Online decentralized decision making with inequality constraints: an ADMM approach

Yuxiao Chen[1], Mario Santillo[2], Mrdjan Jankovic[2], and Aaron D. Ames[1]

Abstract—We discuss an online decentralized decision making problem where the agents are coupled with affine inequality constraints. Alternating Direction Method of Multipliers (ADMM) is used as the computation engine and we discuss the convergence of the algorithm in an online setting. To be specific, when decisions have to be made sequentially with a fixed time step, there might not be enough time for the ADMM to converge before the scenario changes and the decision needs to be updated. In this case, a suboptimal solution is employed and we analyze the optimality gap given the convergence condition. Moreover, in many cases, the decision making problem changes gradually over time. We propose a warm-start scheme to accelerate the convergence of ADMM and analyze the benefit of the warm-start. The proposed method is demonstrated in a decentralized multiagent control barrier function problem with simulation.

Index Terms—Decentralized control; ADMM; Control barrier Functions

I. INTRODUCTION

Multiagent systems and networked systems make up a large portion of the autonomous applications, such as multi-robot systems, autonomous vehicles, and power networks. In many cases, due to the large number of agents and the distributed nature, centralized decision making is not implementable and people resort to decentralized algorithms. These decentralized decision making problems are usually formulated as distributed optimization problems, which have been studied for decades [1], [2]. One typical setup is to optimize over a summation of functions depending on different subsets of the decision variables via local optimization and communication between agents. The algorithm structure and communication protocol largely depend on the coupling graph topology, including the master-worker setting [3], fully connected setting, and fully distributed setting. Time varying topologies and time delays have also been considered in the literature [2], [4].

Alternating Direction Method of Multipliers (ADMM) is one class of optimization algorithms that receive increasing popularity as a simple yet effective framework for distributed optimization [5]. Comparing to the gradient based methods such as [2], it is based on dual ascent, which solves the primal problem locally, and uses the dual variable to coordinate the local optimizations at each node. A common way of applying ADMM to distributed optimization is to formulate it as a consensus problem, where each agent is associated with a local objective function, and are coupled with other agents in the system by a consensus constraint. The idea is to keep local

[1] Yuxiao Chen and Aaron D. Ames are with Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA `chenyx,ames@caltech.edu`
[2] Mario Santillo and Mrdjan Jankovic are with Ford Research and Advanced Engineering, Dearborn, MI, USA `msantil3,mjankov1@ford.com`

copies of variables shared among multiple objective functions and enforcing consistency of the local copies, which fits right into the ADMM formulation. There exist a plethora of different variations of ADMM [6] with different communication topologies and convergence guarantees [7], [8].

In this paper we consider the problem of decentralized decision making with multiple agents coupled by linear inequality constraints. The goal is to search for the solution that minimizes the violation. Examples of such problems include motion planning for multiple agents [9], [10], [11], decentralized model predictive control [12], and decentralized coordination for power grids [13], [14]. Moreover, we consider the case where the decentralized decision making problem is solved repeatedly with gradually changing data, i.e., the inequality constraints and local objective functions changes gradually between successive time steps. In online decision-making problems, the decision need to be made within a limited amount of time, which can be several seconds, or even milliseconds, depending on the application. The motivating example is the obstacle avoidance problem for autonomous vehicles using control barrier functions, where the actions need to be determined in the order of 100 milliseconds. The main contributions of this paper are (1) we propose a consensus-based framework for online decentralized decision making with coupling inequality constraints (2) we apply the Prox-JADMM algorithm with warm-start for online decentralized decision making with changing topology and analyzed the optimality gap for unconverged solutions and the continuity of the optimal solution under changing problem data.

Paper structure. Section II describes the problem setup, Section III reviews the ADMM setup and the convergence result, and Section IV presents the main result, the warm-start Prox-JADMM algorithm for online decentralized decision making. We show the application of the proposed algorithm on a autonomous vehicle obstacle avoidance example in Section V and finally we conclude in Section VI.

II. PRELIMINARIES AND PROBLEM SETUP

We begin with some definitions. A function $f : \mathcal{X} \to \mathbb{R}$ is convex if $\forall x_1, x_2 \in \mathcal{X}, \lambda \in [0, 1], f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$. Given a convex function f, if f is differentiable at x, then $\nabla f(x)$ denote the gradient of f at x; $\partial f(x)$ denotes the subgradient of f, which is defined as

$$\partial f(x) := \{c \in \mathbb{R}^n \mid \forall x_1 \in \mathcal{X}, f(x_1) - f(x) \geq c^{\mathsf{T}}(x_1 - x)\}.$$

When f is differentiable at x, $\partial f = \{\nabla f(x)\}$. A differentiable f is strongly convex with parameter σ if $\forall x, y \in \mathcal{X}, f(y) - f(x) \geq \nabla f(x)^{\mathsf{T}}(y - x) + \frac{\sigma}{2}||x - y||^2$. $|\cdot|$ denotes the element-wise absolute value of a vector or a matrix, $||\cdot||$ denotes the 2-norm of a vector, $||\cdot||_F$ denotes the Frobenius norm.

