

A continuous-time violation-free multi-agent optimization algorithm and its applications to safe distributed control

Xiao Tan, Changxin Liu, Karl H. Johansson, and Dimos V. Dimarogonas

Abstract—In this work, we propose a continuous-time distributed optimization algorithm with guaranteed zero coupling constraint violation and apply it to safe distributed control in the presence of multiple control barrier functions (CBF). The optimization problem is defined over a network that collectively minimizes a separable cost function with coupled linear constraints. An equivalent optimization problem with auxiliary decision variables and a decoupling structure is proposed. A sensitivity analysis demonstrates that the subgradient information can be computed using local information. This then leads to a subgradient algorithm for updating the auxiliary variables. A case with sparse coupling constraints is further considered, and it is shown to have better memory and communication efficiency. For the specific case of a CBF-induced time-varying quadratic program (QP), an update law is proposed that achieves finite-time convergence. Numerical results involving a static resource allocation problem and a safe coordination problem for a multi-agent system demonstrate the efficiency and effectiveness of our proposed algorithms.

I. INTRODUCTION

Distributed optimization has received increasing attention in the last few decades thanks to increasing computational power availability. Depending on how the problem is formulated, distributed optimization can be categorized into the *cost-coupled* and *constraint-coupled* optimization [1]. In the former case, the cost function to be minimized is a summation of local cost functions, each of which depends on a common decision variable \mathbf{x} subject to a common constraint, that is,

$$\min_{\mathbf{x} \in X} \sum_{i \in \{1, 2, \dots, N\}} f_i(\mathbf{x}), \quad (1)$$

where the local cost function f_i and global constraint set X are known to agent i . One such example is the training process of learning algorithms, where the decision variables are the parameters in the learning model and the local cost function is associated with data accessible only to local computational units. Much attention has been focused on this case in the distributed optimization community; see [2], [3] for overviews.

In the constraint-coupled case, each node has its own local decision variable, and the cost function is a summation of local cost functions that depend on the local decision variables only. The constraints in this case are coupled and involve all or parts of the local variables. The problem setup is then given by

$$\min_{(\mathbf{x}_1, \dots, \mathbf{x}_N) \in X} \sum_{i \in \{1, 2, \dots, N\}} f_i(\mathbf{x}_i), \quad (2)$$

This work was supported in part by Swedish Research Council, ERC CoG LEAFHOUND, EU CANOPIES Project, Knut and Alice Wallenberg Foundation, and an NSERC Postdoctoral Fellowship. The authors are with the School of EECS, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (Email: xiaotan, changxin, kallej, dimos@kth.se).

where the local cost function f_i and parts of the constraint set X are known to agent i . Conceptually, we can refer to the stack of all local variables as the global variable. One example of this setup is the economic dispatch problem in power systems, where the local decision variables are constrained by the shared resource limit, and the size of the global decision variable increases with the size of the power grid.

Albeit its wide outreach in real-time decision-making and profound practical implications, the constraint-coupled optimization problem (2) was less attended to compared to the vast literature on cost-coupled optimization problem (1). One viable approach to (2) involves solving its dual problem. Assuming that the optimization problem (2) has zero duality gap and the constraint functions are separable, then the dual problem of (2) has a separable objective function with a common decision variable (the dual variable), aligning the problem with the formulation of (1). Thanks to this observation, many distributed optimization algorithms for (1) can be readily applied to solve (2) by tackling its dual problem [4]–[7]. One major drawback of these duality-based approaches is that, primal convergence is not easily retrieved from dual solutions, known as the primal recovery problem [8]. The remedy usually involves a running average scheme, which leads to a slow convergence rate. Recent works in [9], [10] have proposed various methods to avoid the recovery procedure. Another issue is that, in order to solve for the local primal variable, each node has to solve for a consensus on the dual variable, causing efficiency concerns.

In order to avoid the primal recovery problem as well as enhance efficiency, recently, primal decomposition approach has been pursued as a general distributed solution to the constraint-coupled problem [1]. Instead of looking at the dual problem, primal decomposition approach splits the coupling but separable constraints into local ones using auxiliary variables. The algorithm usually comprises two parts: 1) solving/updating the local primal variables based on local problems, and 2) updating the auxiliary variables. This approach has its roots in large-scale optimization, where the global variable is too large for one computational node to process and a central coordinating node is required for updating the auxiliary variables. See, for example, [11] for more details. It is until recently that [1] proposes a distributed auxiliary update and provides an asymptotic convergence guarantee. [12] further extends this approach and provides a convergence rate analysis.

One issue that has been largely overlooked in the existing distributed optimization literature is primal feasibility throughout solution iterations. Most algorithms mentioned above or surveyed in [2], [3] at best provide an asymptotic convergence guarantee and asymptotic primal feasibility guarantee. This might not be desirable when applying optimization-based algo-

gorithms in real-time safe-critical scenarios, i.e., when transient behavior becomes a crucial safety concern and constraint violation could lead to catastrophic consequences. Conceptually, the problem is challenging since the feasible set is defined over the global decision variable and cannot be evaluated for satisfaction using only local information, which voids the usual practice of projecting the solution onto the feasible set [13].

There are only a few distributed violation-free algorithms for constraint-coupled optimization problems. Recently, [14] proposes a right-hand primal decomposition scheme, yet it only achieves an approximate optimal solution. [15] considers primal feasibility in dual decomposition methods, where the solution still only converges to a neighborhood of optimal solution. [16] proposes a violation-free algorithm with optimality guarantee, but only for a single constraint. A highly relevant work [17] adopts a similar primal decomposition scheme as in this paper, and uses a cascaded law based on projected saddle-point dynamics and safe gradient flow for updating the auxiliary and primal variables. However, only practical convergence can be established due to a regularization step. The violation-free Laplacian-based gradient algorithm in [18] instead focuses on the special case of economic dispatch problem. In summary, the existing violation-free results either give inexact solutions or are tailored to specific problems. One exception is our recent work [19] where the focus is on designing a discrete-time, violation-free, distributed and accelerated optimization algorithm. We note that technicality concerns differ significantly depending on whether the dynamics is continuous-time or discrete-time.

In this work, we propose a distributed, continuous-time optimization algorithm for problem (2) with coupling linear constraints. Our contributions are summarized as follows.

- 1) The coupling constraints hold during the solution evolution, and the iterated variable is shown to converge to the optimal solution under a few mild assumptions;
- 2) The proposed algorithm is more efficient in memory and communication compared to existing results. Each node only needs to store a variable of size $d_i + M$ and transmit a variable of size $2M$ to its neighbors, where d_i is the size of the local primal variable and M the number of coupling constraints. This is notably smaller compared to, e.g., primal decomposition algorithms [1] and dual decomposition algorithms [4];
- 3) We apply the distributed optimization algorithm to the CBF-induced QP as feedback in safe distributed control. To accommodate the slowly time-varying parameters in QP, we further strengthen our proposed algorithm such that finite-time convergence is achieved.

The organization of the paper goes as follows. We first present problem formulation in Section II as well as some leading assumptions. The proposed primal decomposition scheme and the corresponding updates of auxiliary variables are detailed in Section III and Section IV, respectively. The overall algorithm and the convergence analysis are discussed in Section V. Two special cases are further considered in the following sections, focusing on the sparsely coupled case and the CBF-induced QP case. Numerical results are shown in

Section VIII to demonstrate the effectiveness and the favorable properties of our proposed algorithms.

II. PRELIMINARIES AND PROBLEM SETUP

Notations: \mathbb{R} is the set of real numbers, and $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. \mathbb{R}^n denotes the n dimensional Euclidean vector space. Each vector $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$ is interpreted as a column vector if not stated otherwise. $\text{diag}(\cdot)$ is a function that only keeps the diagonal entries and sets the off-diagonal entries zero. $\text{blkdiag}(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n)$ denotes a block diagonal matrix with diagonal blocks $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n$, where $\mathbf{g}_i, i = 1, \dots, n$, can be either a vector or a matrix. For $x \in \mathbb{R}$, $\text{sign}(x) := \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$ For any $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $\text{sign}(\mathbf{x}) := (\text{sign}(x_1), \text{sign}(x_2), \dots, \text{sign}(x_n))$. Vector inequalities are to be interpreted element-wise. $\mathbf{0}$ and $\mathbf{1}$ refer to vectors of proper dimensions with all entries to be 0 or 1, respectively. Given a function $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$, $\partial f(\mathbf{x})$, the subdifferential of f at \mathbf{x} , is defined as the set of vectors \mathbf{s} such that $f(\mathbf{x}') \geq f(\mathbf{x}) + \mathbf{s}^\top (\mathbf{x}' - \mathbf{x})$ for all $\mathbf{x}' \in \mathbb{R}^n$. Such a vector \mathbf{s} is called a subgradient of f at \mathbf{x} . When $f(\mathbf{x})$ is differentiable, the gradient of f is denoted ∇f and the partial derivative $\frac{\partial f}{\partial x_i}$. For two matrices A and B of proper dimensions, we use the convention that $[A, B]$ denotes the horizontal stacking, and $[A; B]$ the vertical stacking. Let $A \in \mathbb{R}^{m \times n}$ be a matrix and $\mathbf{I} \subseteq \{1, 2, \dots, m\}$ be an index set. Then, $A^{\mathbf{I}}$ denotes a matrix composed by the rows indexed in \mathbf{I} . Note that $A^{\mathbf{I}} = I^{\mathbf{I}} A$ where I is an identity matrix of proper dimension.

A. Problem setup

Consider the following multi-agent optimization problem

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} \quad & \sum_{i \in \mathcal{I}} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \begin{cases} \sum_{i \in \mathcal{I}} \mathbf{a}_i^{\mathbf{1}\top} \mathbf{x}_i + b_i^{\mathbf{1}} \leq 0, \\ \vdots \\ \sum_{i \in \mathcal{I}} \mathbf{a}_i^{\mathbf{m}\top} \mathbf{x}_i + b_i^{\mathbf{m}} \leq 0, \\ \vdots \\ \sum_{i \in \mathcal{I}} \mathbf{a}_i^{\mathbf{M}\top} \mathbf{x}_i + b_i^{\mathbf{M}} \leq 0, \end{cases} \end{aligned} \quad (3)$$

where $\mathcal{I} = \{1, 2, \dots, N\}$, and there are M linear coupling constraints indexed by $\mathcal{M} = \{1, 2, \dots, M\}$. Here $\mathbf{a}_i^m, \mathbf{x}_i \in \mathbb{R}^{d_i}, d_i \in \mathbb{N}, b_i^m \in \mathbb{R}, \forall i \in \mathcal{I}, m \in \mathcal{M}$. For brevity, denote $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N]$, $\mathbf{a}^m = [\mathbf{a}_1^m; \mathbf{a}_2^m; \dots; \mathbf{a}_N^m]$, $\mathbf{b}^m = (b_1^m, \dots, b_N^m)$, $m \in \mathcal{M}$, $\mathbf{a}_i = [\mathbf{a}_i^1; \mathbf{a}_i^2; \dots; \mathbf{a}_i^M]$, $\mathbf{b}_i = (b_i^1, \dots, b_i^M)$, $i \in \mathcal{I}$. Only $\mathbf{a}_i, \mathbf{b}_i, f_i$ are known to agent i .

The N computational nodes communicate over a connected and undirected graph $\mathcal{G} = (\mathcal{I}, E)$. $(i, j) \in E$ represents that the agents i, j can communicate with each other. The associated Laplacian matrix [20] is denoted L , and \mathbf{l}_i is its i -th row. We make the following assumptions.

Assumption 1. For all $i \in \mathcal{I}$, each function $f_i(\mathbf{x}_i)$ is convex.

Assumption 2. The optimal value of (3) is finite.

Stacking $\{\mathbf{a}_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ and $\{\mathbf{b}_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ by the order of the agents, we obtain $\mathbf{a}_A = [\mathbf{a}_1; \mathbf{a}_2; \dots; \mathbf{a}_N]$,

$\mathbf{b}_A = [\mathbf{b}_1; \mathbf{b}_2; \dots; \mathbf{b}_N]$. Similarly, stacking $\{\mathbf{a}_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ and $\{b_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ by the order of the constraints, we obtain $\mathbf{a}_C = [\mathbf{a}^1; \mathbf{a}^2; \dots; \mathbf{a}^M]$, $\mathbf{b}_C = [\mathbf{b}^1; \mathbf{b}^2; \dots; \mathbf{b}^M]$. There exists a constant permutation matrix¹ J such that $\mathbf{b}_C = J\mathbf{b}_A$. Note that J only depends on N and M and is independent of the communication graph \mathcal{G} .

In this work, we aim to find a continuous-time distributed algorithm that solves (3) while satisfying all coupling constraints for all time. In [16], a special case of (3) with quadratic cost and a single coupling constraint is investigated. However, the approach in [16] is difficult to generalize to convex programs with multiple coupling constraints, as studied in this work.

Our proposed approach is based on two key observations. 1) By introducing a set of auxiliary variables as decision variables, an equivalent optimization problem can be constructed with separable costs and constraints. 2) When viewing these auxiliary variables as independent variables, the subgradient of the optimal value with respect to these variables can be computed using local information. The first fact motivates the primal decomposition scheme. The second fact motivates the distributed updated law for these variables leveraging subgradient algorithms. The convergence and optimality of this approach are later shown using tools from nonsmooth analysis. Two special cases with sparse coupling constraints or quadratic cost functions are also discussed in the sequel.

III. EQUIVALENT PROBLEM

In this section we present a new optimization problem that is equivalent to (3) but with a separable structure. Similar decomposition schemes were also discussed in [16, Proposition 1] and [17, Proposition 4.1]. Introduce auxiliary decision variables $\{y_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ where $y_i^m \in \mathbb{R}$. Denote $\mathbf{y}^m = (y_1^m, \dots, y_N^m) \in \mathbb{R}^N$, $m \in \mathcal{M}$, and $\mathbf{y} = [\mathbf{y}^1; \mathbf{y}^2; \dots; \mathbf{y}^M]$. Consider

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}^1, \dots, \mathbf{y}^M} \quad & \sum_{i \in \mathcal{I}} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \begin{cases} A^1 \mathbf{x} + L\mathbf{y}^1 + \mathbf{b}^1 \leq \mathbf{0} \\ \vdots \\ A^M \mathbf{x} + L\mathbf{y}^M + \mathbf{b}^M \leq \mathbf{0} \end{cases} \end{aligned} \quad (4)$$

where recall that L is the graph Laplacian matrix, $A^m = \text{blkdiag}(\mathbf{a}_1^m, \dots, \mathbf{a}_N^m) \in \mathbb{R}^{(\sum_i d_i) \times M}$.

