2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Hierarchical optimization framework for network resource allocation under uncertainty

Yifan Liu Ashish Cherukuri

Abstract— **This paper introduces hierarchical optimization algorithms to solve, in a data-driven manner, an uncertain resource allocation problem over a two-layered tree network. In this optimization problem, the root of the tree aims to minimize the cost of procuring a certain resource, the demand for which is uncertain and originates at the leaves of the network. The demand is to be met with a certain probability which is represented as chance-constraints. We assume that data regarding the uncertain demand is available at each leaf of the network and we design a general framework of hierarchical optimization procedures where the chance-constrained allocation problem is solved using the available data with the constraint that the data is not transferred to the root. Our hierarchical procedure, termed the abstraction-allocation framework, is adapted to three data-driven algorithms for solving the chance-constrained problem: the scenario method, sample average approximation, and distributionally robust optimization. In our framework, the middle layer of the network facilitates the abstraction and allocation when information flows from leaves to the root and the other way around, respectively.**

I. Introduction

Network resource allocation problems appear in various application domains, such as, managing the electrical power grid and handling the supply chain logistics. These problems are inevitably affected by uncertainties and this is more so in a modern power grid, where the number of uncertain demand and supply components are increasing rapidly. While a central coordinator can make decisions about how much resource needs to be allocated to which part of the network, this process quickly becomes intractable when the number of uncertain components increase, the data related to them explodes to high volumes, and much of this data is not shared with the central node due to privacy and security concerns. In addition to this challenge, another prevalent characteristic of these resource allocation problems is the natural hierarchical structure of the network, mostly caused by geographical positioning. For example, the solar panels in a neighborhood affect the energy balance and power characteristics of the distribution grid that they are connected to but they have less impact on another city geographically far from them. Bearing the above mentioned challenge and the structure of the problem in mind, in this paper we explore hierarchical optimization routines as a framework for solving chance-constrained resource allocation problem. Our novelty and contribution is the introduction of this framework that can be viewed as a structure that can include further distributed and decentralized subroutines.

The authors are with the Engineering and Technology Institute Groningen, University of Groningen. Email: {yifan.liu, a.k.cherukuri}@rug.nl.

1) Literature review: Distributed and decentralized optimization have been popular for a while, see [1] for a survey of methods, however there is less attention on applying these algorithms to infrastructure networks. Recently in [2], [3] hierarchical approaches for solving deterministic network optimization problems were explored. However, these works did not consider any uncertainty affecting the optimization problems. Concerning decentralized or distributed methods solving stochastic optimization problems over an infrastructure network, some works focus on scenario methods [4], [5] for approximating the uncertain constraints. While the optimization problems in there are richer in the form of constraints and objective function, they do not explore a hierarchical setup. Our attempt in the current paper is to take a step towards bridging these two lines of research.

For our setup, in addition to the popular scenario approach [6], we also consider the sample average approximation [7], and distributionally robust optimization [8], [9], [10]. The later two lead to a mixed-integer optimization problem that need to be solved over a network. In this way, our work also relates to distributed mixed-integer optimization problems [11], [12], [13]. Finally, we refer to [14], [15], where the two-layer architecture, similar to ours, is used in a federated learning setup.

2) Setup and contributions: We consider an undirected two-layered tree network designed for resource allocation, where the root of the network procures a resource from an external source and distributes it to the leaf nodes via the middle layer. The demand of the resource at the leaves is uncertain and the demand needs to be met collectively across the network with high probability. This is encoded using joint chance constraints. The probability distribution of the demand is unknown but the data regarding it is assumed to be known. The key restriction is that the data is only known at the leaves and it is not transferred to the root of the network. We consider three data-driven solution approaches for handling the chance constraints: scenario method, sample average approximation, and distributionally robust optimization. Each of these methods involve solving a deterministic optimization problem for which we design our hierarchical framework.

The hierarchical procedure involves three phases. The first one is the abstraction phase where the middle layer collectively determines how much resource would be required by each of them for satisfying the demand under each of the data-driven methods. The middle layer communicates this information to the root. The second phase consists of solving the optimization problem by the root where the cost

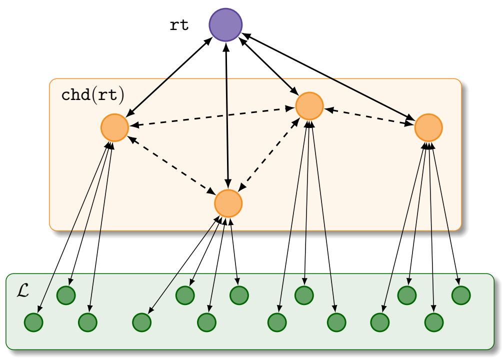

Fig. 1: A two-layer network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for which resource allocation problem is defined. The root of the tree is denoted by $\texttt{rt}$. The children of the root, represented by $\texttt{chd}(\texttt{rt})$, form the middle layer. The leaves of the network form the set $\mathcal{L}$. The solid lines represent the tree network and the dashed lines the communication between the middle layer nodes.

of procuring the resource is minimized. The third phase allocates the resource to the leaves via the middle layer that obtained the optimized resource from the root. We provide the abstraction and allocation procedures for each of the data-driven methods mentioned above. We conclude the paper with a discussion including future extensions.