We consider a fully decentralized setting with N agents, where N may vary over time and the algorithm does not depend on N. Let $x_i \in \mathbb{R}^{n_i}$ be the decision variable for the ith agent, each agent is associated with the local objective function $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$. A coupling linear inequality is a 3-tuple (s, A_s, b_s) where $s \subseteq \{1, 2, ..., N\}$ is a subset of agents, A_s and b_s defines the following coupling constraint:

$$\sum\nolimits_{i \in s} A_s^i x_i \leq b_s,$$

where A_s^i is the block of A_s corresponding to x_i. Two agents x_i and x_j are neighbors of each other if there exists a coupling constraint (s, A_s, b_s) where s contains both i and j, and $\mathcal{N}_i$ denotes the neighbor set of agent i, with cardinality N_i. Let $\mathcal{C}$ denote the set of all coupling constraints, the optimization problem we consider is then:

$$\min_{x_1, ..., x_N} \sum\nolimits_i f_i(x_i)$$
$$\text{s.t. } x_i \in \mathcal{X}_i, \ \forall (s, A_s, b_s) \in \mathcal{C}, \ \sum\nolimits_{i \in s} A_s^i x_i \leq b_s, \quad (1)$$

where $\mathcal{X}_i$ is the domain of x_i, assumed to be convex with a nonempty interior. The compact form of (1) is then

$$\min_{x_1, ..., x_N} \sum\nolimits_i f_i(x_i)$$
$$\text{s.t. } x_i \in \mathcal{X}_i, \ Ax \leq b, \quad (2)$$

where $x = \begin{bmatrix} x_1^\intercal & \cdots & x_N^\intercal \end{bmatrix}^\intercal$ is the vector consisting of all x_i, A and b matrices are obtained by stacking the A_s and b_s matrices on the proper dimensions. In practice, (1) can be infeasible and we solve the following relaxed problem instead:

$$\min_{x_1 \in \mathcal{X}_1, ..., x_N \in \mathcal{X}_N} \sum_i f_i(x_i) + \beta \sum_{(s, A_s, b_s) \in \mathcal{C}} \mathbb{1}^\intercal \max\{\mathbf{0}, \sum_{i \in s} A_s^i x_i - b_s\} \quad (3)$$

where the maximum is taken entry-wise, $\beta > 0$ is the penalty on constraint violation, and $\mathbb{1}$ is a vector of all ones with the proper dimension. The penalty term is a piecewise affine convex function of the $\{x_i\}$.

***Remark* 1.** In [15], the inequality constraint is enforced with a logarithmic penalty. However, since we consider the cases where (2) may be infeasible, a linear penalty is chosen instead. Assuming $\forall x, c_1 \leq \sum_i f_i(x_i) \leq c_2$, if (2) is feasible, then the solution to (3) satisfies $\mathbb{1}^\intercal \max\{\mathbf{0}, \sum_{i \in s} A_s^i x_i - b_s\} \leq \frac{c_2 - c_1}{\beta}$. This means constraint violation due to relaxation can be made arbitrarily small by increasing β.

To solve (3) decentrally, the local copy idea is adopted. To be specific, each agent keeps local copies of the decision variables of its neighbors, and additional consensus constraint is added so that these local copies agree with the actual variable. Let x_j^i denote the local copy of x_j kept by agent i, $\mathbf{x}_i := [x_i^\intercal, {x_{j_1}^i}^\intercal, ..., {x_{j_{N_i}}^i}^\intercal]^\intercal \in \mathbb{R}^{\mathbf{n}_i}$ be the vector consisting of x_i and all the local copies of agent i's neighbors, and let $\mathbf{A}^i \mathbf{x}_i \leq \mathbf{b}^i$ be the compact form of the constraint

$$\forall (s, A_s, b_s) \in \mathcal{C}, i \in s, A_s^i x_i + \sum_{j \in s, j \neq i} A_s^j x_j^i \leq b_s,$$

$\mathbf{A}^i \in \mathbb{R}^{m_i \times \mathbf{n}_i}$ and $\mathbf{b}^i \in \mathbb{R}^{\mathbf{n}_i}$. Then define the augmented local objective function as

$$F_i(\mathbf{x}_i) = f_i(x_i) + \beta(w^i)^\intercal \max\{\mathbf{0}, \mathbf{A}^i \mathbf{x}_i - \mathbf{b}^i\}, \quad (4)$$

where $w^i \in \mathbb{R}^{m_i}$ is a vector with $w_j^i = 1/|s_j|$, s_j being the subset of agents corresponding to the jth row of $\mathbf{A}^i$, and $|s_j|$ being its cardinality. Since the constraint is shared by $|s_j|$ agents, each local objective function gets $1/|s_j|$ of the total penalty. The decentralized optimization problem is then:

$$\min_{\mathbf{x}_1, ..., \mathbf{x}_N} \sum\nolimits_i F_i(\mathbf{x}_i)$$
$$\text{s.t. } \forall i, x_i \in \mathcal{X}_i, \ \forall j \in \mathcal{N}_i, x_j^i = x_j, \quad (5)$$

which shall be solved with ADMM.

III. REVIEW OF ADMM

We briefly review the ADMM framework.

