum_{j \in P} \max(d_j - \theta, 0) (\mathbf{v}_l^H)_j + \Gamma \theta \}$$

This implies that if we set $\theta = d_l$ for any l

$$\min_{\theta \geq 0} \{ \sum_{j \in P} \max(d_j - \theta, 0) (\mathbf{v}_l^H)_j + \Gamma \theta \} \leq \Gamma d_l + \sum_{j=1}^{l} (d_j - d_l) (\mathbf{v}_l^H)_j$$

Continuing from the last equality in (4):

$$\begin{aligned} \mathbf{c}^T \mathbf{v}_l^H + \max_{\{U | U \subseteq P, |U| \leq \Gamma\}} \sum_{j \in U} d_j (\mathbf{v}_l^H)_j &\leq \mathbf{c}^T \mathbf{v}_l^H + \Gamma d_l + \sum_{j=1}^{l} (d_j - d_l)(\mathbf{v}_l^H)_j \\ &= \Gamma d_l + \Omega_l^H \end{aligned}$$

which yields

$$\mathcal{Z}^H \leq \Gamma d_l + \Omega_l^H \text{ for all } l \in P. \square$$

Proof of Theorem 3.2. Recall,

$$\beta = \frac{\mathcal{Z}^H - \mathcal{Z}^{LB}}{\mathcal{Z}^{LB}}.$$

Noting that $\mathcal{Z}^{LB} = \Omega_{t^*}^{LB} + \Gamma d_{t^*}$, and using Lemma 3.4 that says $\mathcal{Z}^H \leq \Gamma d_{t^*} + \Omega_{t^*}^H$ we obtain:

$$\begin{aligned}
\beta &\leq \frac{\Omega_{t^*}^H + \Gamma d_{t^*} - (\Omega_{t^*}^{LB} + \Gamma d_{t^*})}{\Omega_{t^*}^{LB} + \Gamma d_{t^*}} \\
&= \frac{\Omega_{t^*}^H - \Omega_{t^*}^{LB}}{\Omega_{t^*}^{LB} + \Gamma d_{t^*}} \\
&= \frac{\alpha_{t^*}\Omega_{t^*}^{LB}}{\Omega_{t^*}^{LB} + \Gamma d_{t^*}} \\
&= \alpha_{t^*}\left[\frac{\Omega_{t^*}^{LB}}{\Omega_{t^*}^{LB} + \Gamma d_{t^*}}\right] \\
\text{since } \Gamma d_{t^*} \geq 0,\ 0 \leq &\ \frac{\Omega_{t^*}^{LB}}{\Gamma d_{t^*} + \Omega_{t^*}^{LB}} \leq 1, \text{ thus} \\
\beta &\leq \alpha_{t^*}. \square
\end{aligned}$$

4 Approximate Robust Optimization Method Applied to the Robust Connected Facility Location Problem

We now apply the Approximate Robust Optimization method to the robust connected facility location problem. Recall, the robust counterpart of the ConFL problem considers the problem where each assignment cost $\tilde{a}_{ij}$, $\{i, j\} \in \delta(D)$, takes values in the interval $[a_{ij}, a_{ij} + d_{ij}]$, $d_{ij} \geq 0$, and the set of feasible solutions $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ satisfy constraints (1b)-(1i). For convenience we will denote by $\mathcal{X}$ the set of feasible solutions to (1b)-(1i).

Applying the BS framework, we obtain a robust counterpart where we would like to find a solution $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathcal{X}$ that minimizes the maximum cost $\sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} \tilde{a}_{ij} x_{ij}$ such that at most Γ of the coefficients $\tilde{a}_{ij}$ are at their maximum value (and the remaining ones are at their minimum values). Let $\mathcal{W}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} a_{ij} x_{ij}$ (the best case cost of the solution $(\mathbf{x}, \mathbf{y}, \mathbf{z})$). Then, the robust ConFL is given by the following formulation:

$$\begin{aligned} \mathcal{Z} = \text{Minimize} \quad & F_\Gamma(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathcal{W}(\mathbf{x}, \mathbf{y}, \mathbf{z}) + \max_{\{U | U \subseteq \delta(D), |U| \leq \Gamma\}} \sum_{\{i,j\} \in U} d_{ij} x_{ij} \\ \text{subject to:} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathcal{X}. \end{aligned} \tag{5}$$

The second term in the objective function accounts for the uncertainty of the Γ highest possible deviations in the assignment costs for each feasible solution (and thus incorporates the uncertainty in the location (and/or demand) of each customer node). Notice that in any solution to a ConFL instance there is exactly one deviation term for each customer node. Consequently, although the maximum number of uncertain coefficients (i.e. assignment costs) is $|F||D|$; the maximum number of possible assignment costs at their worst case value in a solution is $|D|$. In other words, even though the number of nominal problems may be as large as $|F||D|$, we only need to consider Γ values in the range 0 to $|D|$. In the robust ConFL problem, the nominal problems are deterministic ConFL problems with their assignment costs adjusted by the deviation terms, d_l. (The deviation terms, d_l, are the various assignment cost deviations, d_{ij}, organized in decreasing order and relabeled accordingly.) Then, the nominal problems, G_l, for the robust ConFL problem are given by

$$G_l = \Gamma d_l + \underset{(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{X}}{\text{minimize}} \sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} (a_{ij} + \max\{d_{ij} - d_l, 0\}) x_{ij} \tag{6}$$

and Ω_l simply by

$$\Omega_l = \underset{(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{X}}{\text{minimize}} \sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} (a_{ij} + \max\{d_{ij} - d_l, 0\}) x_{ij} \tag{7}$$

To solve the problems, Ω_l, we use Bardossy and Raghavan [3]'s dual-based local search (DLS) heuristic. The ARO method uses the DLS heuristic to find a (good) heuristic solution $(\mathbf{x_l}, \mathbf{y_l}, \mathbf{z_l})$ for each deterministic ConFL problem, Ω_l, defined by (7). Then, the heuristic solution value $\mathcal{Z}_l^H$ is calculated by evaluating the solution $(\mathbf{x_l}, \mathbf{y_l}, \mathbf{z_l})$ in problem (5). The smallest of these solutions is selected as the heuristic solution to the robust ConFL problem. The algorithm uses the lower bounds Ω_l^{LB} obtained by DLS for each of the nominal problems to calculate a lower bound $\mathcal{Z}^{LB}$ (recall $\min_{l=1,\ldots,p+1} \Gamma d_l + \Omega_l^{LB}$) for the problem.