When organizing the constraints by the order of agents, the problem can be rewritten as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}^1, \dots, \mathbf{y}^M} \quad & \sum_{i \in \mathcal{I}} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \begin{cases} A_1^\top \mathbf{x}_1 + (I_M \otimes \mathbf{l}_1)\mathbf{y} + \mathbf{b}_1 \leq \mathbf{0} \\ \vdots \\ A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i)\mathbf{y} + \mathbf{b}_i \leq \mathbf{0} \\ \vdots \\ A_N^\top \mathbf{x}_N + (I_M \otimes \mathbf{l}_N)\mathbf{y} + \mathbf{b}_N \leq \mathbf{0} \end{cases} \end{aligned} \quad (5)$$

¹A permutation matrix J is a square binary matrix that has exactly one entry of 1 in each row and each column and zeros elsewhere. A permutation matrix is orthogonal, i.e., $JJ^\top = J^\top J = I$.

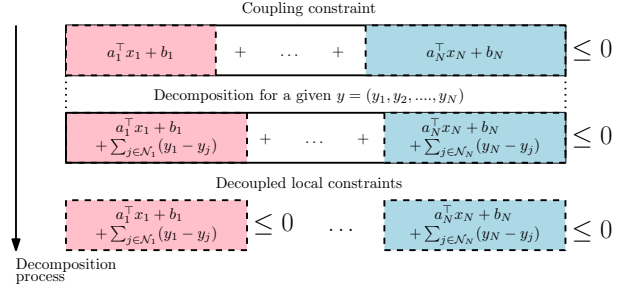


Fig. 1: Illustration of the constraint decomposition scheme. Here only one coupling constraint is considered and the superscript is neglected for simplicity. By introducing an auxiliary variable \mathbf{y} , one coupling constraint becomes N local constraints. Proposition 1 shows that when \mathbf{y} is a decision variable, the two class of constraints represent the same feasible region in \mathbf{x} . Our proposed algorithm locally updates y_i to obtain an “optimal” constraint decomposition in the sense that $\mathbf{a}_i^\top \mathbf{x}_i^* + b_i + \sum_{j \in \mathcal{N}_i} (y_i - y_j) \leq 0$ for all $i \in \mathcal{I}$, where \mathbf{x}_i^* is the optimal solution to the original optimization problem.

where recall that \mathbf{l}_i is the i th row of the graph Laplacian, $A_i = [\mathbf{a}_i^1, \mathbf{a}_i^2, \dots, \mathbf{a}_i^M] \in \mathbb{R}^{d_i \times M}$. One verifies that the optimization problems in (4), (5) are the same up to a permutation of constraints. An illustration of the decomposition approach is shown in Fig. 1.

Proposition 1 (Equivalence). *The two optimization problems in (3) and (4) are equivalent, in the sense that*

- 1) for any feasible solution $(\mathbf{x}', \mathbf{y}')$ to (4), \mathbf{x}' is a feasible solution to (3);
- 2) for any feasible solution \mathbf{x}' to (3), there exists a $\mathbf{y}' \in \mathbb{R}^{NM}$ such that $(\mathbf{x}', \mathbf{y}')$ is a feasible solution to (4);
- 3) the two problems have the same cost function.

Proof. See Appendix. \square

If $\mathbf{y}^m, m \in \mathcal{M}$, are viewed as independent variables instead of decision variables, then the problem (5) has a natural separable structure for a given \mathbf{y} . In the following, we consider a distributed solution to (3), composed of the local optimization problems for agent $i \in \mathcal{I}$

$$\begin{aligned} \min_{\mathbf{x}_i} \quad & f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \begin{cases} \mathbf{a}_i^1 \top \mathbf{x}_i + \sum_{j \in \mathcal{N}_i} (y_i^1 - y_j^1) + b_i^1 \leq 0 \\ \vdots \\ \mathbf{a}_i^M \top \mathbf{x}_i + \sum_{j \in \mathcal{N}_i} (y_i^M - y_j^M) + b_i^M \leq 0 \end{cases} \end{aligned} \quad (6)$$

and some local update law for the variable $y_i^m, i \in \mathcal{I}, m \in \mathcal{M}$, that will be detailed in the next section. We note that this local optimization problem will give an optimal \mathbf{x}_i if and only if $\mathbf{y}^m, m \in \mathcal{M}$ is optimal to the problem in (4) in view of Proposition 1. Nevertheless, for any given \mathbf{y} , we have the following result.

Proposition 2 (Constraint satisfaction). *For a given \mathbf{y} , if $\mathbf{x}_i, i \in \mathcal{I}$, is feasible to the local optimization problem in (6), then the stacked vector $[\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N]$ is a feasible solution to (3).*

Proof. Consider the m -th constraint in (3). By summing up the m -th constraint of (6) for each agent, we have $\sum_{i \in \mathcal{I}} \mathbf{a}_i^{m\top} \mathbf{x}_i + \sum_{i \in \mathcal{I}} \mathbf{l}_i \mathbf{y}^m + \sum_{i \in \mathcal{I}} b_i^m = \sum_{i \in \mathcal{I}} \mathbf{a}_i^{m\top} \mathbf{x}_i + b_i^m \leq 0$, considering that $\sum_{i \in \mathcal{I}} \mathbf{l}_i \mathbf{y}^m = \mathbf{1}^\top L \mathbf{y}^m = 0$. \square

IV. SENSITIVITY ANALYSIS OF LOCAL AND GLOBAL OPTIMIZATION PROBLEMS

In this section, we will investigate how \mathbf{y} affects the optimal value of the local optimization problems. We will propose an updating scheme for \mathbf{y} such that the sum of the optimal values of the local problems always decreases until reaching the optimal value of the centralized problem.

Assumption 3. For any $\mathbf{y} \in \mathbb{R}^{NM}$, Slater's conditions hold for all local optimization problems, i.e., there exists \mathbf{x}_i that is strictly feasible to (6).

One sufficient condition for this assumption to hold is that $\text{Rank}(A_i) = M, \forall i \in \mathcal{I}$. In the following we assume Assumption 3 holds. For a given \mathbf{y} , consider the local problem (6)

$$\begin{aligned} \phi_i(\mathbf{y}) &:= \min f_i(\mathbf{x}_i) \\ \text{s.t. } & A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \leq 0. \end{aligned} \quad (7)$$

Under Assumption 3, $\phi_i(\mathbf{y})$ is well-defined for all $\mathbf{y} \in \mathbb{R}^{NM}$. Given $\mathbf{y} \in \mathbb{R}^{NM}$, define

$$\phi(\mathbf{y}) = \sum_{i \in \mathcal{I}} \phi_i(\mathbf{y}) \quad (8)$$

In general, $\phi(\mathbf{y})$ is not differentiable. In the following, we use tools from nonsmooth analysis [21]. Let the subdifferential, i.e., the set of all subgradients, of $\phi(\mathbf{y})$ be $\partial\phi(\mathbf{y})$. We have the following results characterizing the subdifferential of $\phi(\mathbf{y})$.

Proposition 3 (Subdifferential). Let Assumptions 1, 2, and 3 hold. Denote by $\mathbf{c}_i = (c_i^1, \dots, c_i^M) \in \mathbb{R}^M$ a Lagrange multiplier to (6) for $i \in \mathcal{I}$. Define $\zeta_i^m = \sum_{j \in N_i} (c_i^m - c_j^m)$, $\zeta^m = (\zeta_1^m, \zeta_2^m, \dots, \zeta_N^m)$, $\zeta = [\zeta^1; \zeta^2; \dots; \zeta^M]$. Then

- 1) $\mathbf{c}_i \in \mathbb{R}^M$, not necessarily unique, always exists,
- 2) $\phi(\cdot)$ is convex, but not strongly convex,
- 3) $\zeta \in \partial\phi(\mathbf{y})$.

Proof. From Assumptions 1, 2, and 3, the strong duality theorem for convex inequality constraints [22, Proposition 3.5.1] applies, and thus, the strong duality holds and at least one Lagrange multiplier exists. This proves Point 1).

We show the convexity of $\phi_i(\mathbf{y})$ by definition. For any $\mathbf{y}', \mathbf{y}''$, denote $\mathbf{x}', \mathbf{x}''$ the corresponding optimal solution of (7). It is clear that $\phi_i(\lambda \mathbf{y}' + (1-\lambda)\mathbf{y}'') \leq f_i(\lambda \mathbf{x}' + (1-\lambda)\mathbf{x}'') \leq \lambda f_i(\mathbf{x}') + (1-\lambda)f_i(\mathbf{x}'') = \lambda \phi_i(\mathbf{y}') + (1-\lambda)\phi_i(\mathbf{y}'')$, where the inequalities hold thanks to the linearity of the constraints and the convexity of $f_i(\cdot)$, respectively. Thus, $\phi(\mathbf{y})$ is also convex. $\phi(\mathbf{y})$ not being strongly convex is evident since $\phi(\mathbf{y} + s\mathbf{1}) = \phi(\mathbf{y})$ for any $s \in \mathbb{R}$.

Now consider a perturbed version of (7) with \mathbf{y}' and denote its optimal value $\phi_i(\mathbf{y}')$. From sensitivity analysis [23, Equation 5.57], we know

$$\phi_i(\mathbf{y}') \geq \phi_i(\mathbf{y}) + \mathbf{c}_i^\top (I_M \otimes \mathbf{l}_i)(\mathbf{y}' - \mathbf{y}) \quad (9)$$

for all \mathbf{y}' . This means $(I_M \otimes \mathbf{l}_i^\top) \mathbf{c}_i$ is a subgradient of $\phi_i(\mathbf{y})$. Since $\phi(\mathbf{y})$ is convex, $\partial\phi$ is non-empty and takes compact and convex values [24, Proposition 9]. Based on the calculus of subdifferentials: $\partial(f_1 + f_2)(\mathbf{x}) \supseteq \partial f_1(\mathbf{x}) + \partial f_2(\mathbf{x})$, we know

$$\sum_{i \in \mathcal{I}} (I_M \otimes \mathbf{l}_i^\top) \mathbf{c}_i \in \partial\phi(\mathbf{y}). \quad (10)$$

Note that

$$(I_M \otimes \mathbf{l}_i^\top) \mathbf{c}_i = \begin{bmatrix} \mathbf{l}_i^\top & 0 & & 0 \\ 0 & \mathbf{l}_i^\top & & 0 \\ & & \ddots & \\ 0 & 0 & & \mathbf{l}_i^\top \end{bmatrix} \begin{bmatrix} c_i^1 \\ c_i^2 \\ \vdots \\ c_i^M \end{bmatrix} = \begin{bmatrix} c_i^1 \mathbf{l}_i^\top \\ c_i^2 \mathbf{l}_i^\top \\ \vdots \\ c_i^M \mathbf{l}_i^\top \end{bmatrix}$$

and that

$$\sum_{i \in \mathcal{I}} \mathbf{c}_i^m \mathbf{l}_i^\top = [\mathbf{l}_1^\top \quad \mathbf{l}_2^\top \quad \dots \quad \mathbf{l}_N^\top] \mathbf{c}^m = L^\top \mathbf{c}^m = L \mathbf{c}^m$$

with $\mathbf{c}^m = (c_1^m, c_2^m, \dots, c_N^m)$. We thus obtain

$$\begin{aligned} \sum_{i \in \mathcal{I}} (I_M \otimes \mathbf{l}_i^\top) \mathbf{c}_i &= [L \mathbf{c}^1; L \mathbf{c}^2; \dots; L \mathbf{c}^M] \\ &= (I_M \otimes L) \mathbf{c}_C \in \partial\phi(\mathbf{y}), \end{aligned} \quad (11)$$

where $\mathbf{c}_C = [\mathbf{c}^1; \mathbf{c}^2; \dots; \mathbf{c}^M]$ (ordered by constraints). Equivalently, we have $\zeta \in \partial\phi(\mathbf{y})$. This proves Point 3). \square

Remark 1. Recall that \mathbf{c}_i can be calculated solely based on the local problem at node i . Proposition 3 reveals that although each computational node does not have the full information about the subgradient of $\phi(\mathbf{y})$, each computational node has access to the subgradient of $\phi(\mathbf{y})$ with respect to the local auxiliary variable \mathbf{y}_i by communicating local Lagrange multipliers to its neighbors. Specifically, we have that

$$\frac{\partial\phi(\mathbf{y})}{\partial\mathbf{y}_i^m} = \zeta_i^m = \sum_{j \in N_i} (c_i^m - c_j^m), \forall m \in \mathcal{M} \quad (12)$$

whenever $\phi(\mathbf{y})$ is differentiable.

V. PROPOSED ALGORITHM

In this section, we give details about the proposed continuous-time distributed optimization algorithm. A pseudocode for computational node i is given in Algorithm 1. Note that since the analysis is done in continuous time, we are assuming that the four steps in the loop block are executed at the same time instant, i.e., $\mathbf{y}_i, \mathbf{y}_j, \mathbf{c}_i, \mathbf{c}_j$ in Steps 4 and 6 are computed and transmitted simultaneously. For digital implementation, all agents synchronously communicate \mathbf{y}_i and \mathbf{c}_i in Steps 4 and 6. We also choose a small sampling time to approximate the continuous-time update of \mathbf{y}_i in Step 7.

Theorem 1. Let Assumptions 1, 2 and 3 hold. Denote by $\mathbf{c}_i(\mathbf{y})$ the Lagrange multiplier of (6) given \mathbf{y} . If \mathbf{y}_i is updated according to

$$\dot{\mathbf{y}}_i = -k_0 \sum_{j \in N_i} (\mathbf{c}_i(\mathbf{y}) - \mathbf{c}_j(\mathbf{y})) \text{ for almost all } t, \quad (13)$$

where $k_0 > 0$ is a constant gain and a Caratheodory solution²

²A Caratheodory solution of a differential equation $\dot{\mathbf{x}}(t) = X(\mathbf{x}(t))$ or a differential inclusion $\dot{\mathbf{x}}(t) \in X(\mathbf{x}(t))$ is an absolutely continuous function $\mathbf{x}(t) : [0, \infty) \rightarrow \mathbb{R}^n$ that satisfies the differential equation or the differential inclusion for almost all t , respectively [24].

