

A Randomized Distributed Ellipsoid Algorithm for Uncertain Feasibility Problems

Mohammadreza Chamanbaz, Giuseppe Notarstefano, and Roland Bouffanais

Abstract—In this paper, we consider a network of processors aiming at cooperatively solving a convex feasibility problem in which the constraint set is the intersection of local uncertain sets, each one known only by one processor. We propose a randomized, distributed method—using concepts borrowed from a centralized ellipsoid algorithm—having finite-time convergence and working under asynchronous, time-varying and directed communication topologies. At every communication round, each processor maintains a “candidate” ellipsoid for the global problem and performs two tasks. First, it verifies—in a probabilistic sense—if the center of the candidate ellipsoid is robustly feasible for its local set and, if not, constructs a new ellipsoid with smaller volume. Second, it exchanges its ellipsoid with neighbors, and then selects the one with smallest volume among the collected ones. We show that in a finite number of communication rounds, the processors reach consensus on a common ellipsoid whose center is—with high confidence—feasible for the entire set of uncertainty except a subset having an arbitrary small probability measure. We corroborate the theoretical results with numerical computations in which the algorithm is tested on a multi-core platform of processors communicating asynchronously.

I. INTRODUCTION

Distributed optimization has recently gained significant attention. In this line of research, networks of processors with limited computation/communication capabilities are used to solve “large-scale” optimization problems. There is no central node having full knowledge of the entire problem due to memory, computational power and/or privacy constraints; each node knows only its own constraint set.

Introducing uncertainty in a distributed optimization framework adds more complexity to the problem. There are few papers considering uncertainty in the distributed optimization framework. In [14], a synchronous distributed random projection algorithm with almost sure convergence is proposed for the case where each node knows its local cost function and (uncertain) constraint. The synchronization of local update rules relies on a central clock to coordinate the step size selection. To circumvent this limitation,

M. Chamanbaz is with Department of Electrical and Computer Engineering, Arak University of Technology, Daneshgah Street, Arak, Iran. (email: Chamanbaz@arakut.ac.ir, Chamanbaz@u.nus.edu).

G. Notarstefano is with Department of Engineering, Control Optimization and Robotics lab, Università del Salento, Via Monteroni 73100 Lecce, Italy. (email: giuseppe.notarstefano@unisalento.it).

R. Bouffanais is with Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372 (e-mail: bouffanais@sutd.edu.sg).

This work is supported by European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme, grant agreement No 638992 - OPT4SMART, (GN), the National Research Foundation (NRF), Prime Ministers office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-40) and administered by the National Cybersecurity R&D Directorate(RB), and a grant from National Elites Foundation, Iran (MC).

the same authors in [15] present an asynchronous random projection algorithm in which a gossip-based protocol is used to desynchronize the step size selection. A distributed method based on the scenario approach [7], [8] is introduced in [16] in which random samples are extracted from the uncertain constraint set and a proximal minimization algorithm is used for solving the sampled optimization problem in a distributed way. The number of samples required to guarantee robustness can be large if the probabilistic levels defining robustness of the solution—accuracy and confidence levels—are stringent, thereby (possibly) leading to a computationally demanding sampled-optimization problem at each node. A parallel framework for solving convex optimization problems with one uncertain constraint via the scenario approach has been recently proposed in [19]. In this setup, the sampled-optimization problem is solved by using a primal-dual subgradient (resp. random projection) algorithm over undirected (resp. directed) graphs. We remark that in [19], constraints and cost function of all agents are identical. In [5], a cutting plane consensus algorithm is introduced for solving convex optimization problems where constraints are distributed throughout the network and all processors have a common cost function. If constraints are uncertain, a pessimizing oracle is used for solving the problem under the assumption that constraints are concave with respect to uncertainty vector, while the uncertain set is convex.

In this paper, we propose an *asynchronous* distributed algorithm, based on the centralized ellipsoid method [12], to solve convex feasibility problems (affected by a possibly nonlinear uncertain vector) in a multi-agent network. The centralized algorithm has been discussed in a series of papers including [2], [13], [18]. Here, we consider a *directed* and *time-varying* communication network in which each node has an *uncertain* constraint. The objective of this network system is to reach a consensus—using purely local computation and communication steps—on a solution which robustly satisfies all the constraints scattered across different nodes.