Standard ADMM. The standard ADMM algorithm considers the following problem:

$$\min_{x, z} f(x) + g(z)$$
$$\text{s.t. } Ax + Bz - c = 0, \quad (6)$$

where $x \in \mathbb{R}^n, z \in \mathbb{R}^m$ are the decision variables, $f : \mathbb{R}^n \to \mathbb{R}, g : \mathbb{R}^n \to \mathbb{R}$ are convex objective functions, $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}$ and $c \in \mathbb{R}^p$ defines the linear equality constraint. The ADMM algorithm follows a dual ascent procedure with the following augmented Lagrangian:

$$\mathcal{L}_\rho(x, z, y) = f(x) + g(z) + y^\intercal (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|^2,$$

where y is the Lagrange multiplier of the equality constraint, $\rho > 0$ introduces the quadratic terms that facilitates the convergence of the algorithm. The ADMM updates the follows

$$x^{k+1} := \arg\min_x \mathcal{L}_\rho(x, z^k, y^k)$$
$$z^{k+1} := \arg\min_z \mathcal{L}_\rho(x^{k+1}, z, y^k) \quad (7)$$
$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c).$$

More detail about the ADMM algorithm, including the convergence proof, can be found in [5].

Multiblock ADMM and convergence. The standard ADMM handles the case with $N = 2$ coupled variables, the cases where $N \geq 3$ is often referred to as multiblock ADMM, which solves the following problem:

$$\min_{\mathbf{x}_i} \sum_{i=1}^{N} F_i(\mathbf{x}_i)$$
$$\text{s.t. } \sum_{i=1}^{N} A_i \mathbf{x}_i - c = 0. \quad (8)$$

There are several different setups for multiblock ADMM. A direct extension of the standard ADMM algorithm to the multiblock case is called a Gauss-Seidel ADMM. The Lagrangian for multiblock ADMM can be directly extended from the standard case:

$$\mathcal{L}_\rho(\mathbf{x}_1, ... \mathbf{x}_N, y) = \sum_{i=1}^{N} f_i(\mathbf{x}_i) + y^\intercal (\sum_{i=1}^{N} A_i \mathbf{x}_i - c) + \frac{\rho}{2} \left\| \sum_{i=1}^{N} A_i \mathbf{x}_i - c \right\|^2.$$

The Gauss-Seidel iteration is then

$$\mathbf{x}_1^{k+1} := \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^k, ..., \mathbf{x}_N^k, y^k)$$
$$\mathbf{x}_2^{k+1} := \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{x}_3^k, ..., \mathbf{x}_N^k, y^k)$$
$$\cdots$$
$$\mathbf{x}_N^{k+1} := \arg\min_{\mathbf{x}_N} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, ... \mathbf{x}_{N-1}^{k+1}, \mathbf{x}_N, y^k) \quad (9)$$
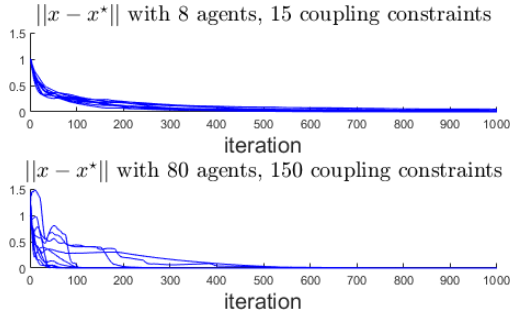$$y^{k+1} := y^k + \rho(\sum_{i=1}^{N} A_i \mathbf{x}_i^{k+1} - c).$$

Fig. 1: Convergence rate with 8 and 80 agents in 10 trials

Note the update of $\mathbf{x}_i$ leverages the updated value of $\mathbf{x}_1, ... \mathbf{x}_{i-1}$, which helps the convergence of the algorithm due to the updated information. See [16] for convergence analysis.

However, as is clear from the algorithm, Gauss-Seidel ADMM cannot be parallelized and is not suitable for a fully decentralized setting. Jacobian ADMM refers to the ADMM algorithms that simultaneously update all x_i with the value from the last iteration, which can be parallelized. However, due to the lack of updated information, Jacobian ADMM is less likely to converge comparing to Gauss-Seidel ADMM. To facilitate convergence, the common strategy is to damp the changing of the variables between iterations, such as adding an underrelaxation [17] and adding a proximal term to the Lagrangian [18]. We shall adopt the latter, named Prox-JADMM, as our ADMM strategy. Prox-JADMM adds a proximal term to primal update, and the ADMM update follows the following procedure:

$$
\begin{aligned}
\mathbf{x}_1^{k+1} &:= \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^k, ..., \mathbf{x}_N^k, y^k) + \left\| \mathbf{x}_1 - \mathbf{x}_1^k \right\|_{P_1}^2 \\
\mathbf{x}_2^{k+1} &:= \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^k, \mathbf{x}_2, \mathbf{x}_3^k, ..., \mathbf{x}_N^k, y^k) + \left\| \mathbf{x}_2 - \mathbf{x}_2^k \right\|_{P_2}^2 \\
&\quad\quad ... \\
\mathbf{x}_N^{k+1} &:= \arg\min_{\mathbf{x}_N} \mathcal{L}_\rho(\mathbf{x}_1^k, ... \mathbf{x}_{N-1}^k, \mathbf{x}_N, y^k) + \left\| \mathbf{x}_N - \mathbf{x}_N^k \right\|_{P_N}^2 \\
y^{k+1} &:= y^k + \gamma\rho(\sum_{i=1}^N A_i \mathbf{x}_i^{k+1} - c),
\end{aligned}
$$
(10)

where $\gamma > 0$, and $\|\cdot\|_{P_i}$ is the 2-norm induced by $P_i \succeq 0$. Note that the dual update for each coupling constraints only depends on the agents involved, thus can be performed decentrally.