3) Notation: Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}$ denote the set of real, nonnegative real, and integer numbers. For a positive integer n, we denote $[n] := \{1, \ldots, n\}$. Given two vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ we denote their concatenation as $(u; v) \in \mathbb{R}^{m+n}$. The number of elements in a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The n-dimensional vector of all ones is represented by $\mathbb{1}_n$.

II. Problem Statement

Consider a two-layered tree network represented by an undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ stand for the nodes and edges of the network, respectively [1]. The root of the tree ($\texttt{rt}$) is responsible for procurement of a certain resource from some external source and the resource is then distributed among the leaves of the network via the middle-layer, see Figure 1 as an example. We consider the case where the demand of the resource is uncertain at each leaf of the network. Considering this setup, we aim to solve the following *joint chance-constrained problem* ($\texttt{JCCP}$):

$$\min_{(y, \{y_\ell\}_{\ell \in \mathcal{L}})} \quad c(y) \tag{1a}$$

$$\text{s.t.} \quad y \geq \sum_{\ell \in \mathcal{L}} y_\ell, \tag{1b}$$

$$\mathbb{P}[y_\ell \geq u_\ell, \forall \ell \in \mathcal{L}] \geq 1 - \epsilon, \tag{1c}$$

where $\mathcal{L} \subset \mathcal{V}$ is the set of leaves of the tree graph $\mathcal{G}$, the objective function $c : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is convex, $\epsilon \in (0, 1)$, and $\mathbb{P}$ is the distribution of the random variable $u := (u_1; u_2; \ldots; u_{|\mathcal{L}|})$. The objective function stands for the cost incurred by the root in procuring the resource for the network, the latter is represented by the decision variable $y \in \mathbb{R}$. The constraint (1b) indicates that the procured resource for the network needs to be larger than the collective requirement of each of the leaves, which is given in (1c). Specifically, all leaves together wish to fulfill the uncertain demand u_ℓ with probability at least $1 - \epsilon$.

Given the problem data c, $\mathbb{P}$, ϵ, and the graph, the root can solve the optimization problem (1) and allocate the resource to leaves as per the obtained optimizer. However, there are two challenges that need to be dealt with. First, the probability distribution $\mathbb{P}$ is often not known and one needs to rely on sample-based methods to approximate the solution of the problem. Second, in a network resource allocation setup as ours, the data is not available at the root. Moreover, the dataset of the entire network can possibly be so large that it cannot be handled by a central computational unit. On the other hand, the entire dataset need not be required to make a meaningful approximation of the optimizer. Thus, keeping these challenges in mind, our work proposes designing a hierarchical optimization routine where we compute the solution of (1) without transferring the entire data across the network. Our routine will be a meta algorithm given in Section III, implementing which requires simple computation in case of some data-driven solutions and further distributed algorithmic subroutines in some others.

III. Hierarchical optimization routines for JCCP

Here, we present our algorithm that will solve problem (1) using samples and hierarchical information exchange. To this end, we assume that N samples $\widehat{\mathcal{U}} := \{\widehat{u}[1], \widehat{u}[2], \ldots, \widehat{u}[N]\}$ of the uncertainty u are available with the constraint that each leaf $\ell \in \mathcal{L}$ only knows $\widehat{\mathcal{U}}_\ell := \{\widehat{u}_\ell[1], \widehat{u}_\ell[2], \ldots, \widehat{u}_\ell[N]\}$ where $\widehat{u}_\ell[i]$ is the component of $\widehat{u}[i]$ corresponding to ℓ. In case these samples are drawn in an i.i.d manner from the distribution $\mathbb{P}$, the solutions obtained from our algorithm will have desirable statistical guarantees. However, our methods themselves do not require this assumption as our focus is on algorithmic aspects.

The general procedure proposed in our work is termed, abstraction-allocation framework, given as Algorithm 1. Below we provide a brief explanation of the steps.

[Informal description]: The procedure admits as input the cost c that is known to the root and samples $\widehat{\mathcal{U}}_\ell$ and violation level ϵ that is known to each of the leaf $\ell \in \mathcal{L}$. The algorithm starts with computing the abstraction $u_{\text{abs},m}$ for each $m \in \texttt{chd}(\texttt{rt})$ in Line 1 using the samples $\{\widehat{\mathcal{U}}_\ell\}_{\ell \in \mathcal{L}_m}$ that the leaves $\mathcal{L}_m$ connected to the node m know and information exchange with other nodes in $\texttt{chd}(\texttt{rt})$ which is denoted in a general way by $\mathcal{I}$. We denote this procedure of generating $u_{\text{abs},m}$ by the map $\texttt{Abst}$. The exact definition of this map will vary based on the data-driven method employed to solve $\texttt{JCCP}$. The computed $u_{\text{abs},m}$ is sent to the root (see Line 2), and the root solves a deterministic optimization problem (2) in Line 3. The optimal

[1]For a graph, a path is an ordered pair of vertices such that each consecutive pair of vertices is an edge. A graph is connected if there is a path between every two vertices. A cycle is a path that starts and ends at the same vertex. A graph is acyclic if it contains no cycle and an acyclic connected graph is called a tree.

solution gives a resource value for each node $m \in \texttt{chd}(\texttt{rt})$ denoted as $\bar{y}_m^{\text{abs}}$, which is sent to m by the root (see Line 4). The last step (Line 5) is the allocation to the leaves, which is executed by the middle-layer. Each node $m \in \texttt{chd}(\texttt{rt})$ allocates y_ℓ^{all} to $\ell \in \mathcal{L}$ which is determined by the map Allo. Similar to the abstraction map, we prescribe the exact definition of Allo later, depending on the data-driven method.