The main ingredients of the proposed algorithm, besides the ellipsoid method [12], are: a (centralized) sequential randomized algorithm, [1], [6], [9], [10], and a constraint-exchange approach for distributed optimization, [17]. The ellipsoid algorithm is used locally at each node to update the candidate solution. Specifically, at every iteration of the algorithm, each node constructs an ellipsoid, which is guaranteed to contain the globally robust feasible set—a set being robustly feasible for all constraints. The center of this ellipsoid is a candidate solution. Feasibility of the candidate solution is checked through a randomized algorithm—

performed locally at each node—by extracting a number of random samples from the set of uncertainty and checking the feasibility of the constraint for the extracted samples. Then, each node collects ellipsoids from in-neighbors and selects the one with smallest-volume. This procedure is performed until all nodes reach a consensus on a common ellipsoid. We prove that if the globally robust feasible set has a nonempty interior, the proposed algorithm converges *in finite time* to a solution being robustly feasible—with high confidence—for the entire set of uncertainty except for a subset having an arbitrary small probability measure.

The effectiveness of this approach is illustrated by means of numerical computations on a distributed (convex) position estimation problem in wireless sensor networks. The centralized problem is formulated in [11]. In particular, we consider a heterogeneous scenario in which (a possibly large) network of sensors with positioning capability—reporting sensor position with some uncertainty—run the distributed algorithm to localize sensors with no positioning capability—equipped only with radio-frequency transmitters.

The rest of the paper is organized as follows. In Section II, the distributed robust feasibility problem is formulated. The distributed randomized algorithm is presented in Section III. The convergence of the proposed randomized algorithm is proved in Section IV. Lastly, an example regarding distributed convex position estimation in a wireless sensor network is included in Section V.

II. PROBLEM FORMULATION AND PRELIMINARIES

We consider a network of processors with limited computation and communication capabilities. Processors communicate according to a time-varying, directed communication graph $\mathcal{G}(t) = \{\{1, \dots, n\}, \mathcal{E}(t)\}$, where $t \in \mathbb{N}$ is a universal time and $\mathcal{E}(t)$ is the set of communication links at time t . Specifically, $(i, j) \in \mathcal{E}(t)$ indicates that i sends information to j . The time-varying set of incoming (resp. outgoing) neighbors of node i at time t , $\mathcal{N}_{\text{in}}(i, t)$ (resp. $\mathcal{N}_{\text{out}}(i, t)$), is defined as the set of nodes from (resp. to) which agent i receives (resp. transmits) information at time t . A directed static graph is said to be *strongly connected* if there exists a directed path (of consecutive edges) between any pair of nodes in the graph. For time-varying graphs we use the notion of *uniform joint strong connectivity* defined next.

Assumption 1 (Uniform joint strong connectivity):

There exists an integer $L \geq 1$ such that the graph $\left(\{1, \dots, n\}, \bigcup_{\tau=t}^{t+L-1} \mathcal{E}(\tau)\right)$ is strongly connected $\forall t \geq 0$.

Processors aim at cooperatively solving the following robust convex feasibility problem

$$\begin{aligned} & \text{find } \theta \\ & \text{subject to: } \theta \in \bigcap_{i=1}^n \Theta_i(q), \quad \text{for all } q \in \mathbb{Q}, \end{aligned} \quad (1)$$

where $\theta \in \mathbb{R}^d$ is the vector of decision variables, \mathbb{Q} is the set of uncertainty and $\Theta_i(q) \doteq \{\theta \in \mathbb{R}^d : f_i(\theta, q) \leq 0\}$,

with $f_i(\cdot, q) : \mathbb{R}^d \rightarrow \mathbb{R}$, is a constraint set known by agent $i \in \{1, \dots, n\}$ only. We refer to $\Theta_i \doteq \{\theta \in \mathbb{R}^d : \theta \in \Theta_i(q), \forall q \in \mathbb{Q}\}$ as the robust feasible set corresponding to agent i . Consistently, the set $\Theta \doteq \{\theta \in \mathbb{R}^d : \theta \in \bigcap_{i=1}^n \Theta_i(q), \forall q \in \mathbb{Q}\}$ is the globally robust feasible set.

We make the following assumption on the constraint sets.