One advantage of solving all $p+1$ nominal problems (as opposed to $p - \Gamma + 2$ as indicated in [1]) is

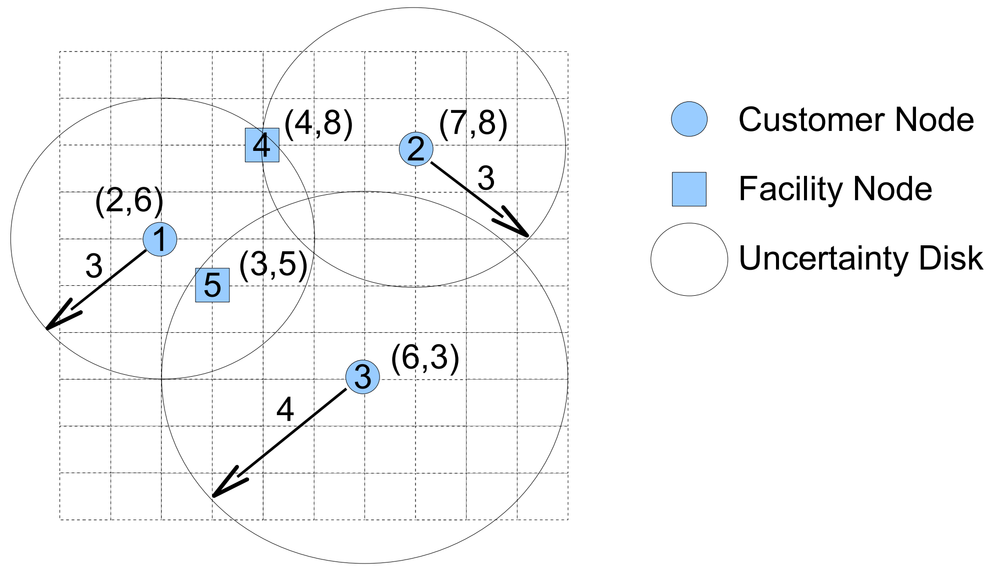

Figure 2: Example of robust ConFL problem

that once we have a heuristic solution $(\mathbf{x_l}, \mathbf{y_l}, \mathbf{z_l})$ and a lower bound Ω_l for each nominal problem; one can quickly find heuristic solutions for different Γ values by recalculating the heuristic solution value $\mathcal{Z}_l^H$ in problem (5) and the lower bounds with the new Γ. This is particularly useful in the context of sensitivity analysis (where a decision maker may rapidly want solutions for a wide range of Γ values).

4.1 Illustration of the ARO Method on the Robust ConFL Problem

In this section we illustrate the ARO method with a small robust ConFL instance. Figure 2 depicts an instance with two facility nodes and three customer nodes with disk uncertain areas. Distances are the Euclidean distances between the coordinates of the nodes. For this example we set the facility opening cost equal to 1 and the cost of the edge between the two potential facilities as twice the Euclidean distance between them.

Table 1 provides information on the interval in which the costs of the assignment edges lie (and also shows the cost of the "tree edge" between the two potential facilities). Here, $a_{ij} = \max\{||\mathbf{i} - \mathbf{j}||_2 - r_j, 0\}$, and $d_{ij} = \min\{||\mathbf{i} - \mathbf{j}||_2 + r_j, 2r_j\}$, where $\mathbf{i}$ and $\mathbf{j}$ are coordinates of node $i \in F$ and $j \in D$, respectively; and r_j is the radius of the uncertainty disk for node $j \in D$. In this example, there are five distinct deviation

Assignment Costs			
Customer	Facility	Minimum Cost	Deviation
1	4	0.00	5.83
1	5	0.00	4.41
2	4	0.00	6.00
2	5	2.00	6.00
3	4	1.39	8.00
3	5	0.00	7.61
Tree Edge Cost			
Facility	Facility	Cost	
4	5	6.32	

Table 1: Assignment and Tree Edge Costs

Assignment Costs							
		Deviation d_l					
Customer	Facility	8.00	7.61	6.00	5.83	4.41	0.00
1	4	0.00	0.00	0.00	0.00	1.42	5.83
1	5	0.00	0.00	0.00	0.00	0.00	4.41
2	4	0.00	0.00	0.00	0.17	1.59	6.00
2	5	2.00	2.00	2.00	2.17	3.59	8.00
3	4	1.39	1.78	3.39	3.56	4.98	9.39
3	5	0.00	0.00	1.61	1.78	3.20	7.61

Table 2: Assignment costs for each nominal problem

values, d_l; consequently, we need to solve six nominal problems: one for each deviation value plus one additional problem for $d_l = 0$, $\mathcal{D} = \{8.00, 7.61, 6.00, 5.83, 4.41, 0\}$.

Table 2 depicts the assignment costs for each nominal problem l, Ω_l, as defined in (7). Each column in the table depicts assignment edge costs for different ConFL (nominal) problem that must be solved. We use the dual-ascent local search (DLS) heuristic to obtain both a heuristic solution $(\mathbf{x_l}, \mathbf{y_l}, \mathbf{z_l})$ and a lower bound Ω_l^{LB} for each nominal problem.