Algorithm 1: Abstraction-allocation framework

Input : Root: cost function c; Leaves: samples $\widehat{\mathcal{U}}_\ell$ and tolerance ϵ

/* Stage 1: Abstraction */

Each $m \in \texttt{chd}(\texttt{rt})$ executes:

1 Computes $u_{\text{abs},m} \leftarrow \texttt{Abst}(\{\widehat{\mathcal{U}}_\ell\}_{\ell \in \mathcal{L}_m}, \mathcal{I})$

2 Sends $u_{\text{abs},m}$ to the root

/* Stage 2: Optimization */

Root executes:

3 Solves:

$$\begin{aligned} \min \quad & c(y) \\ \text{s.t.} \quad & y \geq \textstyle\sum_{m \in \texttt{chd}(\texttt{rt})} \bar{y}_m, \\ & \bar{y}_m \geq u_{\text{abs},m}, \quad \forall m \in \texttt{chd}(\texttt{rt}), \end{aligned} \tag{2}$$

and sets the optimizer as $(y^{\text{abs}}, \{\bar{y}_m^{\text{abs}}\}_{m \in \texttt{chd}(\texttt{rt})})$

4 Sends $\bar{y}_m^{\text{abs}}$ to $m \in \texttt{chd}(\texttt{rt})$

/* Stage 3: Allocation */

Each $m \in \texttt{chd}(\texttt{rt})$ executes:

5 Sets $\{y_\ell^{\text{all}}\}_{\ell \in \mathcal{L}_m} \leftarrow \texttt{Allo}(\bar{y}_m^{\text{abs}})$

6 Sends y_ℓ^{all} to each $\ell \in \mathcal{L}$

Below we proceed to explore the data-driven methods and the Abst and Allo maps that will specify the hierarchical algorithms related to them. We also emphasize on how these maps can be computed with information exchange between each middle-layer node and the leaves associated to it.

A. Scenario Method

The first sample-based procedure that we explore to solve (1) is the scenario method [16]. In here, the chance-constraint (1c) is replaced with a set of constraints, each one of them imposes the constraint $y_\ell \geq u_\ell$ for a data point from $\widehat{\mathcal{U}}_\ell$. In particular, we obtain the following deterministic optimization problem

$$\min_{(y, y^{\text{vec}})} \quad c(y) \tag{3a}$$

$$\text{s.t.} \quad y \geq \mathbb{1}_{|\mathcal{L}|}^\top y^{\text{vec}}, \tag{3b}$$

$$y^{\text{vec}} \geq \widehat{u}[i], \quad \forall i \in [N], \tag{3c}$$

where $y^{\text{vec}} := (y_1; y_2; \dots; y_{|\mathcal{L}|})$ is the vector of all decision variables $\{y_\ell\}$. Note that the above problem is equivalent to

$$\min_{(y, y^{\text{vec}})} \quad c(y) \tag{4a}$$

$$\text{s.t.} \quad y \geq \mathbb{1}_{|\mathcal{L}|}^\top y^{\text{vec}}, \tag{4b}$$

$$y^{\text{vec}} \geq u_{\max}, \tag{4c}$$

where $u_{\max} := (u_{\max,1}; ; u_{\max,2}; \dots, u_{\max,|\mathcal{L}|})$ with $u_{\max,\ell} := \max_i \widehat{u}_\ell[i]$ for all $\ell \in \mathcal{L}$. This simple observation guides us in defining the abstraction and allocation maps that will consequently form Algorithm 1 for the scenario method. We define abstraction as

$$\texttt{Abst}(\{\widehat{\mathcal{U}}_\ell\}_{\ell \in \mathcal{L}_m}, \mathcal{I}) := \sum_{\ell \in \mathcal{L}_m} u_{\max,\ell}. \tag{5}$$

and the allocation as

$$\texttt{Allo}(\bar{y}_m^{\text{abs}}) := \left\{ y_\ell^{\text{all}} := \frac{u_{\max,\ell}}{\sum_{\ell \in \mathcal{L}_m} u_{\max,\ell}} \bar{y}_m^{\text{abs}} \;\middle|\; \ell \in \mathcal{L}_m \right\}. \tag{6}$$

Namely, the abstraction map aggregates the maximum value $u_{\max,\ell}$ among the samples obtained by each leaf node and the allocation map distributes $\bar{y}_m^{\text{abs}}$ proportional to $u_{\max,\ell}$. Note that importantly, there is no need for nodes in $\texttt{chd}(\texttt{rt})$ to communicate with each other. The following result summarizes the formal guarantee of the algorithm under the above defined maps.

Proposition III.1. *(Guarantee for Algorithm 1 under scenario method): The output $(y^{\text{abs}}, \{y_\ell^{\text{all}}\}_{\ell \in \mathcal{L}})$ of Algorithm 1 with the maps* Abst *and* Allo *defined in* (5) *and* (6)*, respectively, is an optimizer of the scenario problem* (3).