Assumption 2 (Convexity): For any fixed value of $q \in \mathbb{Q}$, each set $\Theta_i(q)$, $i \in \{1, \dots, n\}$, is convex.

From the above assumption, it follows immediately that each robust feasible set Θ_i as well as the globally robust feasible set Θ are convex. We remark that, there is no assumption on how uncertainty enters problem (1) making it computationally difficult to solve. Since the uncertain set \mathbb{Q} is an uncountable set, (1) is a semi-infinite feasibility problem involving an infinite number of constraints. Deterministic methods for solving semi-infinite problems are not tractable in cases where uncertainty does not have a simple structure such as affine, multi-affine, or convex. For this reason, we follow a probabilistic approach in which uncertainty is considered as a random variable and constraints are enforced to hold for the entire set of uncertainty except for a subset having an arbitrary small probability measure.

III. DISTRIBUTED SEQUENTIAL RANDOMIZED ELLIPSOID ALGORITHM

In this section, we present a distributed, randomized algorithm for solving—in a probabilistic sense—the robust convex feasibility problem (1). The proposed *Distributed Sequential Randomized Ellipsoid Algorithm* consists of two main steps: a local “verification and shrinking” step and a “communication and selection” step. Specifically, each node has a candidate ellipsoid—containing the globally robust feasible set Θ —whose center is a candidate solution. In the “verification and shrinking” step, each node i first verifies if—with high probability—the candidate solution point belongs to its local robust feasible set Θ_i . If the verification is not successful, meaning that the candidate solution point does not belong to Θ_i , node i runs locally an ellipsoid algorithm based on [12] to shrink the volume of its candidate ellipsoid. Next, in the “communication and selection” step, the node receives ellipsoids from its in-neighbors and selects the one with smallest volume, including its own one. This second part is based on the fact that the ellipsoid of each node contains the globally robust feasible set Θ . The two steps are repeated until all nodes reach the same ellipsoid and hence the same solution point with a desired probabilistic properties in terms of robustness.

More formally, the verification and shrinking step consists of the following procedures. We suppose that there exists an “oracle” being able to declare—in a probabilistic sense—if $\theta_i \in \Theta_i$. Specifically, denote by θ_i the center of the current ellipsoid of node i . Checking if $\theta_i \in \Theta_i$, is not computationally easy in general. Hence, we resort to randomization and consider a randomized oracle being able to verify—with high probability—if $\theta_i \in \Theta_i$. To this end, we assume that q is a random variable and a probability measure \mathbb{P} over the

Borel σ -algebra of \mathbb{Q} is given. At each iteration, we first generate a number of independent and identically distributed (i.i.d) samples from the uncertain set \mathbb{Q} according to the measure \mathbb{P} and using a Monte Carlo approach, feasibility of the candidate solution is examined only at the extracted samples.

Then, if $\theta_i \notin \Theta_i$ the oracle generates a ‘‘separating hyperplane’’ \mathcal{H}^i that separates θ_i from Θ_i .

Given a $\theta_i \in \mathbb{R}^d$ and a positive definite matrix $P_i \in \mathbb{R}^{d \times d}$, consider the ellipsoid of node i

$$\mathbf{E}^i(\theta_i, P_i) = \{\theta \in \mathbb{R}^d : (\theta - \theta_i)^T P_i^{-1} (\theta - \theta_i) \leq 1\}. \quad (2)$$

Let q_{viol} be a point for which $f_i(\theta_i, q_{\text{viol}}) > 0$ (such a point is called ‘‘violation certificate’’), we generate a ‘‘deep-cut’’ separating hyperplane as in [2], based on a suitable subgradient $g_i(\theta_i, q_{\text{viol}})$ of $f_i(\theta, q)$ at point $(\theta_i, q_{\text{viol}})$:

$$\mathcal{H}^i = \{\theta \in \mathbb{R}^d : g_i(\theta_i, q_{\text{viol}})^T (\theta - \theta_i) + f_i(\theta_i, q_{\text{viol}}) \leq 0\}. \quad (3)$$

Hence, a smaller ellipsoid \mathbf{E}_+^i such that $\mathbf{E}^i \cap \mathcal{H}^i \subset \mathbf{E}_+^i$ is constructed. Define

$$\alpha_i(\theta_i, q_{\text{viol}}) = \frac{f_i(\theta_i, q_{\text{viol}})}{\sqrt{g_i(\theta_i, q_{\text{viol}})^T P_i g_i(\theta_i, q_{\text{viol}})}}. \quad (4)$$

The (unique) minimum volume ellipsoid containing the intersection of \mathbf{E}^i and \mathcal{H}^i can be constructed based on the following result from [2], which is a direct application of Theorem 2.1 in [12].

