An Inexact Sample Average Approximation Approach for the Stochastic Connected Facility Location Problem

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

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

The sample average approximation (SAA) approach is a widely used technique, based on Monte-Carlo simulation, often applied to large-scale stochastic optimization problems. In this approach, a set of sample average problems with multiple copies of sampled scenarios are generated and solved exactly. In other words, there is an implicit assumption that the sample average problems are solvable to optimality. In some instances, however, the sample average problems might be NP-hard problems, often difficult or impractical to solve to optimality. In this article, we broaden the scope of the SAA approach and show that even without solving the sample problems to optimality, by combining a heuristic and a lower bounding approach, high-quality solutions with tight confidence bounds on the optimal solution value can be obtained. We demonstrate this "inexact SAA approach" on two problems. First, we apply it to the Stochastic Connected Facility Location (SConFL) problem, the motivating application for this article, that arises in the design of telecommunications networks. As an additional application, we also use it for the Stochastic Uncapacitated Facility Location (SUFL) problem. Our computational results demonstrate the effectiveness of the inexact SAA approach.
NETWORKS, Vol. 70(1), 19–33 2017

Keywords: heuristics; network design; network optimization; stochastic optimization; connected facility location problem; uncapacitated facility location problem; sample average approximation

1. INTRODUCTION

The sample average approximation (SAA) method is an approach for solving large stochastic optimization problems using Monte Carlo simulation. Kleywegt et al. [17], Mak et al. [27], Shapiro and Philpott [33], and Verweij et al. [37] provide good introductions to this approach. In this technique, the expected objective function of the stochastic problem is approximated by a sample average estimate derived from a random sample. The resulting sample average approximating problem (or simply sample average problem), which is a deterministic problem, is then solved exactly by optimization techniques. The process is repeated with different samples to obtain candidate solutions along with statistical estimates of their variability and optimality gaps.

As indicated by Mak et al. [27], this approach assumes that the instances of the sample average problem can be solved for sufficiently large samples to yield "good" bounding information. Unfortunately, in many settings, like the applications considered in this article, it is computationally challenging to even solve the sample average problem. This is particularly, the case for discrete (or integer) optimization problems. Instead of solving the sample average problem exactly, we broaden the scope of the SAA method by applying a heuristic and lower bounding technique to the sample average problem. We then show how to use the heuristic solutions and lower bounds to the sample average problems to construct confidence intervals (or bounds) on the optimal value function of the stochastic program. As we do not solve the sample average problems exactly, we refer to the technique as an "inexact sample average approximation" method.

We first demonstrate the inexact SAA approach on the Stochastic Connected Facility Location (SConFL) problem, the motivating application for this article. The connected facility location (ConFL) problem (see [2]) arises in a number of applications that relate to the design of telecommunication networks as well as data distribution and management problems on networks. This problem combines facility location decisions with network design as the open facilities must be connected through a core network.

Specifically, in the ConFL problem, 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;

Received September 2015; accepted March 2017
Correspondence to: S. Raghavan; e-mail: raghavan@umd.edu
DOI 10.1002/net.21743
Published online 28 April 2017 in Wiley Online Library (wileyonlinelibrary.com).

and $S \subseteq V$, set of potential Steiner nodes, with $D \cup F \cup S = V$. The objective is 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, $f_i \geq 0$ for each facility i; assignment costs, $a_{ij} \geq 0$, for assigning a customer $j \in D$ to a facility $i \in n(j)$ where $n(j)$ is the set of facilities in F that node j can connect to; 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 $\sum_{j \in D} a_{i(j)j} + \sum_{i \in Y} f_i + \sum_{\{i,j\} \in E(T)} b_{ij}$, where $i(j)$ is the facility serving demand node j, Y is the set of open facilities, and T is a Steiner tree connecting the open facilities.

The ConFL problem is deterministic; however, the motivating applications arise in stochastic settings. In particular, both Krick et al. [18] and Nuggehalli et al. [30] describe network information caching problems, which are modeled as ConFL problems. Information must be cached at various nodes of the network to serve its users. However, at the time that the network is designed the number of read and write requests is unknown. Similarly, in the rent-or-buy problem described by Karger and Minkoff [16], which is a special case of the ConFL problem, the assignment costs might not be available until the last moment when the assignment edges are rented. In most cases, the literature assumes average values as an approximation, and addresses the deterministic ConFL problem. In this article, we seek to explore the value of explicitly modeling uncertainty in the assignment costs, and thus consider the Stochastic ConFL (SConFL) problem.

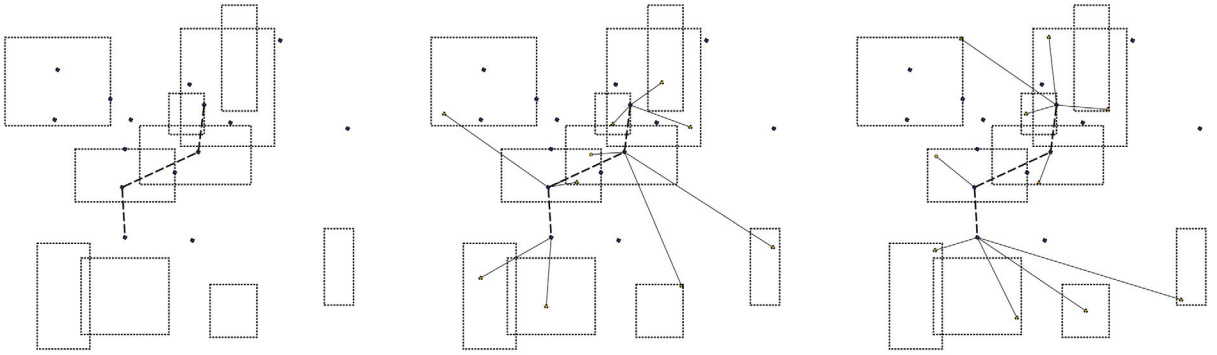
In an instance of the SConFL problem the facility opening costs and the connection costs between them are assumed to be known a priori, while the assignment costs are unknown and dependent upon a set of random scenarios. We are thus in the setting of a two-stage stochastic optimization problem with fixed recourse as introduced by Dantzig [8]. In the first stage, a set of facilities must be opened and a Steiner tree that connects them constructed. In the second stage, uncertainty on the assignment costs is unveiled (i.e., one scenario is realized) and customers must be assigned to open facilities. Figure 1 shows an example where the location of the customers is unknown but limited within a bounded area represented by the rectangles (there are 10 rectangles for the 10 customers). Each of the potential facility locations is identified by the nodes in the figure. Figure 1a shows a core network solution for the first stage of the problem, where a Steiner tree is constructed connecting the facilities. Figure 1b and c show two different realizations of customer locations and their subsequent allocation to open facilities in the second stage. Note that the second stage allocation will be different under different scenarios; that is, customers are assigned to their closest (when assignment costs are a function of distance) open facility depending on their actual location in the second stage. The objective is to design the core network in the first stage that consists of the open facilities and the Steiner tree connecting them, to minimize the overall network cost that includes the core network of the first stage and the expected assignment cost of the second stage.

In the SConFL problem, there are two sources of uncertainty in the assignment costs: (i) customer demand and (ii) customer location (or travel time to potential facilities). Both sources of uncertainty affect the assignment costs; however, they have a very different impact on the objective function of the problem that requires independent discussion. When solely demand, quantities are unknown, we show that the SConFL problem can be optimally solved by replacing the demand quantities by their expected values. The two-stage stochastic problem can nicely be reduced into a one stage problem without recourse. The reason is that demand quantities at a node affect all assignment costs from that node in exactly the same way. Hence, once facilities have been opened, demand nodes are assigned to the closest (per unit of demand) facility (i.e., the quantity of demand does not affect which facility is closest to the customer). There is no recourse in the second stage and the problem can be solved in one stage. The problem is still NP-complete but the value of the stochastic solution is null (i.e., using average demand values as in [18] solves the problem). However, when customer location is the source of uncertainty (in this category we also include other sources of uncertainty in assignment costs that have a similar effect and do not affect all assignment costs in the same direction; i.e., some costs can increase and some can decrease) this is not the case.

For the SConFL problem, there are three different models for the second stage scenarios: (i) a finite-scenario model, where one assumes that there are only a finite number of scenarios that occur with positive probability, and these can be explicitly enumerated; (ii) an independent-activation model, where the scenario-distribution is a product of independent distributions (e.g., in Figure 1 the scenarios are generated by letting each customer node be uniformly distributed in the rectangle bounding its location); and (iii) a black-box model, where nothing is explicitly assumed about the probability distribution, other than the availability of an oracle that can generate scenarios from the unknown distribution.

For the case with a finite number of scenarios, we outline a transformation that essentially replicates each customer (demand) node once for each scenario, to obtain an equivalent but significantly larger deterministic ConFL problem. This can either be solved exactly when feasible to do so (given the size of the problem) using an exact method such as the one

[1] We should note that our definition of the ConFL problem follows Bardossy and Raghavan [2]. They show that at the cost of doubling the number of nodes in the network, 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 Steiner tree star problem [21], and the rent or buy problem [13].



(a) Core Network - First Stage (b) One Scenario - Second Stage Recourse (c) Another Scenario - Second Stage Recourse

FIG. 1. Stochastic connected facility location problem example. [Color figure can be viewed at wileyonlinelibrary.com]

described in Leitner et al. [24] or approximately/heuristically as the dual-based local search (DLS) heuristic described in Bardossy and Raghavan [2] (if we solve the deterministic equivalent problem with the DLS heuristic we get a solution to the SConFL problem along with a lower bound that provides a quality guarantee on how far the solution is from optimality). As the number of scenarios grow solving the deterministic equivalent problem becomes impractical. In such a setting, it is common to apply the SAA method. The SAA method requires (i) creating problems by sampling with a smaller number of scenarios (say N) and solving this sample average problem; and (ii) replicating this process R times (i.e., solving R sample average problems). However, as discussed previously solving the sample average problem exactly for the ConFL problem can be challenging in and of itself; making it hard to apply the SAA method. Consequently, we apply the inexact SAA method that we describe in this article (recall we use the term inexact SAA as we obtain a heuristic solution and a lower bound for each of the sample average problems instead of solving them exactly) to the SConFL problem.