Table 3 shows results of applying the ARO method for $\Gamma = 2$. The second column provides the lower bound obtained for each nominal problem (7), the fourth column provides the lower bound for each nominal problem Ω_l^{LB}, the fifth column computes $\Gamma d_l + \Omega_l^{LB}$, and the sixth column provides the objective value of the solution $(\mathbf{x_l}, \mathbf{y_l}, \mathbf{z_l})$ evaluated in the objective function (5). Recall the overall lower bound can be computed by the lowest of the values in the fifth column, and the overall upper bound (and heuristic solution) from

l	d_l	Solution	Ω_l^{LB}	$\Gamma d_l + \Omega_l^{LB}$	$\mathcal{Z}_H^l$
1	8.00	Open Facility 4	2.39	18.39	16.39
2	7.61	Open Facility 4	2.78	18.00	16.39
3	6.00	Open Facility 4	4.39	16.39	16.39
4	5.83	Open Facility 4	4.73	16.39	16.39
5	4.41	Open Facility 5	7.79	16.61	16.61
6	0.00	Open Facility 5	21.02	21.02	16.61

Table 3: Results of ARO method on Figure 2 example with $\Gamma = 2$.

	Γ			
Solution	0	1	2	3
Open Facility 4	**2.39**	**10.39**	**16.39**	22.21
Open Facility 5	3.00	10.61	16.61	**21.02**

Table 4: Results for $0 \leq \Gamma \leq 3$

the lowest of the values in the sixth column. Thus, when $\Gamma = 2$ the ARO method obtains a lower and upper bound of 16.39 and opens facility 4. Since the upper and lower bounds are equal, in this case the ARO method finds the optimal solution.

Although we set Γ equal to 2, with the heuristic solution and lower bound to each nominal problem at hand, it is easy to vary Γ between 0 and 3 and apply the ARO method to observe how the solution changes. As mentioned earlier this is particularly useful in the context of sensitivity analysis. The ARO method obtained the optimal solution (as the upper and lower bounds were equal) for all values of Γ. Notice that the six nominal problems, Ω_l, produced only two distinct solutions; that is, to either open facility 4 or facility 5. Thus the ARO method will produce one of these two solutions as the heuristic solution to the robust ConFL for a given value of Γ. Table 4 shows the objective value for these two solutions in the robust optimization formulation. To open facility 4 is optimal for every Γ value except 3. In other words, to open facility 4 is a robust solution under a wide range of realizations except when the worst-case scenario takes place.

4.2 Special Case: Disk Uncertainty with Facilities Outside

When all of the potential facilities fall outside the customers' disk uncertainty areas, there is only one d_{ij} value ($= 2r_j$) for each customer $j \in D$ regardless of the facility $i \in F$. We now show that the optimal solution to the robust ConFL problem is identical for all values of Γ. Thus we can solve the robust ConFL problem (for any value of Γ) by simply solving the best case scenario problem.

Theorem 4.1. *If every $i \in F$ falls outside the uncertainty disk for every demand node $j \in D$, the optimal solution to the robust ConFL problem is identical for all values of* Γ.

Proof. By assumption $i \in F$ lies outside the uncertainty disk with radius r_j for all $j \in D$, and assignment cost $\tilde{a}_{ij} \in [a_{ij}, a_{ij} + d_{ij}]$. Since any $i \in F$ lies outside j's uncertainty disk, $d_{ij} = 2r_j, \forall i \in F$. Hence, we can rearrange the order of summation in equation (1a) and noting $\sum_{i \in F} x_{ij} = 1$ (from constraint (1e)) we get: $\sum_{j \in \mathcal{D}} \sum_{i \in F} d_{ij} x_{ij} = \sum_{j \in \mathcal{D}} 2r_j \sum_{i \in F} x_{ij} = \sum_{j \in \mathcal{D}} 2r_j$. Consequently, the deviation term $\max_{\{\mathcal{D} | \mathcal{D} \subset D, |\mathcal{D}| \leq \Gamma\}} \sum_{j \in \mathcal{D}} 2r_j$ is a constant dependent on Γ but unaffected by the problem solution. Thus, the optimal solution to problem (5) is obtained by solving the best-case problem.$\square$

5 Computational Experiments

We now report on a set of computational experiments with the ARO method on the robust ConFL problem. The purpose of these experiments is to assess the effectiveness of the ARO method, in terms of solution quality and computational time, to solve the robust ConFL problem under various uncertainty levels and uncertainty regions. Furthermore, these experiments allow us to observe how the solution changes under different levels of the conservatism parameter Γ. The ARO method (with the DLS heuristic) is coded in C++ and all computations are conducted on a computer with an Intel Core i7-2600 CPU @ 3.40 GHz and 16 GB RAM running Windows 7.

We generated instances by first selecting nodes randomly located on a 100 x 100 square grid. The Euclidean distances were used as a basis for the edge lengths. The assignment edge costs are equal to the edge lengths between demand nodes and facility nodes, while tree edge costs are equal to the edge lengths multiplied by an M factor. The M factor illustrates the significantly higher (in terms of cost per unit distance) connection cost, in practice, of edges in the tree T. For our test instances, we consider three levels for the M factor, $M \in \{3, 5, 7\}$. Each instance has 50 demand nodes, 50 facility nodes and 20 pure potential Steiner nodes. In addition, the facility opening cost was set to 30. These problem parameters cover a wide range of characteristics and were specifically chosen to include the hardest types of ConFL instances (for the DLS heuristic) reported in Bardossy and Raghavan [3].

In order to evaluate how the shape of the uncertainty region affects the ARO method's performance, in particular the computational times, we generated three sets of instances with various uncertainty regions: circular, square and rectangular. In Set 1 the location of demand nodes is represented by an uncertainty disk whose radius is uniformly generated in the range 0 to R. We considered various ranges for the radius and chose $R = 2, 5, 10$, or 20, on each subset of instances. Consequently, for each demand node, j, we defined a center location $\mathbf{j} = (j_1, j_2)$ and an uncertainty radius r_j. Then the minimum assignment cost for demand node j from facility i with coordinates $\mathbf{i} = (i_1, i_2)$ is $a_{ij} = \max\{||\mathbf{i} - \mathbf{j}||_2 - r_j, 0\}$, and the deviation is $d_{ij} = \min\{||\mathbf{i} - \mathbf{j}||_2 + r_j, 2r_j\}$.