The proof can be deduced by noting that following two sets, each corresponding to problems (4) and (2), respectively, are equal:

$$\mathcal{F}_1 := \{y \in \mathbb{R} \,|\, \exists \{y_\ell\}_{\ell \in \mathcal{L}} \text{ s.t. } (y, \{y_\ell\}_{\ell \in \mathcal{L}}) \text{ is feasible for (4)}\},$$

$$\mathcal{F}_2 := \{y \in \mathbb{R} \,|\, \exists \{\bar{y}_m\}_{m \in \texttt{chd}(\texttt{rt})} \text{ s.t. } (y, \{\bar{y}_m\}_{m \in \texttt{chd}(\texttt{rt})}) \text{ is feasible for (2)}\}.$$

The intuition behind the abstraction-allocation mechanism for the scenario method is simple. The result is facilitated by the fact that the inequality defining chance-constraint is straightforward. Our wish is to explore more general cases in future where such abstraction and allocation maps can be defined without any communication between nodes in $\texttt{chd}(\texttt{rt})$. But for now, we investigate other data-driven methods of solving (1) where such communication is necessary.

B. Sample Average Approximation (SAA)

In this method, the probability distribution $\mathbb{P}$ defining the chance constraint (1c) is replaced with the empirical distribution

$$\widehat{\mathbb{P}} := \frac{1}{N} \sum_{i=1}^{N} \delta_{\widehat{u}[i]}$$

defined using the data set $\widehat{\mathcal{U}}$. Here, $\delta_{\widehat{u}[i]}$ stands for the delta function placed at the point $\widehat{u}[i]$. The thus formed optimization problem is termed as the sample average approximation (SAA) of JCCP (1) and is given as

$$\min_{(y, y^{\text{vec}})} \quad c(y) \tag{7a}$$

$$\text{s.t.} \quad y \geq \mathbb{1}_{|\mathcal{L}|}^\top y^{\text{vec}}, \tag{7b}$$

$$\widehat{\mathbb{P}}[y^{\text{vec}} \geq u] \geq 1 - \epsilon. \tag{7c}$$

Here, we recall that y^{vec} collects all y_ℓ variables and $u \sim \widehat{\mathbb{P}}$. To obtain the hierarchical procedure of solving the above problem, we reformulate (7) following [17] and [18] as a mixed-integer linear program (MILP). Introducing for each $i \in [N]$, a binary variable $z_i \in \{0, 1\}$, we write

$$\min_{(z,y,y^{\text{vec}})} \quad c(y) \tag{8a}$$

$$\text{s.t.} \quad y \geq \mathbb{1}_{|\mathcal{L}|}^\top y^{\text{vec}}, \tag{8b}$$

$$y^{\text{vec}} - \widehat{u}[i] + \mathbb{1}_{|\mathcal{L}|} z_i M \geq 0, \ \ \forall i \in [N], \tag{8c}$$

$$\mathbb{1}_N^\top z \leq N\epsilon, \tag{8d}$$

$$z \in \{0, 1\}^N, \tag{8e}$$

where M is a large positive number satisfying $M > 2 \max_{i \in [N]} \|\widehat{u}[i]\|_\infty$. Note that if $z_i = 0$, then the constraint (8c) reads as $y^{\text{vec}} \geq \widehat{u}[i]$. On the other hand, if $z_i = 1$, then $y^{\text{vec}} - \widehat{u}[i] + M\mathbb{1}_{|\mathcal{L}|} \geq 0$, which is trivially satisfied. Thus, the number of samples for which $y^{\text{vec}} \geq \widehat{u}[i]$ is not satisfied is bounded by constraint (8d). The problem (8) is equivalent to (7), which means that the set of values y and y_ℓ components take at the optimizers of both problems are same. To solve (8) in a hierarchical manner, the nodes in $\texttt{chd}(\texttt{rt})$ need to handle constraints (8c) to (8e) collectively by agreeing upon the samples for which $z_i = 0$. To this end, we propose that nodes in $\texttt{chd}(\texttt{rt})$ solve the following problem among themselves:

$$\min_{(z,y^{\text{vec}})} \quad \mathbb{1}_{|\mathcal{L}|}^\top y^{\text{vec}} \tag{9a}$$

$$\text{s.t.} \quad y^{\text{vec}} - \widehat{u}[i] + \mathbb{1}_{|\mathcal{L}|} z_i M \geq 0, \ \ \forall i \in [N], \tag{9b}$$

$$\mathbb{1}_N^\top z \leq N\epsilon, \tag{9c}$$

$$z \in \{0, 1\}^N. \tag{9d}$$

Note that the objective function of the above problem is separable among $\texttt{chd}(\texttt{rt})$, that is, one can write the objective as $\sum_{m \in \texttt{chd}(\texttt{rt})} \sum_{\ell \in \mathcal{L}_m} y_\ell$. However, the constraints (9b)-(9d) are coupled as they all need to agree on the variable z. Therefore, no single node in $\texttt{chd}(\texttt{rt})$ can solve the above problem and so, we appeal to distributed optimization routines. Once $\texttt{chd}(\texttt{rt})$ find the solution of (9), each node in $\texttt{chd}(\texttt{rt})$ computes the abstraction map.

