

A Distributed Ellipsoid Algorithm for Uncertain Convex Problems: A Randomized Approach

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 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 [17], a synchronous distributed random projection algorithm with almost sure convergence is proposed for the case where each node has independent 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, the same authors in [18] 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 [19] in which random samples are extracted from the uncertain constraint set and a proximal minimization algorithm is used to solve 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 [22]. 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 [22], 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 [14], 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], [16], [21]. Here, we consider a *directed* and *time-varying* communication network in which each node has an *uncertain* constraint. The objective of this networked 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 [14], are: a (centralized) sequential randomized algorithm, [1], [6], [10], [12], and a constraint-exchange approach for distributed optimization, [20]. The ellipsoid algorithm is used locally at each node to update the candidate solution. At each iteration of the algorithm, an ellipsoid is constructed, 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—being performed locally at each node—by extracting a number of random samples from the set of uncertainty and checking the feasibility of

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

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).

This work is supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme, grant agreement No 638992 - OPT4SMART, (GN) and by a grant from the Singapore National Research Foundation (NRF) under the ASPIRE project, grant No NCR-NCR001-040 (MC&RB).

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. Finally, 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 [13]. In particular, we consider a heterogeneous scenario in which few sensors with positioning capability—reporting sensor position with some uncertainty—are used as computational nodes to estimate the position of a large number of sensors—equipped only with radio-frequency transmitters—with no positioning capability but having only proximity constraints.

The rest of 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. Some concluding remarks are gathered in Section VI.

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 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}(t)\}$ where $t \in \mathbb{N}$ is a universal time and $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ 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* formally defined next.

Assumption 1 (Uniform joint strong connectivity):

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

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

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 $q \in \mathbb{Q}$, $\Theta_i(q) \doteq \{\theta \in \mathbb{R}^d : f_i(\theta, q) \leq 0\}$ is convex.

From the above assumption it follows immediately that each robust feasible set Θ_i is convex as well as the globally robust feasible set Θ . 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 METHOD

In this section, we present a distributed, randomized algorithm for solving—in a probabilistic sense—the robust convex feasibility problem (1). The distributed algorithm presented here consists of two main steps: “verification and local update” and “communication”. The main idea is the following. Each node has a candidate ellipsoid—containing the globally robust feasible set Θ —whose center is a candidate solution. The algorithm 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 , the node runs locally an ellipsoid algorithm based on [14] to shrink the volume of its candidate ellipsoid. Next, based on the fact that the ellipsoid of each node contains the globally robust feasible set Θ , the node receives ellipsoids from its in-neighbors and selects the one with smallest volume (including its own one). 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.

Consider the ellipsoid of node i

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

and suppose that there exists an “oracle” being able to declare if $\theta_i \in \Theta_i$. Also, if $\theta_i \notin \Theta_i$ the oracle generates a “separating hyperplane” \mathcal{H}^i that separates θ_i from Θ_i . Hence, a smaller ellipsoid \mathbf{E}_+^i can be constructed such that $\mathbf{E}^i \cap \mathcal{H}^i \subset \mathbf{E}_+^i$. Such a separating hyperplane can be constructed based on the subgradient of $f_i(\theta, q)$

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

where $g_i(\theta_i, q_{\text{viol}})$ is a subgradient of $f_i(\theta, q)$ at point $(\theta_i, q_{\text{viol}})$ with q_{viol} being a point for which $f_i(\theta_i, q_{\text{viol}}) > 0$; such a point is called “violation certificate”. Moreover, 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 minimum volume ellipsoid containing the intersection of \mathbf{E}^i and \mathcal{H}^i can be constructed based on the following result from [14].

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

$$\mathbf{E}_+^i = \{\theta \in \mathbb{R}^d : (\theta - \theta_i^+)^T (P_i^+)^{-1} (\theta - \theta_i^+) \leq 1\}, \quad (5)$$

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

$$\theta_i^+ = \theta_i - \tau \frac{P_i g_i}{\sqrt{g_i^T P_i g_i}}, \quad (6)$$

$$P_i^+ = \eta \left(P_i - \sigma \frac{P_i g_i g_i^T P_i}{g_i^T P_i g_i} \right) \quad (7)$$

with

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

Furthermore, the volumes of two consecutive ellipsoids satisfy

$$\frac{\text{Vol}(\mathbf{E}_+^i)}{\text{Vol}(\mathbf{E}^i)} = (\eta^d (1 - \sigma))^{1/2} \leq \exp\left(-\frac{1}{2(d+1)} - \alpha_i\right), \quad (8)$$

where Vol returns the volume of the ellipsoid input.