Theorem 1 ([2]): Consider the ellipsoid \mathbf{E}^i in (2) and the hyperplane \mathcal{H}^i in (3), the minimum volume ellipsoid

$$\mathbf{E}_+^i = \{\theta \in \mathbb{R}^d : (\theta - \theta_i^+)^T (P_i^+)^{-1} (\theta - \theta_i^+) \leq 1\},$$

containing $\mathbf{E}^i \cap \mathcal{H}^i$ can be constructed by setting

$$\theta_i^+ = \theta_i - \tau \frac{P_i g_i}{\sqrt{g_i^T P_i g_i}}, \quad P_i^+ = \eta \left(P_i - \sigma \frac{P_i g_i g_i^T P_i}{g_i^T P_i g_i} \right)$$

with $\tau = \frac{1+d\alpha_i}{d+1}$, $\eta = \frac{d^2}{d^2-1} (1 - \alpha_i^2)$, $\sigma = \frac{2(1+d\alpha_i)}{(d+1)(1+\alpha_i)}$.

Moreover, the volumes of two consecutive ellipsoids satisfy

$$\frac{\text{Vol}(\mathbf{E}_+^i)}{\text{Vol}(\mathbf{E}^i)} = (\eta^d (1 - \sigma))^{1/2} \leq \exp\left(-\frac{1}{2(d+1)} - \alpha_i\right),$$

where $\text{Vol}(\cdot)$ returns the volume of the ellipsoid input.

We define a primitive

$$\mathbf{E}_+^i = \text{ShrinkEllips}(\mathbf{E}^i, q_{\text{viol}}),$$

which for \mathbf{E}^i and q_{viol} constructs \mathbf{E}_+^i based on Theorem 1.

Next, in the ‘‘communication and selection’’ step, each agent updates its ellipsoid according to a deterministic function acting on the collection of ellipsoids available by the agent. That is, calling $\mathbf{E}^{j_1}, \dots, \mathbf{E}^{j_p}$ the collected ellipsoids, agent i computes

$$\mathbf{E}_+^i = \text{SelectMinvol}(\{\mathbf{E}^k\}_{k \in \{j_1, \dots, j_p\}}),$$

which uniquely returns a smallest volume ellipsoid among $\mathbf{E}^{j_1}, \dots, \mathbf{E}^{j_p}$. That is, we suppose that in case more ellipsoids with smallest volume are present, the function selects a unique one.

A pseudocode of our Distributed Randomized Ellipsoid Algorithm is presented in Algorithm 1. The counters ℓ_i and k_i count the number of times verification and local update through Theorem 1 are performed respectively.

Algorithm 1 Distributed Randomized Ellipsoid Algorithm

Input: $f_i(\theta, q)$, probabilistic levels $\varepsilon_i, \delta_i \in (0, 1)$, ellipsoid $\mathbf{E}_1^i \supset \Theta_i$

Output: \mathbf{E}_{sol}

Initialization: Set $\ell_i = 1$ and $k_i = 1$

Evolution:

(i) **Verification and Shrinking:**

- **If** $\mathbf{E}_t^i = \mathbf{E}_{t-1}^i$, **goto** (ii)
- $\ell_i = \ell_i + 1$
- Extract

$$M_{\ell_i} \geq \frac{2.3 + 1.1 \ln \ell_i + \ln \frac{1}{\delta_i}}{\ln \frac{1}{1-\varepsilon_i}}$$

i.i.d samples $\mathbf{q}_{\ell_i} = \{q_{\ell_i}^{(1)}, \dots, q_{\ell_i}^{(M_{\ell_i})}\}$

- **If** $f_i(\theta_i(t), q_{\ell_i}^{(j)}) \leq 0, \forall j \in \{1, \dots, M_{\ell_i}\}$, **goto** (ii); **else**
- set q_{viol} as first sample for which $f_i(\theta_i(t), q_{\ell_i}^{(j)}) > 0$
- $\mathbf{E}_t^i = \text{ShrinkEllips}(\mathbf{E}_t^i, q_{\text{viol}})$
- Set $k_i = k_i + 1$