We report on computational results on a comprehensive set of randomly generated instances using this inexact SAA approach. Our computational results indicate (at least experimentally) the inexact SAA approach has significant computational merits and can obtain approximate/heuristic solutions and tight bounds on the optimal solution value for large two-stage stochastic integer programming problems rapidly. Over 1080 SConFL problem instances, it generates solutions that are on average 2.23% from the optimal (with 99% confidence) taking an average of 137.7 seconds for each problem instance. To demonstrate the applicability of the inexact SAA approach to other large-scale two-stage discrete optimization problems we consider the Stochastic Uncapacitated Facility Location (SUFL) problem (it is similar to the SConFL problem but without the added requirement that open facilities have to be connected). The results of the inexact SAA approach are promising. Over 460 SUFL problem instances it generates solutions that are on average 2.63% from the optimal (with 99% confidence) taking an average of 8.7 seconds for each problem instance.

The rest of this article is organized as follows. In Section 2, we review prior literature on the SAA approach as well as the ConFL problem and related stochastic problems. In Section 3, we describe the inexact SAA method. We discuss the quality of the solution and provide confidence bounds on the optimal solution. In Section 4, we provide a mathematical programming formulation for the SConFL problem. We then describe the deterministic equivalent formulation for uncertain demand and locations in the finite scenario model. In Section 5, we illustrate how to apply the inexact SAA method to the SConFL and SUFL problems with an extensive set of computational experiments. Finally, Section 6, provides concluding remarks and directions for future research.

2. LITERATURE REVIEW

There has been considerable research on two-stage stochastic optimization problems with recourse [see [5]]; however, many of these methods assume linearity on the decisions of the first and second stage decision variables. Integer (and binary) decisions in both stages make the stochastic problem even harder to solve. Ahmed [1] provides a brief introduction to the topic and outlines the difficulties that arise especially when solving two-stage stochastic integer problems. Schultz et al. [32] provide a comprehensive survey of methodologies for two-stage stochastic integer programming. When the number of scenarios is large, as discussed previously, the sample average approximation (SAA) method is an approach for solving large stochastic optimization problems using Monte Carlo simulation. Kleywegt et al. [17] discuss the SAA method in the context of discrete optimization. However, it is implicitly assumed within their article that the sample average problems are solved to optimality (which can be challenging in and of itself for a discrete optimization problem). In this article, we present an approximate solution approach that can be applied in situations when there are a large number of scenarios, to find heuristic solutions and

construct tight confidence intervals on the optimal function value.

Karger and Minkoff [16] introduced the ConFL problem, in an application where they were attempting to solve a network design problem with incomplete information. Gupta et al. [12] 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 [35] described a primal-dual approximation algorithm for the ConFL problem with an approximation ratio of 8.55, which Jung et al. [15] later improved to 6.55. Eisenbrand et al. [9] presented a randomized algorithm that further improves the approximation ratio to 4 (the ratio slightly degrades to 4.23 when the algorithm is derandomized).

With a focus on computationally solving the problem Ljubić [25] 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 [36] 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 [2] proposed a dual-based local search (DLS) heuristic that provides both a tight lower bound and a high-quality solution. The DLS heuristic works in three steps on the ConFL problem. First, it ensures the ConFL problem instance is as defined within their (and this) article (e.g., duplicating a facility node when it does not incur a cost when it is used on the Steiner tree and has no customers connected to it) and then transforms this ConFL problem into a directed Steiner tree problem with a unit degree constraint. Second, it applies a dual-ascent procedure to find both a heuristic solution (i.e., a primal solution) and a lower bound to the ConFL problem (this procedure is actually applied on the directed Steiner tree problem with the unit degree constraint). Third, it improves upon the heuristic solution obtained by dual-ascent through local search moves. The result is a high-quality solution to the ConFL problem with an accompanying measure of its quality provided by the lower bound. Gollowitzer and Ljubić [11] propose several mathematical formulations for the ConFL problem based on direct graphs and compare their linear-programming relaxations. Leitner et al. [24] present a new formulation based on a mixed graph and investigate the associated polytope. In a related article, Leitner et al. [22] 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 [12, 16]). Our article presents the first study of the stochastic variant of the ConFL with uncertainty on the assignment costs.

There have been several articles in the literature that deal with facility location or network design with uncertain demands or edge lengths. Snyder [34] provides a comprehensive review on stochastic and robust facility location models. The earliest articles dealing with stochastic facility location are by Mirchandani [28] and Mirchandani and Odoni [29] where they extend the concept of p-median location to networks whose edge costs are random variables. Their main motivation is the deployment of a service vehicle in a city when the travel times vary randomly and throughout the day due to traffic congestion. The objective of the problem is to minimize the expected travel time to any destination node in the network. Weaver and Church [38] address the same problem and develop a computational procedure. In these articles, no recourse is defined in the problem. Berman [3] and Berman and Odoni [4] add the option of relocating the service vehicle once travel times are revealed (i.e., adding recourse). Berman [3] introduces a heuristic for this problem, that is generalized to multiple facilities by Berman and Odoni [4].

In another set of facility location problems, the uncertainty is in the customer demands. Correia and Saldanha da Gama [7] survey facility location under demand uncertainty. They first discuss the stochastic uncapacitated facility location problem, then review the stochastic capacitated facility location problem including chance-constrained variants. Laporte et al. [20] analyze the capacitated facility location problem with uncertain demand. They state the problem as a two-stage program with recourse where the first stage decisions define the location of the facilities, their capacities, and the allocation decision (i.e., determination of which facility serves each customer); while the second-stage decisions determine the distribution decisions (i.e., quantities delivered to each demand node). Louveaux and Peeters [26] deal with a more general version of the capacitated facility location problem that models both uncertain demands and edge costs. In the first stage, decisions regarding location and capacities of the plants are taken. Next, in the second stage, after demands, prices and costs are revealed, both the allocation and distribution decisions are determined. They propose a deterministic equivalent problem for the stochastic problem, and extend the dual-based procedure of Erlenkotter [10] for the uncapacitated facility location problem and show its effectiveness as a heuristic for this stochastic facility location problem. However, their application of the procedure is limited to very small instances in terms of the number of scenarios (only one, three, or five scenarios).

A different path of research, pursued in the computer science literature, has been the development of approximation algorithms for stochastic facility location problems. Gupta et al. [14] find constant factor approximation algorithms for the stochastic Steiner tree problem and single sink network design problem. Kurz et al. [19] show that the stochastic Steiner tree problem is in the class of fixed-parameter tractable problems, and transfer their results to the directed and prize-collecting variants of the problem. Ravi and Sinha [31] consider two-stage finite scenario stochastic versions of several combinatorial optimization problems including the uncapacitated facility location problem. In their version of the SUFL problem facilities are opened in the first stage. In the second-stage demands are revealed, and it is possible to open additional facilities at a higher cost (so the second stage decisions determine which additional

facilities to open and which open facility to use to serve each customer). They find an 8-approximation algorithm for this variant of the SUFL problem. As the number of scenarios grow in all of these challenging two-stage stochastic combinatorial optimization problems (possibly due to an independent-activation model for scenarios) our proposed inexact SAA method is a viable solution approach to provide high-quality approximate solutions.

3. INEXACT SAMPLE AVERAGE APPROXIMATION METHOD

In this section, we describe the inexact SAA method for a combinatorial two-stage linear stochastic programming problem. In contrast to the commonly applied SAA method in the literature (where the sample average problems are solved exactly), we solve the sample average problems (which are also combinatorial problems) with a heuristic coupled with a lower bound mechanism. Consequently, we refer to this variation of the SAA method as the inexact SAA method.

A combinatorial two-stage linear stochastic programming problem can be formulated as

$$\min_{\mathbf{p}\in P} \left\{ g(\mathbf{p}) = \mathbf{c}^T\mathbf{p} + \mathbb{E}[Q(\mathbf{p},\omega)] \right\} \tag{1}$$

where $Q(\mathbf{p},\omega)$ is the optimal value of the second-stage problem $\min_{\mathbf{x}\in X(\mathbf{p},\omega)} \mathbf{s}(\omega)^T\mathbf{x}$. Here $\mathbf{p}$ is the first-stage decision vector, and P is a discrete set represented as $P = \{\mathbf{p}|\mathbf{B}\mathbf{p} = \mathbf{b}, \mathbf{p} \geq \mathbf{0}$ and integer$\}$. In other words, $\mathbf{B}$ and $\mathbf{b}$ represent the data of the first-stage problem. The second-stage decision vector is $\mathbf{x}$, and $X(\mathbf{p},\omega) = \{\mathbf{x}|\mathbf{A}(\omega)\mathbf{x} = \mathbf{h}(\omega) - \mathbf{C}(\omega)\mathbf{p}, \mathbf{x} \geq \mathbf{0}$ and integral$\}$. In other words, $X(\mathbf{p},\omega)$ is also a discrete set dependent on the scenario ω and the first stage decision vector $\mathbf{p}$. Here $\mathbf{s}(\omega), \mathbf{A}(\omega), \mathbf{C}(\omega)$ and $\mathbf{h}(\omega)$ represent the data of the second-stage problem of the realized scenario ω.

In the SAA method, the expected value function $\mathbb{E}[Q(\mathbf{p},\omega)]$ is approximated by the sample average function $\bar{Q}_N(\mathbf{p}) = \sum_{n=1}^{N} Q(\mathbf{p},\omega^n)/N$, where a sample $\{\omega^1,\omega^2,\ldots,\omega^N\}$ of N sample scenarios is generated from Ω according to probability distribution $P(\Omega)$. The sample average problem

$$u^N = \min_{\mathbf{p}\in P} \left\{ \bar{g}_N(\mathbf{p}) = \mathbf{c}^T\mathbf{p} + \frac{1}{N}\sum_{n=1}^{N} Q(\mathbf{p},\omega^n) \right\}, \tag{2}$$

corresponding to the original two-stage stochastic problem is then solved using a deterministic optimization algorithm. The optimal value u^N and an optimal solution $\hat{\mathbf{p}}$ to the sample average problem provide estimates of their true counterparts in the stochastic program. By generating R independent samples, each of size N, and solving the associated sample average problems, objective values $u^{N1}, u^{N2}, \ldots, u^{NR}$ and candidate solutions $\hat{\mathbf{p}}^1, \hat{\mathbf{p}}^2, \ldots, \hat{\mathbf{p}}^R$ are obtained. Let

$$\bar{u}^N = \frac{1}{R}\sum_{m=1}^{R} u^{Nm} \tag{3}$$

denote the average of the R optimal values of the sample average problems.

