Distributed Solutions to Convex Feasibility Problems with Coupling Constraints

Yingying Xiao, *Student Member, IEEE*, Ji Liu, *Member, IEEE*, and Jianghai Hu, *Member, IEEE*

***Abstract*—In this paper, a distributed approach to convex feasibility problems is proposed. This approach aims to reduce the storage and communication requirements for individual agents by exploiting the sparsity of the constraint couplings across agents: each agent only maintains its own variable together with its desired values for those neighboring agents whose valuations help determining its feasibility; at each iteration, (relaxed) projection and consensus operations are carried out by agents in parallel based on information from only the relevant neighbors. Four algorithms, two synchronous and the other two asynchronous, are proposed and proved to converge asymptotically to a feasible solution starting from any initial guess. The effectiveness of the proposed algorithms is demonstrated through the simulation results on several application examples, e.g. linear programs/equations and network localization.**

***Index Terms*—Convex feasibility problem, distributed approach, sparsely coupled constraints, time-varying communication graph**

I. Introduction

The convex feasibility problem (CFP), also known as the convex intersection problem or constrained consensus problem [1], [2], is the problem of finding a common point that belongs to the intersection of a family of nonempty closed convex sets. As a well known problem in applied mathematics, CFP has found a wide range of practical applications, e.g., image recovery [3]–[5], model predictive control [6], (mobile) sensor networks [7]–[10], robotic teams [11]–[13], social networks [14], electric power grid [15], to name a few.

There has been a tremendous amount of existing literature on the solution of CFP. A majority of existing approaches, especially the earlier ones, are centralized in that a central solver updates a guess of the solution iteratively to satisfy all the convex constraints eventually. A particular popular class of such approaches is the alternative projection method and its variants (e.g. [16]–[18]). Centralized solution algorithms of CFP have the advantage of easy implementation and guaranteed convergence. On the other hand, they often scale poorly as the number of constraints increases.

With a large number of constraints, a natural idea is to partition them into groups for individual agents to maintain. This demands the distributed approaches for solving CFP. Furthermore, in many practical applications, the constraints relevant to an agent often involve the private variables of the agent and its neighbors. Without the need to pass these information to the central solver, distributed solutions can better preserve the privacy of the agents, should such needs arise.

Noteworthy efforts toward this direction include the distributed algorithms for solving linear equations as proposed in [19]–[25], which are subsequently extended to solve nonlinear equations [26]–[28], e.g. paracontractions and strongly quasi-nonexpansive maps, and the projected consensus algorithms for constrained consensus problems [1], [2], [29], [30] and approximate projections [31]. Some of the earlier work on general CFP along this direction can be found in [16], [32]–[34]. CFP can also be formulated as distributed optimization problems; in this context, relevant approaches can be found in, e.g. [35]–[37] to name a few. In all these distributed algorithms, the convex constraints are partitioned and assigned to a group of agents, each of which maintains a local guess of the solution. These local guesses are updated by the agents individually, to satisfy their own constraints through (approximate) projection, and to reach consensus by averaging the guesses of each agent and its neighbors.

A drawback of the aforementioned approaches is that they require each agent to keep and broadcast to its neighbors a vector whose dimension is the same as that of the solution to the problem, which can result in excessive storage and communication requirements on individual agents. For example, Fig. 1 shows a planar network localization problem using relative orientation measurements, which consists of 2 anchors with known locations and 28 free agents whose locations need to be identified. Using the existing algorithms above, each agent needs to maintain and broadcast at each round 56 variables. However, taking the agent labeled by 0 as an example, the constraints relevant to its location only involve the five neighbors within its measurement range, i.e., only 12 of the 56 variables are relevant for the feasibility of its location. With a larger problem size, this disadvantage of the existing algorithms becomes even more severe.

One way to mitigate this issue is to partition not only the

This work is supported by the National Science Foundation under Grant No. 1329875. Part of the material is presented in the 56th IEEE Conference on Decision and Control, December 12-15, 2017, Melbourne, Australia.

Yingying Xiao and Jianghai Hu are with School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: {xiao106, jianghai}@purdue.edu).

Ji Liu is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794-2350, USA (e-mail: ji.liu@stonybrook.edu).

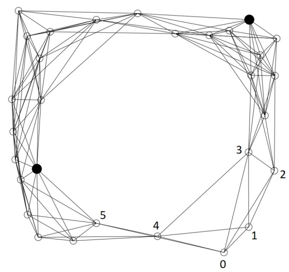


Fig. 1: Network localization with 2 anchors (solid dots) and 28 free agents (small circles).

constraints, but also the solution vector, into different parts and assign them to individual agents. Along this direction, one approach in [38] for solving linear equations $Ax = b$ is that each agent partitions its copy of the solution into multiple blocks and broadcasts periodically or randomly only one of them to its neighbors. This approach reduces the communication load but comes at the expense of convergence rate that heavily depends on how frequently the local copies are broadcasted. Another method in [39] intends to exploit the sparsity of matrix A. Given the partition of x, each agent keeps and broadcasts only the blocks relevant to its own constraints, which often has a much reduced dimension than x. However, this method requires each agent to know not only the index mapping from its own block to the whole variable x but also the index mappings from each of its neighbors to x. This may lead to a large setup overload, and implementation difficulty when such mappings are private information that the agents are not willing to share.

In this paper, a new distributed approach is proposed to solve the general CFP by taking advantage of the possible sparsity of constraint couplings across agents: each agent only maintains its own variable as well as its desired values for the variables of those neighbors whose valuations affect its feasibility; at each round, each agent communicates only with its neighboring agents with constraint couplings (either direction). Such an approach significantly reduces the amount of storage and communication required for individual agents in cases where the couplings are sparse. Four algorithms in this framework are proposed and their convergence to a feasible solution starting from any initial guess is established. Algorithm 1, based on synchronous projection and consensus operation by all agents, converges exponentially fast under some further assumptions. Algorithm 2 extends Algorithm 1 to be asynchronous by allowing agents to independently choose their operations in each iteration while Algorithm 3 generalizes Algorithm 1 by utilizing general, time-varying consensus operations and allowing individual agents to decide if they would like to perform the projection at each round. Algorithm 4, the most general version, combines the relaxations of Algorithms 2 and 3.

The remainder of this paper is organized as follows. The problem is formulated in Section II. Several potential applications are given in Section III. Section IV presents four distributed algorithms and their convergence properties, whose proofs are provided in Section V. Section VI demonstrates the effectiveness of the proposed algorithms through simulation results and Section VII concludes the paper.

Notation: For any integer $m \geq 1$, $\mathcal{I}_m$ denotes the index set $\{1, \ldots, m\}$ and $(x_i)_{i \in \mathcal{I}_m}$ is the column stack of x_i's, $i \in \mathcal{I}_m$, with indices in ascending order. All vectors are treated as column vectors. We use $\mathbf{1}$ to denote the vector of proper size with all entries equal to 1, Id the identity matrix/operator, e the unit vector with only one entry to be 1 in the proper position and all other entries zero, $\text{diag}(A_1, A_2, \ldots, A_m)$ the block diagonal matrix, and $[A]_{ij}$ the entry located at the i-th row and j-th column of matrix A. A vector is stochastic if all entries are nonnegative and sum to 1 and a matrix is stochastic when all of its row vectors are stochastic. For a given set $\mathcal{A}$, $|\mathcal{A}|$ represents its cardinality, $d_{\mathcal{A}}(x)$ denotes the Euclidean distance of x to $\mathcal{A}$, and $\mathcal{A}^o$ is the interior of $\mathcal{A}$. The norm and its special cases, l_1-norm and l_2-norm, are denoted by $|||\cdot|||$, $\|\cdot\|_1$, and $\|\cdot\|$, respectively. For two operators P and Q, the composition $Q(P(x))$ is denoted as $Q \circ P(x)$ and Fix_P represents the fixed point set of P.

II. Problem Formulation

Consider a set of m agents indexed by $\mathcal{I}_m$. Assume each agent $i \in \mathcal{I}_m$ maintains a (local) variable $x_i \in \mathbb{R}^{n_i}$ of its own, which needs to satisfy a constraint of the following form:

$$x_i \in \mathcal{D}_i \left((x_j)_{j \in \mathcal{N}_i^+} \right). \tag{1}$$

Here, $\mathcal{N}_i^+ \subset \mathcal{I}_m \setminus \{i\}$ is a set of agents whose variables are needed to determine the feasible set of x_i; $(x_j)_{j \in \mathcal{N}_i^+}$ is the stacked vector of all the variables of agents in $\mathcal{N}_i^+$; and $\mathcal{D}_i \left((x_j)_{j \in \mathcal{N}_i^+} \right)$ is a subset of $\mathbb{R}^{n_i}$ which may vary with $(x_j)_{j \in \mathcal{N}_i^+}$. Equivalently, the constraint (1) can be written as

$$\left(x_i, (x_j)_{j \in \mathcal{N}_i^+} \right) \in \mathcal{F}_i, \tag{2}$$

where $\mathcal{F}_i$ is a suitably chosen subset of the product space of x_i and $(x_j)_{j \in \mathcal{N}_i^+}$.

The constraint (1) on the variable of agent i is in general non-local as it depends on the variables of other agents in $\mathcal{N}_i^+$. In the case $\mathcal{N}_i^+ = \emptyset$, the feasible set $\mathcal{D}_i$ becomes a fixed subset of $\mathbb{R}^{n_i}$ and the constraint on x_i becomes local. Due to privacy concern, the constraint $\mathcal{F}_i$ (thus $\mathcal{D}_i$) is assumed to be private to each agent $i \in \mathcal{I}_m$ while the local variable x_i is shared with other neighboring agents.

Example 1: Consider the example shown in Fig. 2. There are four agents with the local variables x_i and the local constraints $\mathcal{F}_i$, $i \in \mathcal{I}_4$. In Fig. 2, the local variables are labeled on the right; the local constraints are labeled on the left; the solid lines represent the constraint couplings across agents. Except for agent 1, the local constraint of every other agent is non-local.

A directed graph $\mathcal{G}_d$, called the (constraint) *dependency graph*, can be constructed to represent the interdependency of the agents' feasibility: $\mathcal{G}_d$ has the vertex set $\mathcal{I}_m$ and a directed edge from j to i, denoted as (j, i), whenever the feasible set of x_i depends on x_j. Note that there is no self-loop in $\mathcal{G}_d$. See the left of Fig. 3 for $\mathcal{G}_d$ of Example 1. The aforementioned set $\mathcal{N}_i^+$ is exactly the in-neighborhood of vertex i in $\mathcal{G}_d$; thus

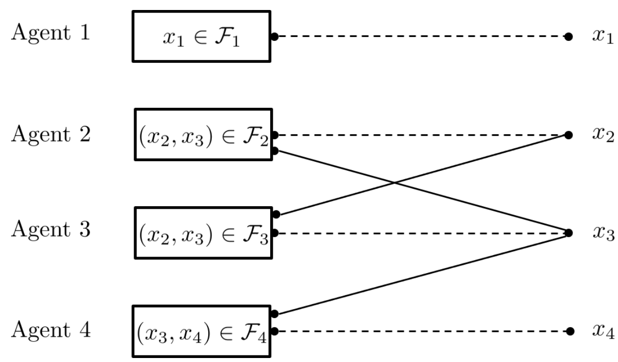


Fig. 2: Dependence Illustration of Example 1.

we call agents indexed by $\mathcal{N}_i^+$ the *in-neighbors* of agent i. Similarly, the out-neighborhood of vertex i in $\mathcal{G}_d$, denoted by $\mathcal{N}_i^- \subset \mathcal{I}_m \setminus \{i\}$, indexes the *out-neighbors* of agent i, namely, agents whose variables' feasibility depends (at least partially) on the value of x_i. The two neighborhoods $\mathcal{N}_i^+$ and $\mathcal{N}_i^-$ may overlap or even be identical (see Example 3 below). Denote by $\mathcal{N}_i := \mathcal{N}_i^+ \cup \mathcal{N}_i^-$ the neighbors of agent i.

Since the agents' constraints are coupled, to ensure feasibility they need to communicate with each other to share their local variables (but not their local constraints due to privacy consideration). The allowable communication among agents is represented by the *communication graph* $\mathcal{G}_c$, which is a directed graph with the vertex set $\mathcal{I}_m$ and an edge set such that a directed edge from j to i exists whenever agent i can receive information from agent j via direct communication.

Assumption 1 (Communicability): The communication graph $\mathcal{G}_c$ contains the union of $\mathcal{G}_d$ and its transpose $\mathcal{G}_d^\top$ [1].

Assumption 1 implies that each agent can have two-way communications (i.e. send information to and receive information from) with any of its in-neighbors and out-neighbors. In other words, the communication is bi-directional between two agents whenever one's feasibility depends on the other's variable. See the right of Fig. 3 for $\mathcal{G}_c$ of Example 1. The following Example 2 demonstrates why the bidirectional communication is necessary.

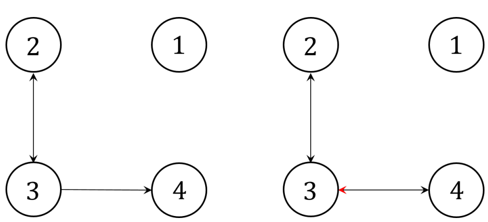


Fig. 3: Dependence graph (left) and communication graph (right) of Example 1.

Example 2: Consider two agents with local variables $x_1, x_2 \in \mathbb{R}$ and local constraints

$$\mathcal{F}_1 : x_1 = x_2,\ x_1 \le 5 \quad \text{and} \quad \mathcal{F}_2 : 4 \le x_2 \le 7,$$

respectively. The dependence graph $\mathcal{G}_d$ has only one edge $(2, 1)$. Assume $x_1(0) = 5$, $x_2(0) = 6$. If agent 2 can not obtain information from agent 1, it will stick to its initial value and never reach consensus with agent 1 on the value of x_2.