We define a primitive $\mathbf{E}_+^i = \text{EllUpdate}(\mathbf{E}^i, q_{\text{viol}})$, which given \mathbf{E}^i and q_{viol} , constructs \mathbf{E}_+^i based on Theorem 1. We highlight that the cut (3) defines a “deep cut” while

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

represents a “neutral” cut. For neutral cuts, the parameter α_i in (8) is zero.

Denote by $\theta_i(t)$ the center of ellipsoid of node i at time t . Checking if $\theta_i(t) \in \Theta_i$ is not computationally easy in general. Hence, we resort to using randomization and consider a randomized oracle being able to verify—with high probability—if $\theta_i(t) \in \Theta_i$. 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. Next, each agent updates its ellipsoid to the smallest volume ellipsoid among its own ellipsoid and the ones from neighbors

$$\mathbf{E}_{t+1}^i = \text{minvol}(\{\mathbf{E}_t^j\}_{j \in \mathcal{N}_{\text{in}}(i,t)} \cup i) \quad (9)$$

where $\text{minvol}(\mathbf{E}^1, \dots, \mathbf{E}^p)$ finds the smallest volume ellipsoid in $\mathbf{E}^1, \dots, \mathbf{E}^p$. A distributed algorithm is presented

in Algorithm 1 which finds a solution that is—with high confidence—globally feasible for the entire set of uncertainty, except a subset having an arbitrary small probability measure.

Algorithm 1 Distributed Randomized Ellipsoid Algorithm

Input: $f_i(\theta, q)$, the probabilistic levels $\varepsilon_i, \delta_i \in (0, 1)$ and an 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 local update:**

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

$$M_{\ell_i} \geq \frac{2.3 + 1.1 \ln \ell_i + \ln \frac{1}{\delta_i}}{\ln \frac{1}{1 - \varepsilon_i}} \quad (10)$$

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

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

(ii) **Communication:**

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

The counters ℓ_i and k_i count the number of times verification and local update through Theorem 1 are performed respectively.

Remark 1 (Communication Load): The amount of data required to be transmitted between processors is very limited. 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. This number is considerably smaller than e.g. [5], [9]. For instance, for a problem with $d = 100$, we only need to transmit 5,150 numbers while the algorithm in [5] needs to transmit twice as many—exactly 10,100—numbers between processors.

Remark 2 (Asynchronicity): The presented distributed randomized algorithm is completely asynchronous. In particular, the time t is just a universal time that does not need to be known by the nodes.

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. This can be regarded as the main result of this paper.

Theorem 2 (Convergence and Properties of the solution): Let Assumptions 1 and 2 hold. Given the probabilistic levels ε_i and δ_i , $i = 1, \dots, n$, let $\varepsilon = \sum_{i=1}^n \varepsilon_i$ and $\delta = \sum_{i=1}^n \delta_i$. Then, the following statements hold

- (i) If the cumulative number of local updates performed by agent i at time t (k_i) exceeds

$$\bar{k}_i \leq \frac{2(d+1) \ln \frac{\text{Vol}(\mathbf{E}_1^i)}{\Phi r^d}}{2\alpha_i^{\min}(d+1) + 1}, \quad (11)$$

where α_i^{\min} is the minimum¹ value of α_i —defined in (4)—along the evolution of Algorithm 1 and Φ is the volume of the d -dimensional unit ball then, Problem (1) is not strictly feasible in the sense that its feasible set Θ does not include a full-dimensional ball of radius r .