Algorithm 1 Distributed Optimization Algorithm

- 1: **Locally stored state:** x_i, \mathbf{y}_i
 - 2: **Initialization:** choose arbitrary \mathbf{y}_i .
 - 3: **loop**
 - 4: Send \mathbf{y}_i to and gather \mathbf{y}_j from neighbors $j \in N_i$
 - 5: Compute (x_i, c_i) as the primal-dual optimizer of (6)
 - 6: Send c_i to and gather c_j from neighbors $j \in N_i$
 - 7: Update \mathbf{y}_i according to (13)
 - 8: **end loop**
-

exists, then

- 1) the Caratheodory solution is unique and converges to the set of optimal \mathbf{y} of the equivalent problem (4),
- 2) the solution to the local problem (6) asymptotically converges to (one of) the optimal solution(s) of the centralized problem in (3),
- 3) and the coupling constraints in (3) hold during the solution evolution.

Proof. From Proposition 3, c_i is well-defined for all \mathbf{y} , but it is not necessarily unique. From the right-hand side of (13), one derives

$$\dot{\mathbf{y}}_i = -k_0 \begin{bmatrix} \sum_{j \in N_i} (c_i^1 - c_j^1) \\ \sum_{j \in N_i} (c_i^2 - c_j^2) \\ \vdots \\ \sum_{j \in N_i} (c_i^M - c_j^M) \end{bmatrix} = -k_0 \begin{bmatrix} l_i c^1 \\ l_i c^2 \\ \vdots \\ l_i c^M \end{bmatrix} \quad (14)$$

for almost all t . Re-order $\{-k_0 l_i c^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ by the order of constraints, we obtain $\dot{\mathbf{y}} = [\dot{\mathbf{y}}^1; \dot{\mathbf{y}}^2; \dots; \dot{\mathbf{y}}^M]$ where

$$\dot{\mathbf{y}}^m = -k_0 [l_1 c^m; l_2 c^m; \dots; l_N c^m] = -k_0 L c^m.$$

Thus, $\dot{\mathbf{y}} = -k_0 (I_M \otimes L) c_C$ for almost all t , where we recall $c_C = [c^1; c^2; \dots; c^M]$ (ordered by constraints). Denote the Caratheodory solution of this differential equation to be $\mathbf{y}(t)$.

Now consider Caratheodory solutions of the differential inclusion $\dot{\mathbf{y}} \in -k_0 \partial \phi(\mathbf{y}) = -\partial k_0 \phi(\mathbf{y})$ in view of Proposition 3, where $\phi(\mathbf{y})$ is defined in (8). From [24, Gradient Differential Inclusion of a Convex Function], we know such a solution exists and is unique. Recall that $\dot{\mathbf{y}}(t) \in -\partial k_0 \phi(\mathbf{y}(t))$ for almost all t , thus $\mathbf{y}(t)$ is also unique.

In order to investigate how $\phi(\mathbf{y}(t))$ evolves with respect to time, we check the set-valued lower Lie derivative along the differential inclusion

$$\underline{L}_{-\partial k_0 \phi(\mathbf{y})} \phi(\mathbf{y}) = \{a \in \mathbb{R} : \text{there exists } \boldsymbol{\xi} \in \partial \phi(\mathbf{y}) \text{ such that } a = \min_{\mathbf{v} \in -\partial k_0 \phi(\mathbf{y})} \boldsymbol{\xi}^\top \mathbf{v}\} \quad (15)$$

For any $\boldsymbol{\xi} \in \partial \phi(\mathbf{y})$, there exists $\mathbf{v} = -k_0 \boldsymbol{\xi} \in -\partial k_0 \phi(\mathbf{y})$ such that $\boldsymbol{\xi}^\top \mathbf{v} = -k_0 \|\boldsymbol{\xi}\|^2 \leq 0$. Hence $\sup \underline{L}_{-\partial k_0 \phi(\mathbf{y})} \phi(\mathbf{y}) \leq 0$ for all \mathbf{y} , and the equality holds if and only if $\mathbf{0} \in \partial \phi(\mathbf{y})$. Following [24, Propositions 13] and the uniqueness of the solutions, we know $\phi(\mathbf{y}(t))$ is nonincreasing. In particular, when $\mathbf{0} \notin \partial \phi(\mathbf{y})$, $\sup \underline{L}_{-\partial k_0 \phi(\mathbf{y})} \phi(\mathbf{y}) < 0$, and thus $\phi(\mathbf{y}(t))$ is strictly decreasing.

Denote $h(t) = \sup \underline{L}_{-\partial k_0 \phi(\mathbf{y}(t))} \phi(\mathbf{y}(t))$, $\alpha = \limsup_{t \rightarrow \infty} h(t)$. Such a limit exists and is bounded

since $h(t)$ is upper bounded by 0. Assume $\alpha < 0$, then, $\exists T > 0$ and $\epsilon = -\alpha/2$ such that $h(t) < \alpha + \epsilon = \alpha/2 < 0$ for all $t \geq T$. This implies that $\phi(\mathbf{y}(t)) < \phi(\mathbf{y}(T)) + \alpha(t-T)/2$ for all $t \geq T$, which contradicts with Assumption 2. Thus, we have $\limsup_{t \rightarrow \infty} h(t) = 0$. This means that a subsequence of $h(t)$ converges to 0 asymptotically, which leads to the conclusion that a subsequence of $\mathbf{y}(t)$ converges to the set $\{\mathbf{y} : \mathbf{0} \in \partial \phi(\mathbf{y})\}$. This proves Point 1).

Since $\mathbf{y}(t)$ converges to the optimal set $\{\mathbf{y} : \mathbf{0} \in \partial \phi(\mathbf{y})\}$ asymptotically, and in view of the equivalence between (3) and (4) from Proposition 1, we know the optimal solution to (3) is obtained asymptotically. This shows Point 2). Point 3) follows from Assumption 3 and Proposition 2. \square

Remark 2. Convergence of the subgradient differential inclusion of a convex function was investigated in [21] and [24, Gradient Differential Inclusion of a Convex Function], where the convergence property has been established. However, we note that here $\phi(\mathbf{y})$ has infinitely many local minima and they may not be bounded. Proofs in [21] and [24] rely on nonsmooth Lyapunov function analysis (with an assumption on the uniqueness of the local minima) or LaSalle's principle (with an assumption on the boundedness of the local minima), both of which are not applicable here. Instead, we did not prove the convergence of $\mathbf{y}(t)$ to a particular optimal point, but to an optimal set $\{\mathbf{y} : \mathbf{0} \in \partial \phi(\mathbf{y})\}$.

Remark 3 (Caratheodory solution). One may wonder why a Caratheodory solution is considered in Theorem 1 and how to guarantee/verify its existence. In Proposition 3, we have established that, for a given \mathbf{y} , the Lagrange multiplier c_i always exists but may not be unique under Assumptions 1, 2, and 3. If $c_i(\mathbf{y})$, as a function of \mathbf{y} , is continuous, then classic solutions to the ODE in (13) exist, and of course a Caratheodory solution exists. We introduce the Caratheodory solution so that we can neglect discontinuities of $c_i(\mathbf{y}(t))$ for a duration of measure zero. Theoretically, for the local optimization problem (7) parameterized by \mathbf{y} , when the cost function is smooth and the Strong Second Order Sufficient Condition (SSOSC) holds at \mathbf{y}' , there exists a neighborhood of \mathbf{y}' such that a unique primal-dual optimizer $(x_i(\mathbf{y}), c_i(\mathbf{y}))$, when viewed as a function of \mathbf{y} , exists and is locally Lipschitz continuous in \mathbf{y} [25]. Other conditions exist for ensuring continuous primal-dual pairs of perturbed optimization problems. See a recent survey [26] and references therein. Numerically, if there are multiple eligible local Lagrange multipliers, one can choose c_i that is close to c_i^- , where c_i^- denotes the variable c_i calculated one step before.

Remark 4 (Differentiability). [27, Corollary 7.3.1] states that the perturbed optimal value function is locally differentiable if and only if the Lagrange multiplier is unique. Thus, one sufficient condition for local differentiability of $\phi_i(\mathbf{y})$ is that the optimization problem in (7) fulfills the Linearly Independent Constraint Qualification (LICQ), which ensures the existence and uniqueness of the Lagrange multiplier [28]. When $\phi_i(\mathbf{y})$'s are differentiable, the subgradient then becomes the usual gradient, i.e., $\partial \phi_i(\mathbf{y}) = \{\nabla \phi_i(\mathbf{y})\}$. In this case, the adaptive law in (13) becomes the well-established gradient flow. This

will be discussed in detail for one special case, namely the control barrier function-induced quadratic programs, in Section VII.

Remark 5 (Memory, computation, and communication efficiency). In Algorithm 1, each computational node i stores variables \mathbf{x}_i and \mathbf{y}_i of dimension $d_i + M$, while broadcasting \mathbf{y}_i and \mathbf{c}_i of dimension $2M$ to neighbors. Compared to the primal decomposition scheme in [1], our approach requires smaller storage ($d_i + M$ versus $d_i + D_i M$) and communication ($2M$ versus $2D_i M$) variables, where D_i is the degree of the communication graph at node i [20]. [17] adopts the same decomposition scheme as in this work, but requires each local node to store $2d_i + 2M$ variables. Additionally, the proposed algorithm requires less computation for local nodes at each iteration, as they only need to solve a single optimization problem with the size of decision variable d_i , unlike the dual decomposition scheme in [4], which involves solving two optimizations with sizes d_i and M .

VI. A SPARSE CASE

What has been discussed so far deals with the most general case, and is applicable in a densely coupled scenario, i.e., every computational node is involved in every coupling constraint. In many cases, the constraint coupling is sparse and, usually, is consistent with the communication topology. For example, in multi-robot applications [29]–[31], collision avoidance is a pair-wise constraint between two robots, and they need to have some information exchange between them in order to work out a safe path. Exploiting the sparsity of the constraint coupling would help us relax conservative conditions, reduce the size of optimization problems on each computational node, and improve communication efficiency with the neighbors. To formally characterize sparsity, we first define the following notations.

Define $\mathcal{I}_m \subseteq \mathcal{I}$ as the index set of the computational nodes involved in the m -th constraint, i.e., $i \in \mathcal{I}_m$ if and only if $\mathbf{a}_i^m \neq \mathbf{0}$ or $b_i^m \neq 0$. Similarly, define $\mathcal{M}_i \subseteq \mathcal{M}$ as the index set of the coupling constraints involving node i , i.e., $m \in \mathcal{M}_i$ if and only if $\mathbf{a}_i^m \neq \mathbf{0}$ or $b_i^m \neq 0$. Define the set of neighboring nodes of node i involved in the m -th constraint as $N_i^m = \{j \in \mathcal{I} : j \in N_i \text{ and } j \in \mathcal{I}_m\}$. In the sparse setting, we assume that each agent knows which neighbors are involved in which relevant constraints, i.e., $N_i^m, \forall m \in \mathcal{M}_i$ is known to agent i .

We call the communication graph $\mathcal{G} = (\mathcal{I}, E)$ is consistent with the m -th constraint if the induced subgraph [20] $\mathcal{G}_m := (\mathcal{I}_m, E_m)$ is connected, where $(i, j) \in E_m$ if and only if $i, j \in \mathcal{I}_m$ and $(i, j) \in E$. Define $\mathcal{G}'_m := (\mathcal{I}, E_m)$.

Assumption 4. The communication graph \mathcal{G} is consistent with all constraints.

Figure 2 illustrates a scenario where there are 4 computational nodes and 2 coupling constraints. Note that both \mathcal{G}_1 (the subgraph in pink) and \mathcal{G}_2 (the subgraph in blue) are connected, then \mathcal{G} is consistent with these two constraints.

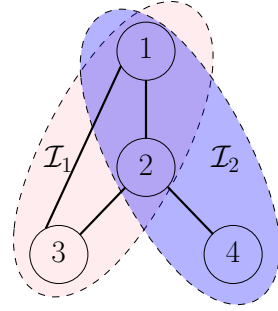


Fig. 2: An illustration of a communication graph satisfying Assumption 4. In this scenario, the agent index sets for each constraint are $\mathcal{I}_1 = \{1, 2, 3\}, \mathcal{I}_2 = \{1, 2, 4\}$, the constraint index sets for each agent are $\mathcal{M}_1 = \mathcal{M}_2 = \{1, 2\}, \mathcal{M}_3 = \{1\}, \mathcal{M}_4 = \{2\}$, and the sets of neighboring nodes with relevant constraint involvement are $N_1^1 = \{2, 3\}, N_1^2 = \{2\}, N_2^1 = \{1, 3\}, N_2^2 = \{1, 4\}, N_3^1 = \{1, 2\}, N_4^2 = \{2\}$, respectively.

Under Assumption 4, we know at most $|\mathcal{M}_i|$ columns of A_i are nonzero. One sufficient condition that fulfills Assumption 3 in this case is thus $\text{Rank}(A_i) = |\mathcal{M}_i|$ for all i .

A. Equivalent optimization problems

Now we show that the original problem in (3) is equivalent to the following optimization problems in a sparse setting.

$$\begin{aligned} & \min_{\mathbf{x}_i, \mathbf{y}_i, i \in \mathcal{I}} \sum_{i \in \mathcal{I}} f_i(\mathbf{x}_i) \\ & \text{s.t. } \mathbf{a}_i^m \top \mathbf{x}_i + \sum_{j \in N_i^m} (y_i^m - y_j^m) + b_i^m \leq 0, \quad (16) \\ & \quad \quad \quad \forall i \in \mathcal{I}, m \in \mathcal{M}_i. \end{aligned}$$

Proposition 4 (Equivalence in a sparse setting). Under Assumptions 3 and 4, the two problems in (3) and (16) are equivalent, in the sense that

- 1) for any feasible solution $(\mathbf{x}', \mathbf{y}')$ to (16), \mathbf{x}' is a feasible solution to (3);
- 2) for any feasible solution \mathbf{x}' to (3), there exists a variable \mathbf{y}' such that $(\mathbf{x}', \mathbf{y}')$ is a feasible solution to (16);
- 3) the two problems have the same cost function.