(ii) **Communication and Selection:**

- $\mathbf{E}_{t+1}^i = \text{SelectMinvol}(\{\mathbf{E}_t^j\}_{j \in \mathcal{N}_{\text{in}}(i,t) \cup i})$
 - **If** \mathbf{E}_{t+1}^i has not changed for $2nL + 1$ times, **return** $\mathbf{E}_{\text{sol}} = \mathbf{E}_{t+1}^i$
-

At this point, it is worth highlighting three main interesting features of the proposed algorithm. First, the local computation at each node involves fairly inexpensive steps. As for verification, even if it may run possibly for a large number of times, it consists of simple, inexpensive inequality checks. The volume shrinking involves simple vector and matrix computations as for Theorem 1, and the selection part consists of just checking the minimum volume among a number of ellipsoids equal to the in-degree of the node. Second, the amount of data required to be transmitted between processors does not depend on the number of agents, but only on the dimension of the space. In particular, each node needs to only transmit a $d \times 1$ vector θ corresponding to the ellipsoid center and a $d \times d$ symmetric matrix P corresponding to the ellipsoid shape matrix. Considering the fact that P is a symmetric matrix, only $d + \frac{d(d+1)}{2}$ numbers need to be transmitted. Third and final, the proposed distributed randomized algorithm is completely asynchronous. In fact, t is just a universal time that does not need to be known by the nodes, and the graph can be time-varying. Thus, if nodes run the computation at different speeds, this can be modeled by having no incoming and outgoing edges in that time interval.

IV. ANALYSIS OF THE DISTRIBUTED RANDOMIZED ELLIPSOID ALGORITHM

We now state a Theorem proving the convergence of Algorithm 1 and quantifying properties of the obtained solution. That is, we prove that all nodes agree on a solution which is globally feasible—with high confidence—for the entire set of uncertainty \mathbb{Q} , except a subset having an arbitrary small probability measure.

Theorem 2 (Algorithm Convergence and Solution Properties): Suppose a network of N agents as in Section II runs Algorithm 1, and let Assumptions 1 and 2 hold. Given the probabilistic levels $\varepsilon_i > 0$ and $\delta_i > 0$, $i \in \{1, \dots, n\}$, let $\varepsilon = \sum_{i=1}^n \varepsilon_i$ and $\delta = \sum_{i=1}^n \delta_i$. Then, the following statements hold

(i) *(Finite-time infeasibility detection)*

If there exists some agent i for which the cumulative number, k_i , of local updates at time t exceeds

$$\bar{k}_i = \left\lceil \frac{2(d+1) \ln \frac{\text{Vol}(\mathbf{E}_i^t)}{\Phi_{r^d}}}{2\alpha_i^{\min}(d+1) + 1} \right\rceil,$$

where α_i^{\min} is the minimum¹ value of α_i —defined in (4)— Φ is the volume of the d -dimensional unit ball and $\lceil x \rceil$ denotes the smallest integer greater than or equal to x . Then, problem (1) is not r -feasible in the sense that its feasible set Θ does not include a full-dimensional ball of radius $r > 0$.

(ii) *(Finite-time convergence for strictly feasible problems)*

If problem (1) is r -feasible, then ellipsoids of all nodes will converge to a common ellipsoid in finite time. That is, there exists $T > 0$ such that $\mathbf{E}_i^t = \mathbf{E}_{\text{sol}}$ for all $i \in \{1, \dots, n\}$ and $t > T$.

(iii) *(Local stopping condition of the distributed algorithm)*

If the ellipsoid of agent i has not changed for $2nL + 1$ communication rounds, then all nodes have a common ellipsoid \mathbf{E}_{sol} .

(iv) *(Probabilistic properties of the solution)*

Let θ_{sol} be the center of \mathbf{E}_{sol} , then the following inequality holds

$$\mathbb{P}^M \left\{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P} \left\{ q \in \mathbb{Q} : \theta_{\text{sol}} \notin \bigcap_{i=1}^n \Theta_i(q) \right\} \leq \varepsilon \right\} \geq 1 - \delta$$

with M being the cardinality of the collection of all agents' multisamples.