It is shown in [18] that when ρ, γ and $\{P_i\}$ satisfies

$$
\begin{cases}
P_i \succ \rho(\frac{1}{\varepsilon_i} - 1) A_i^\mathsf{T} A_i, & i = 1, 2, ..., N \\
\sum_{i=1}^N \varepsilon_i \le 2 - \gamma,
\end{cases}
$$
(11)

for some $\varepsilon_i > 0$, the Prox-JADMM converges with rate $o(1/k)$. However, it is obvious that (11) becomes increasingly difficult to satisfy as N increases. In a fully decentralized problem such as autonomous driving, there may be millions of agents (considering all vehicles in the road system) and typically no agent knows the total number of agents. Fortunately, this convergence proof can be very loose and in practice, we found that the convergence of Prox-JADMM is much better than what the theory predicts, and the convergence rate does not vary much as the number of agents grows. Fig. 1 shows the convergence in 10 randomly generated trials with 8 and 80 agents, and the rate of convergence is similar.

Remark 2. The proposed framework allows for a fully distributed implementation with a synchronized dual update, which can be achieved with a synchronized clock and time stamps. This is due to the fact that the coupling equality constraints are local.

IV. Prox-JADMM for online decentralized decision-making

In this section, we discuss the Prox-JADMM for online decentralized decision-making.

A. Quality of unconverged solution

The optimization problem shown in (5) falls into the multi-block ADMM framework, and as discussed in Section III, in a fully decentralized setting, we use the Prox-JADMM introduced in [18]. Under an online setting, Prox-JADMM may not converge in time, and the unconverged solution is then taken as the solution to the decision making problem after a final correction step, which simply solves the primal update with the current y and without the proximal term:

$$
\forall i = 1, ..., N, \mathbf{x}_i = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_1^k, , ..., \mathbf{x}_{i-1}^k, \mathbf{x}_i, \mathbf{x}_{i+1}^k, ..., \mathbf{x}_N^k, y^k) \quad (12)
$$

where k is the last iteration.

In the setup presented in Section II, each $\mathbf{x}_i$ contains x_i and the local copies x_j^i of its neighbors, x_i is then taken as the local decision. Let $x := [x_1^\mathsf{T}, ... x_N^\mathsf{T}]^\mathsf{T}$ and let y_{ij} be the Lagrange multiplier associated with the equality constraint $x_j^i = x_j$, we have the following theorem that quantifies the optimality gap.

Theorem 1. *Given any unconverged solution $\{\mathbf{x}_i\}$ from Prox-JADMM after the correction in* (12)*, the objective of* (5) *$J(\mathbf{x})$ and the optimal solution $J^\star(\mathbf{x}^\star)$ satisfies*

$$
\begin{aligned}
J(\mathbf{x}) - J^\star(\mathbf{x}^\star) \le &\sum_{(s, A_s, b_s) \in \mathcal{C}} \frac{\beta}{|s|} \sum_{i \in s} \sum_{j \in s, j \ne i} |A_s^j| \cdot |x_j^i - x_j| \\
&- \sum_i \sum_{j \in \mathcal{N}_i} \left((y_{ij}^k)^\mathsf{T}(x_j^i - x_j) + \frac{\rho}{2}||x_j^i - x_j||^2 \right).
\end{aligned}
$$
(13)

Proof. The gap can be split into two parts, $J(\mathbf{x}) - \sum_i F_i(\mathbf{x}_i)$ and $\sum_i F_i(\mathbf{x}_i) - J(\mathbf{x}^\star)$. Let $\bar{\mathbf{x}}_i := [x_i^\mathsf{T}, x_{j_1}^\mathsf{T}, ..., x_{j_{N_i}}^\mathsf{T}]^\mathsf{T}$ be the vector of the true decision variables (as opposed to the local copies) of agent i and its neighbors, by (4), we have

$$
J(\mathbf{x}) = \sum_i F_i(\bar{\mathbf{x}}_i),
$$

$$
\sum_i F_i(\bar{\mathbf{x}}_i) - \sum_i F_i(\mathbf{x}_i) \le \sum_{(s, A_s, b_s) \in \mathcal{C}} \frac{\beta}{|s|} \sum_{i \in s} \sum_{j \in s, j \ne i} |A_s^j| \cdot |x_j^i - x_j|.
$$

This gives rise to the first line of (13). Since $\mathbf{x}_i$s are the optimal solutions to the correction step in (12), we have $\mathcal{L}(\mathbf{x}_1, ..., \mathbf{x}_N, y^k) \le \mathcal{L}(\mathbf{x}_1^\star, ..., \mathbf{x}_N^\star, y^k) = J(\mathbf{x}^\star)$, implying

$$
\sum_i F_i(\mathbf{x}_i) - J(x^\star) \le -\sum_i \sum_{j \in \mathcal{N}_i} \left((y_{ij}^k)^\mathsf{T}(x_j^i - x_j) + \frac{\rho}{2}||x_j^i - x_j||^2 \right).
$$

Combining the two parts proves the bound. $\square$

Theorem 1 shows that the mismatch $x_j^i - x_j$ is an important indicator of the convergence of the algorithm. When the mismatch between local copies and the true values are small, the optimality gap is small.