[1] The transpose graph $\mathcal{G}_d^T$ is obtained by reversing the direction of every edge of $\mathcal{G}_d$.

Example 3: The linear equation $Ax = b$ with

$$A = \left[\begin{array}{cc|c} 1 & 0 & -1 \\ \hline 1 & 1 & 1 \\ 0 & 1 & 1 \end{array}\right] \text{ and } b = \left[\begin{array}{c} 0 \\ \hline 0 \\ -1 \end{array}\right] \tag{3}$$

has a unique solution $x^* = A^{-1}b = (1, -2, 1)$. Partition $x \in \mathbb{R}^3$ into $x = (x_1, x_2)$ where $x_1 \in \mathbb{R}^2$ and $x_2 \in \mathbb{R}$ are the variables of agents 1 and 2, respectively. With the row (constraint) partitions of A and b in (3), the private constraint of agent 1 is underdetermined for x_1: $\begin{bmatrix} 1 & 0 \end{bmatrix} x_1 - x_2 = 0$, while the private constraint of agent 2 is overdetermined for x_2:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

The neighbor sets of the two agents are given by $\mathcal{N}_1^+ = \mathcal{N}_1^- = \{2\}$ and $\mathcal{N}_2^+ = \mathcal{N}_2^- = \{1\}$, resulting in the corresponding dependence graph $\mathcal{G}_d$ with the edge set $\{(1, 2), (2, 1)\}$.

Finally we formulate the problem to be studied in this paper.

Problem 1 (Distributed Feasibility Problem): Design distributed algorithms consistent with the communication graph and maintaining the privacy of individual agents' constraints so that a value of $(x_i)_{i \in \mathcal{I}_m}$ can be (asymptotically) obtained that satisfies the private constraints of all agents.

Denote $x := (x_i)_{i \in \mathcal{I}_m} \in \mathbb{R}^n$ where $n = \sum_{i \in \mathcal{I}_m} n_i$. The following assumptions are imposed throughout this paper.

Assumption 2 (Feasibility): There exists at least one x that satisfies all m constraints in (2).

Assumption 3 (Convexity): The feasible set $\mathcal{F}_i$ in (2) is nonempty, closed and convex for each $i \in \mathcal{I}_m$.

As a consequence of Assumption 3, the feasible set $\mathcal{D}_i\left((x_j)_{j \in \mathcal{N}_i^+}\right)$ in (1) is also convex.

III. Application Examples

Two instances of Problem 1 are presented below.

A. Distributed Solution of Linear Programs/Equations

Let $A \in \mathbb{R}^{\ell \times n}$, $b \in \mathbb{R}^\ell$ be such that the linear program $Ax \le b$ has at least one feasible solution x^*. Suppose that different portions of the variable x and the inequalities are held separately by a group of agents indexed by $\mathcal{I}_m$, i.e., there exist the block partitions $x = (x_1, \cdots, x_m)$,

$$\left[\begin{array}{c|c} A & B \end{array}\right] = \left[\begin{array}{ccc|c} A_{11} & \cdots & A_{1m} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ A_{m1} & \cdots & A_{mm} & b_m \end{array}\right]$$

so that agent $i \in \mathcal{I}_m$ has n_i variables, $x_i \in \mathbb{R}^{n_i}$, and ℓ_i private linear inequality constraints, $A_{i1}x_1 + \cdots + A_{im}x_m \le b_i \in \mathbb{R}^{\ell_i}$. Here, we assume $n_i, \ell_i \ge 0$ with $\sum_i n_i = n$ and $\sum_i \ell_i = \ell$; and "$\le$" denotes entry-wise comparison. Agent i has the neighbor sets $\mathcal{N}_i^+ = \{j \in \mathcal{I} \,|\, A_{ij} \ne 0\}$ and $\mathcal{N}_i^- = \{j \in \mathcal{I} \,|\, A_{ji} \ne 0\}$ and its constraint can be recast as $A_{ii}x_i + \sum_{j \in \mathcal{N}_i^+} A_{ij}x_j \le b_i$. Distributed solution of the above linear program (and as a special case, the linear equation $Ax = b$) is an instance of Problem 1. Example 3 is one such instance.

Example 4: Consider the linear program $-\varepsilon\mathbf{1} \le Ax - b \le \varepsilon\mathbf{1}$ with $x = (x_1, x_2, x_3) \in \mathbb{R}^3$,

$$A = \left[\begin{array}{c|c|c} 1 & 0 & -1 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \end{array}\right], \quad b = \left[\begin{array}{c} 1 \\ \hline -1 \\ \hline 1 \end{array}\right], \quad \mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4}$$

There are three agents with the variables $x_1, x_2, x_3 \in \mathbb{R}$ and the following private constraints, respectively: $|x_1 - x_3 - 1| \le \varepsilon$ for agent 1; $|x_3 + 1| \le \varepsilon$ for agent 2; and $|x_2 + x_3 - 1| \le \varepsilon$ for agent 3. Their neighbor sets are $\mathcal{N}_1^+ = \{3\}$, $\mathcal{N}_1^- = \emptyset$; $\mathcal{N}_2^+ = \mathcal{N}_2^- = \{3\}$; $\mathcal{N}_3^+ = \{2\}$ and $\mathcal{N}_3^- = \{1, 2\}$. Note that the constraint of agent 2 does not involve its own variable x_2, which is allowed in our problem formulation. Further, $x^* = A^{-1}b = (0, 2, -1)$ is a feasible solution for any $\varepsilon \ge 0$.

B. Network Localization

Consider a group of agents (sensors, robots, vehicles) deployed on $\mathbb{R}^2$ with unknown locations $x_i \in \mathbb{R}^2$, $i \in \mathcal{I}$. Suppose each agent $i \in \mathcal{I}$ is equipped with sensors that can measure its relative distance and/or orientation w.r.t. some other agents $j \in \mathcal{N}_i^+$ within its sensing range.

(i) *Relative orientation (Angle-of-Arrival) measurement*: the direction of the vector $x_j - x_i$ is measured against a compass onboard agent i. This imposes a constraint as $\angle(x_j - x_i) \in \Theta_{ij}$, where $\angle$ denotes the phase angle and Θ_{ij} is a singleton $\{\theta_{ij}\}$ if the measurement is precise and an interval $[\theta_{ij} - \delta, \theta_{ij} + \delta]$ if the measurement is imprecise.

(ii) *Relative distance measurement*: the distance $\|x_j - x_i\|$ is measured using, e.g., the strength of signal received by agent i from agent j. This incurs a constraint as $r_1 \le \|x_i - x_j\| \le r_2$.

The private constraint of agent i consists of all the above constraints for $j \in \mathcal{N}_i^+$. The network localization problem is to find the locations of all agents consistent with the measurement data. This is an instance of Problem 1 if $r_1 = 0$.

IV. Proposed Algorithms

We first present an equivalent formulation of Problem 1. Suppose besides its own variable x_i, agent i maintains an additional set of variables, $(x_{ji})_{j\in\mathcal{N}_i^+}$, where x_{ji} represents the value of agent j's variable as *desired* by agent i (which could differ from the actual value of x_j). Define

$$\mathbf{x}_i := \left(x_i, (x_{ji})_{j\in\mathcal{N}_i^+}\right)$$

to be the *augmented variable* of agent i with the dimension $N_i = n_i + \sum_{j\in\mathcal{N}_i^+} n_j$. Then the totality of all $\mathbf{x}_i$'s, denoted by $\mathbf{x} := (\mathbf{x}_i)_{i\in\mathcal{I}_m}$, has dimension $N = \sum_{i\in\mathcal{I}_m} N_i$. For sparse dependency graph $\mathcal{G}_d$, $N \ll mn$. With $\mathbf{x}_i$'s, Problem 1 can be reformulated as follows.

Problem 2: Design distributed algorithms consistent with the communication graph $\mathcal{G}_c$ so that a value of $\mathbf{x}$ is asymptotically obtained that satisfies,

$$\mathbf{x}_i \in \mathcal{F}_i, \quad \forall i \in \mathcal{I}_m, \tag{5}$$

$$x_i = x_{ik}, \quad \forall i \in \mathcal{I}_m,\ \forall k \in \mathcal{N}_i^-. \tag{6}$$

The constraint (5) is from (2) with x_j replaced by x_{ji}, which is local as it only involves agent i's augmented variable $\mathbf{x}_i$. The consensus constraint (6) ensures agent i's variable x_i to be the same as that desired by its out-neighbors, inducing the non-local consensus set

$$\mathcal{C}_i := \{(x_i, (x_{ik})_{k\in\mathcal{N}_i^-}) \,|\, x_i = x_{ik},\ \forall k \in \mathcal{N}_i^-\}. \tag{7}$$

Define

$$\mathcal{A}_1 = \mathcal{F}_1 \times \cdots \times \mathcal{F}_m, \quad \mathcal{A}_2 = M^\top(\mathcal{C}_1 \times \cdots \times \mathcal{C}_m) \tag{8}$$

to be the feasible set and consensus subspace of $\mathbf{x}$, respectively, where $M \in \mathbb{R}^{N\times N}$ is a permutation matrix so that each variable x_i and its desired values by out-neighbors, $x_{ik}, k \in \mathcal{N}_i^-$, are put consecutively in a block in the order of $i = 1, \ldots, m$. Clearly, $\mathcal{A}_1 \cap \mathcal{A}_2$ is the solution set of Problem 2.

It is easy to see that the solutions to Problems 1 and 2 have a one-to-one correspondence; hence they are equivalent. By Assumption 2, Problem 2 has a feasible solution $\mathbf{x}^* = (\mathbf{x}_i^*)_{i\in\mathcal{I}_m}$. Next we present four algorithms to solve Problem 2 (and thus Problem 1).

A. Synchronous Algorithm

The first algorithm iteratively solves Problem 2 with all agents updating synchronously in each iteration. The update at round t consists of two stages: first each agent i updates its augmented variable from $\mathbf{x}_i^t$ to $\mathbf{z}_i^t$ via the (relaxed) projection operator P_i onto its local feasible set $\mathcal{F}_i$ as in (9); then, each agent i simultaneously collects from its out-neighbors their updated desired values of x_i, $(z_{ik}^t)_{k\in\mathcal{N}_i^-}$, to obtain x_i^{t+1} via the consensus operation (10), and broadcasts x_i^{t+1} back to all of its out-neighbors as their updated values x_{ik}^{t+1} as in (11). The iterations above are detailed below and summarized in Algorithm 1.

(i) (Relaxed projection)
Agent i computes $\mathbf{z}_i^t := (z_i, (z_{ji})_{j\in\mathcal{N}_i^+})$ from $\mathbf{x}_i^t$ via the relaxed projection operation

$$\mathbf{z}_i^t = P_i(\mathbf{x}_i^t) := (1 - \alpha_i)\mathbf{x}_i^t + \alpha_i \cdot P_{\mathcal{F}_i}(\mathbf{x}_i^t). \tag{9}$$

Here, $P_{\mathcal{F}_i}$ denotes the orthogonal projection operator onto the local feasible set $\mathcal{F}_i$ and $\alpha_i \in (0, 2)$ is a constant.

(ii) (Consensus)
Agent i collects $(z_{ik}^t)_{k\in\mathcal{N}_i^-}$ from its out-neighbors to update its variable according to

$$\begin{aligned} x_i^{t+1} &= Q_i(z_i^t, (z_{ik}^t)_{k\in\mathcal{N}_i^-}) \\ &:= \tfrac{1}{|\mathcal{N}_i^-|+1}\left(z_i^t + \textstyle\sum_{k\in\mathcal{N}_i^-} z_{ik}^t\right), \end{aligned} \tag{10}$$

and then sends back x_i^{t+1} to out-neighbors for updating

$$x_{ik}^{t+1} = x_i^{t+1},\ k \in \mathcal{N}_i^-. \tag{11}$$

If agent i has no out-neighbors, i.e., $\mathcal{N}_i^- = \emptyset$, the update (10) will be trivial: $x_i^{t+1} = z_i^t$.

In Algorithm 1, all agents update their augmented variables in parallel at each round. Intuitively the relaxed projection (9) helps to improve the satisfaction of the local feasibility constraint (5) while the consensus step (10) together with the broadcast step (11) helps to reach consensus on the value of x_i among agent i and its out-neighbors.

Algorithm 1 Synchronous Algorithm

Initialize $\mathbf{x}^0$ and let $t \leftarrow 0$;
repeat
  for all $i \in \mathcal{I}_m$ **do** {Relaxed projection}
    Agent i computes $\mathbf{z}_i^t$ according to (9);
  end for
  for all $i \in \mathcal{I}_m$ **do** {Consensus}
    Agent i receives z_{ik}^t from all out-neighbors k;
    Agent i computes x_i^{t+1} according to (10);
    Agent i sends back x_i^{t+1} to all out-neighbors for updating x_{ik}^{t+1}as in (11);
  end for
  $t \leftarrow t+1$;
until certain convergence criteria are met
Return $\mathbf{x}^t$.

Note that in Algorithm 1, each agent only communicates with its out-neighbors in the consensus step and this communication is bidirectional (two-way), which is allowed by Assumption 1.

The convergence properties of the synchronous algorithm are characterized by the following two theorems whose proofs will be provided in Section V-A.

Theorem 1: Starting from any initial guess $\mathbf{x}^0$, the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 1 will converge asymptotically to a feasible solution to Problem 2.

Theorem 2: Suppose that there exists a feasible solution $x^{\text{o}} \in \mathbb{R}^n$ to Problem 1 such that for each $i \in \mathcal{I}_m$, $\left(x_i^{\text{o}}, (x_j^{\text{o}})_{j\in\mathcal{N}_i^+}\right) \in \mathcal{F}_i^{\text{o}}$, i.e., an interior point of $\mathcal{F}_i$. Then Algorithm 1 with $\alpha_i = 1$ for all $i \in \mathcal{I}_m$ converges exponentially fast to a feasible solution of Problem 2 starting from any initial point.