- (ii) If Problem (1) is strictly feasible and all nodes perform Algorithm 1, then they will converge to a common ellipsoid in finite time. That is, $\mathbf{E}_t^i = \mathbf{E}_{\text{sol}}$ for all $i \in \{1, \dots, n\}$ and $t > T$.
- (iii) If the ellipsoid of agent i is not changed for $2nL + 1$ communication rounds then, all nodes have a common ellipsoid \mathbf{E}_{sol} .
- (iv) Let θ_{sol} be the center of \mathbf{E}_{sol} , then the following inequality holds

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

with M being the cardinality of collection of multi-samples of all agents.

Proof:

proof of the first statement

We prove by contradiction. Assume that the global feasible set Θ contains a full dimensional ball of radius r and the cumulative number of local updates performed by agent i (k_i) at some time T exceeds \bar{k}_i . In Algorithm 1, ellipsoid \mathbf{E}_t^i is updated through communication—using the `minvol` operator—and through local update using the `EllUpdate` primitive. The `minvol` operator does not increase volume of the updated ellipsoid because it selects the smallest ellipsoid among all incoming neighbors and the one from the agent. The initial ellipsoid \mathbf{E}_1^i is guaranteed to contain the robust feasible set of agent i (Θ_i) and consequently the globally robust feasible set Θ (we note that $\Theta = \bigcap_{i=1}^n \Theta_i$). By construction, the ellipsoid update through Theorem 1 does not remove any part of Θ_i . The ellipsoid update through communication—using the `minvol` operator—also ensures that the updated ellipsoid contains the globally robust feasible set Θ because, it contains the intersection of ellipsoids. Therefore, by construction, the globally robust feasible set Θ must be included in the candidate ellipsoid of all agents at any time, which implies

$$\text{Vol}(\mathcal{B}_d(r)) \leq \text{Vol}(\Theta) \leq \text{Vol}(\mathbf{E}_t^i), \quad \forall t \geq 0, \quad i \in \mathcal{V}, \quad (12)$$

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

where $\mathcal{B}_d(r)$ is the d -dimensional ball of radius r . Any local update through Theorem 1, decreases the volume of the candidate ellipsoid. In particular, the volumes of the two consecutive ellipsoids satisfy (8). Denote by T the time in which the cumulative number of updates performed by agent i (k_i) exceeds \bar{k}_i , e.g. set $k_i = \bar{k}_i + 1$. Applying (8) recursively gives the ratio of volume reduction for the sequence of ellipsoids

$$\begin{aligned} \frac{\text{Vol}(\mathbf{E}_T^i)}{\text{Vol}(\mathbf{E}_1^i)} &\leq \exp\left(-\frac{\bar{k}_i + 1}{2(d+1)} - (\bar{k}_i + 1)\alpha_i^{\min}\right) \\ &< \exp\left(-\frac{\bar{k}_i}{2(d+1)} - \bar{k}_i\alpha_i^{\min}\right) \\ &\leq \frac{\Phi r^d}{\text{Vol}(\mathbf{E}_1^i)} = \frac{\text{Vol}(\mathcal{B}_d(r))}{\text{Vol}(\mathbf{E}_1^i)}, \end{aligned}$$

where the third inequality is obtained by substituting \bar{k}_i with (11). Based on this inequality, the volume of the candidate ellipsoid at time T is smaller than the volume of a full dimensional ball of radius r , that is

$$\text{Vol}(\Theta) \geq \text{Vol}(\mathcal{B}_d(r)) > \text{Vol}(\mathbf{E}_T^i). \quad (13)$$

One can observe that (13) contradicts (12). Therefore, if the cumulative number of local updates exceeds \bar{k}_i at any node i , we can conclude that the globally robust feasible set Θ does not include a full dimensional ball of radius r .

proof of the second statement

We first note that the operator `minvol` does not generate a new ellipsoid in the network. Instead, it selects a minimum volume ellipsoid among the existing ellipsoids. Also, since Problem (1) is strictly feasible, the number of times the candidate ellipsoid of each agent can be locally updated through Theorem 1 is bounded by \bar{k}_i . Therefore, there exists a finite time T after which no agent locally updates its ellipsoid. We note that if there does not exist such a finite time T , there exists a node whose candidate ellipsoid needs to be updated infinitely many times which contradicts the strict feasibility assumption. For $T > t$, since no further local update is possible, agents need to reach a consensus by using the `minvol` operator (9). That is, each agent needs to select the smallest ellipsoid among its own ellipsoid and the ones from neighbors which can be performed in finite time. Therefore, all the nodes converge to a common ellipsoid in finite time.