In Set 2 the location of demand nodes is represented by a square uncertainty region, while in Set 3 the demand nodes are in a rectangular uncertainty region. Again to evaluate the effect of the magnitude of location uncertainty, we created instances with various ϵ_{j1} and ϵ_{j2} deviations for each coordinate axis. These values were uniformly generated between 0 and $R = 2, 5, 10$, and 20. For each demand node we defined a center location $\mathbf{j}$ and an uncertainty deviation ϵ_j (i.e., for square uncertainty region $\epsilon_{j1} = \epsilon_{j2}$.) Similarly to Set 1, we then calculate the minimum assignment cost and maximum deviation for each pair of demand and facility nodes.

We varied Γ between 0 and 50, in steps of 10, to assess its effect on the solution and the performance of the heuristic. We generated 10 instances for each combination of problem characteristics. Note that while the demand node center location will always fall within the 100 x 100 grid by construction, the uncertainty area may extend outside the predefined grid, and consequently, the worst-case location of a demand node may fall outside the grid.

5.1 Results for Set 1 - Disk Uncertainty Area

The ARO method yields high-quality solutions rapidly for the problems in Set 1. Table 5 reports average optimality gaps for different Γ values and maximum uncertainty radius. Recall, we generate 10 instances for each combination of Γ and R. Thus, each row of the table reports on the average over 10 instances. The average optimality gaps are under 3.34% ($M = 3$) and the highest gap for all instances is below 8.76% ($M = 7$). Furthermore, we observe that average gaps decrease for higher values of Γ. On the other hand, for higher magnitudes of location uncertainty the average gaps remain stable. In fact, the highest gap is

		Γ					
M	Maximum Radius	0	10	20	30	40	50
3	2	2.04%	1.98%	1.94%	1.92%	1.90%	1.90%
3	5	2.47%	2.29%	2.18%	2.10%	2.04%	2.03%
3	10	2.73%	2.35%	2.02%	1.88%	1.79%	1.77%
3	20	3.34%	2.61%	1.75%	1.63%	1.48%	1.43%
5	2	2.43%	2.37%	2.32%	2.30%	2.28%	2.27%
5	5	2.57%	2.41%	2.30%	2.23%	2.18%	2.17%
5	10	2.79%	2.46%	2.23%	2.06%	1.99%	1.97%
5	20	3.06%	2.36%	2.00%	1.82%	1.74%	1.71%
7	2	2.83%	2.77%	2.72%	2.69%	2.67%	2.67%
7	5	1.79%	1.70%	1.63%	1.59%	1.57%	1.56%
7	10	2.64%	2.39%	2.12%	2.00%	1.94%	1.92%
7	20	3.23%	2.58%	2.23%	2.03%	1.91%	1.88%

Table 5: Average Optimality Gaps for Set 1

observed for an instance with uncertainty radius bounded by 2 ($M = 7$).

Table 6 shows the average number of nominal problems solved for each instance, the average time the DLS heuristic took to solve the nominal problems and the average processing times. The processing time accounts for the time required to compute $F_{\Gamma}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and select the best solution out of all the solutions obtained for the nominal problems. The processing time is infinitesimally small (less than 0.001 second) for each Γ value. The average number of nominal problems increases with the magnitude of location uncertainty because as the radius of the uncertainty disk increases, facility nodes are more likely to fall within uncertainty disks. In addition, the average running time for the DLS heuristic increases as the number of nominal problems increases. However, the average time is below 5 seconds when the maximum radius is 10 or below 10 seconds when the maximum radius is 20. The maximum time required to solve one of these instances was 9.652 seconds (maximum radius = 20 and $M = 5$).

Recall, with the ARO method it is easy to rapidly compute heuristic solutions for many values of Γ; and thus from a sensitivity analysis perspective it is easy to observe when and how the solution changes for different Γ values. Since we have access to heuristic solutions across a large set of Γ values, we decided to perform some additional robustness/sensitvity analysis to examine how different the DLS heuristic for the best case-scenario was from the ARO heuristic solution across the different Γ values. We take the DLS solution $(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1})$ for the best-case scenario (i.e., the nominal problem Ω_1, which is identical to

M	Maximum Radius	Nominal Problems	DLS (sec)	Processing (sec)	Total (sec)
3	2	52	2.2703	0.0055	2.3033
3	5	59	2.6041	0.0062	2.6410
3	10	77	3.4245	0.0094	3.4807
3	20	141	6.3286	0.0203	6.4501
5	2	53	3.0160	0.0075	3.0609
5	5	57	3.4084	0.0093	3.4640
5	10	76	4.4714	0.0126	4.5471
5	20	139	8.4112	0.0242	8.5567
7	2	52	2.9000	0.0041	2.9245
7	5	57	3.1741	0.0055	3.2069
7	10	74	3.9815	0.0065	4.0204
7	20	128	7.0099	0.0111	7.0763

Table 6: Average Heuristic Times for Set 1

the robust formulation for $\Gamma = 0$); and evaluate its cost as the solution for different Γ values. (In other words we compute $\mathcal{Z}_1^H = F_\Gamma(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1})$ for different Γ values.) We use this to calculate the percentage cost difference (or loss) of the DLS heuristic solution $(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1})$ against the ARO method's solution for different values of Γ.

$$loss\% = \frac{F_\Gamma(\mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1}) - \mathcal{Z}^H}{\mathcal{Z}^H}. \tag{8}$$

Table 7 shows the average percentage loss in cost between the DLS heuristic best-case solution and the ARO heuristic solution. For $M = 7$ for different magnitudes of location uncertainty the average loss is below 0.33%. However, for $M = 3$ the average loss reaches 1.67% and the maximum loss for $\Gamma = 20$ is as high as 4.37% for one instance. Interestingly, we observe that for about half of the instances with low to medium magnitudes of location uncertainty (maximum radius = 2 to 10), the DLS heuristic best-case solution ($\mathcal{Z}_1^H$) is identical to the ARO method heuristic solution ($\mathcal{Z}^H$) for the different levels of conservatism considered. For high magnitudes of location uncertainty (maximum radius = 20) this is not the case (only one instance shows that behavior with $M = 3$). We should note the reason that there is a non-zero value in the column $\Gamma = 0$ is the DLS heuristic finds one solution, while the ARO method evaluates the heuristic solution obtained by all nominal problems to ascertain the best heuristic solution for a given value of Γ (including $\Gamma = 0$).