To arrive at a distributed method for (9), we first provide a reformulation. For each $m \in \texttt{chd}(\texttt{rt})$, consider the variable $\alpha_m \in \mathbb{R}$ that will stand for $\sum_{\ell \in \mathcal{L}_m} y_\ell$. Then, the reduced version of (9) is given as

$$\min_{(z,\{\alpha_m\})} \quad \textstyle\sum_{m \in \texttt{chd}(\texttt{rt})} \alpha_m \tag{10a}$$

$$\text{s.t.} \quad \alpha_m - \mathbb{1}_{|\mathcal{L}_m|}^\top \widehat{u}^{[m]}[i] + z_i |\mathcal{L}_m| M \geq 0, \quad \forall i \in [N], \ \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{10b}$$

$$\mathbb{1}_N^\top z \leq N\epsilon, \ \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{10c}$$

$$z \in \{0, 1\}^N, \tag{10d}$$

where $\widehat{u}^{[m]}[i] := (\widehat{u}_\ell[i])_{\ell \in \mathcal{L}_m}$ stands for the components of $\widehat{u}[i]$ that are held by leaves $\ell \in \mathcal{L}_m$. Problems (10) and (9) are equivalent in the sense that the set of z-components of optimal solutions of both problems are same. That is, if $(z^*, \{\alpha_m^*\})$ is an optimizer of (10), then $(z^*, (y^{\text{vec}})^*)$ with

$$(y^{\text{vec}})_\ell^* := \max_{i \in [N]} \ \widehat{u}_\ell[i] - z_i^* M$$

is an optimizer of (9) and one can verify that $(y^{\text{vec}})^*$ defined in the above manner satisfies $\alpha_m^* = \sum_{\ell \in \mathcal{L}_m} (y^{\text{vec}})_\ell^*$. On the other hand, if $(z^*, (y^{\text{vec}})^*)$ is an optimal solution of (9), then $(z^*, \{\alpha_m^*\})$ with $\alpha_m^* = \sum_{\ell \in \mathcal{L}_m} (y^{\text{vec}})_\ell^*$ is an optimizer of (10). Next, we move to solving (10) in a distributed manner. To this end, we define local variables $(z^{(m)}, \alpha^{(m)})$ for each $m \in \texttt{chd}(\texttt{rt})$, where $z^{(m)} \in \{0, 1\}^N$ is an estimate of z and $\alpha^{(m)} := (\alpha_{\bar{m}}^{(m)})_{\bar{m} \in \texttt{chd}(\texttt{rt})}$ is a vector belonging to $\mathbb{R}^{|\texttt{chd}(\texttt{rt})|}$, where $\alpha_{\bar{m}}^{(m)}$ is the estimate of $\alpha_{\bar{m}}$ held by agent m. Using these variables, the consensus version of (10) is

$$\min_{\{(z^{(m)},\alpha^{(m)})\}} \quad \textstyle\sum_{m \in \texttt{chd}(\texttt{rt})} \mathbf{e}_m^\top \alpha^{(m)} \tag{11a}$$

$$\text{s.t.} \quad \alpha_m^{(m)} - \mathbb{1}_{|\mathcal{L}_m|}^\top \widehat{u}^{[m]}[i] + z_i^{(m)} |\mathcal{L}_m| M \geq 0, \quad \forall i \in [N], \ \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{11b}$$

$$\mathbb{1}_N^\top z^{(m)} \leq N\epsilon, \ \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{11c}$$

$$(z^{(m)}, \alpha^{(m)}) = (z^{(\bar{m})}, \alpha^{(\bar{m})}), \quad \forall m, \bar{m} \in \texttt{chd}(\texttt{rt}), \tag{11d}$$

$$z^{(m)} \in \{0, 1\}^N, \ \alpha^{(m)} \in \mathbb{R}^{|\texttt{chd}(\texttt{rt})|}, \quad \forall m \in \texttt{chd}(\texttt{rt}), \tag{11e}$$

where $\mathbf{e}_m \in \mathbb{R}^{|\texttt{chd}(\texttt{rt})|}$ is a vector with all entries zero except the one corresponding to m. The above problem takes a consensus-MILP for which a distributed algorithm was developed in [19]. The consensus-MILP takes the form

$$\min_{\{x^{(m)}\}} \quad \sum_{m \in \texttt{chd}(\texttt{rt})} c_m^\top x^{(m)}, \tag{12a}$$

$$\text{s.t.} \quad x^{(m)} \in X_m, \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{12b}$$

$$x^{(m)} = x^{(\bar{m})}, \ \forall m, \bar{m} \in \texttt{chd}(\texttt{rt}), \tag{12c}$$

$$x^{(m)} \in \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z}, \ \forall m \in \texttt{chd}(\texttt{rt}), \tag{12d}$$

where X_m is a polyhedron. We comment on the parallelism between (11) and (12). Let $(z^{(m)}, \alpha^{(m)})$ in the former problem be the variable $x^{(m)}$ in the later. The objective function is linear and separable in both problems. The linear constraints (11b) and (11c) are captured in (12b). Finally, consensus and integer constraints in (11d) and (11e) respectively are represented as (12c) and (12d). Having established the similarity, assume now an undirected connected communication graph $\mathcal{G}^{\texttt{chd}} := (\texttt{chd}(\texttt{rt}), \mathcal{E}^{\texttt{chd}})$ among the nodes $\texttt{chd}(\texttt{rt})$. Over this communication network, $\texttt{chd}(\texttt{rt})$ execute Algorithm 1 from [19] and solve problem (11) in a distributed manner.

We omit the details of the algorithm due to space constraints. We do make a note that in literature there are not many works on solving MILP's in a distributed manner and among these, the algorithm in [19] is the only one that considers a separable cost structure with consensus on integer variables and provides a rigorous convergence guarantee for the algorithm.