The proof follows similar steps as in that of Proposition 1 except that now we need to use the Laplacian matrices of the induced subgraphs $\mathcal{G}_m, m \in \mathcal{M}$, which are assumed to be connected under Assumption 4. Details are omitted here for brevity. Similar to Section III, the reformulated problem in (16) has a separable structure, and only local information $y_j^m, j \in N_i^m$ is needed to determine the constraints concerning each node.

It is obvious from (16) that $y_i^m, m \in \mathcal{M} \setminus \mathcal{M}_i$ is not involved in the local optimization problem. Thus, we can pre-set those components in \mathbf{y} to be constant zero. The sparse formulation in (16) also accounts for local constraints that involve only one node. Denoting the node by i , we have $a_j^m, b_j^m = 0$ for any $j \neq i$ and we pre-set $y_k^m = 0$ for any $k \in \mathcal{I}$.

$$\begin{aligned}
\phi_i(\mathbf{y}') &\geq \phi_i(\mathbf{y}) + \sum_{m \in \mathcal{M}_i} c_i^m \sum_{j \in N_i^m} \left((y_i^{\prime,m} - y_j^{\prime,m}) - (y_i^m - y_j^m) \right) \\
&= \phi_i(\mathbf{y}) + \sum_{m \in \mathcal{M}_i} c_i^m \tilde{\mathbf{l}}_{i, \mathcal{G}_m} (\mathbf{y}'^m - \mathbf{y}^m) \\
&\quad + \sum_{m \notin \mathcal{M}_i} c_i^m \tilde{\mathbf{l}}_{i, \mathcal{G}_m} (\mathbf{y}'^m - \mathbf{y}^m), \text{ where } \tilde{L}_i := \begin{bmatrix} \tilde{\mathbf{l}}_{i, \mathcal{G}_1} & 0 & \dots & 0 \\ 0 & \tilde{\mathbf{l}}_{i, \mathcal{G}_2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \tilde{\mathbf{l}}_{i, \mathcal{G}_M} \end{bmatrix} \\
&= \phi_i(\mathbf{y}) + \mathbf{c}_i^\top \tilde{L}_i (\mathbf{y}' - \mathbf{y})
\end{aligned} \tag{17}$$

B. local optimization problems

Following a similar analysis procedure, in the following we will treat \mathbf{y}_i as an independent variable and design a distributed updating law so that \mathbf{y}_i converges to the optimal \mathbf{y}_i^* to (16). For node i and a given \mathbf{y} , a local optimization problem is given by

$$\phi_i(\mathbf{y}) := \min_{\mathbf{x}_i} f_i(\mathbf{x}_i)$$

$$\text{s.t. } \mathbf{a}_i^{m\top} \mathbf{x}_i + \sum_{j \in N_i^m} (y_i^m - y_j^m) + b_i^m \leq 0, \forall m \in \mathcal{M}_i. \tag{18}$$

Define $\mathbf{c}_i = (c_i^1, \dots, c_i^m, \dots, c_i^M)$, where c_i^m is the Lagrange multiplier of (18) if $m \in \mathcal{M}_i$, and is equal to 0 otherwise.

Similar to previous discussions, we have the constraint satisfaction property as follows. The proof is omitted for brevity.

Proposition 5 (Constraint satisfaction). *For any given \mathbf{y} , if $\mathbf{x}_i, i \in \mathcal{I}$ is feasible to the local optimization problem in (18), then $[\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N]$ is a feasible solution to (3).*

Define the summation of the optimal values of the local optimization problems as

$$\phi(\mathbf{y}) = \sum_{i \in \mathcal{I}} \phi_i(\mathbf{y}). \tag{19}$$

Proposition 6 (Subdifferential). *Let Assumptions 1, 2, 3 and 4 hold. Let $\mathbf{c}_i = (c_i^1, \dots, c_i^M) \in \mathbb{R}^m$ be defined below (18). Define $\zeta_i^m = \sum_{j \in N_i^m} (c_i^m - c_j^m)$ if $m \in \mathcal{M}_i$ and $\zeta_i^m = 0$ if $m \notin \mathcal{M}_i$. Let $\boldsymbol{\zeta}^m = (\zeta_1^m, \zeta_2^m, \dots, \zeta_N^m), \boldsymbol{\zeta} = [\boldsymbol{\zeta}^1; \boldsymbol{\zeta}^2; \dots; \boldsymbol{\zeta}^M]$. Then*

- 1) $\mathbf{c}_i \in \mathbb{R}^m$, not necessarily unique, always exists,
- 2) $\phi(\mathbf{y})$ is convex,
- 3) $\boldsymbol{\zeta} \in \partial\phi(\mathbf{y})$.

Proof. The first two claims follow from a similar argument of that of Proposition 3 and are omitted here. We will show Point 3) in the following. From sensitivity analysis, for any \mathbf{y}, \mathbf{y}' , we know (17) holds, where $\tilde{\mathbf{l}}_{i, \mathcal{G}_m}$ is the i -th row of the Laplacian matrix of $\mathcal{G}'_m = (\mathcal{I}, E_m)$. The first inequality in (17) is from [23, Equation 5.57], and the last two equalities follow from the definition of graph Laplacian matrices and noting that when $m \notin \mathcal{M}_i$, $\tilde{\mathbf{l}}_{i, \mathcal{G}_m} = \mathbf{0}$. (17) implies that $\tilde{L}_i^\top \mathbf{c}_i$ is a subgradient of $\phi_i(\mathbf{y})$.

Following the calculus rule for subdifferential, $\sum_i \tilde{L}_i^\top \mathbf{c}_i$ is a subgradient of $\phi(\mathbf{y})$. With a few linear algebraic operations, we know

$$\tilde{L}_i^\top \mathbf{c}_i = \begin{bmatrix} \tilde{\mathbf{l}}_{i, \mathcal{G}_1}^\top & 0 & \dots & 0 \\ 0 & \tilde{\mathbf{l}}_{i, \mathcal{G}_2}^\top & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \tilde{\mathbf{l}}_{i, \mathcal{G}_M}^\top \end{bmatrix} \begin{bmatrix} c_i^1 \\ c_i^2 \\ \dots \\ c_i^M \end{bmatrix} = \begin{bmatrix} c_i^1 \tilde{\mathbf{l}}_{i, \mathcal{G}_1}^\top \\ c_i^2 \tilde{\mathbf{l}}_{i, \mathcal{G}_2}^\top \\ \dots \\ c_i^M \tilde{\mathbf{l}}_{i, \mathcal{G}_M}^\top \end{bmatrix}$$

Thus, the entry in $\sum_i \tilde{L}_i^\top \mathbf{c}_i$ that corresponds to y_i^m is

$$\begin{aligned}
\left[\sum_{k \in \mathcal{I}} c_k^m \tilde{\mathbf{l}}_{k, \mathcal{G}_m}^\top \right]_i &= \sum_{k \in \mathcal{I}} \left[\tilde{\mathbf{l}}_{k, \mathcal{G}_m}^\top \right]_i c_k^m \\
&= \begin{cases} \sum_{j \in N_i^m} (c_i^m - c_j^m) & \text{if } m \in \mathcal{M}_i \\ 0 & \text{if } m \notin \mathcal{M}_i \end{cases}. \tag{20}
\end{aligned}$$

The second equality comes from the fact that $[\tilde{\mathbf{l}}_{k, \mathcal{G}_m}^\top]_i = [L_{\mathcal{G}'_m}]_{k, i}$, the (k, i) -th element of the Laplacian of the graph $\mathcal{G}'_m = (\mathcal{I}, E_m)$. Thus $[\tilde{\mathbf{l}}_{k, \mathcal{G}_m}^\top]_i$ is equal to 0 if $m \notin \mathcal{M}_i$, or $k \neq i \wedge k \notin N_i^m$; -1 if $k \in N_i^m$; and $|N_i^m|$ if $k = i$. Thus we conclude $\boldsymbol{\zeta}$ is a subgradient of $\phi(\mathbf{y})$. \square

The distributed optimization algorithm for node i in the sparse case is given as follows.

Algorithm 2 Sparse Distributed Optimization Algorithm

- 1: **Local stored state:** $\mathbf{x}_i, \mathbf{y}_i$
 - 2: **Initialization:** Choose arbitrary \mathbf{y}_i , and let $y_i^m = 0$ if $m \notin \mathcal{M}_i$.
 - 3: **loop**
 - 4: Send y_i^m to and gather y_j^m from $j \in N_i^m, \forall m \in \mathcal{M}_i$
 - 5: Compute $(\mathbf{x}_i, \mathbf{c}_i)$ from (18)
 - 6: Send c_i^m to and gather c_j^m from $j \in N_i^m, \forall m \in \mathcal{M}_i$
 - 7: Update \mathbf{y}_i according to (21)
 - 8: **end loop**
-

The main difference compared to Algorithm 1 is that 1) now node i receives $\mathbf{y}_j, \mathbf{c}_j$ from and send $\mathbf{y}_i, \mathbf{c}_i$ to neighboring nodes that are relevant, as in set N_i^m in Steps 4 and 7; and 2) only relevant components in \mathbf{y}_j and \mathbf{c}_j are taken into account, as in (18) and (21) in Steps 5 and 6. This leads to fewer variables to communicate among the network and smaller size local optimization problems to solve.

The following theorem states the convergence property.

Theorem 2. *Let Assumptions 1, 2, 3, 4 hold and \mathbf{c}_i as defined below (18). If $\mathbf{y}_i(t)$ is updated according to*

$$\dot{\mathbf{y}}_i^m = \begin{cases} -k_0 \sum_{j \in N_i^m} (c_i^m - c_j^m), & \text{if } m \in \mathcal{M}_i \\ 0 & \text{if } m \notin \mathcal{M}_i \end{cases}, \tag{21}$$

for almost all t , where $k_0 > 0$ is a constant gain and a Caratheodory solution exists, then the properties in Theorem 1 hold.

Proof. In view of (20), we observe that $\dot{\mathbf{y}}_i^m(t)$ given in (21) is exactly the product of (20) together with a constant negative gain. Stacking $\dot{\mathbf{y}}_i^m$ by the order of constraints, we obtain that $\dot{\mathbf{y}} \in -\partial k_0 \phi(\mathbf{y})$ for almost all t from Proposition 6. Thus, the

analyses in the proof of Theorem 1 are also applicable here and are neglected. \square

VII. A SPECIAL CASE: CONTROL BARRIER FUNCTIONS-INDUCED QUADRATIC PROGRAMS

A. Control barrier functions and safe distributed control

In this section, we consider the application of our proposed algorithms in the context of control barrier function-induced safe distributed control. CBF approaches [29]–[31] have already been applied to multi-robot safe coordination with pair-wise constraints, for example, collision avoidance and connectivity maintenance. In the following we introduce a more general formulation that further includes collective constraints.

Consider a multi-agent system with \mathbf{p}_i the state vector of agent i , $i \in \mathcal{I}$. The dynamics of each agent is given by

$$\dot{\mathbf{p}}_i = \mathbf{f}_i(\mathbf{p}_i) + \mathbf{g}_i(\mathbf{p}_i)\mathbf{x}_i,$$

where $\mathbf{f}_i, \mathbf{g}_i$ are locally Lipschitz vector fields, \mathbf{x}_i is the control input to agent i that is yet to be designed. Suppose that this multi-agent system is subject to M coupling state constraints, that is, along the system trajectory, it has to maintain

$$h^m(\mathbf{p}_1, \dots, \mathbf{p}_N) \geq 0, m \in \mathcal{M},$$

and h^m is a smooth function. The constraint h^m is a collective constraint since it restricts the states of all agents. We call the following inequality a CBF condition associated with the m -th constraint

$$\sum_{i \in \mathcal{I}} \mathbf{a}_i^{m\top} \mathbf{x}_i + b_i^m \leq 0, \quad (22)$$

where $\mathbf{a}_i^{m\top} = -\nabla_{\mathbf{p}_i} h^m(\mathbf{p}_1, \dots, \mathbf{p}_N)$, $b_i^m, i \in \mathcal{I}$ are such that $\sum_{i \in \mathcal{I}} b_i^m = -k h^m(\mathbf{p}_1, \dots, \mathbf{p}_N) - \sum_{i \in \mathcal{I}} \nabla_{\mathbf{p}_i} h^m(\mathbf{p}_1, \dots, \mathbf{p}_N) \mathbf{f}_i(\mathbf{p}_i)$, and constant $k > 0$. Here we use linear functions instead of the extended class \mathcal{K} functions for simplicity. Results from CBF literature [32] state that, if a locally Lipschitz control input $\mathbf{x}_i, i \in \mathcal{I}$, is chosen such that (22) holds for all time, then the set $\{(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N) : h^m(\mathbf{p}_1, \dots, \mathbf{p}_N) \geq 0\}$ is forward invariant and the m -th state constraint is always satisfied, provided that it is initially satisfied. In the following analysis we will neglect the explicit state constraints $h^m(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N) \geq 0$ and instead focus on enforcing the CBF conditions for all time.

In CBF literature [16], [32], a commonly used formulation to enforce the CBF conditions is expressed as a quadratic program as follows

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} & \sum_{i \in \mathcal{I}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2 \\ \text{s.t.} & \begin{cases} \sum_{i \in \mathcal{I}} \mathbf{a}_i^{1\top} \mathbf{x}_i + b_i^1 \leq 0, \\ \vdots \\ \sum_{i \in \mathcal{I}} \mathbf{a}_i^{M\top} \mathbf{x}_i + b_i^M \leq 0, \end{cases} \end{aligned} \quad (23)$$

The quadratic program in (23) aims at minimally modifying nominal local controllers $\mathbf{x}_{nom,i}$, which relate to the underlying system tasks, while always respecting M CBF conditions, hereafter referred to as coupling constraints, which relate to system safety.

One challenging problem that naturally arises from this formulation is, for multi-agent systems where agents only have local information, how to solve this optimization problem in a distributed way with guaranteed safety assurance.