Remark 1: The well-known projected consensus algorithm in [1], [2], denoted as Pro-Con, has been proved that the distance of each iterate to the feasible solution set decays exponentially fast. This is weaker than the conclusion of Theorem 2 that the iterates themselves converge exponentially to one feasible solution. Without taking account of the differences in implementation details, the main reason is that the relaxed projection and consensus operations are paracontractions (see Definition 1 in Section V) while the general weight consensus operation adopted in Pro-Con is not.

B. Asynchronous Algorithm

The synchronous operations in Algorithm 1 can be difficult to ensure in practice, which is extended to be asynchronous in this section. At round t, each agent i independently determines whether it will update or not and, if so, chooses one of the following two operations to perform: carrying out the relaxed projection operation (9) to satisfy its local feasibility constraint; reaching consensus on its own variable x_i with a subset of its out-neighbors, denoted by $\mathcal{N}_{i,t}^- \subseteq \mathcal{N}_i^-$, through the averaging step

$$x_i^{t+1} = Q_i^t\left(x_i^t, (x_{ik}^t)_{k\in\mathcal{N}_{i,t}^-}\right) := \tfrac{1}{|\mathcal{N}_{i,t}^-|+1}\left(x_i^t + \textstyle\sum_{k\in\mathcal{N}_{i,t}^-} x_{ik}^t\right) \tag{12}$$

followed by the broadcast step (11) with $\mathcal{N}_i^-$ replaced by $\mathcal{N}_{i,t}^-$. In other words, depending on agent i's update choice at round t, it will belong to one of the three sets, the idle, projection, and consensus sets, denoted by $\mathcal{I}_{\text{idle}}^t$, $\mathcal{I}_P^t$, $\mathcal{I}_Q^t$, respectively, and then perform the corresponding operation. Note that $\mathcal{I}_{\text{idle}}^t$, $\mathcal{I}_P^t$ and $\mathcal{I}_Q^t$ constitutes a partition of I_m. The Algorithm 2 bellow describes this asynchronous version.

Algorithm 2 Asynchronous Algorithm

Initialize $\mathbf{x}^0$ and set $t \leftarrow 0$;
repeat
  for all $i \in \mathcal{I}_m$ **do**
    Agent i idles
    or
    {Relaxed projection}
    Agent i updates $\mathbf{x}_i^{t+1}$ according to (9) with $\mathbf{z}_i^t$ replaced by $\mathbf{x}_i^t$;
    or
    {Partial consensus}
    Agent i receives x_{ik}^t from the out-neighbor k belonging to the subset $\mathcal{N}_{i,t}^- \subseteq \mathcal{N}_i^-$;
    Agent i computes x_i^{t+1} according to (12);
    Agent i sends x_i^{t+1} back to its out-neighbors $k \in \mathcal{N}_{i,t}^-$ as their updated values x_{ik}^{t+1};
  end for
  $t \leftarrow t+1$;
until certain convergence criteria are met
Return $\mathbf{x}^t$.

Remark 2 (Algorithm 2b): A special case of Algorithm 2 is $|\mathcal{N}_{i,t}^-| = 1$ in (12), i.e., agent $i \in \mathcal{I}_Q^t$ performs its consensus operation with only one out-neighbor $k \in \mathcal{N}_i^-$ that is either randomly picked or resulted from some extreme situations. In this case, the equally weighted average (12) can be relaxed to

$$\begin{bmatrix} x_i^{t+1} \\ x_{ik}^{t+1} \end{bmatrix} = (W_{ik} \otimes I_{n_i}) \begin{bmatrix} x_i^t \\ x_{ik}^t \end{bmatrix}. \tag{13}$$

Here $W_{ik} \in \mathbb{R}^{2\times 2}$ is a constant doubly stochastic matrix with strictly positive entires. To carry out the update (13), agent i first collects x_{ik}^t from agent k, then computes the update values for both itself and agent k, and finally sends the latter x_{ik}^{t+1} back to agent k. With this relaxation, agent i may not reach consensus with any out-neighbors, i.e., $x_i^{t+1} \neq x_{ik}^{t+1}, \forall k \in \mathcal{N}_i^-$. We will refer to this relaxed algorithm as Algorithm 2b and show its convergence in Theorem 4.

To establish the convergences of Algorithms 2 and 2b, we impose two assumptions.

Assumption 4 (Semaphore): At round t, for any agent i carrying out the partial consensus operation (12), none of (active) its out-neighbors $\mathcal{N}_{i,t}^-$ in (12) will be performing the relaxed projection operation, i.e., $\mathcal{N}_{i,t}^- \cap \mathcal{I}_P^t = \emptyset,\ \forall i \in \mathcal{I}_Q^t$.

This assumption implies that at each round, each variable either does not change or changes only once resulted from the relaxed projection or partial consensus. This is critically important for establishing the convergences later.

Assumption 5 (Infinite Appearances): (a) For each $i \in \mathcal{I}_m$, $i \in \mathcal{I}_P^t$ for infinitely many $t \in \{0, 1, \ldots\}$; (b) Any pair of neighboring agents is involved in the (partial) consensus operation (12) for an infinite number of times.

Assumption 5 is less restrictive than both periodic and uniformly repeated appearances which require that the two operations in Assumption 5 are involved once and at least once every T rounds, respectively, for a positive integer T. Note that Assumption 5(b) imposes constraints on both $\mathcal{I}_Q^t$ and $\mathcal{N}_{i,t}^-, \forall i \in \mathcal{I}_Q^t$, such that their combinations will guarantee that any neighboring agents have enough communication on their variables to reach consensus.

The following two theorems establish the convergences of Algorithms 2 and 2b, respectively. Their proofs will be given later on in Section V-B.

Theorem 3: Suppose Assumptions 4 and 5 hold. Starting from any initial guess $\mathbf{x}^0$, the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 2 converges asymptotically to a feasible solution to Problem 2.

Theorem 4: Suppose Assumptions 4 and 5 hold. Starting from any initial guess $\mathbf{x}^0$, the sequence $\{\mathbf{x}^t\}$ returned by Algorithm 2b will converge asymptotically to a feasible solution to Problem 2.

C. Generalized Synchronous Algorithm

In this section, we generalize Algorithm 1 in two perspectives: (i) along the spirit of Algorithm 2, each agent independently determines at each round if it will be activated to perform updates and, if so, the type of update to be carried out, (ii) time-varying general weights are adopted in the consensus operation.

For the perspective (i), at round t only agents in the two subsets of $\mathcal{I}_m$, denoted by $\mathcal{I}_P^t$ and $\mathcal{I}_Q^t$, are assumed to perform the relaxed projection and consensus operations of Algorithm 1, respectively. Note that this algorithm remains synchronous in a way that all agents must finish the relaxed projection step before moving to the consensus operation, different from the parallel implementation in Algorithm 2. Therefore, $\mathcal{I}_P^t \cap \mathcal{I}_Q^t$ can be non-empty, i.e., an agent can participate in both the projection and the consensus operations. This extension accommodates the practical situation that some agents may be unable to update due to temporary breakdown or communication blackouts.

For the perspective (ii), the most straightforward generalization is replacing step (10) of agent $i \in \mathcal{I}_Q^t$ by the following:

$$x_i^{t+1} = w_{ii}^t z_i^t + \sum_{k \in \mathcal{N}_i^-} w_{ik}^t z_{ik}^t, \tag{14}$$

where $w_{ii}^t \in \mathbb{R}$ and $w_{ik}^t \in \mathbb{R}, k \in \mathcal{N}_i^-$ are time-varying weights assigned by agent i and satisfy that every weight is bounded from below by $\underline{w} > 0$ and their sum is one. Unfortunately, this generalization does not work in general, even in the simplest case where the weights are constant and $\mathcal{I}_P^t = \mathcal{I}_Q^t = \mathcal{I}_m$, i.e., the extension (i) above is removed. This is shown by Example 5.

Example 5: Consider the linear equation $Ax = b$ where $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ and $b = 0 \in \mathbb{R}^3$, which has a unique solution $x = (x_1, x_2, x_3) = 0$. Suppose it is solved by three agents each in charge of one component of x and one row constraint. Then the augmented variables are $\mathbf{x}_1 = (x_1, x_{21}, x_{31})$, $\mathbf{x}_2 = (x_2, x_{12})$, and $\mathbf{x}_3 = (x_3, x_{13})$. Suppose the stochastic matrices in (14) are given by $w_{11}^t = 0.9$, $w_{12}^t = 0.05$, $w_{13}^t = 0.05$, $w_{21}^t = 0.2$, $w_{22}^t = 0.8$, $w_{31}^t = 0.1$, $w_{33}^t = 0.9$. Assuming that $\mathcal{I}_P^T = \mathcal{I}_Q^t = \mathcal{I}_m$, the iteration of Algorithm 3 can be written as $\mathbf{x}^{k+1} = F\mathbf{x}^k$ for some matrix $F \in \mathbb{R}^{7\times 7}$. It can be verified numerically that F has an eigenvalue 1.1246. Therefore, Algorithm 3 does not converge to the solution 0 starting from some (indeed, almost all) initial guesses $\mathbf{x}^0$.

Instead of (14), the following operation is adopted to replace the consensus step (10) and the broadcast step (11) in Algorithm 1:

$$\left(x_i^{t+1}, (x_{ik}^{t+1})_{k\in\mathcal{N}_i^-}\right) = W_i^t \otimes I_{n_i} \left(z_i^t, (z_{ik}^t)_{k\in\mathcal{N}_i^-}\right). \tag{15}$$

Here, $W_i^t \in \mathbb{R}^{(|\mathcal{N}_i^-|+1)\times(|\mathcal{N}_i^-|+1)}$ is a time-varying weight matrix specified by agent i that satisifies Assumption 6 to be defined below. In the case where agent i at round t receives desired values z_{ik}^t from only a subset of out-neighbors, $\mathcal{N}_{i,t}^- \subset \mathcal{N}_i^-$, the rows and columns of W_i^t corresponding to the other (silent) out-neighbors, i.e., $\mathcal{N}_i^- \setminus \mathcal{N}_{i,t}^-$, are set to proper unit vectors as their desired values of x_i remain unchanged at this round. With this generalized weight matrix W_i^t, the updated values for out-neighbors, $x_{ik}^{t+1}, k \in \mathcal{N}_i^-$, will be different from x_i^{t+1} in general, i.e., agent i does not reach consensus with its out-neighbors on its variable x_i at each round, which is the main difference between (15) and the consensus step in (10). The potential benefits of adopting W_i^t include 1). speeding up the convergence by properly assigning weights, especially when the desired values from some out-neighbors are known to be more accurate/important than others, and 2). accommodating the practical situation that agent i loses communication with some out-neighbors occasionally.

Assumption 6 (Weights Rule): For matrix W_i^t, $\forall t \geq 0$ and $\forall i \in \mathcal{I}_Q^t$,

(a) W_i^t is doubly stochastic;
(b) there is a scalar $\underline{w} > 0$ such that entries of W_i^t corresponding to all agents in $\{i\} \cup \mathcal{N}_{i,t}^-$ are bounded from below by $\underline{w}$, i.e., $[W_i^t]_{kl} \geq \underline{w}$ for all $k, l \in \{i\} \cup \mathcal{N}_{i,t}^-$;
(c) for agent $k \in \mathcal{N}_i^- \setminus \mathcal{N}_{i,t}^-$, the diagonal entries $[W_i^t]_{kk} = 1$ while the other elements in the row and column related to agent k are set to 0.

Assumption 6(b) guarantees that once agent i obtains an out-neighbor's desired value z_{ik}^t, this value will make significant contributions to the consensus outcome. Although Assumption 6(a) requires W_i^t to be doubly stochastic, such a matrix is chosen by agent i alone without any coordination with other agents and will in general be different from those chosen by other agents. In comparison, the traditional double stochasticity assumption (e.g., [40] and Assumption 3 in [1]) needs all of the agents to coordinate to choose a single doubly stochastic matrix.

The generalized synchronous algorithm with the above two extensions is summarized in Algorithm 3. In order to establish its convergence, the following Assumption 7 is imposed.

Algorithm 3 Generalized Synchronous Algorithm

Initialize $\mathbf{x}^0$, and let $t \leftarrow 0$;
repeat
  for all $i \in \mathcal{I}_P^t$ **do** {Relaxed projection}
    Agent i computes $\mathbf{z}_i^t$ according to (9);
  end for
  for all $i \in \mathcal{I}_Q^t$ **do** {Generalized partial consensus}
    Agent i receives z_{ik}^t from out-neighbors $k \in \mathcal{N}_{i,t}^-$;
    Agent i computes x_i^{t+1}, x_{ik}^{t+1} according to (15);
    Agent i sends back x_{ik}^{t+1} to out-neighbors in $\mathcal{N}_{i,t}^-$ as their updated values;
  end for
  $t \leftarrow t+1$;
until certain convergence criteria are met
Return $\mathbf{x}^t$.

Assumption 7 (Uniform Appearances): (a) For each $i \in \mathcal{I}_m$, $i \in \mathcal{I}_P^t$ for infinitely many $t \in \{0, 1, \ldots\}$; (b) There exists a finite integer $T > 0$ such that, for any agent $i \in \mathcal{I}_m$, each of its out-neighbor appears at least once in $\cup_{t=t_0}^{t_0+T} \mathcal{N}_{i,t}^-$ for any integer $t_0 \geq 0$.