As a result of the distributed algorithm from [19], each $m \in \texttt{chd}(\texttt{rt})$ has access to component $((z^{(m)})^*, (\alpha^{(m)})^*)$ of the optimizer of problem (11). Then, the node m sets

$$y^*_\ell = \max_{i\in[N]} \ \widehat{u}_\ell[i] - (z^{(m)})^*_i |\mathcal{L}_m| M$$

for all $\ell \in \mathcal{L}_m$. Using this, the abstraction and allocation maps to solve the sample average approximation (7) of JCCP using Algorithm 1 are given as

$$\texttt{Abst}(\{\widehat{\mathcal{U}}_\ell\}_{\ell\in\mathcal{L}_m}, \mathcal{I}) := \sum_{\ell\in\mathcal{L}_m} y^*_\ell, \tag{13}$$

and

$$\texttt{Allo}(\bar{y}^{\text{abs}}_m) := \left\{ y^{\text{all}}_\ell := \frac{y^*_\ell}{\sum_{\ell\in\mathcal{L}_m} y^*_\ell} \bar{y}^{\text{abs}}_m \;\Big|\; \ell \in \mathcal{L}_m \right\}. \tag{14}$$

We give the formal guarantee of the algorithm under the above defined maps as follows. The proof follows exactly the same way as that of Proposition III.1.

Proposition III.2. *(Guarantee for Algorithm 1 under SAA): The output $(y^{\text{abs}}, \{y^{\text{all}}_\ell\}_{\ell\in\mathcal{L}})$ of Algorithm 1 with the maps* Abst *and* Allo *defined in* (13) *and* (14)*, respectively, is an optimizer of the joint chance-constraint problem using sample average approximation* (7)*.*

C. Distributionally Robust Optimization (DRO)

Here, we consider a distributionally robust (DR) approach of solving the JCCP using ideas from [8], [9], and [10]. The general DR framework involves considering the worst-case value for the objective function or the constraint over a set of distributions and optimizing for it [20]. These sets of distributions are commonly termed as *ambiguity sets*. Given the dataset $\widehat{\mathcal{U}}$ and a suitable radius $\theta > 0$, we define the *Wasserstein ambiguity set* as

$$\mathcal{M}^\theta_N := \left\{ \mathbb{P} \in \mathcal{P}\left(\mathbb{R}^{|\mathcal{L}|}\right) \mid d_{\text{W}}(\mathbb{P}, \widehat{\mathbb{P}}) \le \theta \right\} \tag{15}$$

with $d_{\text{W}}(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\mathbb{P}\in\mathcal{H}(\mathbb{P}_1,\mathbb{P}_2)} \mathbb{E}_{\mathbb{P}}[\|w - v\|]$, where $w \sim \mathbb{P}_1, v \sim \mathbb{P}_2$, and $\mathcal{H}(\mathbb{P}_1, \mathbb{P}_2)$ represents the set of all distributions on $\mathbb{R}^{|\mathcal{L}|} \times \mathbb{R}^{|\mathcal{L}|}$ with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$. Using this definition, we present the DR chance-constrained problem as

$$\begin{aligned} \min_{(y, y^{\text{vec}})} \quad & c(y) && \text{(16a)}\\ \text{s.t.} \quad & y \ge \mathbb{1}^\top_{|\mathcal{L}|} y^{\text{vec}}, && \text{(16b)}\\ & \mathbb{Q}[y^{\text{vec}} \ge u] \ge 1 - \epsilon, \forall \mathbb{Q} \in \mathcal{M}^\theta_N. && \text{(16c)} \end{aligned}$$

The last constraint is the robust one in the sense that the chance-constraint needs to hold for all distributions in the ambiguity set. To solve the problem in a hierarchical way, we first give the reformulation of (16) based on [8, Proposition 2] as an MILP:

$$\begin{aligned} \min_{(z,s,t,y,y^{\text{vec}})} \quad & c(y)\\ \text{s.t.} \quad & y \ge \mathbb{1}^\top_{|\mathcal{L}|} y^{\text{vec}},\\ & \epsilon N t - \mathbb{1}^\top_N s \ge \theta N,\\ & y^{\text{vec}} - \widehat{u}[i] + \mathbb{1}_{|\mathcal{L}|} z_i M \ge \mathbb{1}_{|\mathcal{L}|}(t - s_i), \ \forall i \in [N],\\ & M(1 - z_i) \ge t - s_i, \ \ \forall i \in [N],\\ & z \in \{0,1\}^N, \ t \in \mathbb{R}, s \in \mathbb{R}_{\ge 0}, \end{aligned}$$

where $M > 0$ is a suitably large positive constant. Similar to the previous section on sample average approximation, to determine the abstraction map, the nodes $\texttt{chd}(\texttt{rt})$ would require to solve among themselves the following problem

$$\begin{aligned} \min_{(z,s,t,y^{\text{vec}})} \quad & \mathbb{1}^\top_{|\mathcal{L}|} y^{\text{vec}} && \text{(17a)}\\ \text{s.t.} \quad & \epsilon N t - \mathbb{1}^\top_N s \ge \theta N, && \text{(17b)}\\ & y^{\text{vec}} - \widehat{u}[i] + \mathbb{1}_{|\mathcal{L}|} z_i M \ge \mathbb{1}_{|\mathcal{L}|}(t - s_i), \ \forall i \in [N], && \text{(17c)}\\ & M(1 - z_i) \ge t - s_i, \ \ \forall i \in [N], && \text{(17d)}\\ & z \in \{0,1\}^N, \ t \in \mathbb{R}, s \in \mathbb{R}^N_{\ge 0}. && \text{(17e)} \end{aligned}$$