We assume that $\mathbf{a}_i, b_i, \mathbf{x}_{nom,i}$ only depend on locally obtainable information from the neighbors of the i -th agent, and that the quadratic program is always feasible. These are referred to as the local obtainability and compatibility properties of the CBF conditions as discussed in [16], [33]. One illustrative application of coordinating a multi-agent system with multiple state constraints is shown later in the simulation section.

What sets apart the CBF-induced QP in (23) from the previously discussed distributed optimization formulation is that, since the solution to (23) is applied as feedback to a dynamical system and $\mathbf{x}_{nom,i}, \mathbf{a}_i^m, b_i^m$ are state-dependent, they are time-varying in nature rather than static and evolve along the closed-loop system trajectory. In the following, we will first take a look at a time-invariant instance of (23), i.e., when $\mathbf{x}_{nom,i}, \mathbf{a}_i^m$ and b_i^m are constant, and then analyze the performance of our proposed algorithm in a time-varying setting. We note that the results in this section are also applicable to the sparsely coupled case. This extension is straightforward and neglected for notation simplicity.

Following the previous analysis, with the help of an auxiliary variable $\mathbf{y} \in \mathbb{R}^{NM}$, we have $\phi(\mathbf{y}) = \sum_{i \in \mathcal{I}} \phi_i(\mathbf{y})$ with $\phi_i(\mathbf{y})$ given by the local optimization problems

$$\begin{aligned} \phi_i(\mathbf{y}) = \min_{\mathbf{x}_i} & \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2 \\ \text{s.t.} & A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \leq 0. \end{aligned} \quad (24)$$

Proposition 7. Consider the local QP in (24) with constant $\mathbf{x}_{nom,i}, A_i, \mathbf{y}, \mathbf{b}_i$. Assume that Assumption 3 holds. Then the Lagrange multiplier $\mathbf{c}_i = (c_i^1, c_i^2, \dots, c_i^M)$ exists and satisfies the algebraic equation

$$\begin{aligned} \Lambda_i \mathbf{c}_i = \max & \left(A_i^\top \mathbf{x}_{nom,i} + (I_M \otimes \mathbf{l}_i) \mathbf{y} \right. \\ & \left. + \mathbf{b}_i + (\Lambda_i - A_i^\top A_i) \mathbf{c}_i, \mathbf{0} \right), \end{aligned} \quad (25)$$

where $\Lambda_i = \text{diag}(A_i^\top A_i)$, and the primal-dual optimizer to (24) is $(\mathbf{x}_{nom,i} - A_i \mathbf{c}_i, \mathbf{c}_i)$. The optimal cost is $\frac{1}{2} \|\Lambda_i \mathbf{c}_i\|^2$.

The proof follows from analyzing the Karush–Kuhn–Tucker (KKT) condition of the Lagrangian and is given in Appendix for the sake of completeness. A similar result was reported in [34, Theorem 5]. There however A_i, Λ_i are assumed to be non-singular.

For the time-invariant optimization problem, if the local optimization problem in (24) always fulfills the Linearly Independent Constraint Qualification³ (LICQ) by, for example, A_i being full column rank, then [27, Corollary 7.3.1] states that $\phi_i(\mathbf{y})$ in (24) is differentiable as discussed in Remark 4. In this case, the subgradient reduces to the gradient in the normal sense, and the adaptive law in (13) becomes a gradient flow. Under Assumptions 2, 3 and the convexity of $\phi(\mathbf{y})$, the

³We say that the Linearly Independent Constraint Qualification is satisfied at \mathbf{x}^* for an optimization problem, if the gradients of all active constraints are linearly independent.

convergence guarantee of $\phi(\mathbf{y}(t))$ is well-established in view that it is lower bounded and monotonically decreasing.

B. Finite-time convergence for time-varying optimizations

In this subsection, we will strengthen our proposed algorithms of the previous sections to compensate for the time-varying parameters in the QP. Under LICQ, since the update law in (13) is a gradient flow, one could modify it to a scaled normalized gradient flow as in [35, (6b)], that is,

$$\dot{\mathbf{y}}_i = -k_0 \text{sign} \frac{\partial}{\partial \mathbf{y}_i} \phi(\mathbf{y}) = -k_0 \text{sign} \sum_{j \in N_i} (\mathbf{c}_i - \mathbf{c}_j). \quad (26)$$

[35] shows that the normalized gradient flow achieves finite-time convergence for static strongly convex functions. However, this result is not directly applicable here since $\phi(\mathbf{y})$ is not strongly convex. Moreover, for the CBF-induced QP in (23), $\mathbf{x}_{nom,i}$, \mathbf{a}_i^m , \mathbf{b}_i^m depend on the states of the multi-agent system and are time-varying. Before presenting the main result, we introduce the following Lemma that will simplify the analysis later on.

Lemma 1. *Let \mathbf{w}_C be the vector of $\{w_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$, $w_i^m \in \mathbb{R}$ ordered by constraints. Define $\mathbf{v}_C := I_M \otimes L \mathbf{w}_C$, $\mathbf{v}_A := [I_M \otimes \mathbf{l}_1; I_M \otimes \mathbf{l}_2; \dots; I_M \otimes \mathbf{l}_N] \mathbf{w}_C$, where \mathbf{l}_i is the i -th row of the Laplacian L . Then, $\mathbf{v}_C = J \mathbf{v}_A$, where J is the permutation matrix defined in Section II.A.*

Proof. Define $\mathbf{w}^m = (w_1^m, \dots, w_N^m)$. Let $v_i^m = [L \mathbf{w}^m]_i$. Then it is clear that \mathbf{v}_C is the vector of $\{v_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ ordered by constraints, and \mathbf{v}_A is the vector of $\{v_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ ordered by agents. Thus $\mathbf{v}_C = J \mathbf{v}_A$. \square

The main result is summarized in the theorem below with slowly time-varying $\mathbf{x}_{nom,i}(t)$, $\mathbf{a}_i^m(t)$, $\mathbf{b}_i^m(t)$. Here we note that LICQ is imposed as an easy-to-check sufficient condition for guaranteeing the differentiability of $\phi_i(\mathbf{y})$. Depending on the problem at hand, other conditions may also be applicable.

Theorem 3. *Consider the centralized QP in (23). Assume the LICQ holds for every local optimization problem in (24) at every time instant. Let \mathbf{c}_i be the Lagrange multiplier of the local QP for agent i . Assume further that $\mathbf{x}_{nom,i}(t)$, $\mathbf{a}_i^m(t)$, $\mathbf{b}_i^m(t)$ are slowly time-varying in the sense that there exists a constant $D > 0$ such that $\|\mathbf{d}_i^{\mathbf{l}_i}(t)\| \leq D$ for all $i \in \mathcal{I}$ and all index sets $\mathbf{l}_i \subseteq \mathcal{M}$ with*

$$\mathbf{d}_i^{\mathbf{l}_i}(t) = -\frac{dA_i^{\top, \mathbf{l}_i} A_i^{\top, \mathbf{l}_i, \top}}{dt} \mathbf{c}_i^{\mathbf{l}_i} + \frac{d}{dt} (A_i^{\top, \mathbf{l}_i} \mathbf{x}_{nom,i} + \mathbf{b}_i^{\mathbf{l}_i}). \quad (27)$$

Denote $\underline{\lambda} = \min_{\mathbf{l}_i, i \in \mathcal{I}} \lambda_{\min}((A_i^{\top, \mathbf{l}_i} A_i^{\top, \mathbf{l}_i, \top})^{-1})$, $\bar{\lambda} = \max_{\mathbf{l}_i, i \in \mathcal{I}} \lambda_{\max}((A_i^{\top, \mathbf{l}_i} A_i^{\top, \mathbf{l}_i, \top})^{-1})$. If k_0 is chosen such that

$$k_0 \underline{\lambda} - ND \sqrt{M} \bar{\lambda} \|L\| - \epsilon \geq 0 \quad (28)$$

for some $\epsilon > 0$, then the Filippov solution of the scaled normalized gradient flow in (26) converges to a critical point of $\phi(\mathbf{y})$ in finite time $t_r \leq \sum_{m \in \mathcal{M}} \|L \mathbf{c}^m(0)\|_1 / \epsilon$. Moreover, the coupling constraints in (23) are always satisfied.

Proof. Consider the nonsmooth Lyapunov function

$$V(\mathbf{y}) = \left\| \frac{\partial \phi(\mathbf{y})}{\partial \mathbf{y}} \right\|_1 = \|(I_M \otimes L) \mathbf{c}_C\|_1 = \sum_{m \in \mathcal{M}} \|L \mathbf{c}^m\|_1.$$

Denote at the time t the index set of strictly active constraints of agent i to be $\mathbf{I}_i(t)$ and its complement $\bar{\mathbf{I}}_i(t) = \mathcal{M} \setminus \mathbf{I}_i(t)$.

In the following, we will show that for any constant $\mathbf{I}_i(t)$, $i \in \mathcal{I}$, $V(\mathbf{y})$ decreases monotonically for a system $\Sigma_{\mathbf{I}_i, i \in \mathcal{I}}$ given by (26) together with the algebraic equation

$$\begin{aligned} A_i^{\top, \mathbf{I}_i} A_i^{\top, \mathbf{I}_i, \top} \mathbf{c}_i^{\mathbf{I}_i} &= A_i^{\top, \mathbf{I}_i} \mathbf{x}_{nom,i} + (I^{\mathbf{I}_i} \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i^{\mathbf{I}_i}, \\ \mathbf{c}_i^{\bar{\mathbf{I}}_i} &= \mathbf{0}. \end{aligned} \quad (29)$$

for all $i \in \mathcal{I}$. This equation is obtained by considering only the constraints indexed in \mathbf{I}_i in (25) and noting $\mathbf{c}_i^{\mathbf{I}_i} > \mathbf{0}$. When viewing $\mathbf{I}_i, i \in \mathcal{I}$ as the index of such systems, since $V(\mathbf{y})$ decreases monotonically for all $\Sigma_{\mathbf{I}_i, i \in \mathcal{I}}$ as will be shown later, $V(\mathbf{y})$ effectively acts as a common Lyapunov function [36]. Thus it suffices to consider a system $\Sigma_{\mathbf{I}_i, i \in \mathcal{I}}$, and the time index is dropped to ease the notation.

For arbitrary constant $\mathbf{I}_i, i \in \mathcal{I}$, from (29), we know

$$\begin{aligned} d\mathbf{c}_i^{\mathbf{I}_i}/dt &= (A_i^{\top, \mathbf{I}_i} A_i^{\top, \mathbf{I}_i, \top})^{-1} (I^{\mathbf{I}_i} \otimes \mathbf{l}_i d\mathbf{y}/dt + \mathbf{d}_i^{\mathbf{I}_i}) \\ d\mathbf{c}_i^{\bar{\mathbf{I}}_i}/dt &= \mathbf{0} \end{aligned}$$

Here $\mathbf{d}_i^{\mathbf{I}_i}$ is defined in (27). $A_i^{\top, \mathbf{I}_i} A_i^{\top, \mathbf{I}_i, \top}$ is invertible since LICQ is assumed. Denote by $\Gamma_i' = \begin{pmatrix} (A_i^{\top, \mathbf{I}_i} A_i^{\top, \mathbf{I}_i, \top})^{-1} & 0 \\ 0 & 0 \end{pmatrix}$ and $J_{\mathbf{I}_i}$ the permutation matrix such that $[\mathbf{c}_i^{\mathbf{I}_i}; \mathbf{c}_i^{\bar{\mathbf{I}}_i}] = J_{\mathbf{I}_i} \mathbf{c}_i$. Thus $[I^{\mathbf{I}_i}; I^{\bar{\mathbf{I}}_i}] = J_{\mathbf{I}_i} I_M$. We can write the time derivative of \mathbf{c}_i in a compact form as

$$\begin{aligned} \dot{\mathbf{c}}_i &= J_{\mathbf{I}_i}^{\top} [d\mathbf{c}_i^{\mathbf{I}_i}/dt; d\mathbf{c}_i^{\bar{\mathbf{I}}_i}/dt] \\ &= J_{\mathbf{I}_i}^{\top} (\Gamma_i') J_{\mathbf{I}_i} (I_M \otimes \mathbf{l}_i \dot{\mathbf{y}} + J_{\mathbf{I}_i}^{\top} [\mathbf{d}_i^{\mathbf{I}_i}; \mathbf{0}]) \\ &= \Gamma_i (I_M \otimes \mathbf{l}_i \dot{\mathbf{y}} + \mathbf{d}_i^{\mathbf{I}_i'}) \end{aligned} \quad (30)$$

where $\Gamma_i = J_{\mathbf{I}_i}^{\top} \Gamma_i' J_{\mathbf{I}_i}$, $\mathbf{d}_i^{\mathbf{I}_i'} := J_{\mathbf{I}_i}^{\top} [\mathbf{d}_i^{\mathbf{I}_i}; \mathbf{0}]$.