Clearly, Assumption 7 is stronger than Assumption 5 in part(b) by requiring more frequent consensus operations between neighboring agents. Now we state the convergence result of Algorithm 3 in Theorem 5 below with its proof provided in Section V-C. As will be seen, the convergence analysis of Algorithm 3 is much more challenging than that of Algorithm 1 since the operation (15) is no longer a projection onto the consensus set.

Theorem 5: Suppose that Assumptions 6 and 7 hold. Starting from any initial guess $\mathbf{x}^0$, the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 3 will asymptotically converge to a feasible solution to Problem 2.

D. Generalized Asynchronous Algorithm

The following Algorithm 4 is the asynchronous version of Algorithm 3 in a way that the relaxed projection and the generalized partial consensus operations can be carried out simultaneously rather than consecutively. Its convergence is shown in Theorem 6 below with proof given in Section V-D.

Theorem 6: Suppose Assumptions 4, 6, and 7 hold. Starting at any initial guess $\mathbf{x}^0$, the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 4 converges asymptotically to a feasible solution to Problem 2.

Algorithm 4 Generalized Asynchronous Algorithm

Initialize $\mathbf{x}^0$ and set $t \leftarrow 0$;
repeat
  for all $i \in \mathcal{I}_m$ **do**
    Agent i idles
    or
    {Relaxed projection}
    Agent i updates $\mathbf{x}_i^{t+1}$ according to (9) with $\mathbf{z}_i^t$ replaced by $\mathbf{x}_i^t$;
    or
    {Generalized partial consensus}
    Agent i receives x_{ik}^t from out-neighbors k belonging to the subset $\mathcal{N}_{i,t}^- \subseteq \mathcal{N}_i^-$;
    Agent i computes x_i^{t+1}, x_{ik}^{t+1} according to (15);
    Agent i sends x_{ik}^{t+1} back to its out-neighbors $k \in \mathcal{N}_{i,t}^-$ as their updated values;
  end for
  $t \leftarrow t+1$;
until certain convergence criteria are met
Return $\mathbf{x}^t$.

V. Convergence Proof

We first introduce two useful notions, paracontractions and their subclass, strongly quasi-nonexpansive maps, a key result on paracontractions and their properties.

Definition 1 ([41]): A continuous map $P : \mathbb{R}^n \to \mathbb{R}^n$ is called a paracontraction w.r.t. a norm $|||\cdot|||$ on $\mathbb{R}^n$ if $|||P(x) - y||| < |||x - y|||$ for any $x \notin \mathrm{Fix}_P$ and $y \in \mathrm{Fix}_P$.

The orthogonal projection operator $P_{\mathcal{F}}$ onto a nonempty closed convex set $\mathcal{F}$ is a paracontraction w.r.t. the Euclidean norm. Indeed, for any $\alpha \in (0, 2)$, the relaxed projection operator $(1 - \alpha) \cdot \mathrm{Id} + \alpha \cdot P_{\mathcal{F}}$ with the identity map Id is also a paracontraction [41] and even further a $\frac{2-\alpha}{\alpha}$-strongly quasi-nonexpansive map defined as follows [42].

Definition 2 ([42]): Let $\beta > 0$. A map $P : \mathbb{R}^n \to \mathbb{R}^n$ is β-strongly quasi-nonexpansive w.r.t. a norm $|||\cdot|||$ on $\mathbb{R}^n$ if

$$|||P(x) - y|||^2 \leq |||x - y|||^2 - \beta |||P(x) - x|||^2$$

for any $x \in \mathbb{R}^n$ and $y \in \mathrm{Fix}_P$.

Followed is a useful fact on paracontractions with its straightforward proof omitted.

Lemma 1: Suppose $P_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ is a paracontraction w.r.t. the norm $|||\cdot|||_i$ for $i = 1, \ldots, m$. Then $P = P_1 \times \cdots \times P_m : \mathbb{R}^n \to \mathbb{R}^n$ where $n = \sum_{i=1}^m n_i$ is a paracontraction w.r.t. the norm $|||x||| := (\sum_{i=1}^m |||x_i|||_i^p)^{1/p}$ for $x = (x_1, \ldots, x_m) \in \mathbb{R}^n$ and $p \geq 1$.

A key result proved in [41] is re-stated in the following theorem, which will be the fundamental tool to establish the convergence of Algorithms 1 and 2.

Theorem 7: Suppose $S_i : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, \ldots, \ell$, are paracontractions w.r.t. the norm $|||\cdot|||$ and $\cap_{i=1}^{\ell} \mathrm{Fix}_{S_i} \neq \emptyset$. Starting from any $x^0 \in \mathbb{R}^n$ and for any sequence $\sigma^0, \sigma^1, \ldots \in \{1, \ldots, \ell\}$ so that each index i appears infinitely often, the iteration

$$x^{t+1} = S_{\sigma^t}(x^t), \quad \forall t = 0, 1, \ldots,$$

will converge to a point $x^* = \lim_{t \to \infty} x^t \in \cap_{i=1}^{\ell} \mathrm{Fix}_{S_i}$.

The techniques above are compatible with the general norm $|||\cdot|||$ while the following lemma on paracontractions only applies w.r.t. the l_2-norm $\|\cdot\|$.

Lemma 2: Suppose $P : \mathbb{R}^n \to \mathbb{R}^n$ is a paracontraction w.r.t. the l_2-norm $\|\cdot\|$. Then the operator $\tilde{P} = M^\top P M : \mathbb{R}^n \to \mathbb{R}^n$, where M is a permutation matrix, is also a paracontraction w.r.t. $\|\cdot\|$ with $\mathrm{Fix}_{\tilde{P}} = M^\top \mathrm{Fix}_P$.

In the rest of this section, unless otherwise stated, all norms are Euclidean norm.

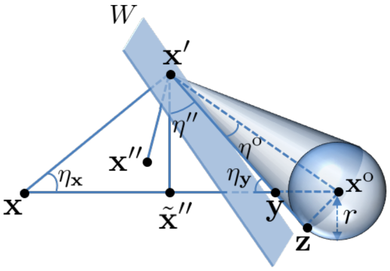


Fig. 4: Proof of Lemma 3.

A. Convergence Proof of Algorithm 1

This section aims to establish the convergence and convergence rate of the synchronous Algorithm 1.

At round t of Algorithm 1, each agent $i \in \mathcal{I}_m$ first computes $\mathbf{z}_i^t = P_i(\mathbf{x}_i^t)$, where P_i is a strongly quasi-nonexpansive map with the fixed point set $\mathcal{F}_i$. Denoting by $\mathbf{z}$ the stacked vector of all $\mathbf{z}_i$, we have

$$\mathbf{z}^t = P\left(\mathbf{x}^t\right) \tag{16}$$

where $P := P_1 \times \cdots \times P_m$ is a paracontraction with $\mathrm{Fix}_P = \mathcal{A}_1$ defined in (8).

The second step, consisting of the consensus operation (10) followed by the broadcast (11), can be expressed as

$$x_{ik}^{t+1} = x_i^{t+1} = Q_i(z_i^t, (z_{ik}^t)_{k \in \mathcal{N}_i^-}),\ k \in \mathcal{N}_i^-,$$

or in a compact form

$$(x_i^{t+1}, (x_{ik}^{t+1})_{k \in \mathcal{N}_i^-}) = \tilde{Q}_i(z_i^t, (z_{ik}^t)_{k \in \mathcal{N}_i^-})$$

where $\tilde{Q}_i(\cdot) := [Q_i, \ldots, Q_i]\,(\cdot)$ is the column concatenation of Q_i's and, with some abuse of notation, Q_i is the matrix $\frac{1}{|\mathcal{N}_i^-|+1}\mathbf{1}^\top \otimes I_{n_i}$ corresponding to the consensus operation in (10). It can be easily seen that, for any $i \in \mathcal{I}_m$, $\tilde{Q}_i$ is exactly the projection operation onto the consensus set $\mathcal{C}_i$ in (7). For simplicity, we reorder the variables of $\mathbf{x}$ as $\tilde{\mathbf{x}} = M\mathbf{x}$, where M is the permutation matrix used in (8). Similarly $\tilde{\mathbf{z}} = M\mathbf{z}$. Then by Lemma 1 the operator $\tilde{Q} : \tilde{\mathbf{z}}^t \mapsto \tilde{\mathbf{x}}^{t+1}$, being $\tilde{Q}_1 \times \cdots \times \tilde{Q}_m$, is a paracontraction w.r.t. the Euclidean norm with the fixed point set $\mathcal{C}_1 \times \cdots \times \mathcal{C}_m$. For the original $\mathbf{x}^t$, it follows that

$$\mathbf{x}^{t+1} = M^\top \tilde{\mathbf{x}}^{t+1} = M^\top \tilde{Q}\left(\tilde{\mathbf{z}}^t\right) = M^\top \tilde{Q} M\left(\mathbf{z}^t\right) = Q\left(\mathbf{z}^t\right),$$

with $Q := M^\top \tilde{Q} M$. By Lemma 2, the operator Q is a paracontraction w.r.t. the Euclidean norm with the fixed point set $\mathcal{A}_2$ in (8).

Proof of Theorem 1:

As discussed above, the sequence $\mathbf{x}^t$ generated by Algorithm 1 is obtained from the iteration

$$\mathbf{x}^{t+1} = Q \circ P(\mathbf{x}^t), \quad t = 0, 1, \ldots,$$

where P and Q are two paracontractions w.r.t. the Euclidean norm whose sets of fixed points are specified by $\mathcal{A}_1$ and $\mathcal{A}_2$ in (8), respectively. By Theorem 7, $\mathbf{x}^t$ will converge to some $\mathbf{x}^* \in \mathcal{A}_1 \cap \mathcal{A}_2$, namely, a solution to Problem 2. ■

To study the convergence rate of Algorithm 1, we need the following result.

Lemma 3: Suppose that $\mathcal{E}_1$ and $\mathcal{E}_2$ are two closed convex subsets of $\mathbb{R}^d$ whose intersection $\mathcal{E}_1 \cap \mathcal{E}_2$ contains an interior point $\mathbf{x}^{\mathrm{o}}$ of $\mathcal{E}_1$, i.e., there exists $r > 0$ such that the closed ball $\mathcal{B}(\mathbf{x}^{\mathrm{o}}, r)$ centered at $\mathbf{x}^{\mathrm{o}}$ with the radius r is contained in $\mathcal{E}_1$. Let $P_{\mathcal{E}_1}$ and $P_{\mathcal{E}_2}$ be the projection operators onto these two sets, respectively. Then, for any $\mathbf{x} \in \mathcal{E}_2 \setminus \mathcal{E}_1$, we have

(a) $d_{\mathcal{E}_1 \cap \mathcal{E}_2}(\mathbf{x}) \le \frac{\|\mathbf{x}-\mathbf{x}^{\mathrm{o}}\|}{r} \cdot d_{\mathcal{E}_1}(\mathbf{x})$,
(b) $d_{\mathcal{E}_1}\left(P_{\mathcal{E}_2}(P_{\mathcal{E}_1}(\mathbf{x}))\right) \le \gamma \cdot d_{\mathcal{E}_1}(\mathbf{x})$,

with the constant $\gamma := \frac{\sqrt{\|\mathbf{x}-\mathbf{x}^{\mathrm{o}}\|^2 - r^2}}{\|\mathbf{x}-\mathbf{x}^{\mathrm{o}}\|} \in [0, 1)$.

Proof: Let $\mathbf{x} \in \mathcal{E}_2 \setminus \mathcal{E}_1$ be arbitrary and denote $\mathbf{x}' = P_{\mathcal{E}_1}(\mathbf{x})$ and $\mathbf{x}'' = P_{\mathcal{E}_2}(\mathbf{x}')$ (see Fig. 4). Without loss of generality assume $\mathbf{x}' \notin \mathcal{E}_2$ (otherwise $\mathbf{x}' = \mathbf{x}''$ resulting zero in the left-hand side and both conclusions are trivial), which implies that $\mathbf{x} \notin \mathcal{B}(\mathbf{x}^{\mathrm{o}}, r)$. Hence $\mathbf{x} \neq \mathbf{x}'$ and $\mathbf{x}' \neq \mathbf{x}''$. Since $\mathbf{x}', \mathbf{x}^{\mathrm{o}} \in \mathcal{E}_1$, the line segment $\overline{\mathbf{x}'\mathbf{x}^{\mathrm{o}}}$ between $\mathbf{x}'$ and $\mathbf{x}^{\mathrm{o}}$ is contained entirely in $\mathcal{E}_1$.

The fact $\mathbf{x}' = P_{\mathcal{E}_1}(\mathbf{x})$ implies that 1) there is a supporting hyperplane W of $\mathcal{E}_1$ that passes through $\mathbf{x}'$ and is orthogonal to $\overline{\mathbf{x}\mathbf{x}'}$, 2) the angle that $\overline{\mathbf{x}'\mathbf{x}^{\mathrm{o}}}$ and $\overline{\mathbf{x}'\mathbf{x}}$ make at $\mathbf{x}'$ is obtuse and thus $\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\| < \|\mathbf{x} - \mathbf{x}^{\mathrm{o}}\|$, and 3) the points $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{x}^{\mathrm{o}}$ constitute a plane W^{o} that is orthogonal to W. Note that $\mathbf{x}$ is on one side of W while the convex hull $\mathcal{C} \subset \mathcal{E}_1$ of the point $\mathbf{x}'$ and the ball $\mathcal{B}(\mathbf{x}^{\mathrm{o}}, r)$ is on the other side.