This procedure produces up to R different candidate solutions. Out of these R different candidate solutions, we have to select one as the approximation to the optimal solution of the original stochastic program. One generally accepted strategy is to generate a sample problem with a significantly large number of scenarios, $N' >> N$. Then, it is natural to take $\hat{\mathbf{p}}^*$ as one of the optimal solutions $\hat{\mathbf{p}}^1, \hat{\mathbf{p}}^2, \ldots, \hat{\mathbf{p}}^R$ of the R sample average problems that has the smallest estimated objective value, that is,

$$\hat{\mathbf{p}}^* \in \arg\min \left\{ \bar{g}_{N'}(\hat{\mathbf{p}}) | \hat{\mathbf{p}} \in \left\{ \hat{\mathbf{p}}^1, \hat{\mathbf{p}}^2, \ldots, \hat{\mathbf{p}}^R \right\} \right\} \tag{4}$$

where $\{\omega^1,\omega^2,\ldots,\omega^{N'}\}$ is the sample of scenarios chosen to evaluate the candidate solutions.

When a heuristic is used to obtain upper bounds on the sample average problems, we adapt the SAA method as follows. We generate R heuristic candidate solutions $\mathbf{p}_H^1, \mathbf{p}_H^2, \ldots, \mathbf{p}_H^R$ to the sample average problems with their corresponding objective values, $u_H^{N1}, u_H^{N2}, \ldots, u_H^{NR}$. Then, we take as the heuristic solution to the stochastic program the heuristic solution $\mathbf{p}_H^*$ that has the smallest estimated objective value in the sample problem with N' scenarios, that is,

$$\mathbf{p}_H^* \in \arg\min \left\{ \bar{g}_{N'}(\mathbf{p}_H) | \mathbf{p}_H \in \left\{ \mathbf{p}_H^1, \mathbf{p}_H^2, \ldots, \mathbf{p}_H^R \right\} \right\}. \tag{5}$$

To provide quality bounds on this heuristic solution, we also compute lower bounds, $u_{LB}^{N1}, u_{LB}^{N2}, \ldots, u_{LB}^{NR}$ on each of the sample problems. We discuss these bounds in the next section.

3.1. Quality of the Solution

Kleywegt et al. [17] provide performance bounds on the quality of the solution obtained by the SAA method when applied to discrete stochastic optimization problems. In this section, we follow and extend their argument to provide performance bounds on the quality of the solution produced by the inexact SAA approach.

Given a first-stage feasible solution $\mathbf{p} \in P$, we have to evaluate the quality of this solution viewed as a candidate for solving the true stochastic problem (i.e., taking into account all possible scenarios). As the solution $\mathbf{p}$ is feasible, we clearly have that $g(\mathbf{p}) \geq u^*$, where $u^* = \min_{\mathbf{p}\in P} g(\mathbf{p})$ is the optimal value of the stochastic problem, and $g(\mathbf{p})$ is the true stochastic objective function. The quality of $\mathbf{p}$ can be measured by the optimality gap

$$\text{gap}(\mathbf{p}) := g(\mathbf{p}) - u^*. \tag{6}$$

A confidence interval on the true value of $g(\mathbf{p})$ can be estimated by Monte Carlo sampling; that is, an independent and identically distributed random sample $\omega^j, j = 1,\ldots,N'$, of ω is generated and $g(\mathbf{p})$ is estimated by the corresponding sample average $\bar{g}_{N'}(\mathbf{p}) = \mathbf{c}^T\mathbf{p} + \bar{Q}_{N'}(\mathbf{p})$, where $\mathbf{c}^T\mathbf{p}$ is the cost of the first-stage decisions, and $\bar{Q}_{N'}(\mathbf{p})$ is the average of

the second-stage problem over the sampled scenarios. At the same time the sample variance

$$\sigma^2_{N'}(\mathbf{p}) := \frac{1}{N'(N'-1)} \sum_{j=1}^{N'} [Q(\mathbf{p}, \omega^j) - \bar{Q}_{N'}(\mathbf{p})]^2 \quad (7)$$

of $\bar{Q}_{N'}(\mathbf{p})$ (and thus $\bar{g}_{N'}(\mathbf{p})$) is calculated. Then we can calculate an approximate $100(1-\alpha)\%$ confidence upper bound for $g(\mathbf{p})$ by

$$U_{N'}(\mathbf{p}) := \bar{g}_{N'}(\mathbf{p}) + z_\alpha \sigma_{N'}(\mathbf{p}). \quad (8)$$

This bound is justified by the Central Limit Theorem with the critical value $z_\alpha = \Phi^{-1}(1-\alpha)$, where $\Phi(z)$ is the cumulative distribution function of the standard normal distribution.

To calculate a lower bound for u^* we proceed as follows. Denote by u^N_{LB} the lower bound yielded by the approximate solution procedure to the sample average problem based on a sample of size N. Note that u^N_{LB} is a function of the (random) sample and hence is random. To obtain a lower bound for u^* observe that $\mathbb{E}[\bar{g}_N(\mathbf{p})] = g(\mathbf{p})$, that is, the sample average $\bar{g}_N(\mathbf{p})$ is an unbiased estimator of the expectation $g(\mathbf{p})$. We also have that for any $\mathbf{p} \in P$ the inequality $\bar{g}_N(\mathbf{p}) \geq \inf_{\mathbf{p}' \in P} \bar{g}_N(\mathbf{p}') \geq u^N_{LB}$ holds, so for any $\mathbf{p} \in P$, we have

$$g(\mathbf{p}) = \mathbb{E}[\bar{g}_N(\mathbf{p})] \geq \mathbb{E}[\inf_{\mathbf{p}' \in P} \bar{g}_N(\mathbf{p}')] \geq \mathbb{E}[u^N_{LB}]. \quad (9)$$

By taking the minimum over $\mathbf{p} \in P$ of the left-hand side of the above inequality, we obtain $u^* \geq \mathbb{E}[u^N_{LB}]$.

We can estimate $\mathbb{E}[u^N_{LB}]$ by solving the sample average problems several times and averaging the lower bounds calculated by the approximate solution procedure. Suppose we generate R independent sample average problems, each with N scenarios, and obtain a lower bound for each problem. Let $u^{N1}_{\mathrm{LB}}, u^{N2}_{\mathrm{LB}}, \ldots, u^{NR}_{\mathrm{LB}}$ be the computed lower bound values for these sample average problems. Then,

$$\bar{u}^{N,R}_{\mathrm{LB}} := \frac{1}{R} \sum_{j=1}^{R} u^{Nj}_{\mathrm{LB}} \quad (10)$$

is an unbiased estimator of $\mathbb{E}[u^N_{\mathrm{LB}}]$. As the samples, and hence $u^{N1}_{\mathrm{LB}}, u^{N2}_{\mathrm{LB}}, \ldots, u^{NR}_{\mathrm{LB}}$, are independent, we can estimate the variance of $\bar{u}^{N,R}_{LB}$ by

$$\sigma^2_{NR} := \frac{1}{R(R-1)} \sum_{j=1}^{R} \left(u^{Nj}_{\mathrm{LB}} - \bar{u}^{N,R}_{\mathrm{LB}} \right)^2. \quad (11)$$

A confidence $100(1-\alpha)\%$ lower bound for $\mathbb{E}[u^N_{\mathrm{LB}}]$ is then given by

$$L_{N,R} := \bar{u}^{N,R}_{\mathrm{LB}} - t_{\alpha,\nu} \sigma_{NR}, \quad (12)$$

where $\nu = R - 1$ and $t_{\alpha,\nu}$ is the α-critical value of the t-distribution with ν degrees of freedom. As $u^* \geq \mathbb{E}[u^N_{LB}]$, we have that $L_{N,R}$ gives a valid statistical lower bound for u^* as well. Consequently,

$$\hat{\mathrm{gap}}(\mathbf{p}) := U_{N'}(\mathbf{p}) - L_{N,R} \quad (13)$$

gives a statistically valid (with confidence at least 1–2 α) bound on the true gap($\mathbf{p}$). Alternatively, we can express this gap as a percentage by

$$\hat{\mathrm{gap}}(\mathbf{p})[\%] := \frac{U_{N'}(\mathbf{p}) - L_{N,R}}{L_{N,R}} \times 100[\%], \quad (14)$$

with the following interpretation: the heuristic solution is within $\hat{\mathrm{gap}}(\mathbf{p})\%$ from the true optimal solution with 1–2 α confidence. It should be noted that the lower bound $L_{N,R}$ is somewhat conservative and depends on the quality of the lower bounding mechanism.

This result is particularly important for problems such as the ConFL problem that are extremely challenging to solve to optimality (see [24, 25]). It should be clear that $\hat{\mathrm{gap}}(\mathbf{p})\%$ for the inexact SAA will depend on the quality of the both the upper and lower bounds obtained by the heuristic for the sample average problems, the sample average problem size (N), and the variability in the solution values. This suggests that large two-stage stochastic integer programming problems where *good heuristics* and *tight lower bounding mechanisms* are at hand may be good candidates for the inexact SAA approach.

4. DETERMINISTIC EQUIVALENT FORMULATION FOR THE SCONFL PROBLEM

In this section provide an integer programming formulation for the SConFL problem. Next, when the uncertainty is solely due to customer demands, we show that the SConFL problem is equivalent to a deterministic ConFL problem where each customer's demand is equal to its average demand in the SConFL problem. When there is location uncertainty, we show that the finite scenario case can be transformed into a larger deterministic ConFL problem where each demand node is replicated as many times as the number of scenarios in the problem.