proof of the third statement

Consider nodes i and p , since the graph is uniformly jointly strongly connected, for any pairs of nodes u and v and for any $t \geq 0$, there exists a time-dependent path from u to v [15]—a path $u, \ell_1, \dots, \ell_k, v$ such that $(\ell_t, \ell_{t+1}) \in \mathcal{E}(t+1)$ —of length at most nL . We recall that n is the number of nodes and L is defined in Assumption 1. If $\ell_1 \in \mathcal{N}_{\text{out}}(i, t_0)$ then $\text{Vol}(\mathbf{E}_{t_0+1}^{\ell_1}) \leq \text{Vol}(\mathbf{E}_{t_0}^i)$. Iterating this argument, we get $\text{Vol}(\mathbf{E}_{t_0+nL}^p) \leq \text{Vol}(\mathbf{E}_{t_0}^i)$. Again, since the graph is uniformly jointly strongly connected, there is a time-varying path of length at most nL from node p to node i . Therefore,

$$\text{Vol}(\mathbf{E}_{t_0}^i) \geq \text{Vol}(\mathbf{E}_{t_0+nL}^p) \geq \text{Vol}(\mathbf{E}_{t_0+2nL}^i)$$

If $\text{Vol}(\mathbf{E}_{t_0}^i) = \text{Vol}(\mathbf{E}_{t_0+2nL}^i)$, that is the ellipsoid of agent i has not changed for $2nL + 1$ communication rounds and considering the point that node p can be any node of the graph, then all nodes have the same volume and hence the same ellipsoid.

proof of the fourth statement

We first note that using [10, Theorem 1], [6, Theorem 3] and [12, Theorem 5.3] we can show that—at any iteration t —if the sample size is selected based on (10) and the verification step is successful, that is $q_{\text{vio1}} = \emptyset$, then

$$\mathbb{P}^M \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{q \in \mathbb{Q} : \theta_i(t) \notin \Theta_i(q)\} \leq \varepsilon_i \} \geq 1 - \delta_i,$$

recall that $\theta_i(t)$ is the center of ellipsoid of node i at time t . The above inequality is a centralized result and holds only for the agent's own constraint $\Theta_i(q)$. Also since for θ_{sol} , the verification has to be successful, then

$$\mathbb{P}^M \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{q \in \mathbb{Q} : \theta_{\text{sol}} \notin \Theta_i(q)\} \leq \varepsilon_i \} \geq 1 - \delta_i. \quad (14)$$

Nevertheless, we are interested in bounding the probability by which $\theta_{\text{sol}} \notin \bigcap_{i=1}^n \Theta_i(q)$, i.e.

$$\mathbb{P}^M \left\{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{q \in \mathbb{Q} : \theta_{\text{sol}} \notin \bigcap_{i=1}^n \Theta_i(q)\} \leq \varepsilon \right\}. \quad (15)$$

Define the following events

$$\begin{aligned} \text{Bad}_i &\doteq \{ \theta_{\text{sol}} \notin \Theta_i(q), \forall q \in \mathbb{Q} \} \\ \text{Bad} &\doteq \{ \theta_{\text{sol}} \notin \bigcap_{i=1}^n \Theta_i(q), \forall q \in \mathbb{Q} \}. \end{aligned}$$

Equations (14) and (15) can be written in terms of the events Bad_i and Bad

$$\mathbb{P}^M \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{\text{Bad}_i\} \leq \varepsilon_i \} \geq 1 - \delta_i \quad (16)$$

$$\mathbb{P}^M \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{\text{Bad}\} \leq \varepsilon \} \quad (17)$$

respectively. One can observe that

$$\theta_{\text{sol}} \notin \bigcap_{i=1}^n \Theta_i(q) \Rightarrow \exists i \in \{1, \dots, n\} : \theta_{\text{sol}} \notin \Theta_i(q).$$