As shown in Fig. 4, let $\overline{\mathbf{x}'\mathbf{z}} \subset W^{\mathrm{o}}$ be the line segment that is tangential to the sphere $\partial\mathcal{B}(\mathbf{x}^{\mathrm{o}}, r)$ at the point $\mathbf{z}$ and intersects $\overline{\mathbf{x}\mathbf{x}^{\mathrm{o}}}$ at a point $\mathbf{y}$, and let $\tilde{\mathbf{x}}''$ be a point on the line segment $\overline{\mathbf{x}\mathbf{x}^{\mathrm{o}}} \subset \mathcal{E}_2$ such that $\overline{\mathbf{x}'\tilde{\mathbf{x}}''} \perp \overline{\mathbf{x}\mathbf{x}^{\mathrm{o}}}$. Then $d_{\mathcal{E}_1}(\mathbf{x}'') \le \|\mathbf{x}'' - \mathbf{x}'\| \le \|\mathbf{x}' - \tilde{\mathbf{x}}''\|$, with the two inequalities following from the fact that $\mathbf{x}' \in \mathcal{E}_1$ and $\tilde{\mathbf{x}}'' \in \mathcal{E}_2$ are not necessarily the projection points of $\mathbf{x}''$ onto $\mathcal{E}_1$ and $\mathbf{x}'$ onto $\mathcal{E}_2$, respectively.

The angles between the line segments $\overline{\mathbf{x}'\mathbf{y}}$ and $\overline{\mathbf{x}'\mathbf{x}^{\mathrm{o}}}$, $\overline{\mathbf{x}'\tilde{\mathbf{x}}''}$ and $\overline{\mathbf{x}'\mathbf{y}}$, $\overline{\mathbf{x}\mathbf{x}'}$ and $\overline{\mathbf{x}\tilde{\mathbf{x}}''}$, $\overline{\mathbf{y}\mathbf{x}'}$ and $\overline{\mathbf{y}\tilde{\mathbf{x}}''}$, are denoted by η^{o}, η'', $\eta_{\mathbf{x}}$, $\eta_{\mathbf{y}}$, respectively. Obviously, $\sin(\eta^{\mathrm{o}}) = r/\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|$, $\eta'' \le 90° - \eta^{\mathrm{o}}$ and $\eta_{\mathbf{y}} \ge \eta^{\mathrm{o}}$.

By the geometric relationship in the plane W^{o}, we have

$$\begin{aligned}
d_{\mathcal{E}_1 \cap \mathcal{E}_2}(\mathbf{x}) &\le \|\mathbf{x} - \mathbf{y}\| = \|\mathbf{x} - \tilde{\mathbf{x}}''\| + \|\tilde{\mathbf{x}}'' - \mathbf{y}\| \\
&= \|\mathbf{x} - \mathbf{x}'\| \cos(\eta_{\mathbf{x}}) + \|\mathbf{x} - \mathbf{x}'\| \sin(\eta_{\mathbf{x}}) \tan(\eta'') \\
&\le \|\mathbf{x} - \mathbf{x}'\| \cos(\eta_{\mathbf{x}}) + \|\mathbf{x} - \mathbf{x}'\| \sin(\eta_{\mathbf{x}}) \tan(90° - \eta^{\mathrm{o}}) \\
&= \|\mathbf{x} - \mathbf{x}'\| \frac{\sin(\eta_{\mathbf{x}} + \eta^{\mathrm{o}})}{\sin(\eta^{\mathrm{o}})} \le \|\mathbf{x} - \mathbf{x}'\| / \sin(\eta^{\mathrm{o}}) \\
&= d_{\mathcal{E}_1}(\mathbf{x}) \frac{\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|}{r} \le d_{\mathcal{E}_1}(\mathbf{x}) \frac{\|\mathbf{x} - \mathbf{x}^{\mathrm{o}}\|}{r},
\end{aligned}$$

which is the desired conclusion (a).

Since the angle that $\overline{\mathbf{x}'\mathbf{x}^{\mathrm{o}}}$ and $\overline{\mathbf{x}'\mathbf{x}}$ make at $\mathbf{x}'$ can not be acute, $\eta_{\mathbf{x}} + \eta_{\mathbf{y}} \le 90°$ holds, implying that $0 \le \eta_{\mathbf{x}} \le 90° - \eta_{\mathbf{y}} \le 90° - \eta^{\mathrm{o}} \le 90°$. Then $\sin(\eta_{\mathbf{x}}) \le \sin(90° - \eta^{\mathrm{o}}) = \cos(\eta^{\mathrm{o}}) = \sqrt{\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|^2 - r^2} / \|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|$. Therefore,

$$\begin{aligned}
\frac{d_{\mathcal{E}_1}(\mathbf{x}'')}{d_{\mathcal{E}_1}(\mathbf{x})} &\le \frac{\|\mathbf{x}' - \tilde{\mathbf{x}}''\|}{\|\mathbf{x}' - \mathbf{x}\|} = \sin(\eta_x) \le \frac{\sqrt{\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|^2 - r^2}}{\|\mathbf{x}' - \mathbf{x}^{\mathrm{o}}\|} \\
&\le \frac{\sqrt{\|\mathbf{x} - \mathbf{x}^{\mathrm{o}}\|^2 - r^2}}{\|\mathbf{x} - \mathbf{x}^{\mathrm{o}}\|} = \gamma.
\end{aligned}$$

Combined with the trivial case that $\mathbf{x}' = \mathbf{x}''$ when $\mathbf{x} \in \mathcal{B}(\mathbf{x}^{\mathrm{o}}, r)$, the conclusion (b) is proved. ■

Using Lemma 3, we are ready to prove the exponential convergence rate of Algorithm 1.

Proof of Theorem 2:

As shown in the proof of Theorem 1, given $\alpha_i = 1, \forall i \in \mathcal{I}_m$, the sequence $\{\mathbf{x}^t\}$ generated by Algorithm 1 satisfies the condition that $\mathbf{x}^t \in \mathcal{A}_2$ and $\mathbf{x}^{t+1} = P_{\mathcal{A}_2}(P_{\mathcal{A}_1}(\mathbf{x}^t))$, $\forall t = 0, 1, \ldots$, with $\mathcal{A}_1$ and $\mathcal{A}_2$ defined in (8), and that $\lim_{t\to\infty} \mathbf{x}^t = \mathbf{x}^* \in \mathcal{A}_1 \cap \mathcal{A}_2$ as a consequence.

By our assumption on $x^{\rm o}$ and setting $x_{ik}^{\rm o} = x_i^{\rm o}, \forall k \in \mathcal{N}_i^-$, the corresponding $\mathbf{x}^{\rm o}$ has the properties that $\mathbf{x}^{\rm o} \in \mathcal{A}_2$ by construction, and each $\mathbf{x}_i^{\rm o} \in \mathcal{F}_i^{\rm o}$ which leads to the fact that $\mathbf{x}^{\rm o}$ is an interior point of $\mathcal{A}_1$. Thus by combining two conclusions in Lemma 3 and the fact that $\|\mathbf{x}^t - \mathbf{x}^{\rm o}\|$ is nonincreasing and hence bounded, $d_{\mathcal{A}_1\cap\mathcal{A}_2}(\mathbf{x}^t)$ decays to zero exponentially fast.

Let $t \geq 0$ be arbitrary and denote $\mathbf{y}^t := \mathcal{P}_{\mathcal{A}_1\cap\mathcal{A}_2}(\mathbf{x}^t)$. Since $\mathbf{y}^t \in \mathcal{A}_1 \cap \mathcal{A}_2$, $\|\mathbf{x}^{t+s} - \mathbf{y}^t\|$ is nonincreasing in s for $s \geq 0$, resulted from the facts that $\mathbf{x}^{t+s+1} = P_{\mathcal{A}_2}(P_{\mathcal{A}_1}(\mathbf{x}^{t+s}))$ and that both $P_{\mathcal{A}_1}$ and $P_{\mathcal{A}_2}$ are paracontractions with $\mathbf{y}^t$ being one of their fixed points. Thus $\|\mathbf{x}^t - \mathbf{y}^t\| \geq \lim_{s\to\infty} \|\mathbf{x}^{t+s} - \mathbf{y}^t\| = \|\mathbf{x}^* - \mathbf{y}^t\|$, which leads to

$$\|\mathbf{x}^t - \mathbf{x}^*\| \leq \|\mathbf{x}^t - \mathbf{y}^t\| + \|\mathbf{x}^* - \mathbf{y}^t\| \leq 2\|\mathbf{x}^t - \mathbf{y}^t\| = 2d_{\mathcal{A}_1\cap\mathcal{A}_2}(\mathbf{x}^t).$$

Therefore, $\mathbf{x}^t$ converges to $\mathbf{x}^*$ exponentially fast. ■

B. Convergence Proof of Algorithm 2

Similarly to Algorithm 1, we will show in the following that the augmented variable $\mathbf{x}^t$ is updated at each round of Algorithm 2 by the composition of two paracontractions.

Proof of Theorem 3:

Under Assumption 4, the update at round t can be written as

$$\mathbf{x}^{t+1} = Q^t \circ P^t \left(\mathbf{x}^t\right).$$

Here, the operator P^t is defined as

$$P^t = P_1^t \times \cdots \times P_m^t, \tag{17}$$

where $P_i^t : \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$ is the P_i defined in (9) if agent i performs the relaxed projection at this round, and otherwise the identity map. The operator Q^t is defined as

$$Q^t = M^\top \tilde{Q}^t M$$

where M is the same permutation matrix defined in (8) and

$$\tilde{Q}^t = \tilde{Q}_1^t \times \cdots \times \tilde{Q}_m^t$$

with each $\tilde{Q}_i^t : \mathbb{R}^{n_i(1+|\mathcal{N}_i^-|)} \to \mathbb{R}^{n_i(1+|\mathcal{N}_i^-|)}$ being

$$\tilde{Q}_i^t = (M_i^t)^\top \left[Q_i^t, \cdots, Q_i^t, I_{n_i}, \cdots, I_{n_i}\right] M_i^t. \tag{18}$$

if agent i performs the partial consensus, and otherwise the identity map. In (18), M_i^t is a permutation matrix that puts the agents in $\{i\} \cup \mathcal{N}_{i,t}^-$ at the front of the group $\{i\} \cup \mathcal{N}_i^-$; with a little abuse of notation, Q_i^t being the matrix $\frac{1}{|\mathcal{N}_{i,t}^-|+1}\mathbf{1}^\top \otimes I_{n_i}$ corresponding to the operation (12) appears $|\mathcal{N}_{i,t}^-| + 1$ times. By repeatedly applying Lemmas 1 and 2, we know P^t and Q^t are paracontractions w.r.t. the Euclidean norm.

It is easy to see that the number of possible operators P^t and Q^t for $t = 0, 1, \ldots$, is finite. Moreover, P^t's and Q^t's have the common fixed point sets $\mathcal{A}_1$ and $\mathcal{A}_2$ defined in (8), respectively. Hence, under Assumption 5, $\mathbf{x}^t$ will converge to some $\mathbf{x}^* \in \mathcal{A}_1 \cap \mathcal{A}_2$ as a consequence of Theorem 7. ■

The convergence of Algorithm 2b can be proven similarly.

Proof of Theorem 4:

Since the matrix $W_{ik} \in \mathbb{R}^{2\times 2}$ used in the update (13) is doubly stochastic, it can be explicitly expressed as

$$W_{ik} = \begin{bmatrix} 1-\beta_{ik} & \beta_{ik} \\ \beta_{ik} & 1-\beta_{ik} \end{bmatrix}$$

with $\beta_{ik} \in (0, 1)$. The corresponding update is

$$\begin{bmatrix} x_i^{t+1} \\ x_{ik}^{t+1} \end{bmatrix} = (1-2\beta_{ik}) \begin{bmatrix} x_i^t \\ x_{ik}^t \end{bmatrix} + 2\beta_{ik} \begin{bmatrix} (x_i^t + x_{ik}^t)/2 \\ (x_i^t + x_{ik}^t)/2 \end{bmatrix},$$

which is a $(2\beta_{ik})$-relaxed projection onto the consensus set $\{(x_i, x_{ik}) | x_i = x_{ik}\}$, and therefore a paracontraction w.r.t. the Euclidean norm. The remaining proof is exactly the same as that of Theorem 3. ■

In general, the convergence of Algorithms 2 and 2b is not exponential.

C. Convergence Proof of Algorithm 3

The convergence analysis of Algorithm 3 is more challenging than Algorithms 1 and 2 because the operation in (15) is no longer a paracontraction. Instead, our proof will utilize the property of strongly quasi-nonexpansive maps.

For any agent $i \in \mathcal{I}_m$, its augmented variable $\mathbf{x}_i$'s update at round t of Algorithm 3 can be summarized as

$$\mathbf{z}_i^t = P_i^t(\mathbf{x}_i^t) \tag{19}$$

$$\mathbf{x}_i^{t+1} = \sum_{k\in\mathcal{I}_m} \left(Q^t\right)_{ik} \mathbf{z}_k^t, \tag{20}$$

where

$$P_i^t = \begin{cases} P_i, & \text{if } i \in \mathcal{I}_P^t, \\ \mathrm{Id}, & \text{otherwise;} \end{cases} \tag{21}$$

and Q^t is a doubly stochastic matrix whose block in the i-th row and k-th column, denoted as $(Q^t)_{ik}$, $\forall i, k \in \mathcal{I}_m$, will be defined shortly. In sum, the dynamics is

$$\mathbf{z}^t = P^t(\mathbf{x}^t) \tag{22}$$

$$\mathbf{x}^{t+1} = Q^t \mathbf{z}^t. \tag{23}$$

To define Q^t, we first reorder $\mathbf{x}$ and $\mathbf{z}$ as $\tilde{\mathbf{x}} = M\mathbf{x}$ and $\tilde{\mathbf{z}} = M\mathbf{z}$, respectively, using the same permutation matrix M in (8). Then the consensus step (15) will result in

$$\tilde{\mathbf{x}}^{t+1} = W^t \tilde{\mathbf{z}}^t, \tag{24}$$

with $W^t := \mathrm{diag}\left(W_1^t \otimes I_{n_1}, W_2^t \otimes I_{n_2}, \cdots, W_m^t \otimes I_{n_m}\right)$. If $i \notin \mathcal{I}_Q^t$, i.e., agent i is not activated to perform the consensus update (15) at round t, W_i^t is set to be the identity map Id. For $\mathbf{x}$, the following holds

$$\mathbf{x}^{t+1} = M^\top \tilde{\mathbf{x}}^{t+1} = M^\top W^t \tilde{\mathbf{z}}^t = M^\top W^t M \mathbf{z}^t = Q^t \mathbf{z}^t,$$

under the definition $Q^t := M^\top W^t M$. As a consequence of Assumption 6(a), W^t, and hence Q^t, is doubly stochastic.