The proof is omitted for the sake of space and will be provided in a forthcoming document.

Remark 1 (Extension to Optimization): Extension of Algorithm 1 to optimization problems is not trivial. In a centralized setup one can use an “objective cut”, which removes part of the robust feasible set Θ with larger objective value than the query point. However, in a distributed framework

¹As stated in [2][Remark 1], α_i^{\min} can be hard to estimate beforehand, therefore a running estimate can be used.

agents do not have full knowledge of the (intersection) set Θ , thus an objective cut cannot be performed locally by the agents. A naïve way to extend the distributed framework presented here to solve optimization problems, is to perform the objective cut only if all nodes have converged to the same ellipsoid whose center is globally probabilistically feasible. This ensures that only the part of Θ with larger objective than the ellipsoid's center is removed from the search space. The drawback of this approach is that all nodes need to converge in order to perform a new objective cut.

V. APPLICATION EXAMPLE: DISTRIBUTED LOCALIZATION IN WIRELESS SENSOR NETWORKS

To illustrate the distributed algorithm presented in Section III, we solve the problem of distributed convex position estimation in wireless sensor networks. A centralized solution to this problem is originally formulated in [11].

Consider a two-dimensional² space containing m heterogeneous wireless sensors which are randomly placed over a given environment. One can think of a scenario where all sensors are dropped from the air and scattered about an unknown area. The sensors are of two classes:

- (i) Wireless sensors capable of positioning themselves up to a given accuracy, i.e., there is uncertainty in the position reported by this class of sensors. They are equipped with processors with limited computation/communication capabilities and play the role of computational nodes in the distributed optimization framework. These n sensors can communicate with each other based on a metric distance. That is, two sensors which are close enough can establish a bidirectional communication link. Henceforth, these sensors are called “known sensors”.
- (ii) Wireless sensors with no positioning capabilities whatsoever. These $m - n$ sensors are only equipped with a short-range transmitter having a rotationally symmetric communication range, periodically transmitting an ID, specific to each sensor. Henceforth, they are called “unknown sensors”.

The objective of the network is to “estimate”—in a distributed fashion—the position of unknown sensors. We further consider a heterogeneous setup in which some of the sensors with positioning capabilities are equipped with laser transceiver providing an estimate of relative angle to unknown sensors which are within the range of the laser transceiver. This “angular constraint” can be represented by the intersection of three half-spaces, two to bound the angle and one corresponding to the transceiver range as presented in Fig. 1. Sensors with known position, which are not equipped with laser transceiver, can place a “radial constraint” for those unknown sensors that are within the communication range. If the signal from an unknown sensor can be sensed by a known sensor, a proximity constraint exists between them, that is, the unknown sensor is placed in

²Here we consider a two-dimensional space, nevertheless extension to a three-dimensional space is straightforward.

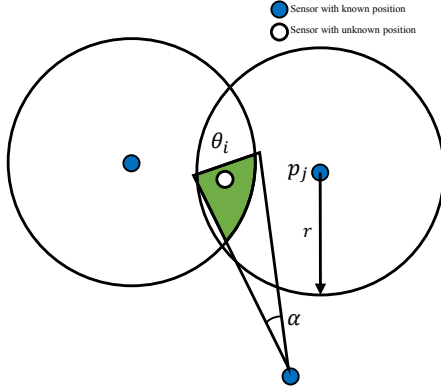


Fig. 1. Sensors with known position are computational nodes in the distributed optimization setup. Their objective is to estimate the position of unknown sensors by locally applying angular and radial constraints and exchanging information with their neighbors. The green (shaded) region represents the feasible set of unknown positions.

a circle with radius r centered at the known sensor's position with r being the communication range. This constraint can be formulated as $\|\theta_i - p_j\|_2 \leq r$, where $\theta_i \in \mathbb{R}^{2 \times 1}$ is the position of i th unknown sensor and $p_j \in \mathbb{R}^{2 \times 1}$ is the position of j th known sensor. Using Schur-complement [4], the proximity constraint can be expressed in terms of a linear matrix inequality (LMI), i.e.,

$$\|\theta_i - p_j\|_2^2 \leq r^2 \Leftrightarrow F(\theta_i) \doteq \begin{bmatrix} rI_2 & (\theta_i - p_j) \\ (\theta_i - p_j)^T & r \end{bmatrix} \geq 0,$$

where I_2 is the identity matrix of size 2×2 .