Let $\mathbf{z} = I_M \otimes L \mathbf{c}_C$. Then $V(\mathbf{y}) = \|\mathbf{z}\|_1$. The generalized time derivative is given by

$$\frac{d}{dt} V(\mathbf{y}) \in \sum_{i \in \mathcal{I}^=} \text{sign}(z_i) \dot{z}_i + \sum_{i \in \mathcal{I}^=} \text{SIGN}(z_i) \dot{z}_i \quad (31)$$

where $\mathcal{I}^= = \{i \in \mathcal{I} : z_i = 0\}$, $\mathcal{I}^{\neq} = \{i \in \mathcal{I} : z_i \neq 0\}$, $\text{SIGN}(z) := \begin{cases} 1, & z > 0; \\ [-1, 1], & z = 0; \\ -1, & z < 0. \end{cases}$ Since we are dealing with Filippov solutions, we can disregard the case which $z_i = 0$ holds for isolated time instants of measure zero. If $z_i = 0$ holds along an interval of time of positive measure, then, in the sense of Filippov, \dot{z}_i exists at those time instants and $\dot{z}_i = 0$. Based on this fact, we have that for almost all t ,

$\frac{d}{dt} V(\mathbf{z}) = \sum_{i \in \mathcal{I}^{\neq}} \text{sign}(z_i) \dot{z}_i + \sum_{i \in \mathcal{I}^=} \text{sign}(z_i) \dot{z}_i$, that is,

$$\frac{d}{dt} V(\mathbf{y}) = \text{signz}^{\top} \dot{\mathbf{z}} = \text{signz}^{\top} (I_M \otimes L) J \dot{\mathbf{c}}_A \quad (32)$$

Define $\boldsymbol{\sigma} = (I_M \otimes L) \text{signz}$. After some calculations, one finds $\boldsymbol{\sigma} = [L \text{sign} L \mathbf{c}^1; L \text{sign} L \mathbf{c}^2; \dots; L \text{sign} L \mathbf{c}^M]$. Denote by $\sigma_i^m = [L \text{sign} L \mathbf{c}^m]_i$. Then $\boldsymbol{\sigma}$ can be seen as $\{\sigma_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ ordered by constraints. Let $\boldsymbol{\sigma}_i = (\sigma_i^1, \sigma_i^2, \dots, \sigma_i^M)$, and $\boldsymbol{\sigma}_A = J^{\top} \boldsymbol{\sigma} = [\boldsymbol{\sigma}_1; \boldsymbol{\sigma}_2; \dots; \boldsymbol{\sigma}_N]$ be the vector of $\{\sigma_i^m\}_{i \in \mathcal{I}, m \in \mathcal{M}}$ ordered by agents. From Lemma 1, we have

$$\boldsymbol{\sigma}_A = [I_M \otimes \mathbf{l}_1; I_M \otimes \mathbf{l}_2; \dots; I_M \otimes \mathbf{l}_N] \text{signz}. \quad (33)$$

Furthermore, from (26), $\dot{\mathbf{y}}_A = -k_0 \text{sign}([I_M \otimes \mathbf{l}_1; I_M \otimes \mathbf{l}_2; \dots; I_M \otimes \mathbf{l}_N] \mathbf{c}_C)$, thus $\dot{\mathbf{y}} = -k_0 \text{sign}(I_M \otimes L \mathbf{c}_C) = -k_0 \text{sign}(\mathbf{z})$ in view of Lemma 1 and the definition of \mathbf{z} . Further considering (33), we have $[I_M \otimes \mathbf{l}_1; I_M \otimes \mathbf{l}_2; \dots; I_M \otimes \mathbf{l}_N] \dot{\mathbf{y}} = -k_0 \boldsymbol{\sigma}_A$. Following (30), we thus obtain

$$\begin{aligned} \dot{\mathbf{c}}_A &= [\dot{\mathbf{c}}_1; \dot{\mathbf{c}}_2; \dots; \dot{\mathbf{c}}_N] \\ &= \text{blkdiag}(\Gamma_1, \dots, \Gamma_N) (-k_0 \boldsymbol{\sigma}_A + [\mathbf{d}_1^{\mathbf{I}_1'}; \dots; \mathbf{d}_N^{\mathbf{I}_N'}]) \end{aligned} \quad (34)$$

From (32), the generalized time derivative is

$$\begin{aligned} \frac{d}{dt} V(\mathbf{y}) &= \boldsymbol{\sigma}_A^\top \dot{\mathbf{c}}_A \\ &= \sum_{i \in \mathcal{I}} \boldsymbol{\sigma}_i^\top \Gamma_i (-k_0 \boldsymbol{\sigma}_i + \mathbf{d}_i^{\mathbf{I}_i'}) \end{aligned} \quad (35)$$

Consider the following two cases:

- 1) $L \mathbf{c}^m = \mathbf{0}$ for all $m \in \mathcal{M}$. In this case, $V(\mathbf{y}) = 0$, and thus from the first-order optimality condition one critical point of $\phi(\mathbf{y})$ has been reached.
- 2) $L \mathbf{c}^m \neq \mathbf{0}$ for some $m \in \mathcal{M}$. This implies that $\|L \text{sign} L \mathbf{c}^m\| \geq 1$ [37, Proposition 2.1]. Without loss of generality, assume that $|\sigma_1^1| \geq 1$ and $\mathbf{I}_1 = \{1, 2, \dots, M'\}$. These assumptions are not restrictive as they can be fulfilled by permuting the numbering of the agents and constraints. This leads to

$$\Gamma_1 = \begin{pmatrix} (A_1^{\top, \mathbf{I}_1} A_1^{\top, \mathbf{I}_1})^{-1} & 0 \\ 0 & 0 \end{pmatrix} \quad (36)$$

and $\|\boldsymbol{\sigma}_1^{\mathbf{I}_1}\| \geq 1$. The generalized time derivative becomes

$$\begin{aligned} \frac{d}{dt} V(\mathbf{y}) &\leq -k_0 \boldsymbol{\sigma}_1^\top \Gamma_1 \boldsymbol{\sigma}_1 + \sum_{i \in \mathcal{I}} \boldsymbol{\sigma}_i^\top \Gamma_i \mathbf{d}_i^{\mathbf{I}_i'} \\ &\leq -k_0 \lambda_{\min}((A_1^{\top, \mathbf{I}_1} A_1^{\top, \mathbf{I}_1})^{-1}) \\ &\quad + \sum_{i \in \mathcal{I}} \|\boldsymbol{\sigma}_i\| \|\Gamma_i\| \|\mathbf{d}_i^{\mathbf{I}_i'}\| \end{aligned} \quad (37)$$

for almost all t . The first inequality comes from the positive semi-definiteness of Γ_i , and the second inequality follows from the positive definiteness of $(A_1^{\top, \mathbf{I}_1} A_1^{\top, \mathbf{I}_1})^{-1}$ and $\|\boldsymbol{\sigma}_1^{\mathbf{I}_1}\| \geq 1$. Note that $\boldsymbol{\sigma}_i \in \mathbb{R}^M$, $[\boldsymbol{\sigma}_i]_m = [L \text{sign} L \mathbf{c}^m]_i$, then $\|\boldsymbol{\sigma}_i\| \leq \sqrt{M} \|L\|$, $\|\mathbf{d}_i^{\mathbf{I}_i'}\| = \|d_i^{\mathbf{I}_i}\| \leq D$. Thus, from (28),

$$\frac{d}{dt} V(\mathbf{y}) \leq -k_0 \lambda + ND \sqrt{M \lambda} \|L\| \leq -\epsilon \quad (38)$$

Based on these discussions, we know $\frac{d}{dt} V \leq -\epsilon$ for almost all t as long as $L \mathbf{c}^m \neq \mathbf{0}$ for some $m \in \mathcal{M}$ for arbitrary systems $\Sigma_{i, i \in \mathcal{I}}$. Hence, leveraging the theory of common Lyapunov functions, it gives a trivial upper estimate of the convergence time as $\sum_{m \in \mathcal{M}} \|L \mathbf{c}^m(0)\|_1 / \epsilon$. From Proposition 2, we further conclude that the coupling constraints are fulfilled for all time. \square

Theorem 3 shows that the solution optimal to the slowly time-varying quadratic program (23) is obtained in a violation-free, finite-time, and distributed manner. These properties are favorable for implementing the CBF-induced quadratic programs as in (23) in a safe, (point-wise) optimal, and distributed way.

One could easily recognize that the update law in (26) and the analysis in Theorem 3 are similar to those in [16], where a simple case with a single coupling constraint is considered. We note that this extension is not trivial due to the nonlinear relations between \mathbf{c}_i and \mathbf{y} in (25). Compared to [16], here we distinguish active constraints from inactive ones in order to obtain an analytical relation between changes in \mathbf{c}_i and \mathbf{y} , and show that the Lyapunov function monotonically decreases for any active constraints, i.e., it is a common Lyapunov function.

VIII. SIMULATIONS

In this section, we demonstrate the effectiveness of the proposed algorithms in a static optimization problem and a multi-agent safe coordination application. All the simulations are conducted in Matlab R2022b on an Intel Core i7-1365U Windows laptop, and the primal-dual pairs are solved by the built-in `quadprog` function with default parameters. We apply forward Euler integration for simulating the continuous-time dynamics with a sampling time 0.01s.

A. A static online optimization problem

In this subsection, we consider an optimization problem that involves $N = 9$ agents indexed in $\mathcal{I} = \{1, 2, \dots, 9\}$ with the communication graph shown in Fig. 3. Here each agent i has 6 decision variables $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,6})$, where $(x_{i,1}, x_{i,2}, x_{i,3})$ represents the demands of 3 products, and $(x_{i,4}, x_{i,5}, x_{i,6})$ the supplies of 3 resources. The optimization problem is given by

$$\begin{aligned} \min_{\mathbf{x}_1, \dots, \mathbf{x}_9} & \sum_{i \in \mathcal{I}} (\mathbf{x}_i^\top Q \mathbf{x}_i + \sum_{j=4,5,6} h_{i,j} x_{i,j}) \\ \text{s.t.} & \begin{cases} \sum_{i \in \mathcal{I}} 2x_{i,2} + x_{i,3} - x_{i,4} \leq 0, \\ \sum_{i \in \mathcal{I}} 2x_{i,1} + x_{i,3} - x_{i,5} \leq 0, \\ \sum_{i \in \mathcal{I}} x_{i,1} + x_{i,2} - x_{i,6} \leq 0, \\ x_{i,1} \geq h_{i,1}, x_{i,2} \geq h_{i,2}, x_{i,3} \geq h_{i,3}, \forall i \in \mathcal{I} \end{cases} \end{aligned} \quad (39)$$

where $h_{i,j} = \text{ceil}(10 \sin(ij) + 20)$, $i \in \mathcal{I}, j = 1, 2, \dots, 6$, Q is symmetric and $\mathbf{x}_i^\top Q \mathbf{x}_i = (2x_{i,2} + x_{i,3} - x_{i,4})^2 + (2x_{i,1} + x_{i,3} - x_{i,5})^2 + (x_{i,1} + x_{i,2} - x_{i,6})^2$. The first three inequality constraints represent that the total supply for the resources needs to exceed the total demand of producing 3 different products; the last three inequality constraints represent the local production output requirements. In the cost function, we penalize the derivation between the local supply and local demand, which can be interpreted as the cost of transporting or storing the deficient or surplus resources, and take into account the local cost of obtaining those resources. We assume that the local production output requirement $h_{i,j}, j = 1, 2, 3$ and the local cost coefficient $h_{i,j}, j = 4, 5, 6$ are only known to agent i .

It is clear that the local cost function $f_i(\mathbf{x}_i) = \mathbf{x}_i^\top Q \mathbf{x}_i + \sum_{j=4,5,6} h_{i,j} x_{i,j}$ is convex but not strongly convex, and the local coefficient matrix per the definition in (6)

$$A_i^\top = \begin{pmatrix} 0 & 2 & 1 & -1 & 0 & 0 \\ 2 & 0 & 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}$$

is of full rank. This fulfills Assumption 3 on Slater's condition.

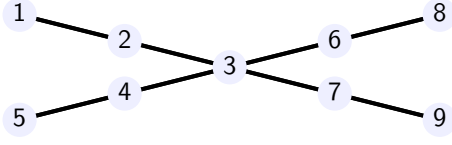


Fig. 3: Communication graph

From the structure of the problem, since the last three constraints only involve its own decision variable, the algorithms for the sparse case apply here and that the related \mathbf{y}^m equals zero. Our proposed approach introduces 27 auxiliary scalar variables, with each node needing to communicate 6 scalar variables with its neighbors. In comparison, the primal decomposition approach in [1] requires 48 auxiliary scalar variables and 24 scalar variables to be communicated by node 3. By choosing $y_i^m(0) = 0, \forall i \in \mathcal{I}, m \in \mathcal{M}, k_0 = 1$ with sampling-time 0.01s, we obtain the results shown in Fig. 4. One observes that 1) the total cost error is monotonically decreasing; 2) the local Lagrange multipliers corresponding to each coupling constraint converge to a common value; and 3) the three coupling constraints are always fulfilled during the solution iterations. This violation-free behavior could be desirable in this scenario because one does not need to wait for the convergence of the optimization algorithm before applying the solutions, and the intermediate solution, though suboptimal, still guarantees that the supply-demand requirement is respected.

B. An application to safe distributed control for MAS

In the second example, we show one application of the proposed algorithm for safe distributed control of a multi-agent system. Denote by $\mathbf{p}_i = (p_{i,1}, p_{i,2}) \in \mathbb{R}^2$ the state and $\mathbf{p}_{i,d} \in \mathbb{R}^2$ the desired state of agent i . We assume that the dynamics of each agent is given by $\dot{\mathbf{p}}_i = \mathbf{x}_i$, where \mathbf{x}_i is the velocity command to agent i . The coordination task is to transform the formation of the multi-agent system from the initial formation $\mathbf{p}_{\text{initial}} = [\mathbf{p}_1(0); \mathbf{p}_2(0); \dots; \mathbf{p}_9(0)] = (4, 6, 2, 8, 0, 10, -2, 8, -4, 6, 2, 12, -2, 12, 4, 14, -4, 14)$ (shown in red in Fig. 5) to the target formation $\mathbf{p}_{\text{final}} = [\mathbf{p}_{1,d}; \mathbf{p}_{2,d}; \dots; \mathbf{p}_{9,d}] = (60, 16, 60, 13, 60, 10, 60, 7, 60, 4, 63, 13, 63, 7, 66, 16, 66, 4)$. The communication graph is in Fig. 3. Assume that agent $i, i \in \mathcal{I}$ has access to the desired relative position $\mathbf{p}_{ij,d} = \mathbf{p}_{j,d} - \mathbf{p}_{i,d}, j \in N_i$ and its absolute position \mathbf{p}_i , and agent 3, referred to as the leader, knows its target position $\mathbf{p}_{3,d}$. An intuitive distributed protocol for fulfilling this task is given by

$$\mathbf{x}_{nom,i}(\mathbf{p}_i, \{\mathbf{p}_j\}_{j \in N_i}) = \sum_{j \in N_i} \mathbf{p}_{ij,d} - \mathbf{p}_{ij} + \delta_i(\mathbf{p}_{i,d} - \mathbf{p}_i),$$

where $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$, $\delta_i = 1$ if agent i is the leader and $\delta_i = 0$ otherwise.