With the dynamics (22) and (23), we next establish the convergence by showing first the intermediate values $\mathbf{z}^t$ will converge to a point in $\mathcal{A}_1 \cap \mathcal{A}_2$, i.e., a solution to Problem 2.

To proceed, define agent i's displacement vector as

$$\mathbf{e}_i^t := P_i^t(\mathbf{x}_i^t) - \mathbf{x}_i^t = \mathbf{z}_i^t - \mathbf{x}_i^t, \tag{25}$$

with P_i^t in (21). Note that if $P_i^t = P_i$, then

$$\|\mathbf{e}_i^t\| = \|(1-\alpha_i)\mathbf{x}_i^t + \alpha_i P_{\mathcal{F}_i}\left(\mathbf{x}_i^t\right) - \mathbf{x}_i^t\| \\ = \alpha_i \|P_{\mathcal{F}_i}\left(\mathbf{x}_i^t\right) - \mathbf{x}_i^t\| = \alpha_i\, d_{\mathcal{F}_i}\left(\mathbf{x}_i^t\right).$$

The following lemma shows that $\mathbf{e}_i^t$ will converge to zero.

Lemma 4: Suppose Assumption 6 holds. The displacement vector $\mathbf{e}^t := (\mathbf{e}_i^t)_{i\in\mathcal{I}_m} \to 0$ as $t \to \infty$.

Proof: Let $\mathbf{y} \in \mathcal{A}_1 \cap \mathcal{A}_2$ be a solution of Problem 2. Then for all $t = 0, 1, \ldots$, $\mathbf{y} = Q^t \mathbf{y}$ because the matrix Q^t is stochastic. As discussed at the beginning of this section, P_i defined in (9) is a β_i-strongly quasi-nonexpansive map with $\beta_i := \frac{2-\alpha_i}{\alpha_i}$, i.e.,

$$\|P_i(\mathbf{x}_i^t) - \mathbf{y}_i\|^2 \le \|\mathbf{x}_i^t - \mathbf{y}_i\|^2 - \beta_i \|P_i(\mathbf{x}_i^t) - \mathbf{x}_i^t\|^2.$$

When replacing P_i by the identity map Id, the above inequality still holds for the same β_i. It then follows that

$$\|\mathbf{z}_i^t - \mathbf{y}_i\|^2 \le \|\mathbf{x}_i^t - \mathbf{y}_i\|^2 - \underline{\beta}\|\mathbf{e}_i^t\|^2,$$

with $\underline{\beta} := \min_{i\in\mathcal{I}_m} \beta_i = \min_{i\in\mathcal{I}_m} \left\{\frac{2-\alpha_i}{\alpha_i}\right\}$. Combining with (20) we have

$$\underline{\beta}\|\mathbf{e}_i^t\|^2 \le \left\| \sum_{k\in\mathcal{I}_m} (Q^{t-1})_{ik}\mathbf{z}_k^{t-1} - \mathbf{y}_i \right\|^2 - \left\|\mathbf{z}_i^t - \mathbf{y}_i\right\|^2. \quad (26)$$

Define an element-wise convex map $\Gamma : \mathbb{R}^N \to \mathbb{R}^N$ such that $\Gamma(x)_l = x_l^2$, $l \in \mathcal{I}_N$. Then (26) will lead to

$$\begin{aligned} \underline{\beta}\|\mathbf{e}^t\|^2 &= \sum_{i\in\mathcal{I}_m} \underline{\beta}\|\mathbf{e}_i^t\|^2 \\ &\le \mathbf{1}^\top\Big\{\Gamma\big(Q^{t-1}\mathbf{z}^{t-1} - \mathbf{y}\big) - \Gamma\big(\mathbf{z}^t - \mathbf{y}\big)\Big\} \\ &= \mathbf{1}^\top\Big\{\Gamma\left(Q^{t-1}\left(\mathbf{z}^{t-1} - \mathbf{y}\right)\right) - \Gamma\big(\mathbf{z}^t - \mathbf{y}\big)\Big\} \\ &\le \mathbf{1}^\top\Big\{Q^{t-1}\Gamma\big(\mathbf{z}^{t-1} - \mathbf{y}\big) - \Gamma\big(\mathbf{z}^t - \mathbf{y}\big)\Big\} \\ &= \mathbf{1}^\top\Gamma\big(\mathbf{z}^{t-1} - \mathbf{y}\big) - \mathbf{1}^\top\Gamma\big(\mathbf{z}^t - \mathbf{y}\big) \\ &= \|\mathbf{z}^{t-1} - \mathbf{y}\|^2 - \|\mathbf{z}^t - \mathbf{y}\|^2, \end{aligned} \quad (27)$$

where the second inequality and the second last equality follow from the convexity of Γ and the doubly stochasticity of Q^{t-1}, respectively. For any time instant $\bar{t} > 0$, summing the above inequalities for $t = 1, \ldots, \bar{t}$ will result in

$$\sum_{t=1}^{\bar{t}} \underline{\beta}\|\mathbf{e}^t\|^2 \le \|\mathbf{z}^0 - \mathbf{y}\|^2 - \|\mathbf{z}^{\bar{t}} - \mathbf{y}\|^2 \le \|\mathbf{z}^0 - \mathbf{y}\|^2.$$

Since the inequality holds for arbitrarily large $\bar{t}$, we have

$$\sum_{t=1}^{\infty} \underline{\beta}\|\mathbf{e}^t\|^2 \le \|\mathbf{z}^0 - \mathbf{y}\|^2 < \infty.$$

With $\underline{\beta} > 0$, this directly leads to $\lim_{t\to\infty} \mathbf{e}^t = 0$. ■

With Lemma 4, we next show that $\mathbf{z}^t$ will asymptotically satisfy the consensus constraint in (6) as $t \to \infty$.

Lemma 5: Suppose Assumptions 6 and 7 hold. Then $\forall i \in \mathcal{I}_m$ and $\forall k \in \mathcal{N}_i^-$, $\lim_{t\to\infty} \|z_i^t - z_{ik}^t\| = 0$.

Proof: In this proof we focus on the reordered variables $\tilde{\mathbf{x}}_i = \left(x_i, (x_{ik})_{k\in\mathcal{N}_i^-}\right)$ and $\tilde{\mathbf{z}}_i = \left(z_i, (z_{ik})_{k\in\mathcal{N}_i^-}\right)$ for an arbitrary $i \in \mathcal{I}_m$, whose dynamics according to (15) can be written as $\tilde{\mathbf{x}}_i^{t+1} = \left(W_i^t \otimes I_{n_i}\right)\tilde{\mathbf{z}}_i^t$. Combined with the fact $\tilde{\mathbf{z}}_i^{t+1} = \tilde{\mathbf{x}}_i^{t+1} + \tilde{\mathbf{e}}_i^{t+1}$, it follows that

$$\tilde{\mathbf{z}}_i^{t+1} = \left(W_i^t \otimes I_{n_i}\right)\tilde{\mathbf{z}}_i^t + \tilde{\mathbf{e}}_i^{t+1}.$$

For $s \le t$, repeatedly applying the above equation yields

$$\tilde{\mathbf{z}}_i^{t+1} = \left(\Phi_i^{t,s} \otimes I_{n_i}\right)\tilde{\mathbf{z}}_i^s + \sum_{r=s+1}^{t} \left(\Phi_i^{t,r} \otimes I_{n_i}\right)\tilde{\mathbf{e}}_i^r + \tilde{\mathbf{e}}_i^{t+1}, \quad (28)$$

where $\Phi_i^{t,s} := W_i^t W_i^{t-1} \cdots W_i^{s+1} W_i^s$ and $\Phi_i^{t,s} = W_i^t$ when $s = t$. Obviously, $\Phi_i^{t,s}$ is doubly stochastic under Assumption 6(a).

Under Assumption 7(b), the sequence of graphs $\{\mathcal{G}_i^t\}$ associated with the matrix sequence $\{W_i^t\}$ is repeatedly jointly strongly connected (see Appendix for the definitions). Together with Assumption 6 on W_i^t, this implies that, for any fixed s, every entry of $\Phi_h^{t,s}$ will converge to $1/\left(1 + |\mathcal{N}_i^-|\right)$ exponentially fast as $t \to \infty$ as shown by [1, Prop. 1]. More precisely,

$$\left| [\Phi_i^{t,s}]_{kl} - \frac{1}{1 + |\mathcal{N}_i^-|} \right| \le c\lambda^{t-s}$$

for all $k, l \in \mathcal{I}_{1+|\mathcal{N}_i^-|}$. Here, the constants $c > 0$ and $\lambda \in [0, 1)$ are determined by the cardinality of $\mathcal{N}_i^-$, $\underline{w}$ in Assumption 6(b) and T from Assumption 7(b) (see [1, Prop. 1]).

Following (28), for $k \in \mathcal{I}_{1+|\mathcal{N}_i^-|}$, the k-th subvector in $\tilde{\mathbf{z}}_i^{t+1}$ (namely, z_i^{t+1} and its desired values by its out-neighbors) is given by

$$[\tilde{\mathbf{z}}_i^{t+1}]_k = \sum_{l=1}^{1+|\mathcal{N}_i^-|} [\Phi_i^{t,s}]_{kl}[\tilde{\mathbf{z}}_i^s]_l + \sum_{r=s+1}^{t} \sum_{l=1}^{1+|\mathcal{N}_i^-|} [\Phi_i^{t,r}]_{kl}[\tilde{\mathbf{e}}_i^r]_l \\ + [\tilde{\mathbf{e}}_i^{t+1}]_l.$$

Define y_i^{t+1} to be the average of z_i^{t+1} and its desired values from out-neighbors. Then

$$\begin{aligned} y_i^{t+1} &= \frac{1}{1 + |\mathcal{N}_i^-|}\left(\mathbf{1}^\top \otimes I_{n_i}\right)\tilde{\mathbf{z}}_i^{t+1} \\ &= \frac{1}{1 + |\mathcal{N}_i^-|}\Big\{\left(\mathbf{1}^\top \otimes I_{n_i}\right)\tilde{\mathbf{z}}_i^s + \sum_{r=s+1}^{t}\left(\mathbf{1}^\top \otimes I_{n_i}\right)\tilde{\mathbf{e}}_i^r \\ &\quad + \left(\mathbf{1}^\top \otimes I_{n_i}\right)\tilde{\mathbf{e}}_i^{t+1}\Big\} \\ &= \frac{1}{1 + |\mathcal{N}_i^-|}\Big\{\sum_{l=1}^{1+|\mathcal{N}_i^-|}[\tilde{\mathbf{z}}_i^s]_l + \sum_{r=s+1}^{t}\sum_{l=1}^{1+|\mathcal{N}_i^-|}[\tilde{\mathbf{e}}_i^r]_l \\ &\quad + \sum_{l=1}^{1+|\mathcal{N}_i^-|}[\tilde{\mathbf{e}}_i^{t+1}]_l\Big\}. \end{aligned}$$

Note that the second equality follows from (28) and the fact that $\Phi_i^{t,s}$ is doubly stochastic. Then for $k \in \mathcal{I}_{1+|\mathcal{N}_i^-|}$, we have

$$\begin{aligned}
&\left\|[\tilde{\mathbf{z}}_i^{t+1}]_k - y_i^{t+1}\right\|_1 \\
&= \Bigg\| \sum_{l=1}^{1+|\mathcal{N}_i^-|} \Big([\Phi_i^{t,s}]_{kl} - \frac{1}{1+|\mathcal{N}_i^-|}\Big)[\tilde{\mathbf{z}}_i^s]_l \\
&\quad + \sum_{r=s+1}^{t} \sum_{l=1}^{1+|\mathcal{N}_i^-|} \Big([\Phi_i^{t,r}]_{kl} - \frac{1}{1+|\mathcal{N}_i^-|}\Big)[\tilde{\mathbf{e}}_i^r]_l \\
&\quad + [\tilde{\mathbf{e}}_i^{t+1}]_k - \frac{1}{1+|\mathcal{N}_i^-|} \sum_{l=1}^{1+|\mathcal{N}_i^-|} [\tilde{\mathbf{e}}_i^{t+1}]_l \Bigg\|_1 \\
&\le c\lambda^{t-s}\|\tilde{\mathbf{z}}_i^s\|_1 + \sum_{r=s+1}^{t} c\lambda^{t-r}\|\tilde{\mathbf{e}}_i^r\|_1 + \|\tilde{\mathbf{e}}_i^{t+1}\|_1 \\
&\le cc_1\lambda^{t-s}\|\tilde{\mathbf{z}}_i^s\|_2 + \sum_{r=s+1}^{t} cc_1\lambda^{t-r}\|\tilde{\mathbf{e}}_i^r\|_2 + c_1\|\tilde{\mathbf{e}}_i^{t+1}\|_2
\end{aligned}$$

where $c_1 = \sqrt{1+|\mathcal{N}_i^-|}$, following from the fact $\|x\|_1 \le \sqrt{n}\|x\|_2$, $x \in \mathbb{R}^n$. By Lemma 4, $\forall \epsilon > 0$, there exists an s such that for all $t \ge s$, $\|\tilde{\mathbf{e}}_i^t\| < \epsilon$. This leads to

$$\left\|[\tilde{\mathbf{z}}_i^{t+1}]_k - y_i^{t+1}\right\|_1 \le cc_1\lambda^{t-s}\|\tilde{\mathbf{z}}_i^s\|_2 + cc_1\epsilon\frac{1-\lambda^{t-s}}{1-\lambda} + c_1\epsilon.$$

Since ϵ can be arbitrarily small, the following holds

$$\lim_{t\to\infty} \left\|[\tilde{\mathbf{z}}_i^{t+1}]_k - y_i^{t+1}\right\|_1 = 0,$$

i.e., z_i^t and z_{ik}^t for $k \in \mathcal{N}_i^-$ reach consensus asymptotically. ■

Finally, we establish the convergence of the generalized synchronous algorithm stated in Theorem 5.