We first define a cutset formulation for the deterministic ConFL (i.e., a ConFL problem with known demand quantities and assignment costs). 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 (15a) has three terms: the facility opening cost, the core tree cost and the assignment cost. Constraints (15b)–(15d) impose the condition that the open facilities are connected by a Steiner tree, while constraints (15e) and (15f) ensure that each demand node is assigned to an open facility.

Cutset formulation for the Deterministic ConFL problem:

$$\text{Minimize} \sum_{i\in F} f_i z_i + \sum_{\{i,j\}\in E(F\cup S)} b_{ij} y_{ij} + \sum_{j\in D}\sum_{i\in n(j)} a_{ij} x_{ij} \tag{15a}$$

$$\text{subject to} \sum_{\{i,j\}\in E(R)} y_{ij} \le \sum_{l\in R\setminus k} z_l, \forall R \subset (S\cup F), |R| \ge 3, \quad \forall k \in R \tag{15b}$$

$$y_{ij} \le z_i, y_{ij} \le z_j, \forall \{i,j\} \in E(S\cup F) \tag{15c}$$

$$\sum_{\{i,j\}\in E(S\cup F)} y_{ij} = \sum_{l\in(S\cup F)} z_l - 1 \tag{15d}$$

$$\sum_{i\in n(j)} x_{ij} = 1, \forall j \in D \tag{15e}$$

$$x_{ij} \le z_i, \forall j \in D, \forall i \in n(j) \tag{15f}$$

$$x_{ij} \in \{0,1\}, \forall j \in D, \forall i \in n(j) \tag{15g}$$

$$y_{ij} \in \{0,1\}, \forall \{i,j\} \in E(S\cup F) \tag{15h}$$

$$z_l \in \{0,1\}, \forall l \in S\cup F. \tag{15i}$$

In the stochastic version of the ConFL problem, assignment costs, a_{ij}, are uncertain and depend on the realization of a random variable ω or scenario. Then, $a_{ij}(\omega)$ represents the assignment cost under scenario ω, and π_ω is the probability of occurrence of scenario ω.

In the SConFL problem the first-stage decisions are the set of open facilities, $\mathbf{z}$, and the Steiner tree that connects them, $\mathbf{y}$ (i.e., the first stage decision variables $\mathbf{p}$ in the generic formulation (1) are $(\mathbf{z},\mathbf{y})$ in the SConFL problem), and the second-stage decisions involve the allocation of customers to open facilities, $\mathbf{x}$. Given that the second-stage decisions only depend on the open facilities (i.e., $\mathbf{z}$) and not on the tree that connects them (i.e., $\mathbf{y}$), we denote the second stage objective by $Q(\mathbf{z},\omega)$ and the feasible discrete set of solutions to the second stage by $X(\mathbf{z},\omega)$. The analogous cutset formulation of the SConFL problem as a two-stage stochastic program with fixed recourse is described below.

Cutset formulation for the Stochastic ConFL problem:

$$\text{Minimize} \sum_{i\in F} f_i z_i + \sum_{\{i,j\}\in E(F\cup S)} b_{ij} y_{ij} + \mathbb{E}_\omega(Q(\mathbf{z},\omega)) \tag{16}$$

subject to (15b), (15c), (15d), (15h), and (15i).

In the stochastic version of the ConFL problem, the assignment cost is unknown in the first stage and hence we replace the third term in the objective function (15a) by its expected value, (i.e., the expected value of the second stage decision problem), yielding (16). In other words, the assignment decision for each demand node is determined in the second stage when the recourse minimization problem (17) is solved.

$$Q(\mathbf{z},\omega) = \text{Minimize } q(\mathbf{z},\omega)(\mathbf{x}) = \sum_{j\in D}\sum_{i\in n(j)} a_{ij}(\omega) x_{ij} \tag{17}$$

subject to (15e), (15f), and (15g).

Clearly, once open facilities are defined in the first stage, the recourse problem reduces to an assignment problem where demand nodes are assigned to the open facility with lowest assignment cost. To determine the best open facility for each demand node we must wait until assignment costs are realized; consequently, the solution to the assignment problem may vary for each scenario. Note, however, the feasible region $X(\mathbf{z},\omega)$ only depends on $\mathbf{z}$ and not on ω. Consequently, we remove the dependence on ω and denote the feasible discrete set of solutions to the second stage by $X(\mathbf{z})$.

4.1. SConFL Problem with Uncertain Demands

In the case, where uncertainty on assignment costs is due to unknown demand quantities, we assume that the per unit assignment cost (denoted by k_{ij}) is fixed and known beforehand. Here, when scenario ω is unveiled, we mean that the demand quantity $d_j(\omega)$ is discovered for each demand node j, and hence the assignment cost $a_{ij}(\omega) = k_{ij} d_j(\omega)$ is revealed. In this setting, the assignment costs are the only random input parameter in the problem. For this specific realization of events, we show that the value of the stochastic solution is null. In other words, the optimal solution for this stochastic ConFL is the optimal solution of a deterministic ConFL problem when average demands are assumed.

Theorem 4.1. *The optimal solution of the stochastic ConFL with uncertain demands is equal to the optimal solution of the deterministic ConFL obtained by replacing all random variables by their expected values.*

To prove Theorem 4.1, we use the following two lemmas.

Lemma 4.2. *Given facility locations* $\mathbf{z}$ *from the first stage, the optimal allocation solution,* $\mathbf{x}^*$*, to the recourse problem,* $Q(\mathbf{z},\omega)$*, for the SConFL with uncertain demands is invariant to demand realizations.*

Proof of Lemma 4.2. Let $\mathbf{x}^* \in X(\mathbf{z})$ be an optimal solution for $Q(\mathbf{z},\tilde{\omega})$ for some $\tilde{\omega} \in \Omega$. Then,

$$\begin{aligned} &q(\mathbf{z},\tilde{\omega})(\mathbf{x}^*) \\ &= \sum_{j\in D}\left(\sum_{i\in n(j)} a_{ij}(\tilde{\omega}) x^*_{ij}\right) = \sum_{j\in D}\left(\sum_{i\in n(j)} k_{ij} d_j(\tilde{\omega}) x^*_{ij}\right) \\ &= \sum_{j\in D} d_j(\tilde{\omega})\left(\sum_{i\in n(j)} k_{ij} x^*_{ij}\right) \le \sum_{j\in D} d_j(\tilde{\omega})\left(\sum_{i\in n(j)} k_{ij} x_{ij}\right), \\ &\quad \forall \mathbf{x} \in X(\mathbf{z}). \end{aligned}$$

In vector notation where $\mathbf{k}_j = \{k_{1j}, k_{2j}, \ldots, k_{|n(j)|j}\}$ and $\mathbf{x}_j = \{x_{1j}, x_{2j}, \ldots, x_{|n(j)|j}\}$,

$$\sum_{j\in D} d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}} \mathbf{x}^*_j) \le \sum_{j\in D} d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}} \mathbf{x}_j), \forall \mathbf{x} \in X(\mathbf{z}). \tag{18}$$

In fact, inequality (18) implies that the inequality holds not only for the summation, but also for each individual term in it. If there were a demand node $j \in D$, such that $d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^*) > d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j')$ for some $\mathbf{x}_j'$, then we could replace j's assignment in $\mathbf{x}^*$ and obtain a lower objective function. Consequently, $d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^*) \leq d_j(\tilde{\omega})(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j), \forall j \in D, \forall \mathbf{x} \in X(\mathbf{z})$; which implies $\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^* \leq \mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j, \forall j \in D, \forall \mathbf{x} \in X(\mathbf{z})$.

As $\mathbf{x}^* \in X(\mathbf{z})$, it is a feasible solution for $Q(\mathbf{z}, \omega), \forall \omega \in \Omega$. Now, assume that $\mathbf{x}^*$ is not an optimal solution to $Q(\mathbf{z}, \omega)$ for some $\omega \in \Omega$. Then, there exists an $\mathbf{x}' \neq \mathbf{x}^* \in X(\mathbf{z})$ such that $q(\mathbf{z}, \omega)(\mathbf{x}') < q(\mathbf{z}, \omega)(\mathbf{x}^*)$ for some $\omega \in \Omega$.

$$\begin{aligned} q(\mathbf{z},\omega)(\mathbf{x}') = \textstyle\sum_{j\in D} d_j(\omega)(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j') &< \textstyle\sum_{j\in D} d_j(\omega)(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^*) \\ \Rightarrow d_j(\omega)(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j') &< d_j(\omega)(\mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^*), \exists j \in D \\ \Rightarrow \mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j' &< \mathbf{k}_j^{\mathrm{T}}\mathbf{x}_j^*, \exists j \in D \Rightarrow\Leftarrow \end{aligned}$$

This yields a contradiction equation proving Lemma 4.2. ∎

Lemma 4.3. *Given a first-stage decision,* $\mathbf{z}$*, the expected value of the recourse program,* $Q(\mathbf{z}, \omega)$*, for the SConFL with uncertain demands equals the objective function value of the recourse program with expected demands.*

Proof of Lemma 4.3. $\mathbb{E}_\omega(Q(\mathbf{z}, \omega)) = \sum_{\omega\in\Omega} \pi_\omega Q(\mathbf{z}, \omega) = \sum_{\omega\in\Omega} \pi_\omega \min_{\mathbf{x}\in X(\mathbf{z})} \sum_{j\in D} \sum_{i\in n(j)} k_{ij} d_j(\omega) x_{ij}$. By Lemma 4.2, we can take the minimization outside the first summation. Moreover, we can rearrange the order of summations. Thus,

$$\begin{aligned} \mathbb{E}_\omega(Q(\mathbf{z},\omega)) &= \min_{\mathbf{x}\in X(\mathbf{z})} \sum_{\omega\in\Omega} \pi_\omega \sum_{j\in D} \sum_{i\in n(j)} k_{ij} d_j(\omega) x_{ij} \\ &= \min_{\mathbf{x}\in X(\mathbf{z})} \sum_{j\in D} \sum_{i\in n(j)} k_{ij} \left(\sum_{\omega\in\Omega} \pi_\omega d_j(\omega) \right) x_{ij} \\ &= \min_{\mathbf{x}\in X(\mathbf{z})} \sum_{j\in D} \sum_{i\in n(j)} k_{ij} \mathbb{E}_\omega(d_j) x_{ij}. \end{aligned}$$ ∎

Proof of Theorem 4.1. Theorem 4.1 now directly follows from Lemma 4.3. ∎

4.2. SConFL Problem with Uncertain Locations

When variability in assignment costs is due to uncertainty on customers' location or other factors that do not affect assignment costs proportionally, simplifying the problem by replacing the random variables by their expected value does not lead to good solutions. This is because the location of the closest open facility depends on the realized scenario. In this case, we transform the SConFL problem into a deterministic ConFL problem with multiple copies of demand nodes. Then, any high-quality method for the ConFL problem can be applied to solve the deterministic equivalent problems (when the number of scenarios is limited).