In the above problem, the objective function is separable, however, nodes in $\texttt{chd}(\texttt{rt})$ need to agree upon variables (z, s, t) to handle the constraints collectively. Thus, we again make use of the distributed algorithm proposed in the previous section. To this end, we first rewrite the above problem in a reduced form as done in (10). We obtain:

$$\begin{aligned} \min_{(z,s,t,\{\alpha_m\})} \quad & \textstyle\sum_{m\in\texttt{chd}(\texttt{rt})} \alpha_m && \text{(18a)}\\ \text{s.t.} \quad & \epsilon N t - \mathbb{1}^\top_N s \ge \theta N, && \text{(18b)}\\ & \alpha_m - \mathbb{1}^\top_{|\mathcal{L}_m|} \widehat{u}^{[m]}[i] + z_i |\mathcal{L}_m| M \ge |\mathcal{L}_m|(t - s_i),\\ & \qquad \forall i \in [N], \ \ \forall m \in \texttt{chd}(\texttt{rt}), && \text{(18c)}\\ & M(1 - z_i) \ge t - s_i, \ \ \forall i \in [N], && \text{(18d)}\\ & z \in \{0,1\}^N, \ t \in \mathbb{R}, s \in \mathbb{R}^N_{\ge 0}. && \text{(18e)} \end{aligned}$$

The above problem is equivalent to (17). It is in the sense that the set of (z, s, t)-components of optimal solutions of both problems are same. Given this reduced form, we proceed to write the consensus-MILP form for which the distributed algorithm from [19] can be employed by $\texttt{chd}(\texttt{rt})$ to solve the problem. Let $(z^{(m)}, s^{(m)}, t^{(m)}, \alpha^{(m)})$ for each $m \in \texttt{chd}(\texttt{rt})$ be local variables, where $z^{(m)} \in \{0,1\}^N$, $s^{(m)} \in \mathbb{R}^N_{\ge 0}$, and $t^{(m)} \in \mathbb{R}$ are estimates of z, s, and t, respectively, and $\alpha^{(m)} := (\alpha^{(m)}_{\bar{m}})_{m\in\texttt{chd}(\texttt{rt})}$ with $\alpha^{(m)}_{\bar{m}}$ being the estimate of $\alpha_{\bar{m}}$ held by m. Then, we obtain the consensus-MILP:

$$\begin{aligned} \min \quad & \textstyle\sum_{m\in\texttt{chd}(\texttt{rt})} \mathbf{e}^\top_m \alpha^{(m)} && \text{(19a)}\\ \text{s.t.} \quad & \epsilon N t^{(m)} - \mathbb{1}^\top_N s^{(m)} \ge \theta N, \ \forall m \in \texttt{chd}(\texttt{rt}), && \text{(19b)}\\ & \alpha^{(m)}_m - \mathbb{1}^\top_{|\mathcal{L}_m|} \widehat{u}^{[m]}[i] + z^{(m)}_i |\mathcal{L}_m| M \ge |\mathcal{L}_m|(t^{(m)} - s^{(m)}_i),\\ & \qquad \forall i \in [N], \ \ \forall m \in \texttt{chd}(\texttt{rt}), && \text{(19c)}\\ & M(1 - z^{(m)}_i) \ge t^{(m)} - s^{(m)}_i, \forall i \in [N],\\ & \qquad \forall m \in \texttt{chd}(\texttt{rt}), && \text{(19d)}\\ & (z^{(m)}, s^{(m)}, t^{(m)}, \alpha^{(m)}) = (z^{(\bar{m})}, s^{(\bar{m})}, t^{(\bar{m})}, \alpha^{(\bar{m})}),\\ & \qquad \forall m, \bar{m} \in \texttt{chd}(\texttt{rt}), && \text{(19e)}\\ & z^{(m)} \in \{0,1\}^N, \ \alpha^{(m)} \in \mathbb{R}^{|\texttt{chd}(\texttt{rt})|}, \ s^{(m)} \in \mathbb{R}^N_{\ge 0},\\ & \qquad t^{(m)} \in \mathbb{R}, \ m \in \texttt{chd}(\texttt{rt}). && \text{(19f)} \end{aligned}$$

Given the above form, the distributed algorithm from [19] can be implemented and each $m \in \texttt{chd}(\texttt{rt})$ obtains the component $((z^{(m)})^*, (s^{(m)})^*, (t^{(m)})^*, (\alpha^{(m)})^*)$ of the optimizer of (19). Using this, the abstraction map is given as

$$\texttt{Abst}(\{\widehat{\mathcal{U}}_\ell\}_{\ell \in \mathcal{L}_m}, \mathcal{I}) := \sum_{\ell \in \mathcal{L}_m} y_\ell^*, \tag{20}$$

where $y_\ell^* = \widehat{u}_\ell[i] - (z_i^{(m)})^* |\mathcal{L}_m| M + |\mathcal{L}_m|((t^{(m)})^* - (s_i^{(m)})^*)$. Furthermore, the allocation is

$$\texttt{Allo}(\bar{y}_m^{\text{abs}}) := \left\{ y_\ell^{\text{all}} := \frac{y_\ell^*}{\sum_{\ell \in \mathcal{L}_m} y_\ell^*} \bar{y}_m^{\text{abs}} \;\middle|\; \ell \in \mathcal{L}_m \right\}. \tag{21}$$

The formal guarantee of the algorithm under the above defined maps is given as follows.