Suppose that the multi-agent team needs to satisfy 2 coupling state constraints during the formation transition, given

by

$$h_1(t, \mathbf{p}_1, \dots, \mathbf{p}_9) = 30 + t - 0.1 \sum_{i \in \mathcal{I}} (p_{i,1} + p_{i,2}) \geq 0, \quad (40a)$$

$$h_2(t, \mathbf{p}_1, \dots, \mathbf{p}_9) = 10 + t - 0.1 \sum_{i \in \mathcal{I}} (p_{i,1} - p_{i,2}) \geq 0. \quad (40b)$$

These two inequalities impose time-varying upper bounds on the collective summation of all coordinates as in (40a) and the collective difference between the first and the second coordinates as in (40b), respectively. When viewing the two coordinates of \mathbf{p}_i as two different local sources and the overall task as transitioning from one resource allocation scheme to another, one may interpret the first constraint as a gradual relaxation on the source limit, and the second one as the gradual relaxation on the difference limit between the resources.

Following the control barrier function-induced control design procedures [38], the centralized safety-assuring controller is given by

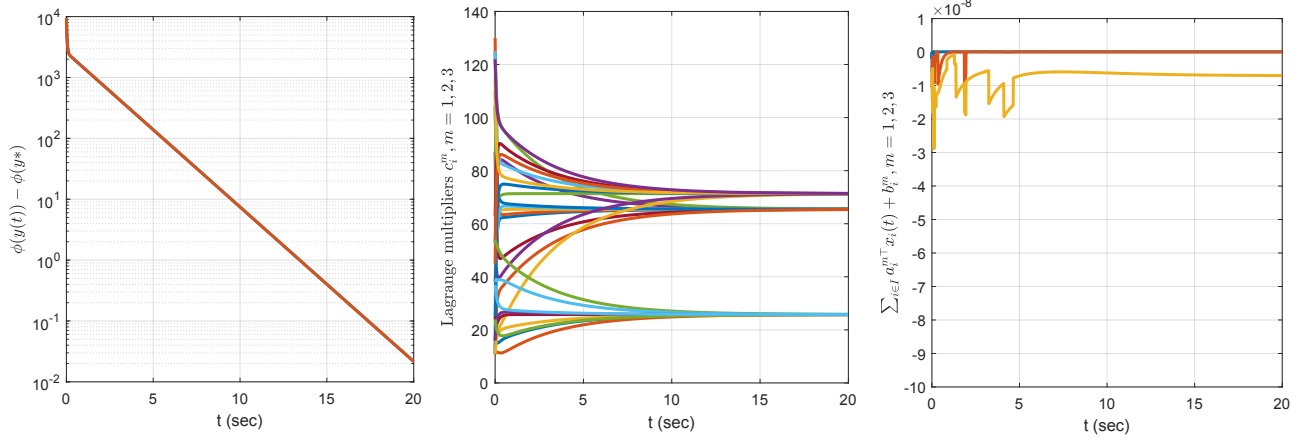
$$\begin{aligned} & \min_{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_9} \sum_{i \in \mathcal{I}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2 \\ \text{s. t. } & \sum_{i \in \mathcal{I}} [0.1 \ 0.1] \mathbf{x}_i - h_1(t, \mathbf{p}_1, \dots, \mathbf{p}_9) - 1 \leq 0, \\ & \sum_{i \in \mathcal{I}} [0.1 \ -0.1] \mathbf{x}_i - h_2(t, \mathbf{p}_1, \dots, \mathbf{p}_9) - 1 \leq 0. \end{aligned} \quad (41)$$

Here we choose the extended class \mathcal{K} function to be an identity function for simplicity. It is clear that (41) is a special case of (23) with 2 coupling inequality constraints. Denote the corresponding auxiliary variables by $\mathbf{y}^1, \mathbf{y}^2$. Based on previous analysis, the local quadratic program-based controller is given by

$$\begin{aligned} & \min_{\mathbf{x}_i} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2 \\ \text{s. t. } & [0.1 \ 0.1] \mathbf{x}_i + p_{i,1} + p_{i,2} + \mathbf{l}_i \mathbf{y}^1 - 31/9 \leq 0, \\ & [0.1 \ -0.1] \mathbf{x}_i + p_{i,1} - p_{i,2} + \mathbf{l}_i \mathbf{y}^2 - 11/9 \leq 0. \end{aligned} \quad (42)$$

where \mathbf{l}_i is the i -th row of the Laplacian matrix. These auxiliary variables are updated locally according to (26). In this case we know $A_i^\top = 0.1 \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is of full column rank and the LICQ of the local optimization problem is always fulfilled. Moreover, recall that $\mathbf{x}_{nom,i}(\mathbf{p}(t))$ and $\mathbf{y}^m(t), m = 1, 2$ are locally Lipschitz in time, we know the control input from this local optimization problem is also locally Lipschitz in time [39, Theorem 3.1]. Together with the violation-free property from Theorem 3, our proposed algorithm guarantees system safety.

In Fig. 5 we present the trajectories of four different cases: 1) the nominal case, where the controller $\mathbf{x}_{nom,i}$ is applied, 2) the centralized case, where we assume a central node exists that computes the safe controller from (41) and sends the control signal to each agent; 3) the distributed case, where our proposed safe controller in (42) is implemented; 4) the naive distributed case, where the controller from (42) is used with all $\mathbf{y}^1, \mathbf{y}^2$ zero. The last case is often used in CBF literature when system designers focus on guaranteeing safety and would sacrifice point-wise optimality.



(a) Time evolution of the total cost minus the optimal cost. (b) Time evolution of the local Lagrange multipliers corresponding to the three coupling constraints. (c) Time evolution of the value of the first Lagrange multiplier corresponding to the three coupling constraints.

Fig. 4: Numerical results involving 9 agents solving a static optimization problem.

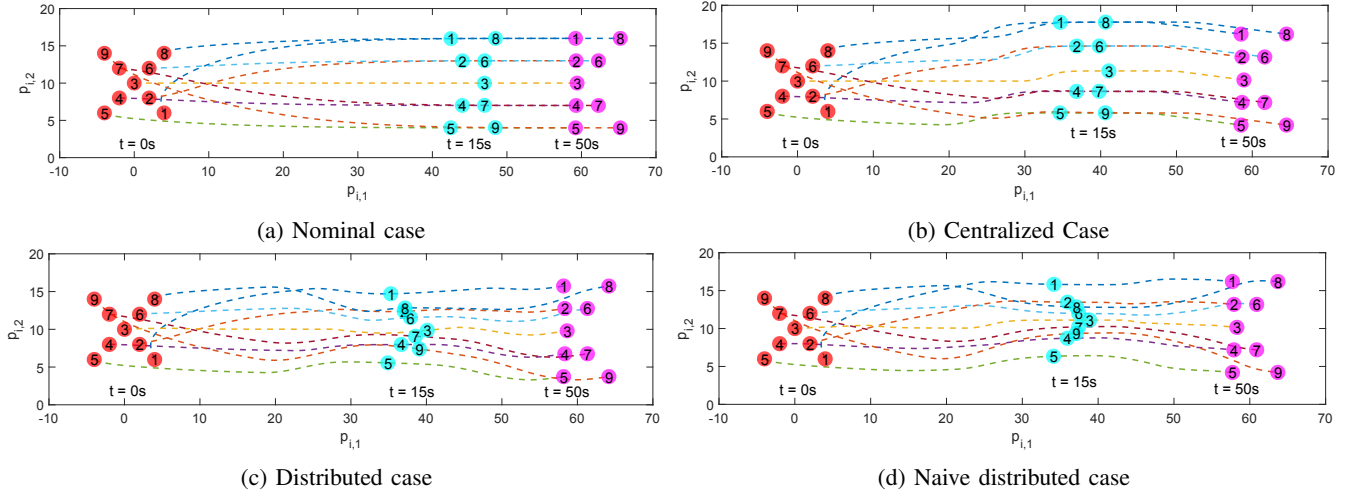


Fig. 5: Safe distributed control for coordinating 9 agents.

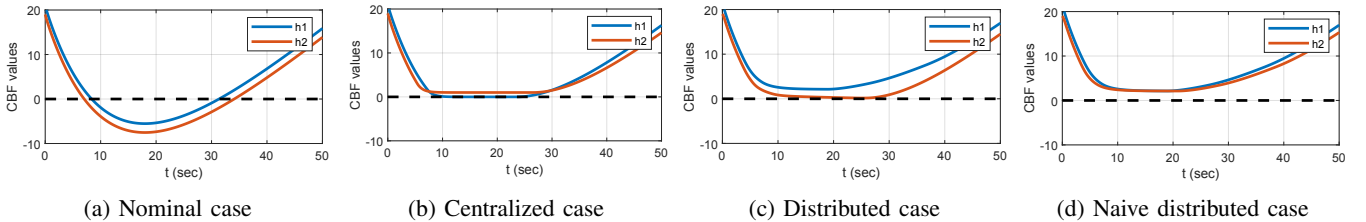


Fig. 6: Time evolution of the CBF values when coordinating 9 agents.

In the simulation, we choose $k_0 = 1$ and replace $\text{sign}(\cdot)$ with a saturation function $\text{sat}(v) = 1, \text{ if } v > 0.1; 10v, \text{ if } -0.1 \leq v \leq 0.1; -1, \text{ if } v < -0.1$ for eliminating the numerical chattering. One observes that the formation task is fulfilled by all four control schemes. In Fig. 6, it becomes clear that the two coupling state constraints are violated in the nominal case, and respected in the other three cases. Moreover, the distributed case with the proposed algorithm shows to be less conservative compared to the naive distributed scheme as the trajectory approaches closer to the safety boundary. To better quantify the performance of the proposed algorithm, we further compare the point-wise cost $\sum_{i \in \mathcal{I}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2$ in Fig. 7

along the state trajectory when our proposed algorithm is applied, i.e., the trajectory in Fig. 5(c). At any time t , one could view the curves corresponding to the naive distributed case and the centralized case in Fig. 7 as the initial cost and the optimal cost of the frozen-time optimization problem, respectively. Compared to the naive implementation, our proposed algorithm achieves safe distributed control with much less point-wise cost.

One key prerequisite that can restrict applicability of the proposed approach is Assumption 3. Even though we provide sufficient conditions for it to hold in dense and sparse cases, it can still be limiting. We will explore how to online monitor lo-

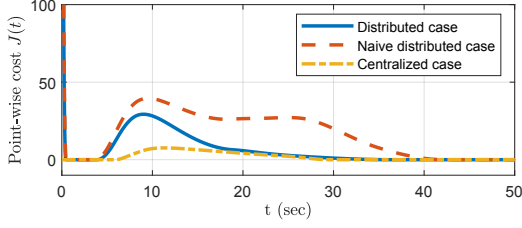


Fig. 7: Time evolution of the cost $\sum_{i \in \mathcal{I}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2$ along the state trajectory when the proposed algorithm is applied.

cal feasibility and adapt the local parameter y_i with feasibility guarantees in our future work.

IX. CONCLUSION

In this work, we propose a continuous-time violation-free, distributed algorithm for constraint-coupled optimization problems and demonstrate its applicability for control barrier function (CBF)-induced safe distributed control design. To decompose the global constraint coupled optimization problem into equivalent local ones, auxiliary variables are introduced, whose differences indicate the allocation of the constraints among all agents. Moreover, the subgradient information with respect to the auxiliary variables can be obtained using local information, which leads to a subgradient algorithm-like update law design with asymptotic optimality and all-time constraint satisfaction guarantee. For safe distributed control design, we further strengthen our algorithms to achieve finite-time convergence to cope with slowly time-varying parameters, leading to a safety-assured and point-wise optimal controller. Two numerical examples are provided to illustrate the favorable properties of the proposed algorithms: a static resource allocation optimization example and a multi-agent coordination example with two coupling state constraints.

APPENDIX

Proof of Proposition 1. To prove Point 1), given any feasible solution $(\mathbf{x}', \mathbf{y}')$ to (4), for an arbitrary m -th constraint in (3), we know $\mathbf{1}^\top (A^m \mathbf{x}' + L \mathbf{y}' + \mathbf{b}^m) = \sum_{i \in \mathcal{I}} \mathbf{a}_i^{m \top} \mathbf{x}' + b_i^m \leq 0$ thanks to the fact that $\mathbf{1}^\top L = \mathbf{0}$ for L being the Laplacian matrix of any connected and undirected graph \mathcal{G} [20]. Thus \mathbf{x}' also satisfies the m -th constraint in (3).

For Point 2), given any feasible solution \mathbf{x}' to (3), for an arbitrary m -th constraint in (3), define $\mathbf{v}^m = A^m \mathbf{x}' + \mathbf{b}^m$, $\mathbf{w}^m = \frac{\mathbf{1}_N^\top \mathbf{v}^m}{N} \mathbf{1}_N$. Thus, $\mathbf{w}^m \leq \mathbf{0}$ is obtained from the fact that \mathbf{w}^m has the same value in each entry and $\mathbf{1}_N^\top \mathbf{w}^m = \mathbf{1}_N^\top \mathbf{v}^m = \sum_{i \in \mathcal{I}} \mathbf{a}_i^{m \top} \mathbf{x}' + b_i^m \leq 0$. We further know that there exists a $\mathbf{y}^m \in \mathbb{R}^N$ such that $L \mathbf{y}^m + \mathbf{v}^m = \mathbf{w}^m$ thanks to the fact that $\text{Range}(L) = \{\mathbf{z} \in \mathbb{R}^N : \mathbf{1}_N^\top \mathbf{z} = 0\}$ for a connected undirected graph [20]. Thus, such a $(\mathbf{x}', \mathbf{y}^1, \dots, \mathbf{y}^M)$ is also feasible to the problem in (4).