We assume there is a finite number of scenarios, $\omega \in \Omega$ with positive probability, π_ω that fully determine the entire set of possible assignment costs, $a_{ij}(\omega) j \in D$ and $\forall i \in n(j)$. That is, $a_{ij}(\omega)$ is the cost of assigning demand node j to facility node i when scenario ω is realized. Then, the SConFL problem is equivalent to a deterministic ConFL problem with $|\Omega|$ copies of each demand node—one copy for each scenario—and with assignment costs, a_{ij_ω} equal to $\pi_\omega a_{ij}(\omega)$, where j_ω is the copy of demand node $j \in D$ for scenario $\omega \in \Omega$ and $i \in n(j)$.[2]

Deterministic equivalent formulation for the Stochastic ConFL problem:

$$\text{Minimize} \sum_{i\in F} f_i z_i + \sum_{\{i,j\}\in E(F\cup S)} b_{ij} y_{ij} + \sum_{\omega\in\Omega} \sum_{j\in D} \sum_{i\in n(j)} \pi_\omega a_{ij}(\omega) x_{ij}(\omega) \tag{19a}$$

$$\text{subject to} \sum_{i\in n(j)} x_{ij}(\omega) = 1, \forall j \in D, \forall \omega \in \Omega \tag{19b}$$

$$x_{ij}(\omega) \leq z_i, \forall j \in D, \forall i \in n(j), \forall \omega \in \Omega \tag{19c}$$

$$x_{ij}(\omega) \in \{0,1\}, \forall j \in D, \forall i \in n(j) \tag{19d}$$

$$\text{and (15b), (15c), (15d), (15h), (15i).}$$

We should note that although, we have a deterministic equivalent problem in hand, its size grows rapidly with the number of possible scenarios. For example, a problem with only two facilities, $|F| = 2$, and three demand nodes, $|D| = 3$, where any demand node can be assigned to any facility node with an uncertain assignment cost, a_{ij}, that can be either low or high would have a total of 64 scenarios to consider, $|2|^{|D|^{|F|}}$. When the location of demand nodes are independent from each other, the number of scenarios increases rapidly and to simply solve the deterministic equivalent problem with multiple demand node copies is impractical and computationally infeasible. The inexact SAA presented in Section 3 addresses this situation (and ones in which the number of scenarios are exponential or uncountable) by solving approximately the sample average problems. In particular, we approximately solve the sample average problems using Bardossy and Raghavan [2]'s DLS heuristic. The DLS heuristic also provides a lower bound on the optimal solution value (in this case for the sample average problem). The results in Bardossy and Raghavan [2] suggest that the DLS heuristic provides extremely tight bounds and runs very rapidly. These two indicators suggest it might work well in the inexact SAA approach.

5. COMPUTATIONAL EXPERIMENTS

In this section, we apply the inexact SAA method. First, we consider the SConFL problem (the motivating application

[2] This strategy of transforming a stochastic combinatorial problem into a deterministic equivalent problem on a larger graph (where one copy of a node is created for each scenario) has also been applied in other stochastic network design problems; see Bomze et al. [6], Correia and Saldanha da Gama [7], Kurz et al. [19], Louveaux and Peeters [26].

for this approach). In addition, we consider the SUFL problem. All our computational experiments are conducted on a Windows 7 machine with an Intel Core i7-3770 processor with a speed of 3.40 GHz and with 32 gigabytes of RAM.

5.1. Problem Generation and Characteristics

We first describe how we generate instances for our test problems and their corresponding sample average problems. We use a similar approach to generate both the SConFL and SUFL problem instances. The only difference is that in the SUFL problem instances we do not generate any Steiner nodes. Our ConFL problem instances follow the more common convention that facility nodes that do not serve customers do not incur a facility opening cost.

We start by generating a 100×100 square grid. The location of each demand, facility, and Steiner node (in the SConFL problem instances) is selected randomly on the grid. Furthermore, to represent the uncertainty in the assignment costs we assume that the exact location of each demand node is uncertain and in a random box around its coordinate $\mathbf{j} = (j_1, j_2)$ on the grid. This is defined by an error term, $\mathbf{e} = (\varepsilon_{j1}, \varepsilon_{j2})$, drawn from a discrete uniform distribution according to a given variability, v. In our first set of instances, v ranges from 5 to 20 in steps of 5; that is, if $v = 5$ then ε_{j1} and ε_{j2} are uniformly distributed between -5 and 5. To generate the sample average problems, we generate scenarios where the exact location of the demand nodes is randomly determined within its box-uncertainty region.

The Euclidean distances rounded up to their nearest integer values 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 (in the SConFL problem) 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 of edges in the tree T. The number of demand nodes and facility nodes vary between 10 and 90 in steps of 10, with the total number of demand and facility nodes equal to 100. The number of Steiner nodes is 20 for all SConFL problem instances. The facility opening costs are equal to 30 and the same for all the facility nodes. These problem parameters cover a wide range of characteristics and were specifically chosen to include the hardest types of ConFL problem instances (for the DLS heuristic) reported in Bardossy and Raghavan [2].

5.2. Parameter Selection for the Inexact SAA Approach

Most of our experiments are on the SConFL problem, and we use it to select the parameters N, R, and N' in the inexact SAA approach. There is a trade-off between sample size N (i.e., number of scenarios per sample average problem), the number of replications R (i.e., number of sample average problems) and the computational effort. The larger the sample size N, the more closely the sample problem will resemble the stochastic problem and the longer it will take to solve. Similarly, the larger the number of replications R, the lower the variance of the lower bound and the tighter the lower limit of the confidence interval. As indicated by Kleywegt et al. [17], "if the computational complexity of solving the SAA problem increases faster than linearly in the sample size N, it may be more efficient to choose a smaller sample size N and to generate and solve several SAA problems". We explore this premise in this section and find the complexity of solving larger ConFL problem increases faster than linearly. A third parameter of the SAA method is the number of scenarios N' used to test each of the solutions generated by solving the sample average problems. In this case, a larger number of scenarios N' decreases the sample variance of the upper bound which can yield a tighter upper limit for the confidence interval.

TABLE 1. Average 99% confidence interval percentage gaps and computational times (in seconds) while varying R and keeping N fixed at 20

		R						
M		10	15	20	25	30	35	40
3	Gap	3.33%	3.21%	3.12%	3.09%	3.08%	3.06%	3.05%
	Time (s)	66.17	99.27	132.42	165.52	198.65	231.78	264.68
5	Gap	4.09%	3.97%	3.90%	3.84%	3.66%	3.55%	3.47%
	Time (s)	81.59	122.59	163.32	203.94	244.64	285.29	325.87
7	Gap	3.40%	3.23%	3.15%	3.11%	2.99%	2.89%	2.88%
	Time (s)	99.82	149.79	199.71	249.35	299.61	349.54	399.72

The number of replications, R, enters into the computation of the lower bound. To improve the lower bound defined by Equation (12), one must decrease the sample variance of $\mathbb{E}[u_{LB}^N]$ either increasing the number of scenarios, N, for each sample average problem, or the number of replications, R. Tables 1 and 2 show average 99%-confidence interval gaps (as a percentage) and computational times for various sample sizes, N, and number of replications, R, respectively. In Table 1, R is varied while N is kept fixed. In these instances, there are 50 demand nodes, 50 facility nodes, and has variability up to ± 10 in demand node coordinates. Each entry in the table is the average of ten instances. For example, when $M = 3$ and $R = 10$ (with N fixed at 20) the average 99% confidence interval percentage gap is 3.33% and the average running time is 66.17 seconds. Table 1 shows that the computational time increases linearly while the average gap decreases slowly with the number of replications. Interestingly, while increasing the number of replications does not improve the quality of the upper bound, it does decrease the lower bound sample variance and consequently improves the lower bound. The steady improvement in the gap for additional replications has a clear computational cost and considering the trade-off we selected the number of replications R to 20 for our computational experiments.

In Table 2, N is varied while R is kept fixed. It shows that the computational time grows faster than a linear function of sample size N; while the effect of the sample size N on the gaps is somewhat unpredictable. The gaps seem to decrease

TABLE 2. Average 99% confidence interval percentage gaps and computational times (in seconds) while varying N and keeping R fixed at 20

		N						
M		10	15	20	25	30	35	40
3	Gap	3.51%	3.41%	3.12%	3.19%	3.19%	3.13%	3.08%
	Time (s)	57.70	92.59	132.59	186.26	232.65	295.89	355.08
5	Gap	3.71%	3.72%	3.66%	3.65%	3.77%	3.70%	3.81%
	Time (s)	65.63	106.63	162.55	235.02	295.22	380.32	469.81
7	Gap	3.13%	3.01%	3.00%	3.44%	3.31%	3.30%	3.24%
	Time (s)	77.89	128.20	199.81	287.79	372.80	484.11	606.35

when N is increased from 10 to 15 to 20 but for larger sample sizes the gaps increase and decrease without a clear pattern. Larger sample problems did not yield significantly better candidate solutions; consequently, for our experiments we settled with $N = 20$.