Proposition III.3. *(Guarantee for Algorithm 1 under DRO): The output $(y^{\text{abs}}, \{y_\ell^{\text{all}}\}_{\ell \in \mathcal{L}})$ of Algorithm 1 with the maps* Abst *and* Allo *defined in* (20) *and* (21)*, respectively, is an optimizer of the joint chance-constraint problem using distributionally robust optimization* (16)*.*

Remark III.1. *(Communication and information exchange):* The abstraction-allocation framework when applied to the scenario method, SAA, and DRO, creates a different requirement on the information to be exchanged among various agents. In scenario, the data at leaves need not be shared with anyone, the middle layer need to communicate with each other, and there is only one round of information transfer up and down the tree. On the other hand, in SAA and DRO, the data needs to be shared with the middle layer, the middle layer needs to communicate with each other for several iterations, but the information only travels once up and once down through the tree. This last aspect is central in all our methods and prevents several interactions between the root and the leaves. In future, we wish to further explore problems that can be solved with such a one-round restriction on information flow up and down the tree. •

IV. Conclusions

We have presented a hierarchical framework for solving a simple uncertain network resource allocation problem. The details of the hierarchical routine, that is the abstraction and allocation procedures, are presented for three data-driven methods of handling chance constraints. We make the following remarks regarding our framework and its potential future extensions:

- In our proposed framework, we only had one round of communication up and down the tree. This was facilitated by the simple nature of the function appearing in the probabilistic constraint. There are other problems that can be tackled under this single-round communication. Beyond this, we wish to explore in future the range of problems that would require iterative communication throughout the network.
- We only considered a two-layer tree network and scalar decision variables. In a typical electrical power network, for example, the tree has multiple levels at the distribution grid and there are other restrictions on the power flow that for example, control the voltage around the nominal point. It would be quite interesting to study the abstraction-allocation routines for such constraints.
- Chance-constraints are one way of handling resource satisfaction under uncertainty. Adapting our methods to a more flexible and coherent risk-metric such as the conditional value-at-risk is another interesting and relevant direction to pursue.

References

[1] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed Optimization for Smart Cyber-Physical Networks," *Foundations and Trends® in Systems and Control*, vol. 7, pp. 253–383, Dec. 2019. Publisher: Now Publishers, Inc.

[2] S. Shin, P. Hart, T. Jahns, and V. M. Zavala, "A hierarchical optimization architecture for large-scale power networks," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1004–1014, 2019.

[3] S. Shin, V. M. Zavala, and M. Anitescu, "Decentralized schemes with overlap for solving graph-structured optimization problems," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 3, pp. 1225–1236, 2020.

[4] A. Falsone, K. Margellos, M. Prandini, and S. Garatti, "A scenario-based approach to multi-agent optimization with distributed information," *IFAC-PapersOnLine*, vol. 53, pp. 20–25, Jan. 2020.

[5] G. Minelli, A. Falsone, and M. Prandini, "A steady-state optimal coordination strategy for DERs systems with guaranteed probabilistic performance," *IFAC-PapersOnLine*, vol. 56, pp. 7090–7095, Jan. 2023.

[6] G. C. Calafiore, "Random convex programs," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3427–3464, 2010.

[7] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[8] C. Zhi, D. Kuhn, and W. Wiesemann, "Data-driven chance constrained programs over Wasserstein balls," *Operations Research*, 2022.

[9] W. Xie, "On distributionally robust chance constrained programs with Wasserstein distance," *Mathematical Programming*, vol. 186, no. 1, pp. 115–155, 2021.

[10] A. R. Hota, A. Cherukuri, and J. Lygeros, "Data-driven chance constrained optimization under Wasserstein ambiguity sets," in *American Control Conference*, (Philadelphia, PA), pp. 1501–1506, July 2019.

[11] A. Falsone, K. Margellos, and M. Prandini, "A decentralized approach to multi-agent MILPs: Finite-time feasibility and performance guarantees," *Automatica*, vol. 103, pp. 141–150, May 2019.

[12] A. Testa, A. Rucco, and G. Notarstefano, "Distributed mixed-integer linear programming via cut generation and constraint exchange," *IEEE Transactions on Automatic Control*, vol. 65, pp. 1456–1467, Apr. 2020.

[13] A. Camisa, I. Notarnicola, and G. Notarstefano, "Distributed primal decomposition for large-scale MILPs," *IEEE Transactions on Automatic Control*, vol. 67, pp. 413–420, Jan. 2022.

[14] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2022.

[15] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical federated learning with quantization: Convergence analysis and system design," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 2–18, 2023.

[16] M. C. Campi and S. Garatti, *Introduction to the Scenario Approach*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2018.

[17] J. Luedtke, S. Ahmed, and G. L. Nemhauser, "An integer programming approach for linear programs with probabilistic constraints," *Mathematical Programming*, vol. 122, no. 2, pp. 247–272, 2007.

[18] A. Ruszczyński, "Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra," *Mathematical Programming*, vol. 93, pp. 195–215, 2002.

[19] Z. Liu and O. Stursberg, "Distributed optimization for mixed-integer consensus in multi-agent networks," in *2022 European Control Conference (ECC)*, pp. 2196–2202, 2022.

[20] H. Rahimian and S. Mehrotra, "Frameworks and results in distributionally robust optimization," *Open Journal of Mathematical Optimization*, vol. 3, pp. 1–85, 2022.