Point 3) is obvious in view of the two problems. Based on the previous analysis, (3) and (4) share the same set of feasible solutions with respect to \mathbf{x} . Noting that they also share the same cost function, we conclude that the two problems are equivalent. \square

Proof of Proposition 7. For problem (24), we consider its Lagrangian

$$\mathcal{L}(\mathbf{x}_i, \mathbf{c}_i) = \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_{nom,i}\|^2 + \mathbf{c}_i^\top \left(A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \right)$$

where $\mathbf{c}_i = (c_i^1, c_i^2, \dots, c_i^M) \in \mathbb{R}^M$ is the Lagrange multiplier. Since Slater's condition holds and the problem (24) is a linearly constrained convex optimization problem, \mathbf{x}_i is an optimal solution if and only if \mathbf{c}_i exists and satisfies the following Karush–Kuhn–Tucker (KKT) conditions [40, Theorem 10.6]

$$\mathbf{x}_i - \mathbf{x}_{nom,i} + A_i \mathbf{c}_i = \mathbf{0} \quad (43a)$$

$$A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \leq \mathbf{0} \quad (43b)$$

$$\mathbf{c}_i \geq \mathbf{0} \quad (43c)$$

$$\mathbf{c}_i \circ \left(A_i^\top \mathbf{x}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \right) = \mathbf{0} \quad (43d)$$

where \circ denotes Hadamard Product, i.e., pointwise multiplication. To solve the KKT system, we obtain from (43a) that

$$\mathbf{x}_i = \mathbf{x}_{nom,i} - A_i \mathbf{c}_i = \mathbf{x}_{nom,i} - \sum_{m=1}^M c_i^m \mathbf{a}_i^m \quad (44)$$

Substituting it to (43b) and (43d), we know

$$A_i^\top \mathbf{x}_{nom,i} - A_i^\top A_i \mathbf{c}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \leq \mathbf{0} \quad (45a)$$

$$\mathbf{c}_i \circ \left(A_i^\top \mathbf{x}_{nom,i} - A_i^\top A_i \mathbf{c}_i + (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i \right) = \mathbf{0} \quad (45b)$$

Next, we consider two disjoint cases for the m -th coupled inequality. Before proceeding, we denote $\mathcal{C}_m := \{1, \dots, m-1, m+1, \dots, M\}$. We assume that the Lagrangian variables corresponding to the index set \mathcal{C}_m are known, i.e., $c_i^p, p \in \mathcal{C}_m$ are known. Define $\mathbf{x}_{nom,i}^{m'} = \mathbf{x}_{nom,i} - \sum_{p \in \mathcal{C}_m} c_i^p \mathbf{a}_i^p$, $\mathbf{b}_i' = (I_M \otimes \mathbf{l}_i) \mathbf{y} + \mathbf{b}_i$. Then $\mathbf{x}_i = \mathbf{x}_{nom,i}^{m'} - c_i^m \mathbf{a}_i^m$ from (44). Define $\Lambda_i = \text{Diag}(A_i^\top A_i) \in \mathbb{R}^{M \times M}$. Consider the following cases:

- 1) $[A_i^{m \top} \mathbf{x}_{nom,i}^{m'} + \mathbf{b}_i']_m \leq 0$. It is clear from (43d) that $c_i^m = 0$. (46)

Then $c_i^m = 0$ together with $c_i^p, p \in \mathcal{C}_m$ is the Lagrange multiplier.

- 2) $[A_i^{m \top} \mathbf{x}_{nom,i}^{m'} + \mathbf{b}_i']_m > 0$. Suppose $c_i^m = 0$ in this case, then $\mathbf{x}_i = \mathbf{x}_{nom,i}^{m'}$ yet it fails the constraint $A_i^{m \top} \mathbf{x}_i + \mathbf{b}_i' \leq \mathbf{0}$. By contradiction, we have $c_i^m > 0$. Therefore, to satisfy (43d), there must hold

$$[A_i^\top \mathbf{x}_{nom,i}^{m'} + \mathbf{b}_i' - c_i^m A_i^\top \mathbf{a}_i^m]_m = 0. \quad (47)$$

Thus,

$$\mathbf{a}_i^{m \top} \mathbf{a}_i^m c_i^m = [A_i^{m \top} \mathbf{x}_{nom,i}^{m'} + \mathbf{b}_i']_m \quad (48)$$

Recall $[\Lambda_i]_{m,m} = \mathbf{a}_i^{m \top} \mathbf{a}_i^m$, $\mathbf{x}_{nom,i}^{m'} = \mathbf{u}_{nom,i} - A_i \mathbf{c}_i + c_i^m \mathbf{a}_i^m$. Thus

$$[\Lambda_i]_{m,m} c_i^m = [A_i^\top (\mathbf{u}_{nom,i} - A_i \mathbf{c}_i) + \mathbf{b}_i' + \Lambda_i \mathbf{c}_i]_m \quad (49)$$

Summarizing the two cases in (46) and (49), we obtain $c_i^m, m \in \mathcal{M}$ satisfies

$$[\Lambda_i]_{m,m} c_i^m = \max \left([A_i^\top (\mathbf{x}_{nom,i} - A_i \mathbf{c}_i) + \mathbf{b}_i' + \Lambda_i \mathbf{c}_i]_m, 0 \right).$$

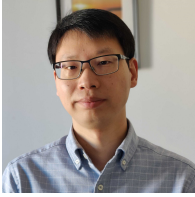
Stacking it together, we conclude that

$$\Lambda_i \mathbf{c}_i = \max \left(A_i^\top \mathbf{x}_{nom,i} + \mathbf{b}'_i + (\Lambda_i - A_i^\top A_i) \mathbf{c}_i, \mathbf{0} \right) \quad (50)$$

Up to now we have shown that if $(\mathbf{x}_i, \mathbf{c}_i)$ satisfies the KKT condition, then \mathbf{c}_i satisfies (50). In the following we show that whenever \mathbf{c}_i solves (50), the pair $(\mathbf{x}_i, \mathbf{c}_i)$ with $\mathbf{x}_i = \mathbf{x}_{nom,i} - A_i \mathbf{c}_i$ fulfills the KKT conditions. One verifies that (43a) and (43c) are trivially satisfied. What remains to show is (43b) and (43d). Denote by $c_i^p, p \in \mathcal{C}_m$ as part of the solution to (50). Now $c_i^p, p \in \mathcal{C}_m$ are not assumed to be the correct multipliers but will be proved so later. Recall $\mathbf{x}_{nom,i}^{m'} = \mathbf{x}_{nom,i} - \sum_{p \in \mathcal{C}_m} c_i^p \mathbf{a}_i^p$. If $[A_i^\top \mathbf{x}_{nom,i}^{m'} + \mathbf{b}'_i]_m \leq 0$, we know (50) dictates $c_i^m = 0$. If $[A_i^\top \mathbf{x}_{nom,i}^{m'} + \mathbf{b}'_i]_m > 0$, then (50) dictates $\mathbf{a}_i^{m\top} \mathbf{a}_i^m c_i^m = [A_i^\top \mathbf{x}_{nom,i}^{m'} + \mathbf{b}'_i]_m$. In both cases, the parts in (43b) and (43d) corresponding to the m -th constraint are fulfilled. This analysis holds for all $m \in \mathcal{M}$. This concludes the proof. \square

REFERENCES

- [1] I. Notarnicola and G. Notarstefano, "Constraint-coupled distributed optimization: a relaxation and duality approach," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 483–492, 2019.
- [2] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.
- [3] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [4] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.
- [5] A. Simonetto and H. Jamali-Rad, "Primal recovery from consensus-based dual decomposition for distributed convex optimization," *Journal of Optimization Theory and Applications*, vol. 168, pp. 172–197, 2016.
- [6] D. Mateos-Núñez and J. Cortés, "Distributed saddle-point subgradient algorithms with Laplacian averaging," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2720–2735, 2016.
- [7] X. Li, G. Feng, and L. Xie, "Distributed proximal algorithms for multiagent optimization with coupled inequality constraints," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1223–1230, 2020.
- [8] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757–1780, 2009.
- [9] Y. Su, Q. Wang, and C. Sun, "Distributed primal-dual method for convex optimization with coupled constraints," *IEEE Transactions on Signal Processing*, vol. 70, pp. 523–535, 2021.
- [10] A. Falsone and M. Prandini, "Augmented lagrangian tracking for distributed optimization with equality and inequality coupling constraints," *Automatica*, vol. 157, p. 111269, 2023.
- [11] L. S. Lasdon, *Optimization theory for large systems*. Courier Corporation, 2002.
- [12] H. Wang, A. Papachristodoulou, and K. Margellos, "Distributed control design and safety verification for multi-agent systems," *arXiv preprint arXiv:2303.12610*, 2023.
- [13] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3310–3315, 2015.
- [14] X. Wu, S. Magnússon, and M. Johansson, "Distributed safe resource allocation using barrier functions," *Automatica*, vol. 153, p. 111051, 2023.
- [15] H. Abeynanda, C. Weeraddana, G. Lanel, and C. Fischione, "On the primal feasibility in dual decomposition methods under additive and bounded errors," *IEEE Transactions on Signal Processing*, vol. 71, pp. 655–669, 2023.
- [16] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2021.
- [17] P. Mestres and J. Cortes, "Distributed and anytime algorithm for network optimization problems with separable structure," in *2023 IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 5457–5462.
- [18] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 226–237, 2015.
- [19] C. Liu, X. Tan, X. Wu, D. V. Dimarogonas, and K. H. Johansson, "Achieving violation-free distributed optimization under coupling constraints," *arXiv preprint arXiv:2404.07609*, 2024.
- [20] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [21] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth analysis and control theory*. Springer Science & Business Media, 2008, vol. 178.
- [22] D. P. Bertsekas, *Nonlinear programming*. Belmont: Athena Scientific., 1995.
- [23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [24] J. Cortes, "Discontinuous dynamical systems," *IEEE Control systems magazine*, vol. 28, no. 3, pp. 36–73, 2008.
- [25] S. M. Robinson, "Strongly regular generalized equations," *Mathematics of Operations Research*, vol. 5, no. 1, pp. 43–62, 1980.
- [26] P. Mestres, A. Allibhoy, and J. Cortés, "Robinson's counterexample and regularity properties of optimization-based controllers," *arXiv preprint arXiv:2311.13167*, 2023.
- [27] M. Florenzano and C. Le Van, *Finite dimensional convexity and optimization*. Springer Science & Business Media, 2001, vol. 13.
- [28] G. Wachsmuth, "On LICQ and the uniqueness of Lagrange multipliers," *Operations Research Letters*, vol. 41, no. 1, pp. 78–80, 2013.
- [29] L. Wang, A. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [30] V. N. Fernandez-Ayala, X. Tan, and D. V. Dimarogonas, "Distributed barrier function-enabled human-in-the-loop control for multi-robot systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7706–7712.
- [31] P. Mestres, C. Nieto-Granda, and J. Cortés, "Distributed safe navigation of multi-agent systems using control barrier function-based controllers," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6760 – 6767, 2024.
- [32] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. European Control Conf.*, 2019, pp. 3420–3431.
- [33] X. Tan and D. V. Dimarogonas, "Compatibility checking of multiple control barrier functions for input constrained systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 939–944.
- [34] M. Santilli, G. Oliva, and A. Gasparri, "Distributed finite-time algorithm for a class of quadratic optimization problems with time-varying linear constraints," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 4380–4386.
- [35] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, 2006.
- [36] D. Liberzon, *Switching in systems and control*. Springer, 2003, vol. 190.
- [37] M. Franceschelli, A. Pisano, A. Giua, and E. Usai, "Finite-time consensus with disturbance rejection by discontinuous local interactions in directed graphs," *IEEE transactions on Automatic Control*, vol. 60, no. 4, pp. 1133–1138, 2014.
- [38] 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.
- [39] W. W. Hager, "Lipschitz continuity for constrained processes," *SIAM Journal on Control and Optimization*, vol. 17, no. 3, pp. 321–338, 1979.
- [40] A. Beck, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 2014.



Xiao Tan (Member, IEEE) received his B.Eng. and M.Sc. degrees from Beihang University, China, in 2015 and 2018, respectively, and his Ph.D. degree in Electrical Engineering from KTH Royal Institute of Technology, Sweden, in 2023. He is currently a post-doctoral research associate at Amber Lab, California Institute of Technology, USA. His research interests lie in the areas of correct, safe, and secure control design for dynamical systems, with applications to autonomous and multi-robot systems.



Dimos V. Dimarogonas (Fellow Member, IEEE) was born in Athens, Greece, in 1978. He received the Diploma degree in electrical and computer engineering and the Ph.D. degree in mechanical engineering from the National Technical University of Athens, Athens, Greece, in 2001 and 2007, respectively. Between 2007 and 2010, he held Postdoctoral positions with the Department of Automatic Control, KTH Royal Institute of Technology and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology. His current research interests include multi-agent systems, hybrid systems and control, robot navigation and manipulation, human-robot interaction, and networked control. Prof. Dimarogonas serves as an Associate Editor of *Automatica* and a Senior Editor of *IEEE Transactions on Control of Network Systems*. He was a recipient of the ERC starting Grant in 2014, the ERC Consolidator Grant in 2019, and the Knut och Alice Wallenberg Academy Fellowship in 2015.



Changxin Liu (Member, IEEE) received his Ph.D. in mechanical engineering from the University of Victoria, Victoria, BC, Canada, in 2021. From 2021 to 2024, he worked as a postdoctoral researcher at the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. In October 2024, he joined the Department of Automation at East China University of Science and Technology, Shanghai, China, where he is currently a Professor. His research interests include distributed optimization and control, machine

learning, and combinatorial optimization, with applications to the process industry.



Karl H. Johansson (Fellow Member, IEEE) is Swedish Research Council Distinguished Professor in Electrical Engineering and Computer Science at KTH Royal Institute of Technology in Sweden and Founding Director of Digital Futures. He earned his MSc degree in Electrical Engineering and PhD in Automatic Control from Lund University. He has held visiting positions at UC Berkeley, Caltech, NTU and other prestigious institutions. His research interests focus on networked control systems and cyber-physical systems with applications in transportation,

energy, and automation networks. For his scientific contributions, he has received numerous best paper awards and various other distinctions from IEEE, IFAC, and other organizations. He has been awarded Distinguished Professor by the Swedish Research Council, Wallenberg Scholar by the Knut and Alice Wallenberg Foundation, Future Research Leader by the Swedish Foundation for Strategic Research. He has also received the triennial IFAC Young Author Prize, IEEE CSS Distinguished Lecturer, IFAC Outstanding Service Award, and IEEE CSS Hendrik W. Bode Lecture Prize. His extensive service to the academic community includes being President of the European Control Association, IEEE CSS Vice President Diversity, Outreach & Development, and Member of IEEE CSS Board of Governors and IFAC Council. He has served on the editorial boards of *Automatica*, *IEEE TAC*, *IEEE TCNS* and many other journals. He has also been a member of the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He is Fellow of both the IEEE and the Royal Swedish Academy of Engineering Sciences.