Hence, the event Bad can be written as the union of events Bad_i , $i = 1, \dots, n$, that is $\text{Bad} = \text{Bad}_1 \cup \text{Bad}_2 \cup \dots \cup \text{Bad}_n$. Invoking Boole's inequality [11] (also known as Bonferroni inequality), we have

$$\mathbb{P}\{\text{Bad}\} \leq \sum_{i=1}^n \mathbb{P}\{\text{Bad}_i\}. \quad (18)$$

Replacing $\mathbb{P}\{\text{Bad}\}$ in (17) with the right-hand-side term of

(18) we arrive at

$$\begin{aligned} &\mathbb{P}^M \left\{ \mathbf{q} \in \mathbb{Q}^M : \sum_{i=1}^n \mathbb{P}\{\text{Bad}_i\} \leq \varepsilon \right\} \\ &= \mathbb{P}^M \left\{ \mathbf{q} \in \mathbb{Q}^M : \sum_{i=1}^n \mathbb{P}\{\text{Bad}_i\} \leq \sum_{i=1}^n \varepsilon_i \right\} \\ &\geq \mathbb{P}^M \left\{ \bigcap_{i=1}^n \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{\text{Bad}_i\} \leq \varepsilon_i \} \right\} \\ &\geq 1 - \sum_{i=1}^n \mathbb{P}^M \{ \mathbf{q} \in \mathbb{Q}^M : \mathbb{P}\{\text{Bad}_i\} > \varepsilon_i \} \\ &\geq 1 - \sum_{i=1}^n \delta_i \\ &= 1 - \delta. \end{aligned}$$

We remark that the third inequality comes from the fact that if $\mathbb{P}\{\text{Bad}_i\} \leq \varepsilon_i$, $\forall i = 1, \dots, n$ then, one can ensure that $\sum_{i=1}^n \mathbb{P}\{\text{Bad}_i\} \leq \sum_{i=1}^n \varepsilon_i$. The fourth inequality also is due to the fact that $\mathbb{P}\{\bigcap_i A_i\} = 1 - \mathbb{P}\{\bigcup_i A_i^c\}$ where A_i^c is the complement of the event A_i . ■

Remark 3 (Extension to Optimization): Extension of Algorithm 1 to consider optimization problems is not trivial. In a centralized setup one, can use the objective cut which removes part of the robust feasible set with larger objective value than the query point. However, in a distributed framework where agents do not have full knowledge of the set Θ , the use of the objective cut can remove the entire Θ from the search space, thus leading to a possible failure of the distributed algorithm. 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 ellipsoids center is removed from the search space. The drawback of this approach is that one has to wait for all nodes to converge again in order to perform the next 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 [13]. 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

²Here we consider a two-dimensional space nevertheless, extension to a three-dimensional space is trivial and straight forward.

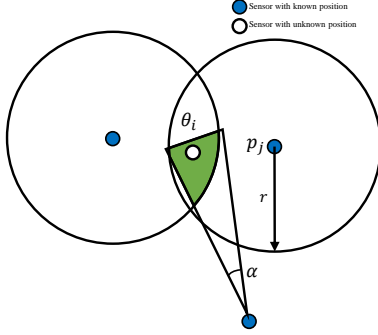


Fig. 1. Sensors with known position are computational nodes in the distributed optimization setup. Their objective is to estimate the unknown position by locally applying angular and radial constraints and exchanging information with their neighbors. The green (shaded) region represents the feasible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

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, this class of sensors are called “known sensors”.

- (ii) Wireless sensors with no positioning capabilities whatsoever. These m 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 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 presented by the intersection of three half-spaces, two to bound the angle and one corresponding to transceiver range as presented in Fig. 1. Sensors with known position which are not equipped with laser transceiver, can place “radial constraint” for the unknown sensors within the communication range. If 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 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)

$$\|\theta_i - p_j\|_2^2 \leq r^2 \Leftrightarrow F(\theta_i) \doteq \begin{bmatrix} rI_2 & (\theta_i - p_j) \\ (\theta_i - p_j)^T & r \end{bmatrix} \geq 0, \quad (19)$$