B. Online decision-making with Prox-JADMM

Online decision-making is very common in engineering systems. Our motivating example is a control barrier function quadratic programming (QP), which is determined by the state of the agents and needs to be solved for every time step. Similar problems include the optimal power flow (OPF) problem for power systems and decentralized MPC. To distinguish the sequential decision making from the iterations of ADMM between time steps, we let $f_i[t]$, $\mathcal{C}[t]$ denote the local objective functions and the constraint set at time t. The ADMM algorithm would take multiple iterations to obtain a solution within one time step, then solve an updated problem with $f_i[t+1]$ and $\mathcal{C}[t+1]$. Typically the decision making problem satisfies some continuity condition, i.e., the problem would gradually change over time. To be more precise, let

$$F(x) = \sum_i f_i(x_i) + \beta \mathbb{1}^\intercal \max\{\mathbf{0}, Ax - b\}, \quad (14)$$

be defined as the total objective on $\mathcal{X} := \mathcal{X}_1 \times ... \times \mathcal{X}_N$. We begin with the following assumptions.

Assumption 1. The change of the local objective function f_i and the constraint set $\mathcal{C}$ over successive time steps is bounded such that there exists $\kappa > 0$ that $F[t+1] - F[t]$ is Lipschitz continuous with constant κ within $\mathcal{X}$, i.e.,

$$\forall t = 0, 1, ..., \forall x, x' \in \mathcal{X},$$
$$|(F[t+1] - F[t])(x') - (F[t+1] - F[t])(x)| \le \kappa ||x - x'||.$$

Assumption 2. For all agents, the local objective function f_i is twice differentiable and strongly convex with parameter σ, and ∇f_i is Lipschitz with constant ν.

Proposition 1. Under Assumptions 1, 2, let $x^\star[t]$ be the optimal solution to (5) at time t, then $||x^\star[t+1] - x^\star[t]|| \le \sigma^{-1}\kappa$.

Proof. Proposition 1 is directly adopted from Proposition 4.32 from [19]. Since f_is are strongly convex with parameter σ, F is also strongly convex with parameter σ, thus satisfies the second order growth condition with parameter σ. Then the result follows by taking ε to 0. □

Proposition 1 gives continuity condition for the optimal solution of (5), but the dual variable is also critical to the convergence of ADMM. Although strong duality guarantees the uniqueness of $x^\star$, the dual variables that satisfy the Karush-Kuhn-Tucker (KKT) condition may not be unique, which is mainly due to the nondifferentiability of the max function. One solution is to use a smooth approximation of the max function, for example, with the *LogSumExp* function. The smooth approximation of F_i is defined as

$$\bar{F}_i(\mathbf{x}_i) = f_i(x_i) + \frac{\beta}{c}(w^i)^\intercal \log(e^{c(\mathbf{A}^i \mathbf{x}_i - \mathbf{b}^i)} + \mathbb{1}), \quad (15)$$

where $\exp$ and $\log$ are taken entrywise and $c > 0$ is a parameter that tunes the smoothness of the approximation. The larger c is, the closer the approximation is to the original function. The total objective after smoothing is then

$$\bar{F}(x) = \sum_{i=1}^N f_i(x_i) + \frac{\beta}{c}\mathbb{1}^\intercal \log(e^{c(Ax-b)} + \mathbb{1}). \quad (16)$$
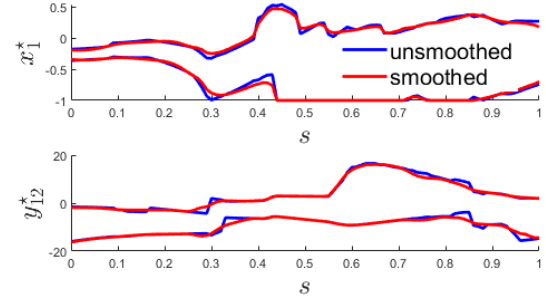

Fig. 2: Optimal primal and dual values with and without *LogSumExp* smoothing

Proposition 2. *Under Assumptions 2 and assume $\bar{F}[t+1] - \bar{F}[t]$ is Lipschitz with constant κ for all t, let $x^\star[t], y^\star[t]$ be the solution of (5) with F_i replaced by $\bar{F}_i$ defined in (15) at time t, let $|s|^{\max}$ be the maximum number of agents coupled by a single constraint in $\mathcal{C}$, then $||x^\star[t+1] - x^\star[t]|| \le \sigma^{-1}\kappa$, and $||y^\star[t+1] - y^\star[t]|| < ||A||_F^2 \frac{|s|^{\max}-1}{|s|^{\max}} \beta c \sigma^{-1}\kappa$.*