		Γ					
M	Maximum Radius	0	10	20	30	40	50
3	2	0.44%	0.43%	0.42%	0.41%	0.41%	0.41%
3	5	0.03%	0.02%	0.02%	0.02%	0.02%	0.02%
3	10	0.47%	0.45%	0.47%	0.44%	0.43%	0.40%
3	20	0.92%	0.88%	1.30%	1.61%	1.66%	1.64%
5	2	0.31%	0.30%	0.30%	0.29%	0.29%	0.29%
5	5	0.27%	0.26%	0.25%	0.24%	0.24%	0.24%
5	10	0.68%	0.58%	0.56%	0.55%	0.53%	0.52%
5	20	0.96%	0.97%	1.14%	1.12%	1.07%	1.06%
7	2	0.33%	0.32%	0.32%	0.32%	0.31%	0.31%
7	5	0.32%	0.29%	0.28%	0.28%	0.27%	0.27%
7	10	0.30%	0.32%	0.33%	0.30%	0.29%	0.29%
7	20	0.28%	0.29%	0.32%	0.32%	0.32%	0.33%

Table 7: Average Percentage Loss of the DLS Heuristic Best-Case Solution for Set 1

		Γ					
M	R Value	0	10	20	30	40	50
3	2	1.79%	1.71%	1.63%	1.59%	1.58%	1.57%
3	5	1.98%	1.73%	1.56%	1.51%	1.48%	1.46%
3	10	2.45%	1.85%	1.69%	1.71%	1.61%	1.58%
3	20	3.40%	2.03%	1.32%	1.27%	1.18%	1.17%
5	2	3.56%	3.46%	3.40%	3.32%	3.29%	3.28%
5	5	2.40%	2.19%	2.03%	1.98%	1.93%	1.90%
5	10	3.09%	2.72%	2.56%	2.44%	2.28%	2.24%
5	20	4.28%	2.99%	2.71%	2.30%	2.07%	1.99%
7	2	1.95%	1.90%	1.87%	1.87%	1.84%	1.84%
7	5	2.25%	2.10%	1.94%	1.88%	1.81%	1.80%
7	10	2.27%	1.84%	1.75%	1.67%	1.61%	1.59%
7	20	3.38%	2.44%	1.92%	1.71%	1.74%	1.72%

Table 8: Average Optimality Gaps for Set 2

M	R Value	Nominal Problems	DLS (sec)	Processing (sec)	Total (sec)
3	2	2485	113.3821	0.2217	114.7126
3	5	2484	113.2993	0.2618	114.8703
3	10	2478	115.1680	0.2966	116.9476
3	20	2480	115.7032	0.4209	118.2287
5	2	2480	126.9626	0.2339	128.3663
5	5	2484	129.2902	0.2464	130.7684
5	10	2483	123.0530	0.2489	124.5462
5	20	2477	126.3557	0.3126	128.2313
7	2	2474	143.9196	0.1850	145.0294
7	5	2473	140.6016	0.1935	141.7627
7	10	2477	141.6860	0.2590	143.2401
7	20	2473	144.9919	0.2354	146.4044

Table 9: Average Heuristic Times for Set 2

5.2 Results for Set 2 - Square Uncertainty Area

The ARO method also provides high-quality solutions for Set 2. Table 8 shows the average optimality gaps are under 4.28% ($M = 5$) and the maximum gap is below 6.93% ($M = 5$). The average gaps also decrease as Γ increases and there is no significant difference in the average gaps for higher magnitudes of location uncertainty. For low values of Γ, the average gap is slightly higher for higher uncertainty levels. However, this tendency fades away as Γ increases.

Table 9 shows the average number of nominal problems and times for instances in Set 2. The maximum computational time required by any instance is below 2.56 minutes ($M = 7$). With regards to the running time of the ARO method, the instances in Set 2 (and Set 3) take a significantly greater amount of time than the instances in Set 1. The main reason is that to solve the robust counterpart of these instances, we have to solve a significantly larger number of nominal problems. The average number of nominal problems for each instance in Set 2 and 3 is above 2400 (and this number remains about the same even as the R value changes). Recall that the maximum number of possible nominal problems in these instances is 2500. Consequently, the total computational times are much higher and they average around 2 minutes.

Table 10 shows the average difference between the cost of the DLS heuristic best-case solution and the ARO heuristic solution. The ARO heuristic solution is slightly better than the DLS heuristic best-case solution for low magnitudes of location uncertainty. However, for the greatest level of location uncertainty,

		Γ					
M	R Value	0	10	20	30	40	50
3	2	0.54%	0.53%	0.53%	0.54%	0.53%	0.52%
3	5	0.35%	0.32%	0.42%	0.42%	0.40%	0.39%
3	10	0.37%	0.75%	0.71%	0.70%	0.75%	0.73%
3	20	1.17%	1.75%	2.74%	2.83%	2.82%	2.74%
5	2	0.32%	0.33%	0.34%	0.34%	0.33%	0.33%
5	5	0.77%	0.73%	0.76%	0.75%	0.74%	0.73%
5	10	0.66%	0.78%	0.81%	0.81%	0.77%	0.76%
5	20	0.79%	1.28%	1.58%	1.70%	1.80%	1.76%
7	2	0.20%	0.20%	0.22%	0.22%	0.22%	0.22%
7	5	0.58%	0.59%	0.69%	0.67%	0.68%	0.68%
7	10	0.40%	0.49%	0.46%	0.45%	0.42%	0.42%
7	20	1.35%	1.23%	1.42%	1.39%	1.37%	1.34%

Table 10: Average Percentage Loss of the DLS Heuristic Best-Case Solution for Set 2

the DLS heuristic best-case solution can be on average as high as 2.84% more costly than the ARO heuristic solution. Over the whole set, the most costly DLS heuristic best-case solution is 5.39% more costly than its ARO heuristic solution counterpart.