Proof of Theorem 5:

From (27) we have

$$\|\mathbf{z}^t - \mathbf{y}\|^2 \le \|\mathbf{z}^{t-1} - \mathbf{y}\|^2. \tag{29}$$

Thus the sequence $\{\|\mathbf{z}^t - \mathbf{y}\|^2\}$ is non-increasing for any $\mathbf{y} \in \mathcal{A}_1 \cap \mathcal{A}_2$. In particular, this implies that the sequence $\{\mathbf{z}^t\}$ is bounded and has accumulation points.

Next we prove the accumulation point is unique. Let $\mathbf{z}^*$ be a point that $\mathbf{z}^t$ converges to along the time subsequence $\{t_s\}$. As a consequence of Lemma 5, $\mathbf{z}^* \in \mathcal{A}_2$.

In the first case, assume $\mathbf{z}^* \in \mathcal{A}_1$, implying that $\mathbf{z}^* \in \mathcal{A}_1 \cap \mathcal{A}_2$. Let $\hat{\mathbf{z}}^* \ne \mathbf{z}^*$ be a distinct accumulation point that $\{\mathbf{z}^t\}$ converges to along the time subsequence $\{\hat{t}_s\}$. Without loss of generality, assume $\hat{t}_s > t_s$ for all s. Then by replacing $\mathbf{y}$ in (29) with $\mathbf{z}^*$, we have $\|\mathbf{z}^{\hat{t}_s} - \mathbf{z}^*\|^2 \le \|\mathbf{z}^{t_s} - \mathbf{z}^*\|^2$. As $s \to \infty$, we have $\|\hat{\mathbf{z}}^* - \mathbf{z}^*\|^2 \le 0$, which contradicts the assumption that $\hat{\mathbf{z}}^* \ne \mathbf{z}^*$. Therefore there is only one accumulation point.

In the second case, assume $\mathbf{z}^* \notin \mathcal{A}_1$. Then there exists an integer $r \in \mathcal{I}_m$ such that $\mathbf{z}^*$ violates a total of r out of the m feasibility constraints in (5). Without loss of generality, the first r constraints are assumed to be violated, i.e., $d_{\mathcal{F}_i}(\mathbf{z}_i^*) > 0$ for all $i \in \mathcal{I}_r$. Pick any δ such that $0 < \delta \le \min_{i\in\mathcal{I}_r} d_{\mathcal{F}_i}(\mathbf{z}_i^*)$. Then as a consequence of Lemma 4, there exists a large enough integer $K > 0$ such that for all $t \ge K$, $\|\mathbf{e}^t\| \le \delta\underline{\alpha}/8$ with $\underline{\alpha} := \min_{i\in\mathcal{I}_m} \alpha_i$. Suppose at time $t_1 \ge K$, $\mathbf{z}^{t_1} \in \mathcal{B}(\mathbf{z}^*, \delta/4)$. This implies that for all $i \in \mathcal{I}_r$,

$$d_{\mathcal{F}_i}(\mathbf{z}_i^{t_1}) \ge d_{\mathcal{F}_i}(\mathbf{z}_i^*) - \|\mathbf{z}_i^{t_1} - \mathbf{z}_i^*\| \ge \delta - \frac{\delta}{4} = \frac{3\delta}{4}.$$

At the same time, the next iteration value $\mathbf{x}^{t_1+1}$ satisfies

$$\begin{aligned}
&\|\mathbf{x}^{t_1+1} - \mathbf{z}^{t_1}\| = \|Q^{t_1}\mathbf{z}^{t_1} - Q^{t_1}\mathbf{z}^* + \mathbf{z}^* - \mathbf{z}^{t_1}\| \\
&\le \|Q^{t_1} - I\|\|\mathbf{z}^{t_1} - \mathbf{z}^*\| \le (\|Q^{t_1}\| + 1)\|\mathbf{z}^{t_1} - \mathbf{z}^*\| \\
&= 2\|\mathbf{z}^{t_1} - \mathbf{z}^*\| \le \delta/2.
\end{aligned}$$

Here, we use the fact that $\|Q^{t_1}\| = 1$ for the doubly stochastic matrix Q^{t_1}. Combining the above two results, we obtain $\forall i \in \mathcal{I}_r$,

$$d_{\mathcal{F}_i}\left(\mathbf{x}_i^{t_1+1}\right) \ge d_{\mathcal{F}_i}\left(\mathbf{z}_i^{t_1}\right) - \|\mathbf{x}_i^{t_1+1} - \mathbf{z}_i^{t_1}\| \ge \frac{3\delta}{4} - \frac{\delta}{2} = \frac{\delta}{4}.$$

In the next relaxed projection update, if $i \in \mathcal{I}_r \cap \mathcal{I}_P^{t_1+1}$, i.e., agent $i \in \mathcal{I}_r$ is activated to carry out projection at round t_1+1, the resulted displacement vector $\mathbf{e}^{t_1+1}$ satisfies

$$\|\mathbf{e}^{t_1+1}\| \ge \|\mathbf{e}_i^{t_1+1}\| = \alpha_i d_{\mathcal{F}_i}\left(\mathbf{x}_i^{t_1+1}\right) \ge \frac{\underline{\alpha}\delta}{4},\ i \in \mathcal{I}_r,$$

which contradicts the previous assumption that $\|\mathbf{e}^t\| \le \delta\underline{\alpha}/8$ for any $t \ge K$. Therefore, we must have $\mathcal{I}_r \cap \mathcal{I}_P^{t_1+1} = \emptyset$. This implies that the iteration from $\mathbf{z}^{t_1}$ to $\mathbf{z}^{t_1+1}$ is through the operator $P^{t_1+1} \circ Q^{t_1}$ where P^{t_1+1} satisfies that $P_i^{t_1+1} = \mathrm{Id}$ for $i \in \mathcal{I}_r$. Equivalently, we can view this step as one iteration of Algorithm 3 applied to a new problem, which is the same as Problem 2 except that the feasible sets $\mathcal{F}_1, \dots, \mathcal{F}_r$ are relaxed to be the entire spaces of proper dimensions while $\mathcal{F}_{r+1}, \dots, \mathcal{F}_m$ remain unchanged. Since $\mathbf{z}^*$ is in the consensus subspace $\mathcal{A}_2$ and satisfies the constraints $\mathcal{F}_{r+1}, \dots, \mathcal{F}_m$, it is a solution to the relaxed problem. By following the same arguments we used previously to derive (27), we can show that

$$\|\mathbf{z}^{t_1+1} - \mathbf{z}^*\| \le \|\mathbf{z}^{t_1} - \mathbf{z}^*\| \le \delta/4.$$

In other words, $\mathbf{z}^{t_1+1} \in \mathcal{B}(\mathbf{z}^*, \delta/4)$. By repeating the above steps and induction, we conclude that the sequence $\{\mathbf{z}^t\}$ will stay inside the closed ball $\mathcal{B}(\mathbf{z}^*, \delta/4)$ for all $t \ge t_1$. Since the choice of $\delta > 0$ can be arbitrarily small, there will be no other accumulation points besides $\mathbf{z}^*$.

In summary, the accumulation point of $\{\mathbf{z}^t\}$ is unique, i.e., $\lim_{t\to\infty} \mathbf{z}^t = \mathbf{z}^*$. Also $\lim\limits_{t\to\infty} \mathbf{x}^t = \mathbf{z}^*$ holds based on the facts that $\mathbf{x}^t = \mathbf{z}^t - \mathbf{e}^t$ from (25) and $\lim\limits_{t\to\infty} \mathbf{e}^t = 0$ in Lemma 4.

Now we show that $\mathbf{z}^* \in \mathcal{A}_1$. With P_i^t defined in (21), under Assumption 7(a), let $\{\tau\}$ be the subsequence of $\{t\}$ that $P_i^\tau = P_i$. Then $\{\mathbf{x}_i^\tau\}$ and $\{\mathbf{e}_i^\tau\}$ are subsequences of $\{\mathbf{x}_i^t\}$ and $\{\mathbf{e}_i^t\}$, respectively. Since $\lim\limits_{t\to\infty} \mathbf{x}_i^t = \mathbf{z}_i^*$, $\forall i \in \mathcal{I}_m$, we have

$$d_{\mathcal{F}_i}\left(\mathbf{z}_i^*\right) = \lim_{\tau\to\infty} d_{\mathcal{F}_i}\left(\mathbf{x}_i^\tau\right) = (1/\alpha_i)\lim_{\tau\to\infty}\|\mathbf{e}_i^\tau\| = 0,$$

where the second equality follows from the argument after (25). Therefore, $\mathbf{z}_i^* \in \mathcal{F}_i$, $\forall i \in \mathcal{I}_m$, or equivalently, $\mathbf{z}^* \in \mathcal{A}_1$. This completes the proof. ■

D. Convergence Proof of Algorithm 4

Under Assumption 4, the relaxed projection and generalized partial consensus operations operate on two disjoint sets of variables. Therefore the operation at round t can be split into two steps:

$$\mathbf{z}^t = P^t(\mathbf{x}^t),$$
$$\mathbf{x}^{t+1} = Q^t \mathbf{z}^t,$$

where $\mathbf{z}^t$ denotes the intermediate value resulted from all the relaxed projections of this round, $P^t = P_1^t \times \cdots \times P_m^t$ with $P_i^t = P_i$ defined in (9) if agent i performs the relaxed projection at this round and $P_i^t = \mathrm{Id}$ if otherwise, and Q^t is the same stochastic matrix defined in (20).

Obviously, the augmented variable $\mathbf{x}^t$'s dynamics is identical to that of Algorithm 3. Under Assumptions 6 and 7, the convergence proof of Algorithm 4 is the same as that of Algorithm 3 and therefore omitted here.

VI. Simulation Results

In this section, the simulation results of several numerical examples are presented. In all examples, the initial values $x_{ik}^0 = x_i^0$ for all $i \in \mathcal{I}_m, k \in \mathcal{N}_i^-$.

We firstly apply Algorithms 1 and 3 to solve Example 4 with $\varepsilon = 0, 0.01, 0.5$, respectively, which is a linear equation when $\varepsilon = 0$ and linear programs for others. For comparison, in Algorithm 3 all agents are assumed to take part in both the projection and consensus operations for all rounds. The parameter α_i in the relaxed projection (9) is $\alpha_i = 1.5$, $i \in \{1, 2, 3\}$ in both Algorithms 1 and 3 and the weight matrices in (15) of Algorithm 3 are, for all $t = 1, 2, \ldots$

$$W_1^t = \begin{bmatrix}1\end{bmatrix}, \ W_2^t = \begin{bmatrix}0.1 & 0.9\\ 0.9 & 0.1\end{bmatrix}, \ W_3^t = \begin{bmatrix}0.04 & 0.48 & 0.48\\ 0.48 & 0.04 & 0.48\\ 0.48 & 0.48 & 0.04\end{bmatrix}.$$

The results are shown in Fig. 5, where $\mathbf{x}^*$ is the augmented variable corresponding to the unique solution $x^* = (0, 2, -1)$ when $\varepsilon = 0$, and the converged feasible solution when $\varepsilon \neq 0$. As can be seen, Algorithm 3 with proper assigned weights converges significantly faster than Algorithm 1 in all cases.

We next consider the network localization problem in section III-B. Thirty agents are randomly placed inside a planar region. Among them, two are anchors who know their exact locations (at least two anchors are needed to localize the network [12]). The other free agents need to estimate their positions based on the relative orientation measurements from their neighbors that are within a certain range. In Fig. 6, (a) shows the random initial guesses of free agents' locations and (i) shows all agents' true locations and the relative orientation measurements (one edge represents a pair of measurements). Assuming there is no measurement error (i.e., $\delta = 0$), the iterative results of applying Algorithm 1 with $\alpha_i \equiv 1.9$ are plotted in Fig. 6, and as can be seen the algorithm converges to the ground truth in about 50 iterations.