Proof. Following a similar argument made in the proof of Proposition 1, $||x^\star[t+1] - x^\star[t]|| \le \sigma^{-1}\kappa$. The dual variable corresponds to the equality constraint such that all the local copies are equal to the actual variable. Let y_{ij} be the dual variable of the constraint $x_j^i = x_j$, then $y_{ij}^\star[t]$ satisfies $y_{ij}^\star[t] = -\frac{\partial \bar{F}_i(\mathbf{x}_i^\star[t])}{\partial x_j^i}$. The Hessian of the partial derivative is $\frac{\partial^2 \bar{F}_i}{\partial (x_j^i)^2} = \sum_{n=1}^{m_i} \frac{e^{\mathbf{A}_n^i \mathbf{x}_i - \mathbf{b}_n^i} \beta c}{(e^{\mathbf{A}_n^i \mathbf{x}_i - \mathbf{b}_n^i} + 1)^2 |s_n|}(\mathbf{A}_{n,j}^i)^\intercal \mathbf{A}_{n,j}^i$, where $\mathbf{A}_n^i$ is the nth row of $\mathbf{A}^i$, and $\mathbf{A}_{n,j}^i$ is the block corresponding to x_j^i. This implies that $\frac{\partial \bar{F}_i}{\partial x_j^i}$ has a Lipschitz constant less than $\sum_{n=1}^{m_i} \frac{\beta c}{|s_n|}||\mathbf{A}_{n,j}^i||_F^2$, i.e., $||y_{ij}^\star[t-1] - y_{ij}^\star[t]|| \le \sum_{n=1}^{m_i} \frac{\beta c}{|s_n|}||\mathbf{A}_{n,j}^i||_F^2 ||x_j^\star[t+1] - x_j^\star[t]||$. Sum up for all i, j,

$$||y^\star[t+1] - y^\star[t]|| < \sum_{(s,A_s,b_s)\in\mathcal{C}} \frac{(|s|-1)\beta c}{|s|}\sum_{i\in s}||A_s^i||_F^2 ||x_i^\star[t+1] - x_i^\star[t]||.$$

Using Cauchy-Schwartz inequality, we get

$$||y^\star[t+1] - y^\star[t]|| < ||A||_F^2 \frac{|s|^{\max} - 1}{|s|^{\max}} \beta c ||x^\star[t+1] - x^\star[t]||$$
$$\le ||A||_F^2 \frac{|s|^{\max} - 1}{|s|^{\max}} \beta c \sigma^{-1}\kappa. \quad \square$$

In practice, we found that smoothing may be unnecessary if the dual variable in the original problem is smooth enough w.r.t. the problem data. Fig. 2 shows the result of a numeric experiment with 8 agents, each $x_i \in \mathbb{R}^2$, f_i quadratic, and the coupling constraints are pairwise. We let the coupling constraint $Ax \le b$ gradually change from the initial value to the final value via linear interpolation: $A = \lambda A^1 + (1 - \lambda)A^2$, $b = \lambda b^1 + (1-\lambda)b^2$, and let λ change from 0 to 1, where A^1 and A^2 share the same sparsity pattern. Fig. 2 shows the optimal value of x_1 and y_{12} as λ changes. The smoothing made both $x^\star$ and $y^\star$ smoother, yet the unsmoothed curve is already continuous with a reasonably small Lipschitz constant.

Remark 3. The disadvantage of smoothing in this case is that it makes the local optimization from a quadratic programming to an optimization with the *LogSumExp* function, which is still convex, but there lacks specialized solvers for it. Therefore, when y is smooth enough w.r.t. the problem data without smoothing, we choose to not use smoothing.

The sensitivity analysis above shows that when the problem data $\{f_i\}$ and $\mathcal{C}$ change gradually over time, the optimal value also changes continuously. This motivate us to warm start the ADMM iteration with the solution of both the primal and dual variables (though unconverged) from last the iteration. In [18], convergence was proved by showing the distance from the current solution to the optimal solution is monotonically decreasing. Although no quantitative analysis of the convergence rate was given, one would expect that if the change of the optimal solution between successive time steps is bounded, the convergence of ADMM can catch up and maintain a certain maximum distance to the optimal solution, which is demonstrated in Section V. When the topology of the communication network changes, the dual variable associated with the newly added coupling constraint is simply initialized to be zero.

The warm start online Prox-JADMM is summarized in Algorithm 1. M is the maximum number of iterations allowed between time steps (determined by the computation speed and communication speed). At the beginning of each time step, the primal and dual variables are warm started with the (possibly unconverged) values from the last time step.

V. Application to multiagent CBF

The motivating example is a multiagent control barrier function (CBF) problem that arises in the control of autonomous vehicles. Control barrier function [20] considers a control affine dynamic system described by a differential equation

$$\dot{x} = f(x) + g(x)u, x \in \mathbb{R}^n, u \in \mathcal{U} \subseteq \mathbb{R}^m, \quad (17)$$

where x is the state, subject to a safety constraint $x \notin \mathcal{X}_d$, u is the input.

Remark 4. To avoid confusion, x denotes system state, following the convention of control literature, the input u is the decision variable of the ADMM problem introduced later.