5.3 Results for Set 3 - Rectangular Uncertainty Area

Table 11 shows the average optimality gaps for Set 3. The ARO method also provides high-quality solutions for Set 3. The average optimality gaps are under 3.35% ($M = 5$) slightly below the average optimality gaps observed in Set 2. Similarly, the highest gap in Set 3 is below 5.86% ($M = 5$). For smaller R values ($R = 2, 5$) the average gaps show an overall tendency to decrease as Γ increases. However, for larger magnitudes of location uncertainty ($R = 10, 20$) the average gaps first decrease with Γ and then increase for the largest values of Γ.

Table 12 shows the average number of nominal problems is slightly higher in Set 3; and consequently, the computational times are proportionally higher. The average total time per instance is approximately 2 minutes, with the maximum of 2.62 minutes. The running times for Set 2 and 3 increase as M increases.

Table 13 shows the average difference between the cost of the DLS heuristic best-case solution and the ARO heuristic solution.

		Γ					
M	R Value	0	10	20	30	40	50
3	2	1.72%	1.68%	1.64%	1.61%	1.60%	1.59%
3	5	1.83%	1.63%	1.62%	1.49%	1.37%	1.42%
3	10	2.27%	1.87%	1.74%	1.62%	1.64%	1.65%
3	20	2.62%	1.66%	1.29%	1.30%	1.48%	1.77%
5	2	2.16%	2.07%	2.01%	1.93%	1.91%	1.92%
5	5	2.74%	2.57%	2.39%	2.31%	2.15%	2.08%
5	10	2.18%	2.07%	1.99%	1.90%	1.84%	1.89%
5	20	3.35%	2.59%	2.28%	2.22%	2.19%	2.43%
7	2	2.22%	2.17%	2.13%	2.10%	2.05%	2.03%
7	5	1.91%	1.75%	1.69%	1.63%	1.61%	1.62%
7	10	2.48%	2.18%	1.99%	2.08%	2.07%	1.91%
7	20	1.39%	1.26%	1.47%	1.56%	1.73%	1.78%

Table 11: Average Optimality Gaps for Set 3

M	R Value	Nominal Problems	DLS (sec)	Processing (sec)	Total (sec)
3	2	2494	113.7649	0.2043	114.9907
3	5	2494	113.3844	0.2932	115.1435
3	10	2489	113.9181	0.3421	115.9708
3	20	2491	116.6032	0.5112	119.6702
5	2	2487	128.1238	0.3060	129.9597
5	5	2483	132.4882	0.3175	134.3932
5	10	2478	131.8056	0.3044	133.6323
5	20	2480	135.0287	0.4783	137.8983
7	2	2483	141.2065	0.1878	142.3334
7	5	2489	140.5241	0.2387	141.9561
7	10	2483	143.2032	0.2417	144.6533
7	20	2485	147.3074	0.2670	148.9095

Table 12: Average Heuristic Times for Set 3

		Γ					
M	R Value	0	10	20	30	40	50
3	2	0.53%	0.52%	0.53%	0.54%	0.53%	0.52%
3	5	0.54%	0.52%	0.56%	0.62%	0.65%	0.65%
3	10	0.64%	0.60%	0.75%	0.97%	1.02%	0.94%
3	20	1.00%	1.83%	2.63%	2.97%	3.21%	3.17%
5	2	0.33%	0.35%	0.36%	0.41%	0.42%	0.43%
5	5	0.56%	0.63%	0.66%	0.67%	0.69%	0.70%
5	10	0.94%	0.98%	1.07%	1.22%	1.36%	1.37%
5	20	1.39%	1.23%	1.55%	1.81%	2.09%	2.17%
7	2	0.27%	0.28%	0.26%	0.25%	0.25%	0.24%
7	5	0.31%	0.33%	0.31%	0.38%	0.40%	0.40%
7	10	0.88%	0.82%	0.84%	0.95%	1.09%	1.25%
7	20	0.81%	0.99%	1.16%	1.52%	1.56%	1.58%

Table 13: Average Percentage Loss of the DLS Heuristic Best-Case Solution for Set 3

5.4 Smaller Data Set and Comparison to Compact Robust Formulation

The first three data sets provide a significant amount of information regarding the performance of the ARO method on the robust ConFL problem. As an alternative to solving all of the nominal problems to solve the robust optimization problem (5); one can solve a single mixed-integer program (MIP) that models the deviation in the objective function of problem (5) (see Theorem 1 in [7]). To get a sense of the speedup in applying the ARO method, as well as to visually display solutions for some non trivial but smaller instances, we created a fourth set of 10 test instances. They contain 25 demand nodes, 25 facility nodes, and 10 facility nodes. For these instances we fixed the facility opening cost to 30, M = 3, and Γ = 10. The uncertainty area is rectangular with R value of 20.