In summary, the gap improvements from additional replications was greater than the one obtained by increasing the number of scenarios per sample average problem. Given a budget of computational time, these results indicate that increasing the number of replications might be the preferred strategy to produce tighter confidence interval around the true optimal cost and improve the quality of the lower bound. This behavior was representative of the entire set of problems tested; hence, we set $N = 20$ and $R = 20$ for our remaining computational experiments.

Earlier, we mentioned that we can use a large number of scenarios to assess the quality of a solution and calculate an approximate $100(1-\alpha)\%$ confidence upper bound by using equation (8). Clearly, the sample variance is one of the key factors that determines the width of such a bound. Figure 2a shows how the sample variance of $\bar{Q}_{N'}(\mathbf{z})$ changes as the number of scenarios, N', increases. This figure corresponds to one instance of the problem studied in Tables 1 and 2 with $N = 20$, $R = 20$, and $M = 3$. (Notice, although there were 20 replications only 11 unique solutions were obtained.) The sample variance decreases abruptly at the beginning but later it starts to level off around the 2000 scenarios. Sample variance $\bar{Q}_{N'}(\mathbf{z})$ only affects the upper limit of the confidence interval (which in turn affects the confidence interval percentage gap). Figure 2b shows the change in the 99%-confidence interval percentage gap as the number of scenarios increases. The gap decreases abruptly up to 1500 scenarios and then stays stable within 0.05%. Although the variance decreases as the number of scenarios, N', increases; there is variability in the total cost of the solution (which together with the sample variance affect the value of the upper limit of the confidence interval). Thus the 99%-confidence interval percentage gap does not necessarily decrease and stays relatively stable in our computational experiments. Based on these observations, we decided to fix $N' = 2000$ in our computational experiments. Evaluating more scenarios would only increase computational time with practically no gain in terms of the quality of the bound. Once again, these results were representative of the entire set of problem instances.

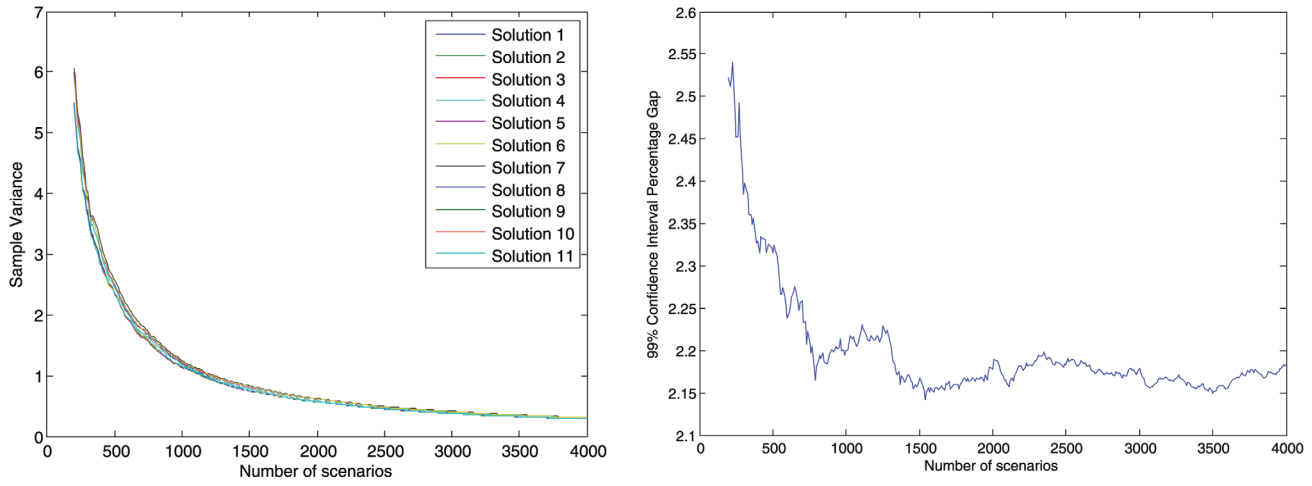
5.3. SConFL Problem Results

Tables 3 and 4 show our computational results for the SConFL problem using the inexact SAA method and the DLS heuristic. Each entry in Table 3 shows the average gap and average time over 10 instances, while Table 4 shows the maximum gap and maximum computational time for those 10 instances. The values reported are average 99% confidence interval percentage gaps. That is, with 99% confidence the optimal value of the true stochastic problem is $\hat{gap}(\mathbf{z})\%$ from the (upper bound) solution obtained by the inexact SAA method. Overall, these gaps follow the behavior observed for the deterministic instances in Bardossy and Raghavan [2]. Lower gaps are observed for either high proportions of demand nodes or facility nodes. On the contrary, higher gaps are observed for balanced instances with similar numbers of demand nodes and facility nodes. Furthermore, these confidence interval gaps increase for higher levels of uncertainty. We can provide two explanations for this behavior. First, as the uncertainty level increases, the optimality gaps (i.e., percentage gaps between the upper and lower bounds) obtained by DLS increase; we observe this behavior for the individual sample average problems. Second, as the uncertainty level increases, the sample variance increases as well, and the width of the confidence interval increases. The magnitude of the 99% confidence interval percentage gaps are quite reasonable. Their average value ranges between 0.28% to 4.27%; while the maximum value is 7.17%.

In terms of computational times, we observe that for all values of M and v as the proportion of demand nodes increases the computational time increases reaching a peak at 30 demand nodes and 70 facility nodes and then decreases again. Overall, computational times increase as the M factor increases. The behavior of the DLS heuristic and the inexact SAA method is quite stable as the maximum running times are close to the averages; indicating that the run time across the 10 instances does not exhibit much variability. The maximum running time across the 1080 instances is 282.27 seconds (less than 5 min).

While the gaps of the inexact SAA method applied to the SConFL problem are quite reasonable, natural questions arise. How much have we gained in terms of running time (and thus our ability to solve large-scale problems)? How much have we lost in terms of the gaps? To assess this issue, we obtained a state-of-the-art code capable of solving large ConFL instances to optimality [23]. We used it within the SAA method and compared its results to those obtained by the inexact SAA method.

To conduct this comparison, we considered a subset of our test instances with $v = 10$. Table 5 summarizes the results of this comparison. It shows that the (exact) SAA method yields solutions with average 99% confidence interval percentage gaps less than 1%; and maximum 99% confidence interval percentage gaps less than 1.39%. Conversely, the



(a) Sample variance of $\bar{Q}_{N'}(\mathbf{z})$ (b) 99% Confidence Interval Percentage Gap

FIG. 2. Effect of increasing number of scenarios, N'. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3. Average 99% confidence interval percentage gaps and times (in seconds) for the Inexact SAA method applied to the SConFL problem

			v							
			5		10		15		20	
M	$\|D\|$	$\|F\|$	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)
	10	90	0.54%	108.58	0.98%	110.06	1.06%	111.94	1.87%	113.50
	20	80	2.40%	159.55	2.66%	161.28	2.60%	163.29	3.29%	168.71
	30	70	2.63%	169.52	2.67%	172.60	3.55%	176.26	3.90%	178.55
	40	60	2.34%	155.47	2.94%	158.19	3.55%	160.46	4.14%	163.49
3	50	50	2.86%	130.18	3.27%	133.14	3.42%	136.77	3.85%	138.73
	60	40	2.86%	98.36	3.00%	99.43	3.11%	101.46	3.33%	103.94
	70	30	1.77%	69.30	1.96%	70.46	2.14%	71.54	2.53%	72.56
	80	20	0.73%	50.86	1.07%	48.64	1.32%	48.73	1.61%	48.55
	90	10	0.28%	38.60	0.55%	39.31	0.74%	38.52	0.76%	39.26
	10	90	0.41%	141.15	0.94%	142.48	1.38%	144.23	1.54%	146.30
	20	80	1.41%	190.87	1.64%	193.96	1.86%	197.17	1.93%	200.34
	30	70	2.22%	206.35	2.33%	209.70	2.72%	215.23	3.05%	220.36
	40	60	3.19%	196.05	3.28%	198.73	3.42%	202.38	3.78%	204.06
5	50	50	3.35%	159.62	3.75%	162.36	3.75%	165.55	4.11%	168.76
	60	40	2.69%	123.41	2.86%	123.47	3.07%	124.33	3.37%	126.47
	70	30	2.20%	87.94	2.43%	88.36	2.59%	88.93	2.80%	89.75
	80	20	1.22%	61.42	1.65%	60.88	2.05%	60.20	1.99%	60.15
	90	10	0.44%	40.31	0.63%	41.21	0.82%	41.39	0.93%	42.20
	10	90	0.46%	170.67	0.85%	171.63	1.40%	173.23	1.54%	174.98
	20	80	0.33%	235.50	0.77%	239.51	0.88%	242.81	1.28%	245.90
	30	70	1.57%	245.71	1.60%	249.34	2.00%	254.21	2.34%	258.37
	40	60	2.54%	230.22	2.68%	234.37	2.95%	239.18	3.40%	242.39
7	50	50	3.16%	196.59	3.01%	200.92	3.41%	203.84	3.59%	207.51
	60	40	3.43%	152.18	3.75%	152.18	3.99%	155.16	4.27%	157.55
	70	30	2.82%	109.93	2.89%	108.65	2.93%	110.98	3.11%	113.08
	80	20	1.41%	72.79	1.54%	72.31	1.77%	73.84	1.98%	76.02
	90	10	0.56%	40.75	0.76%	41.21	0.90%	41.05	1.03%	41.16

inexact SAA method shows average 99% confidence interval percentage gaps up to 3.75%, and maximum 99% confidence interval percentage gaps up to 6.00% (achieved when $v = 10$, $M = 7$, $|D| = 50$, $|F| = 50$). Regarding the computational times, the (exact) SAA method seems to be very sensitive to the input data and times vary greatly between instances. The computational times for the SAA method average 16 min over the 270 test instances, with some instances taking as long as 2 h. Conversely, the computational times for the inexact SAA method are quite robust and average 2 min over the