In order to solve this problem in a distributed manner, all unknown positions are collected in $\theta \in \mathbb{R}^{(m-n) \times 2}$, i.e., $\theta \doteq [\theta_1, \dots, \theta_{m-n}]^T$ —recall that m is the total number of sensors and n is the number of sensors with known positions. As shown in Fig. 1, the position of unknown sensors are often sensed and hence constrained by several known sensors. This is used to shrink the feasibility set of unknown positions. We remark that in a distributed setup, constraints regarding the position of an unknown sensor are distributed in different computational nodes and there is no agent having full knowledge of all constraints.

Known sensors are capable of localizing themselves. Nevertheless, their localization is not accurate and there is an uncertainty associated with the position vector reported by each node. This uncertainty can be modeled as a norm-bounded vector

$$\|p_j - \bar{p}_j\|_2 \leq \rho, \quad (5)$$

where \bar{p}_j is the nominal position reported by the sensor j , p_j is its actual position and ρ is the radius of the uncertain set. In other words, the uncertainty is defined by an ℓ_2 ball centered at the nominal point \bar{p}_j with radius ρ .

For simulation purposes, we consider a scenario where 30 sensors are placed randomly in a 10×10 square environment and only 10 of them are capable of positioning themselves up

to a given accuracy. The sensors with known position have processing and communication capabilities. In particular, two sensors can establish a two-way communication link if their distance is less than 3. The communication range of unknown sensors is also considered to be 3. Among the 10 known sensors, half of them are able to place angular constraints—using laser transceiver—on their relative angle to unknown sensors within the laser range. The angle representing the accuracy of the laser transceiver—the angle α in Fig. 1—is 20 degrees. Finally, the uncertainty radius ρ in (5) is 0.05 and the distribution of the uncertainty is selected to be uniform due to its worst-case nature [3]. We remark that in the considered scenario, the dimension of the decision variable θ is 40 and the number of computation nodes n is 10; there are also 20 uncertain parameters rendering the problem nontrivial to solve. The design parameters ε_i and δ_i , $i = 1, \dots, 10$ are selected to be 0.01 and 10^{-10} respectively, leading to $\varepsilon = 0.1$ and $\delta = 10^{-9}$. The initial ellipsoid is selected to be a 40 dimensional ball of radius 10 centered at the point (5, 5).

We used a workstation with 12 cores and 48 GB RAM to emulate the distributed framework. Each one of the 10 computational nodes runs Algorithm 1 in an independent Matlab environment and communication is performed by sharing files between different Matlab environments. This accounts for asynchronicity of Algorithm 1. Figure 2 shows the evolution of ellipsoid volume and the distance of the ellipsoid center from a probabilistically robust feasible point (θ_{sol}) for all the nodes. As expected, volumes are non-increasing over time and they all converge to a common value. The center of each node ellipsoid is its candidate solution. It can be seen from Fig. 2 that the candidate solutions of all nodes converge to the same point as well. The insets in Fig. 2 also show that both the volumes and centers of the ellipsoids converge in finite time, consistently with the theoretical result proven in Theorem 2. As a final remark, it is worth pointing out once again that the horizontal axis of Fig. 2 represents a universal time which does not need to be known by the nodes.

VI. CONCLUSION

In this paper, we proposed an asynchronous, randomized distributed algorithm for solving robust convex feasibility problems in which the uncertain constraint sets are distributed in a network of processors communicating according to a directed and time-varying graph. The distributed algorithm, run by the network nodes, relies on the centralized ellipsoid algorithm [12]. Each node has a candidate ellipsoid guaranteed to contain the global robust feasible set—the set being robustly feasible for all the nodes. Each node first verifies—in a Monte Carlo setup—if the center of its ellipsoid is feasible with high probability. If not, it updates its ellipsoid with a smaller one by using the ellipsoid algorithm [12]. Next, it collects ellipsoids of all its neighbors and updates its candidate ellipsoid with the smallest volume among the collected ones. The two steps are iteratively performed until all nodes converge to the same