Formulation (9) shows the *compact formulation* of the robust ConFL problem as an MIP. We implemented constraint set (9e) by means of a single commodity flow (SCF) formulation for the ConFL problem. It is certainly possible to implement constraint set (9e) using a multicommodity flow formulation or using an exponential set of GSECs (as in (1b)). In terms of a quick comparative implementation the multicommodity flow formulation is not viable because it rapidly blows up and is not computationally tractable. The GSEC based formulation requires the implementation of a branch-and-cut procedure which is specialized and beyond the scope of this paper. Hence, we implemented (9e) by means of a SCF formulation in CPLEX 12.5.

$$\text{Minimize} \sum_{i \in F} f_i z_i + \sum_{\{i,j\} \in E(S \cup F)} b_{ij} y_{ij} + \sum_{i \in F, j \in D} a_{ij} x_{ij} + \sum_{i \in F, j \in D} \pi_{ij} + \Gamma \delta \tag{9a}$$

subject to

$$\pi_{ij} + \delta - d_{ij} x_{ij} \geq 0 \; \forall i \in F, \forall j \in D \tag{9b}$$

$$\delta \geq 0 \tag{9c}$$

$$\pi_{ij} \geq 0 \; \forall i \in F, \forall j \in D \tag{9d}$$

$$(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathcal{X} \tag{9e}$$

We limited the size of our instances to 60 nodes because CPLEX 12.5 could not handle larger instances within a reasonable time. We limited the running time for all methods to 1800 seconds. Table 14 shows the results obtained by Formulation (9), and the ARO heuristic. For comparative purposes we also provide the running time for one nominal problem with the SCF formulation. This was then used to get a very rough estimate of the time it would take to solve the problem if the SCF formulation (from a computational perspective we would not recommend the single commodity flow formulation for exact approaches, and would recommend using a state-of-the-art branch-and-cut code as in [10, 19] to solve nominal ConFL instances) was used.

Table 14 shows the total cost of the best solution found within the 1800 seconds allotted time for Formulation (9), the optimality gap of the solution using its best-known lower bound, and the total time in seconds. Table 14 also shows the total cost, the optimality gap of the solution (using the ARO method's lower bound) and the total time in seconds for the ARO heuristic. As can be seen from the table, the upper bounds found by both the compact formulation and the ARO method are comparable. However, the compact formulation using the SCF provides weak lower bounds. In fact for instance 7 we let Formulation (9) run for two hours and obtained a slightly better solution than the ARO heuristic; yet this compact formulation reported a gap of 17.3% (but when the solution is measured against the ARO lower bound, the gap is in fact 2.8%). On the other hand, the ARO heuristic obtained a solution with an optimality gap of 2.9% in 45 seconds. The last

	Formulation (9)			ARO Heuristic			Nominal Problem (SCF)		
Inst	Cost	Gap	Time	Cost	Gap	Time	Time	p	Est. Time
1	903.42	8.67%	1801.74	909.84	0.71%	49.40	3.91	619	2420.40
2	1014.05	13.99%	1801.50	994.16	0.16%	52.48	6.51	618	4024.26
3	938.61	17.09%	1803.82	940.91	0.24%	47.32	35.93	616	22133.31
4	914.36	16.64%	1800.94	914.36	0.37%	44.07	11.24	621	6977.91
5	994.01	19.50%	1801.17	979.81	1.33%	101.68	56.19	620	34837.42
6	929.75	27.57%	1802.09	931.38	0.67%	43.92	24.54	616	15115.20
7	934.63	19.25%	1802.47	916.57	2.92%	45.56	10.65	619	6593.27
8	779.79	12.69%	1800.59	779.79	0.00%	44.05	6.62	616	4080.70
9	919.96	19.24%	1803.65	924.13	0.45%	42.05	25.00	620	15497.11
10	962.79	8.67%	1804.67	959.82	0.24%	46.60	13.56	624	8460.81

Table 14: Comparison of Results for Set 4

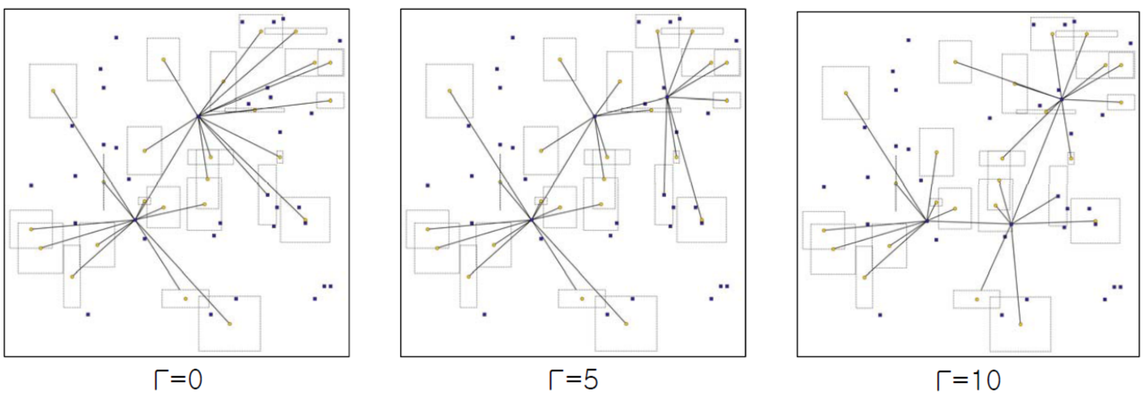

Figure 3: Example of robust ConFL problem - Instance 7

few columns of the table show the total time required to solve one nominal instance to optimality using the SCF formulation, the total number of nominal problems, and an estimate of the total time required should one solve the robust problem to optimality applying the BS algorithm with the SCF formulation for each nominal problem.