TABLE 4. Maximum 99% confidence interval percentage gaps and times (in seconds) for the inexact SAA method applied to the SConFL problem

			v							
			5		10		15		20	
M	$\|D\|$	$\|F\|$	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)
	10	90	0.72%	123.99	1.39%	125.23	1.80%	126.45	2.25%	126.98
	20	80	3.62%	175.57	3.51%	176.43	3.76%	176.25	6.18%	185.48
	30	70	4.06%	183.97	4.22%	181.77	5.02%	187.54	7.17%	187.05
	40	60	3.88%	164.44	4.56%	164.06	4.83%	164.56	5.30%	169.03
3	50	50	4.70%	138.78	5.25%	142.36	5.04%	142.05	5.35%	143.02
	60	40	4.06%	105.81	3.77%	105.96	3.79%	106.86	4.54%	108.74
	70	30	4.07%	72.91	3.59%	74.31	3.57%	75.50	3.72%	77.18
	80	20	1.29%	59.49	2.01%	55.36	2.08%	50.41	2.31%	50.33
	90	10	0.58%	41.50	1.26%	41.54	1.61%	42.05	1.36%	42.45
	10	90	0.85%	166.89	1.35%	167.35	1.82%	167.61	2.97%	167.59
	20	80	2.82%	213.59	2.82%	215.98	3.26%	216.74	3.15%	220.24
	30	70	5.13%	223.04	4.56%	228.44	4.95%	232.74	6.04%	239.52
	40	60	5.26%	207.57	5.03%	210.47	4.69%	213.76	6.12%	215.83
5	50	50	5.34%	167.25	5.36%	175.63	6.07%	178.28	6.37%	178.56
	60	40	4.91%	136.76	4.91%	134.48	5.13%	132.29	4.61%	131.17
	70	30	3.73%	98.38	3.81%	97.99	4.23%	98.58	4.64%	100.30
	80	20	2.64%	70.19	3.64%	71.75	3.78%	69.38	2.94%	73.35
	90	10	0.92%	46.63	1.16%	46.54	1.49%	45.27	1.42%	46.00
	10	90	0.75%	185.59	1.26%	186.54	2.00%	187.31	2.25%	188.28
	20	80	0.47%	270.62	1.04%	273.41	1.18%	276.95	1.75%	278.24
	30	70	2.70%	265.43	3.25%	271.51	4.10%	277.62	4.29%	282.27
	40	60	4.95%	244.97	5.55%	251.48	5.68%	260.06	6.69%	266.13
7	50	50	6.61%	212.78	6.00%	213.30	6.66%	211.80	6.80%	213.41
	60	40	5.76%	159.72	5.81%	160.57	6.44%	164.32	6.09%	167.45
	70	30	4.69%	117.20	3.97%	117.20	3.93%	120.05	4.23%	123.52
	80	20	2.13%	80.40	3.17%	79.67	3.63%	84.60	3.39%	85.64
	90	10	1.09%	50.84	1.99%	50.90	1.94%	50.76	2.32%	51.01

270 test instances, with no instance taking more than 5 min to solve. In general problems with fewer candidate facility nodes are faster to solve with both the SAA and inexact SAA method.

We wanted to get a better sense of whether the greater gap with the inexact SAA method was due to the quality of the upper or lower bound. To this end, we evaluated the upper end of the confidence interval generated by the inexact SAA method against the lower end of the confidence interval generated by the (exact) SAA method. This is shown in the last two columns of Table 5. This comparison shows gaps closer to the SAA method, indicating that the DLS produces high-quality solutions, while the lower bounds may not be as tight. In fact, on average the upper end of the confidence interval for the inexact SAA method is 0.30% greater than the upper end of the confidence interval for the SAA method; while the lower end of the confidence interval of the inexact SAA method is 1.22% lower than the lower end of the confidence interval for the SAA method.

5.4. *SUFL Problem Results*

The SUFL problem can be viewed as a special case of the SConFL problem without the requirement that open facilities have to be connected. Correia and Saldanha da Gama [7] state the SUFL problem and provide a deterministic equivalent formulation which we arrive at from Formulation (19) by eliminating the y_{ij} variables.

Deterministic equivalent formulation for the SUFL problem:

$$\text{Minimize} \sum_{i\in F} f_i z_i + \sum_{\omega\in\Omega}\sum_{j\in D}\sum_{i\in n(j)} \pi_\omega a_{ij}(\omega) x_{ij}(\omega) \qquad (20)$$

subject to (19b), (19c), (19d) and (15i).

We apply the inexact SAA method to the SUFL problem. To get a heuristic solution and lower bound for each sample average problem we used dual-ascent. We converted the UFL problem to a directed Steiner tree problem and applied our dual-ascent algorithm. Table 6 describes these results. The table reports on the average 99% confidence interval percentage gap and the average computational time (averaged over ten instances). The results of the inexact SAA approach are promising. Over 460 problem instances it generates solutions that are on average 2.63% from the optimal (with 99% confidence) taking an average of 8.7 s

TABLE 5. Comparing the quality of solutions and computational time for the Inexact SAA Method and (Exact) SAA Method applied to the SConFL problem (on instances with $v = 10$)

			SAA (Leitner et al. [23])				Inexact SAA (DLS)					
			Gap		Time (s)		Gap		Time (s)		Gap to SAA Lower Confidence Limit	
M	$\|D\|$	$\|F\|$	Average	Max	Average	Max	Average	Max	Average	Max	Average	Max
	10	90	0.98%	1.39%	888.60	1692.59	0.98%	1.39%	110.06	125.23	0.98%	1.39%
	20	80	0.73%	1.06%	156.23	296.15	2.66%	3.51%	161.28	176.43	1.34%	1.97%
	30	70	0.59%	0.92%	1186.81	2934.26	2.67%	4.22%	172.60	181.77	0.92%	1.50%
	40	60	0.58%	1.02%	1662.28	4459.16	2.94%	4.56%	158.19	164.06	1.29%	2.86%
3	50	50	0.61%	0.85%	3235.62	6692.01	3.27%	5.25%	133.14	142.36	1.05%	1.90%
	60	40	0.56%	0.70%	2086.72	4533.44	3.00%	3.77%	99.43	105.96	1.06%	1.71%
	70	30	0.48%	0.68%	717.26	1616.15	1.96%	3.59%	70.46	74.31	0.71%	1.13%
	80	20	0.43%	0.63%	171.52	343.16	1.07%	2.01%	48.64	55.36	0.52%	0.98%
	90	10	0.39%	0.65%	69.77	107.59	0.55%	1.26%	39.31	41.54	0.45%	0.98%
	10	90	0.94%	1.35%	1308.83	1647.14	0.94%	1.35%	142.48	167.35	0.94%	1.35%
	20	80	0.66%	1.03%	400.29	1198.11	1.64%	2.82%	193.96	215.98	0.73%	1.26%
	30	70	0.47%	0.61%	211.48	441.17	2.33%	4.56%	209.70	228.44	0.60%	1.01%
	40	60	0.46%	0.63%	614.30	1456.21	3.28%	5.03%	198.73	210.47	1.33%	2.75%
5	50	50	0.48%	0.75%	1900.01	6188.95	3.75%	5.36%	162.36	175.63	1.00%	1.63%
	60	40	0.50%	0.72%	1765.26	4362.83	2.86%	4.91%	123.47	134.48	0.66%	1.19%
	70	30	0.44%	0.62%	808.04	2176.64	2.43%	3.81%	88.36	97.99	0.82%	1.91%
	80	20	0.41%	0.52%	226.09	505.20	1.65%	3.64%	60.88	71.75	0.78%	2.99%
	90	10	0.32%	0.47%	70.58	115.82	0.63%	1.16%	41.21	46.54	0.38%	0.65%
	10	90	0.85%	1.26%	1603.79	1872.76	0.85%	1.26%	171.63	186.54	0.85%	1.26%
	20	80	0.77%	1.04%	2069.57	2592.70	0.77%	1.04%	239.51	273.41	0.77%	1.04%
	30	70	0.46%	0.62%	396.27	1145.23	1.60%	3.25%	249.34	271.51	0.58%	1.13%
	40	60	0.48%	0.61%	310.61	483.29	2.68%	5.55%	234.37	251.48	0.94%	2.65%
7	50	50	0.39%	0.62%	751.39	4069.91	3.01%	6.00%	200.92	213.30	0.72%	1.40%
	60	40	0.31%	0.53%	1267.10	5152.87	3.75%	5.81%	152.18	160.57	1.22%	3.54%
	70	30	0.39%	0.56%	625.07	1460.82	2.89%	3.97%	108.65	117.20	1.00%	1.90%
	80	20	0.36%	0.58%	297.52	737.58	1.54%	3.17%	72.31	79.67	0.51%	1.10%
	90	10	0.30%	0.45%	83.46	154.21	0.76%	1.99%	41.21	50.90	0.42%	1.24%

TABLE 6. Average 99% confidence interval gaps and times (in seconds) for the inexact SAA method applied to the SUFL problem

		v							
		5		10		15		20	
$\|D\|$	$\|F\|$	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)
10	90	0.86%	3.28	2.34%	3.52	3.14%	4.21	5.32%	4.32
20	80	2.35%	6.23	3.38%	6.80	4.56%	8.07	6.97%	8.07
30	70	3.01%	8.29	3.44%	8.93	4.21%	10.43	6.94%	10.70
40	60	2.69%	9.81	3.26%	10.19	4.75%	11.90	5.93%	12.01
50	50	2.01%	10.34	3.18%	10.75	3.60%	11.90	4.33%	12.32
60	40	1.50%	10.16	2.07%	10.45	2.38%	11.21	3.22%	11.56
70	30	0.70%	10.04	1.23%	9.90	1.35%	10.42	1.66%	10.51
80	20	0.35%	8.09	0.68%	8.06	0.63%	8.35	0.82%	8.27
90	10	0.30%	6.01	0.44%	5.95	0.55%	6.08	0.62%	5.98

for each instance. We notice that the run times of the inexact SAA are quite stable and marginally affected by the level of uncertainty (v).

To assess the quality of the gaps provided by the inexact SAA method for the SUFL problem and to get a better sense of the speedup obtained using the inexact SAA method we also implemented the SAA method where the sample average problems are solved exactly with CPLEX. Table 7 displays these results, reporting on the average 99% confidence interval percentage gap and the average computational time (over ten instances). In contrast to the inexact SAA method, the SAA approach generates solutions that are on average 0.94% from the optimal (with 99% confidence) taking an average of 475 s for each instance.