A CBF is then a function $h : \mathbb{R}^n \to \mathbb{R}$ that satisfies

$$\forall\, x \in \mathcal{X}_0, \qquad\qquad\qquad h(x) \geq 0,$$
$$\forall\, x \in \mathcal{X}_d, \qquad\qquad\qquad h(x) < 0 \quad (18)$$
$$\forall\, x \in \{x \mid h(x) \geq 0\}, \quad \exists\, u \in \mathcal{U} \text{ s.t. } \dot{h} + \alpha(h) \geq 0,$$

where $\mathcal{X}_0$ is the set of initial states, $\alpha(\cdot)$ is a class-$\mathcal{K}$ function, i.e., $\alpha(\cdot)$ is strictly increasing and satisfies $\alpha(0) = 0$. The last condition is called the CBF condition, which guarantees that any state within the safe set ($h(x) \geq 0$) remains safe, and

---

Algorithm 1 Warm start online Prox-JADMM

1: $t \leftarrow 0$, $k \leftarrow 0$
2: $\{\mathbf{x}_i^0\} \leftarrow \mathbf{0}$, $y^0 \leftarrow \mathbf{0}$
3: **while NOT** TERMINATE **do**
4:     Obtain $\{F_i\}[t]$, calculate $\mathcal{L}$
5:     **for** $k = 0 : M$ **do**
6:         Prox-JADMM update following (10)
7:     **end for**
8:     Correction step following (12)
9:     $t \leftarrow t + 1$, $k \leftarrow 0$, $\{\mathbf{x}_i^0\} \leftarrow \{\mathbf{x}_i^M\}$, $y^0 \leftarrow y^M$
10: **end while**

---

any state outside the safe set converges back to the safe set exponentially. Given a legacy controller $u^0 : \mathbb{R}^n \to \mathcal{U}$, the following CBF QP solves for the minimum intervention over $u^0(x)$ while enforcing the CBF condition:

$$u^\star = \underset{u \in \mathcal{U}}{\arg\min} \left\| u - u^0(x) \right\|^2$$
$$\text{s.t. } \nabla h \cdot (f(x) + g(x)u) + \alpha(h) \geq 0, \quad (19)$$

which is a QP subject to a linear inequality constraint. The feasibility of (19) when $h(x) \geq 0$ is guaranteed if the set $\{x \mid h(x) \geq 0\}$ is a control invariant set. This is sufficient to guarantee safety, since under (19), any nonnegative h will remain nonnegative. In particular, [10] presents a CBF based on backup policies, which is applicable to multiagent problems and the feasibility of the decentralized CBF QP is guaranteed when $h \geq 0$. To be specific, consider N agents with dynamics $x_i = f_i(x_i) + g_i(x_i)u_i$, where x_i and u_i are the state and input of agent i. The CBF for multiagent collision avoidance can be decomposed as follows:

$$h(x_1, ... x_N) = \min_{i=1,..,N} \left[h_i(x_i), \min_{j \in \mathcal{N}_i} h_{ij}(x_i, x_j) \right], \quad (20)$$

where h_i is the local CBF for agent i, h_{ij} is the CBF for obstacle avoidance between agent i and j, defined for all of agent i's neighbors $\mathcal{N}_i$. The multiagent CBF QP is then

$$\min_{u_1, .., u_N} \sum_i \left\| u_i - u_i^0 \right\|^2$$
$$\text{s.t. } \forall i, \nabla h_i \cdot f_i(x_i, u_i) + \alpha(h_i) \geq 0,$$
$$\forall j \in \mathcal{N}_i, \ \nabla h_{ij} \cdot (f_i(x_i) + g(x_i)u_i + f_j(x_j) + g(x_j)u_j) + \alpha(h_{ij}) \geq 0, \quad (21)$$

which falls into the form of (1). It is shown in [10] that when all the h_i's and h_{ij}'s are positive, (21) is always feasible, and a feasible solution can be obtained by a fully decentralized optimization. However, when $h < 0$, there is no guarantee of feasibility, and the decentralized algorithm without communication proposed in [10] may perform badly. [21] proposed a presumed cooperation algorithm that also uses local copies, but consensus cannot be achieved without communication. This motivates us to use ADMM to improve the performance via consensus. As mentioned in Section II, the goal is to minimize the violation of the CBF condition. The local optimization for agent i is then the following:

$$\min_{\mathbf{u}_i} \left\| u_i - u_i^0 \right\|^2 + \beta \max\{0, -\nabla h_i \cdot f_i(x_i, u_i) - \alpha(h_i)\}$$
$$+ \frac{\beta}{2} \sum_{j \in \mathcal{N}_i} \max\{\mathbf{0}, -\nabla h_{ij} \cdot (f_i(x_i) + g_i(x_i)u_i + f_j(x_j) + g_j(x_j)u_j^i) - \alpha(h_{ij})\}$$
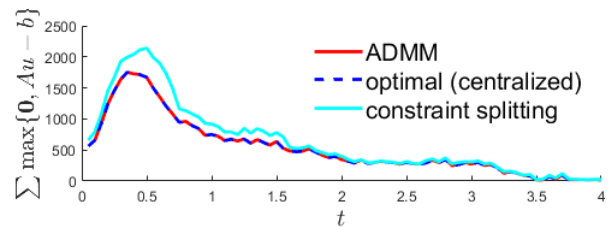$$\text{s.t. } u_i \in \mathcal{U}_i \quad (22)$$