Figures 3 and 4 provide a visual representation of the solution for different Γ values. This is another significant benefit of the ARO method. The ARO allows the decision-maker to rapidly observe how the (approximate) solution to the problem changes as the level of conservatism changes. For instance 9, we observe that the ARO method's (robust) solution changes from low conservatism values to medium conservatism values and then changes back for high conservatism values. This type of information can be

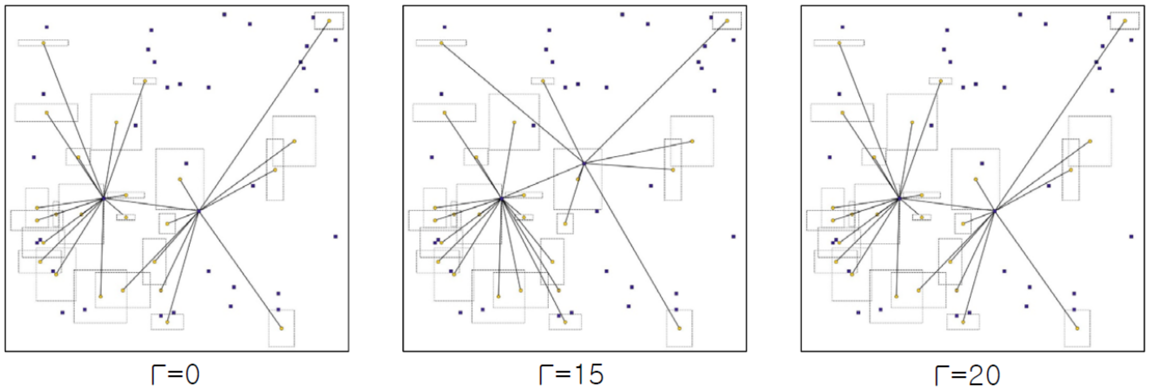

Figure 4: Example of robust ConFL problem - Instance 9

particularly useful to a decision maker who may be unsure of what value of Γ to choose. Having a solution that is near optimal for a large set of conservatism values can be useful in making a final decision on the "robust" solution to choose.

6 Conclusions

In this paper we introduced the robust ConFL problem. It more accurately models the practical problems that motivated the ConFL [11, 14]. To address the issue of solving large-scale robust variants of challenging (NP-hard) discrete optimization problems, we introduced the approximate robust optimization method. To work well in practice, this approach requires both high quality upper and lower bounds for each nominal problem. We applied the ARO method to the robust ConFL problem, using a dual-based local search heuristic that gives high quality upper and lower bounds on the deterministic ConFL [3]. A more significant contribution is the ARO method—along with its application on the robust ConFL problem. Secondly, we proposed an ARO method by extending the BS robust optimization approach using heuristics and lower bounding procedures for the nominal problems. And lastly, we demonstrated this approach using a dual-based local search heuristic and found that it yielded high-quality solutions for the robust ConFL problem.

The ARO method found high-quality solutions to the robust ConFL very rapidly. In fact, were we to solve each nominal problem exactly using a state-of-the-art method as in [10, 19] it would require a significantly greater computational effort[3] (with the benefit of course being we would have solved the robust ConFL to optimality). Since the ARO method expands the scope of the BS method in a computationally practical manner— allowing one to apply heuristics to the nominal problems (when there is a good lower bound at hand) and obtain high-quality solutions to the robust optimization problem rapidly—we hope this approach will be adopted by researchers for other robust optimization problems.

References

[1] Eduardo Álvarez-Miranda, Ivana Ljubić, and Paolo Toth. A note on the Bertsimas & Sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4):349–360, 2013.

[2] I. Aron and P. Van Hentenryck. On the complexity of the robust spanning tree problem with interval data. *Operations Research Letters*, 32(1):36–49, 2004.

[3] M.G. Bardossy and S. Raghavan. Dual-based local search for the connected facility location and related problems. *INFORMS Journal on Computing*, 22(4):584–602, 2010.

[4] O. Baron, J. Milner, and H. Naseraldin. Facility Location: A Robust Optimization Approach. *Production and Operations Management*, 20(5):772–785, 2011.

[5] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[6] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.

[7] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.

[8] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[9] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1174–1183, 2008.

[10] S. Gollowitzer and I. Ljubic. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2):435–449, 2010.

[11] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 389–398, 2001.

[3] Leitner et al. [19] are able to solve to optimality instances similar to the ones presented here in an average of 68 seconds. With approximately 2400 nominal instances, one can estimate about 2400 minutes to apply the BS method with an exact algorithm as opposed to less than 2.5 minutes (for near-optimality) with the ARO method.

[12] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden. Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 606–615, 2003.

[13] H. Jung, M. K. Hasan, and K. Y. Chwa. Improved primal-dual approximation algorithm for the connected facility location problem. *Lecture Notes in Computer Science*, 5165:265–277, 2008.

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

[15] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, Boston, 1997.

[16] Y. Lee, L. Lu, Y. Qiu, and F. Glover. Strong formulations and cutting planes for designing digital data service networks. *Telecommunication Systems*, 2(1):261–274, 1993.

[17] Y. Lee, S. Y. Chiu, and J. Ryan. A branch and cut algorithm for a Steiner tree-star problem. *Journal on Computing*, 8(3):194–201, 1996.

[18] Markus Leitner, Ivana Ljubić, Juan-José Salazar-González, and Markus Sinnl. On the asymmetric connected facility location polytope. In Alexander Schrijver, editor, *Combinatorial Optimization*, pages 371–383. Springer, 2014.

[19] Markus Leitner, Ivana Ljubic, Juan-José Salazar-González, and Markus Sinnl. The connected facility location polytope: valid inequalities, facets and a computational study. *Optimization online*, 2014.

[20] I. Ljubić. A hybrid VNS for connected facility location. *Lecture Notes in Computer Science*, 4771: 157–169, 2007.

[21] L.V. Snyder. Facility location under uncertainty: A review. *IIE Transactions*, 38(7):547–564, 2006.

[22] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157, 1973.

[23] C. Swamy and A. Kumar. Primal–dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[24] A. Tomazic and I. Ljubić. A GRASP algorithm for the connected facility location problem. *2008 International Symposium on Applications and the Internet*, pages 257–260, 2008.

[25] Hande Yaman, Oya Ekin KaraşAn, and Mustafa Ç PıNar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29(1):31–40, 2001.

[26] Jianming Zhu, Jun Huang, Degang Liu, and Jiye Han. Connected distribution center location problem under traffic network in emergency management. In *Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on*, pages 160–163. IEEE, 2012.