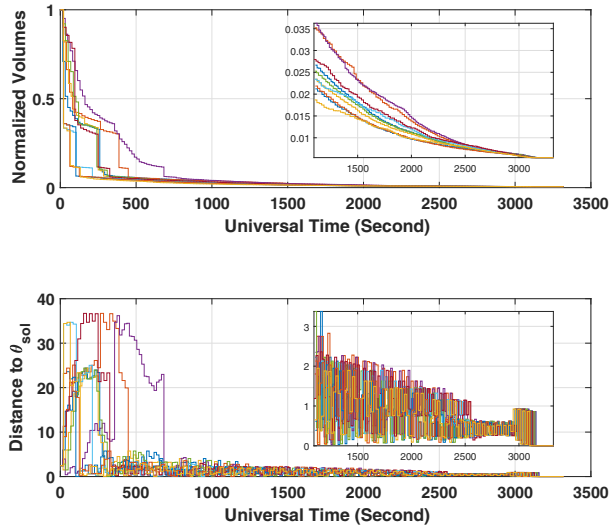


Fig. 2. Top: Volume of ellipsoids of all the nodes along the evolution of the distributed algorithm (normalized with volume of the initial ellipsoid). Bottom: Distance of the center of ellipsoids (candidate solutions) to θ_{sol} for all nodes in the network.

candidate ellipsoid with the desired probabilistic robustness. We analyzed the finite-time convergence of the proposed algorithm and showed its performance on a robust position estimation problem in wireless sensor networks.

REFERENCES

- [1] T. Alamo, R. Tempo, A. Luque, and D.R. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52:160–172, 2015.
- [2] A. Ataei and Q. Wang. A probabilistic ellipsoid algorithm for linear optimization problems with uncertain LMI constraints. *Automatica*, 52:248–254, 2015.
- [3] E. Bai, R. Tempo, and M. Fu. Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis. *Mathematics of Control, Signals and Systems*, 11:183–196, 1998.
- [4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] M. Bürger, G. Notarstefano, and F. Allgöwer. A polyhedral approximation framework for convex and robust distributed optimization. *IEEE Transactions on Automatic Control*, 59:384–395, 2014.
- [6] G. Calafiore and F. Dabbene. A probabilistic analytic center cutting plane method for feasibility of uncertain lmis. *Automatica*, 43:2022–2033, 2007.
- [7] G.C. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46, 2004.
- [8] G.C. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51:742–753, 2006.
- [9] M. Chamanbaz, F. Dabbene, R. Tempo, V. Venkataramanan, and Q. G. Wang. Sequential randomized algorithms for convex optimization in the presence of uncertainty. *IEEE Transactions on Automatic Control*, 61:2565–2571, 2016.
- [10] F. Dabbene, P. Shcherbakov, and B. Polyak. A randomized cutting plane method with probabilistic geometric convergence. *SIAM Journal on Optimization*, 20:3185–3207, 2010.
- [11] L. Doherty, K. S. J. pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *Proc. 20th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1655–1663, 2001.
- [12] D. Goldfarb and M.J. Todd. Modifications and implementation of the ellipsoid algorithm for linear programming. *Mathematical Programming*, 23(1):1–19, 1982.
- [13] S. Kanev, B. De Schutter, and M. Verhaegen. An ellipsoid algorithm for probabilistic robust controller design. *Systems & Control Letters*, 49:365–375, 2003.
- [14] S. Lee and A. Nedić. Distributed random projection algorithm for convex optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7:221–229, 2013.
- [15] S. Lee and A. Nedić. Asynchronous gossip-based random projection algorithms over networks. *IEEE Transactions on Automatic Control*, 61:953–968, 2016.
- [16] K. Margellos, A. Falsone, S. Garatti, and M. Prandini. Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *arXiv preprint arXiv:1603.02239*, 2016.
- [17] G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: Theory and applications. *IEEE Transactions on Automatic Control*, 56(10):2247–2261, 2011.
- [18] T. Wada and Y. Fujisaki. Sequential randomized algorithms for robust convex optimization. *IEEE Transactions on Automatic Control*, 60:3356–3361, 2015.
- [19] K. You and R. Tempo. Networked Parallel Algorithms for Robust Convex Optimization via the Scenario Approach. *arXiv preprint arXiv:1607.05507*, 2016.