Fig. 3: Total constraint violation of the constraint splitting method used in [10], the Prox-JADMM, and the optimal solution solved with centralized optimization
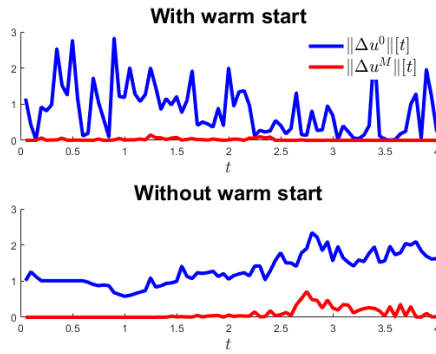
Fig. 4: Effect of warm start on convergence

where $\mathbf{u}_i$ contains u_i and u_j^i, the local copies of the input of all its neighbors. Note that the penalty for violating h_{ij}s are multiplied by $\frac{1}{2}$ since every h_{ij} is shared by two agents, thus the penalty is also shared by two agents.

The CBF condition is determined by the state of the agents, changing gradually over time. Therefore, it is straightforward to find the Lipschitz constant that characterizes the continuity of the online decision-making problem. The strong convexity comes from the quadratic local cost of each agent.

Remark 5. Comparing to the cooperative control methods, the Prox-JADMM CBF approach is fully decentralized, the agents do not know the total number of agents, they simply exchange messages with their neighbors and solve the local update in a synchronous and parallel fashion. Moreover, The agents do not need to share their local objective functions.

We consider the merging case in autonomous driving, in which the CBF may start negative since the vehicles on the two lanes can detect each other only when they are close to the merging point, and their current path may lead to collision. Dubin's car model is used for the vehicle dynamics with acceleration and yaw rate as input. The time step for updating the control input is 50ms, and the maximum iteration M for the Prox-JADMM is 30, chosen based on the sampling time for control update. The CBF QP only includes h_i and h_{ij} that are smaller than a threshold (large h indicates the constraint is satisfied with a big margin), which localizes the problem and bounds the size of the local optimization.

Fig. 3 shows the performance comparison of the Prox-JADMM algorithm with the benchmark in [10] and the optimal solution. The total constraint violation of Prox-JADMM is always smaller than the benchmark, and very close to the optimal solution, as Theorem 1 predicts. Fig. 4 shows the effect of warm start where $\|\Delta u\| \doteq \sum_{i,j} \|u_j^i - u_j\|$, the violation of the consensus constraint, which is shown to be an important indicator of the convergence of ADMM. The blue and red curves show the consensus violation at the beginning and at the end (after M iterations) of the ADMM iterations, respectively. The warm start help reducing Δu after the update of the optimization problem and ensures good convergence of the ADMM, whereas the convergence can be bad without the warm start.

A video of the highway sim can be found youtu.be/9qjHuDkpd2E and a python realization of the Prox-JADMM algorithm with linear inequality constraint can be found github.com/chenyx09/LCADMM.

VI. Conclusion

We present a Prox-JADMM algorithm for online decentralized decision making with coupling inequality constraints. We show that under gradual change of the problem data, the optimal solution changes continuously, and warm starting the ADMM helps accelerate convergence between time steps. The algorithm is applied to a decentralized control barrier function problem and the simulation shows that the decentralized algorithm achieves near optimal performance.

References

[1] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[3] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[4] S. Yang, Q. Liu, and J. Wang, "Distributed optimization based on a multiagent system in the presence of communication delays," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 5, pp. 717–728, 2016.

[5] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[6] Y. Chen, H. Peng, and J. W. Grizzle, "Validating noncooperative control designs through a lyapunov approach," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 527–539, 2018.

[7] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[8] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *International conference on machine learning*, 2014, pp. 1701–1709.

[9] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[10] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[11] R. M. Murray, "Recent research in cooperative control of multivehicle systems," 2007.

[12] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *2011 50th IEEE conference on decision and control and European control conference*. IEEE, 2011, pp. 7488–7493.

[13] Z. Li, W. Wu, B. Zeng, M. Shahidehpour, and B. Zhang, "Decentralized contingency-constrained tie-line scheduling for multi-area power grids," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 354–367, 2016.

[14] X. Fang, B.-M. Hodge, H. Jiang, and Y. Zhang, "Decentralized wind uncertainty management: Alternating direction method of multipliers based distributionally-robust chance constrained optimal power flow," *Applied Energy*, vol. 239, pp. 938–947, 2019.

[15] J. Wang and N. Elia, "A control perspective for centralized and distributed convex optimization," in *2011 50th IEEE conference on decision and control and European control conference*. IEEE, 2011, pp. 3800–3805.

[16] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the admm with multiblock variables," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1478–1497, 2015.

[17] B. He, L. Hou, and X. Yuan, "On full jacobian decomposition of the augmented lagrangian method for separable convex programming," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2274–2312, 2015.

[18] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with o (1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[19] J. F. Bonnans and A. Shapiro, *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.

[20] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[21] M. Santillo and M. Jankovic, "Collision free navigation with interacting, non-communicating obstacles," *arXiv preprint arXiv:2008.12092*, 2020.