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 a sensor with unknown position 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

$$\|p_j - \bar{p}_j\|_2 \leq \rho, \quad (20)$$

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 uncertain set. In other words, the uncertainty is defined in 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 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 laser transceiver—the angle α in Fig. 1—is 20 degrees. Finally, the uncertainty radius ρ in (20) is 0.05 and 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 computational node (10 in our case) runs Algorithm 1 in an independent Matlab environment and communication is performed by sharing files between different Matlab environment. This accounts for asynchronicity of Algorithm 1. Figure 2 presents the evolution of volume of ellipsoids in the network and distance of their centers to the robust feasible point (the final solution) for all the nodes in the network. As expected, volumes are non-increasing over time and they all converge to a common value. Center of ellipsoid of each node is its candidate solution. It can be seen from Fig. 2 that candidate solution of all nodes converge to the same point as well. We

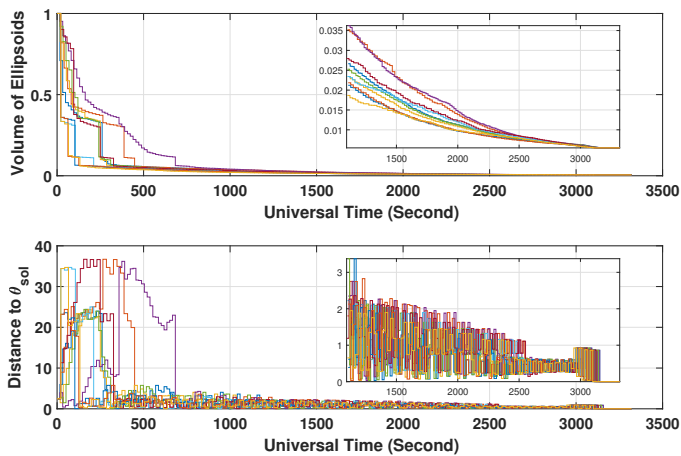


Fig. 2. Top: Volume of ellipsoids of all the nodes along the evolution of the distributed algorithm (normalized with volume of the initial ellipsoid). Bottom: Distance of the center of ellipsoids (candidate solutions) to θ_{sol} for all nodes in the network.

further remark 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 each node is developed based on the ellipsoid algorithm [14]. 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 [14]. 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 candidate ellipsoid with the desired probabilistic robustness. We analyzed the finite-time convergence of the proposed algorithm. The proposed algorithm is used for solving the problem of robust convex position estimation 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] L. Carlone, V. Srivastava, F. Bullo, and G. Calafiore. Distributed random convex programming via constraints consensus. *SIAM Journal on Control and Optimization*, 52:629–662, 2014.
- [10] M. Chamanz, 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.
- [11] L. Comtet. *Sieve Formulas*, pages 176–203. Springer Netherlands, Dordrecht, 1974.
- [12] F. Dabbene, P. Shcherbakov, and B. Polyak. A randomized cutting plane method with probabilistic geometric convergence. *SIAM Journal on Optimization*, 20:3185–3207, 2010.
- [13] 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.
- [14] D. Goldfarb and M.J. Todd. Modifications and implementation of the ellipsoid algorithm for linear programming. *Mathematical Programming*, 23(1):1–19, 1982.
- [15] J. Hendrickx. *Graphs and networks for the analysis of autonomous agent systems*. PhD thesis, Universit Catholique de Louvain, Louvain, Belgium, 2008.
- [16] S. Kanev, B. De Schutter, and M. Verhaegen. An ellipsoid algorithm for probabilistic robust controller design. *Systems & Control Letters*, 49:365–375, 2003.
- [17] S. Lee and A. Nedić. Distributed random projection algorithm for convex optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7:221–229, 2013.
- [18] S. Lee and A. Nedić. Asynchronous gossip-based random projection algorithms over networks. *IEEE Transactions on Automatic Control*, 61:953–968, 2016.
- [19] 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.
- [20] G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: Theory and applications. *IEEE Transactions on Automatic Control*, 56(10):2247–2261, 2011.
- [21] T. Wada and Y. Fujisaki. Sequential randomized algorithms for robust convex optimization. *IEEE Transactions on Automatic Control*, 60:3356–3361, 2015.
- [22] K. You and R. Tempo. Networked Parallel Algorithms for Robust Convex Optimization via the Scenario Approach. *arXiv preprint arXiv:1607.05507*, 2016.