In Fig. 7, we compare the convergence rates of Algorithm 1 with three different settings of α_i: $\alpha_i \equiv 0.5$, $\alpha_i \equiv 1$, $\alpha_i \equiv 1.9$ and Pro-Con algorithm in [1], [2]. For a fair comparison, the Pro-Con algorithm adopts equal weights as

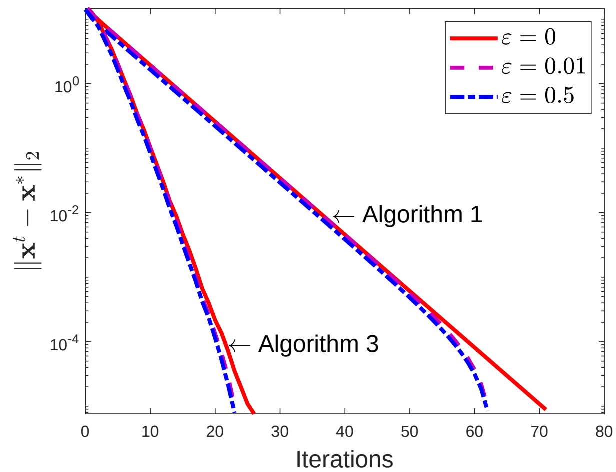
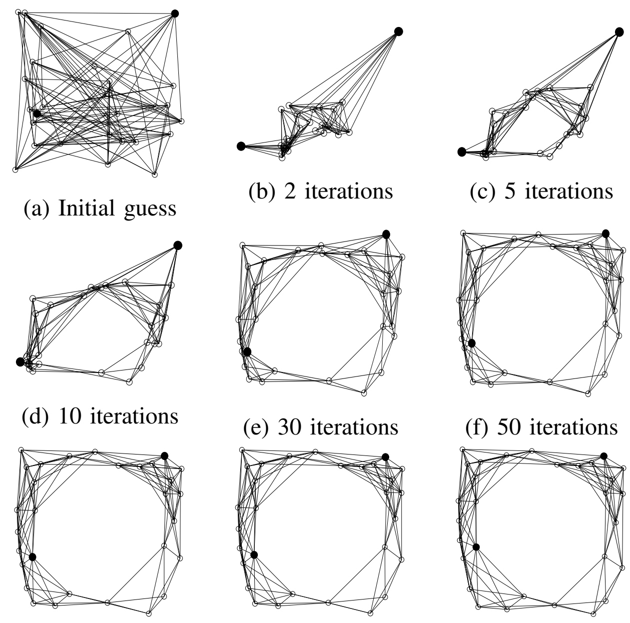


Fig. 5: Results of Example 4: plots of $\|\mathbf{x}^t - \mathbf{x}^*\|$ vs iterations t when applying Algorithm 1 and Algorithm 3.



(a) Initial guess (b) 2 iterations (c) 5 iterations

(d) 10 iterations (e) 30 iterations (f) 50 iterations

(g) 80 iterations (h) 100 iterations (i) Ground truth

Fig. 6: Results of applying Algorithm 1 to the network localization problem with 2 anchors among 30 agents.

that in the consensus operation of Algorithm 1. At least for this example, regardless of α_i being used, Algorithm 1 converges much faster than the Pro-Con algorithm despite the fact that the later one demands each agent to store and exchange with neighbors a whole copy of the variable x, resulting in more information storage and communication for all agents. An intuitive explanation of the performance difference is as follows. In the Pro-Con algorithm, agent i maintains a copy of x. However, in the copy, only the part involved in the local constraint $\mathcal{F}_i$ will be updated/improved via the local projection step, while the other part remains unchanged but still gets delivered to neighboring agents for their consensus step, potentially hindering the algorithm.

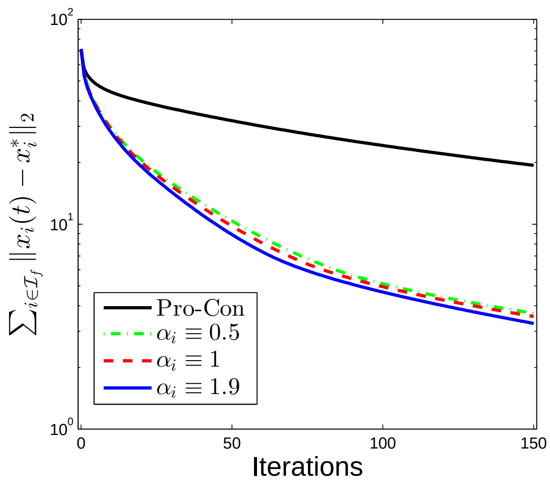


Fig. 7: Comparison of the convergence rates of Algorithm 1 with different α_i for the network localization problem. The value $\sum_{i\in\mathcal{I}_f}\|x_i^t - x_i^*\|_2$ versus iteration number t is plotted.

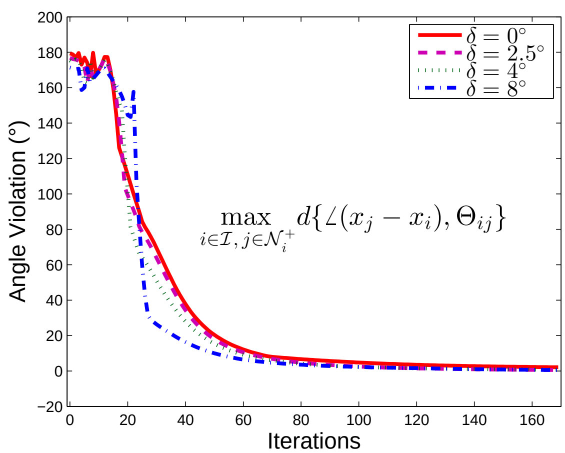


Fig. 8: Comparison of the convergence rates of Algorithm 1 for the network localization problem with measurement errors.

Finally, suppose the relative orientation measurements are inaccurate ($\delta \neq 0$). Setting $\delta = 0°$, $2.5°$, $4°$, $8°$, respectively, the results of applying Algorithm 1 are shown in Fig. 8, which plots the sum of constraint violations $d\{\angle(x_j^t - x_i^t), \Theta_{ij}\}$ vs. iteration number t. As can be seen, with a larger error range δ and hence a larger feasible set, the algorithm converges faster to a feasible solution, which is not the ground truth in general.

VII. Conclusion And Future Works

In this paper, we propose a distributed approach for solving the convex feasibility problems with coupling constraints that can significantly reduce the storage and communication requirements for each agent. Four associated distributed algorithms are developed, whose convergence properties have been established and also demonstrated through numerical examples.

As future directions, we will study the possibility of extending the developed algorithms to the more general cases of directed communication, non-doubly-stochastic weighting matrices, and distributed optimization problem with locally coupled objective functions and constraints. Characterizing the convergence rates of the algorithms will also be explored.

Appendix
Useful Notions in Graph Theory

For a stochastic matrix $A \in \mathbb{R}^{m\times m}$, its *associated graph* $\mathcal{G}$ is defined to have the vertex set $\mathcal{I}_m$ and a directed edge (j, i) from vertices j to i whenever the entry in i-th row and j-th column is positive, i.e., $[A]_{ij} > 0$. A finite sequence of graphs $\mathcal{G}_1, \dots, \mathcal{G}_T$ with the same vertex set $\mathcal{I}_m$ is said to be *jointly strongly connected* if their union $\mathcal{G}_1 \cup \dots \cup \mathcal{G}_T$ is strongly connected. Here the union $\mathcal{G}_1 \cup \dots \cup \mathcal{G}_T$ is the directed graph with the same vertex set $\mathcal{I}_m$ and an edge set that is the union of individual graph's edge set. An infinite sequence of graphs $\{\mathcal{G}_t\}$ is *repeatedly jointly strongly connected* if there exists a length $T > 0$ such that every T successive graphs from $\{\mathcal{G}_t\}$ is jointly strongly connected.

References

[1] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[2] A. Nedić and J. Liu, "On convergence rate of weighted-averaging dynamics for consensus problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 766–781, 2017.

[3] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[4] M. Jiang and G. Wang, "Convergence studies on iterative algorithms for image reconstruction," *IEEE Transactions on Medical Imaging*, vol. 22, no. 5, pp. 569–579, 2003.

[5] A. Cichocki and S.-i. Amari, *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002, vol. 1.

[6] A. Bemporad, D. Bernardini, and P. Patrinos, "A convex feasibility approach to anytime model predictive control," *arXiv preprint arXiv:1502.07974*, 2015.

[7] J. Ash and L. Potter, "Sensor network localization via received signal strength measurements with directional antennas," in *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*, 2004, pp. 1861–1870.

[8] L. Doherty, L. El Ghaoui *et al.*, "Convex position estimation in wireless sensor networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2001, pp. 1655–1663.

[9] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, 2006.

[10] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1856–1871, 2009.

[11] G. Zhu and J. Hu, "Distributed continuous-time protocol for network localization using angle-of-arrival information," in *Proc. American Control Conf.*, 2013, pp. 1006–1011.

[12] ——, "A distributed continuous-time algorithm for network localization using angle-of-arrival information," *Automatica*, vol. 50, no. 1, pp. 53–63, 2014.

[13] M. Cao, A. S. Morse, and B. D. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 575–600, 2008.

[14] C. Sueur, J.-L. Deneubourg, and O. Petit, "From social network (centralized vs. decentralized) to collective decision-making (unshared vs. shared consensus)," *PLoS one*, vol. 7, no. 2, p. e32566, 2012.

[15] S. Yang, S. Tan, and J.-X. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4416–4426, 2013.

[16] H. H. Bauschke and J. M. Borwein, "On projection algorithms for solving convex feasibility problems," *SIAM review*, vol. 38, no. 3, pp. 367–426, 1996.

[17] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, and G. T. Herman, "On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints," *Computational Optimization and Applications*, vol. 51, no. 3, pp. 1065–1088, 2012.

[18] A. J. Zaslavski, "Convex feasibility problems," in *Approximate Solutions of Common Fixed-Point Problems*. Springer, 2016, pp. 341–384.

[19] S. Mou and A. Morse, "A fixed-neighbor, distributed algorithm for solving a linear algebraic equation," in *Control Conference (ECC), 2013 European*. IEEE, 2013, pp. 2269–2273.

[20] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2863–2878, 2015.

[21] G. Shi, B. D. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659–2674, 2017.

[22] H.-T. Cao, T. E. Gibson, S. Mou, and Y.-Y. Liu, "Impacts of network topology on the performance of a distributed algorithm solving linear equations," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 1733–1738.

[23] B. D. Anderson, S. Mou, A. S. Morse, and U. Helmke, "Decentralized gradient algorithm for solution of a linear equation," *Numerical Algebra, Control & Optimization*, vol. 6, no. 3, pp. 319–328, 2016.

[24] X. Wang, S. Mou, and D. Sun, "Improvement of a distributed algorithm for solving linear equations," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3113–3117, 2017.

[25] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed algorithm for least squares solutions," *IEEE Transactions on Automatic Control*, 2019.

[26] D. Fullmer, L. Wang, and A. S. Morse, "A distributed algorithm for computing a common fixed point of a family of paracontractions," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 552–557, 2016.

[27] D. Fullmer, J. Liu, and A. S. Morse, "An asynchronous distributed algorithm for computing a common fixed point of a family of paracontractions," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 2620–2625.

[28] J. Liu, D. Fullmer, A. Nedić, T. Başar, and A. S. Morse, "A distributed algorithm for computing a common fixed point of a family of strongly quasi-nonexpansive maps," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 686–690.

[29] S. Khoshfetrat Pakazad, M. S. Andersen, and A. Hansson, "Distributed solutions for loosely coupled feasibility problems using proximal splitting methods," *Optimization Methods and Software*, vol. 30, no. 1, pp. 128–161, 2015.

[30] K. Lu, G. Jing, and L. Wang, "Distributed algorithms for solving convex inequalities," *IEEE Transactions on Automatic Control*, 2017.

[31] Y. Lou, G. Shi, K. H. Johansson, and Y. Hong, "Approximate projected consensus for convex intersection computation: Convergence analysis and critical error angle," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1722–1736, 2014.

[32] R. Aharoni and Y. Censor, "Block-iterative projection methods for parallel computation of solutions to convex feasibility problems," *Linear Algebra Appl.*, vol. 120, pp. 165–175, 1989.

[33] Y. Censor, D. Gordon, and R. Gordon, "Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems," *Parallel computing*, vol. 27, no. 6, pp. 777–808, 2001.

[34] S.-S. Chang, J. Kim, and X. Wang, "Modified block iterative algorithm for solving convex feasibility problems in banach spaces," *Journal of Inequalities and Applications*, vol. 2010, no. 1, p. 869684, 2010.

[35] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[37] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[38] X. Gao, J. Liu, and T. Başar, "Stochastic communication-efficient distributed algorithms for solving linear algebraic equations," in *Control Applications (CCA), 2016 IEEE Conference on*. IEEE, 2016, pp. 380–385.

[39] S. Mou, A. S. Morse, Z. Lin, L. Wang, and D. Fullmer, "A distributed algorithm for efficiently solving linear equations," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 6791–6796.

[40] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005, pp. 63–70.

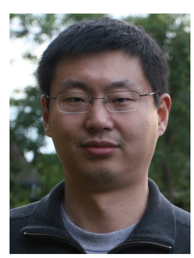
[41] L. Elsner, I. Koltracht, and M. Neumann, "Convergence of sequential and asynchronous nonlinear paracontractions," *Numerische Mathematik*, vol. 62, no. 1, pp. 305–319, 1992.

[42] A. Cegielski, *Iterative methods for fixed point problems in Hilbert spaces*. Springer, 2012, vol. 2057.

Yingying Xiao received the B.S. and M.S. degrees in aerospace engineering from Harbin Institute of Technology, Harbin, China, in 2012 and 2014, respectively. She is currently working toward a Ph.D. degree in School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. Her research interests include distributed computation, multi-agent systems, control application in buildings.

Ji Liu received the B.S. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2013.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Stony Brook University, Stony Brook, NY, USA. Prior to joining Stony Brook University, he was a Postdoctoral Research Associate at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA, and the school of Electrical, Computer and Energy Engineering at Arizona State University, Tempe, AZ, USA. His current research interests include distributed control and computation, multi-agent systems, social networks, epidemic networks, and power networks.



Jianghai Hu (S'99-M'04) received the B.E. degree in automatic control from Xi'an Jiaotong University, P.R. China, in 1994, and the M.A. degree in Mathematics and the Ph.D. degree in Electrical Engineering from the University of California, Berkeley, in 2002 and 2003, respectively. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Purdue University. His research interests include hybrid systems, multi-agent systems, control of systems with uncertainty, and control applications. He is an Associate Editor of *SIAM Journal on Control and Optimization* and *Nonlinear Analysis: Hybrid Systems*. He served as the IEEE Control System Society Electronic Publications Chair and editor of the monthly E-letters from 2016 to 2018.