TABLE 7. Average 99% confidence interval gaps and times (in seconds) for the SAA method applied to the SUFL problem

		v							
		5		10		15		20	
$\|D\|$	$\|F\|$	Gap	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap)	Time (s)
10	90	0.64%	197.09	1.29%	273.38	1.70%	350.06	2.29%	396.65
20	80	0.66%	487.78	1.18%	680.20	1.55%	913.97	2.16%	1109.60
30	70	0.55%	646.20	1.00%	820.65	1.28%	1165.66	1.68%	1500.63
40	60	0.48%	661.96	0.94%	843.31	1.20%	1238.24	1.53%	1391.68
50	50	0.48%	230.72	0.93%	291.78	1.13%	408.88	1.43%	503.40
60	40	0.43%	317.17	0.76%	411.47	0.93%	495.58	1.19%	587.55
70	30	0.32%	181.45	0.63%	196.03	0.74%	218.46	0.93%	243.42
80	20	0.26%	71.26	0.55%	73.17	0.54%	77.89	0.67%	75.25
90	10	0.30%	14.45	0.44%	14.04	0.53%	14.42	0.62%	14.32

Going further and comparing the gaps on each instance we found on average the upper end of the confidence interval for the inexact SAA method is 1.02% greater than the upper end of the confidence interval for the SAA method; while the lower end of the confidence interval of the inexact SAA method is 0.63% lower than the lower end of the confidence interval for the SAA method. This suggests that there may be some slight room to improve on the upper bound solution produced by the dual-ascent heuristic. Indeed, we implemented the dual-ascent heuristic for the UFL without any additional local search phase (as we did for the ConFL) that adds and drops facilities.

6. CONCLUSIONS

In this article, we proposed an inexact SAA method that broadens the scope and increases the applicability of the SAA method to large two-stage stochastic integer programs. To apply it, one only needs a good heuristic and a tight lower bounding procedure for the sample average problems to yield tight confidence intervals on the stochastic problem.

We considered the SConFL problem that arises in applications in data management and telecommunications. We studied the impact of two alternate sources of assignment cost uncertainty and described the value of the stochastic solution for each case. For uncertain demand quantities, we showed that the problem can be solved optimally by replacing uncertain values by their average values. In the more general case, where variability in costs occurs on the assignment edges, we showed how to transform the SConFL problem into a deterministic equivalent ConFL problem with a larger number of customer nodes; which then can be solved by already proposed methods, such as the DLS heuristic, when the number of scenarios is limited. For SConFL problems with a large number of scenarios, that are impractical to solve by solving their deterministic equivalent problem, we illustrated the inexact SAA approach with strong results and tight confidence intervals on the objective function value.

As the inexact SAA method expands the scope and allows one to apply heuristics to the sample average problems (when there is a good lower bound at hand), we expect (and hope) this approach will be adopted by other researchers as a computational solution procedure of choice for a wide range of two-stage stochastic integer programs, where *good heuristics* and *tight lower bounding mechanisms* are at-hand. Certainly, additional questions remain to be answered from a theoretical perspective. For example, it may be useful to characterize the number of samples for the sample average problem and the number of replications for a given level of convergence. Furthermore, one can explore variance reduction techniques, such as common random streams, to improve the computational efficiency of our method. These are topics for future research.

REFERENCES

[1] S. Ahmed, "Two-stage stochastic integer programming: A brief introduction," Wiley encyclopedia of operations research and management science, J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, and J.C. Smith (Editors), Wiley, 2010.

[2] M.G. Bardossy and S. Raghavan, Dual-based local search for the connected facility location and related problems, INFORMS J Comput 22 (2010), 584–602.

[3] O. Berman, Dynamic positioning of mobile servers on networks, Technical report TR-144, Operations Research Center, MIT, 1978.

[4] O. Berman and A.R. Odoni, Locating mobile servers on a network with Markovian properties, Networks 12 (1982), 73–86.

[5] J.R. Birge and F. Louveaux, Introduction to stochastic programming, Springer, Berlin, 1997.

[6] I. Bomze, M. Chimani, M. Jünger, I. Ljubić, P. Mutzel, and B. Zey, "Solving two-stage stochastic Steiner tree problems by two-stage branch-and-cut," Algorithms and Computations, Lecture Notes in Computer Science, O. Cheong, K.Y. Chwa and K. Park (Editors), Vol. 6506, Springer, Berlin, 2010, pp. 427–439.

[7] I. Correia and F. Saldanha da Gama, "Facility location under uncertainty," Location science, G. Laporte, S. Nickel, and F. Saldanha da Gama (Editors), Chapter 8, Springer, Berlin, 2015, pp. 177–203.

[8] G.B. Dantzig, Linear programming under uncertainty, Manage Sci 1 (1955), 197–206.

[9] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer, Approximating connected facility location problems via random facility sampling and core detouring, Proc 19th Ann ACM-SIAM Symp Discrete Algorithms, San Francisco, CA, 2008, pp. 1174–1183.

[10] D. Erlenkotter, A dual-based procedure for uncapacitated facility location, Oper Res 26 (1978), 992–1009.

[11] S. Gollowitzer and I. Ljubić, MIP models for connected facility location: A theoretical and computational study, Comput & Oper Res 38 (2011), 435–449.

[12] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, Provisioning a virtual private network: a network design problem for multicommodity flow, Proc 33rd Ann ACM Symp Theory Comput, Heraklion, Crete, Greece, 2001, pp. 389–398.

[13] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden, Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem, Proc 44th Ann IEEE Symp Foundations Comput Sci, Cambridge, MA, 2003, pp. 606–615.

[14] A. Gupta, R. Ravi, and A. Sinha, LP rounding approximation algorithms for stochastic network design, Math Oper Res 32 (2007), 345–364.

[15] H. Jung, M.K. Hasan, and K.Y. Chwa, Improved primal-dual approximation algorithm for the connected facility location problem, Proc 2nd Ann Int COCOA, Combinatorial Optimization and Applications, Lecture Notes in Computer Science, Vol. 5165, 2008, 265–277.

[16] D.R. Karger and M. Minkoff, Building Steiner trees with incomplete global knowledge, Proc 41st Ann IEEE Symp Foundations of Computer Science, Redondo Beach, CA, 2000, pp. 613–623.

[17] A. Kleywegt, A. Shapiro, and T. Homem-de Mello, The sample average approximation method for stochastic discrete optimization, SIAM J Optim 12 (2002), 479–502.

[18] C. Krick, H. Räcke, and M. Westermann, Approximation algorithms for data management in networks, Theory Comput Syst 36 (2003), 497–519.

[19] D. Kurz, P. Mutzel, and B. Zey, "Parameterized algorithms for stochastic Steiner tree problems," Mathematical and Engineering Methods in Computer Science, Lecture Notes in Computer Science, Vol. 7721, Springer, Berlin, 2013, pp. 143–154.

[20] G. Laporte, F. Louveaux, and L. Van Hamme, Exact solution to a location problem with stochastic demands, Transportation Sci 28 (1994), 95–103.

[21] Y. Lee, S.Y. Chiu, and J. Ryan, A branch and cut algorithm for a Steiner tree-star problem, INFORMS J Computi 8 (1996), 194–201.

[22] M. Leitner, I. Ljubić, J. Salazar-González, and M. Sinnl, "On the asymmetric connected facility location polytope," Combinatorial optimization, A. Schrijver (Editor), Springer, Berlin, 2014, pp. 371–383.

[23] M. Leitner, I. Ljubić, J. Salazar-González, and M. Sinnl, An algorithmic framework for the exact solution of tree-star problems, Eur J Oper Res 261 (2017), 54–66.

[24] M. Leitner, I. Ljubić, J. Salazar-González, and M. Sinnl, The connected facility location polytope, Discr Appl Math (2017), to appear.

[25] I. Ljubić, A hybrid VNS for connected facility location, Proc 4th Int Conf on Hybrid Metaheuristics, Hybrid Metaheuristics, Lecture Notes in Computer Science, Vol. 4771, Springer, 2007, pp. 157–169.

[26] F. Louveaux and D. Peeters, A dual-based procedure for stochastic facility location, Oper Res 40 (1992), 564–573.

[27] W.K. Mak, D.P. Morton, and R.K. Wood, Monte Carlo bounding techniques for determining solution quality in stochastic programs, Oper Res Lett 24 (1999), 47–56.

[28] P.B. Mirchandani, Analysis of stochastic networks in emergency service systems, Technical report, IRP-TR-15-75, Operations Research Center, MIT, 1975.

[29] P.B. Mirchandani and A.R. Odoni, Locations of medians on stochastic networks, Transp Sci 13 (1979), 85–97.

[30] P. Nuggehalli, V. Srinivasan, and C.F. Chiasserini, Energy-efficient caching strategies in ad hoc wireless networks, Proc 4th ACM Int Symp Mobile ad hoc Networking & Comput, Annapolis, MD, 2003, pp. 25–34.

[31] R. Ravi and A. Sinha, Hedging uncertainty: Approximation algorithms for stochastic optimization problems, Math Program 108 (2006), 97–114.

[32] R. Schultz, L. Stougie, and M. Vlerk, Two-stage stochastic integer programming: A survey, Statistica Neerlandica 50 (2008), 404–416.

[33] A. Shapiro and A. Philpott, A tutorial on stochastic programming, Unpublished Manuscript 2007.

[34] L.V. Snyder, Facility location under uncertainty: A review, IIE Trans 38 (2006), 547–564.

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

[36] A. Tomazic and I. Ljubić, A GRASP algorithm for the connected facility location problem, The 2008 Int Symp Appl Internet, Turku, Finland, 2008, pp. 257–260.

[37] B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro, The sample average approximation method applied to stochastic routing problems: A computational study, Comput Optim Appl 24 (2003), 289–333.

[38] J.R. Weaver and R.L. Church, Computational procedure for location problems on stochastic networks, Transportation Sci 17 (1983), 168–180.