2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed and Anytime Algorithm for Network Optimization Problems with Separable Structure

Pol Mestres Jorge Cortés

Abstract— **This paper considers the problem of designing a dynamical system to solve constrained optimization problems in a distributed way and in an anytime fashion (i.e., such that the feasible set is forward invariant). For problems with separable objective function and constraints, we design an algorithm with the desired properties and establish its convergence. Simulations illustrate our results.**

I. Introduction

Distributed optimization methods are a popular tool for solving several engineering problems like parameter estimation, resource allocation in communication networks, source localization, etc. In several of these applications, feedback controllers are often implemented as the solution of such problems on a physical plant. In this context, the safe operation requirements of the system are encoded as constraints of the optimization problem. This approach is very versatile, but the implementation of such controllers faces several challenges. On the one hand, the algorithm solving the optimization problem may be terminated at any time, and hence feasibility must be maintained throughout its execution. We refer to an algorithm with this property as anytime. Moreover, it must retain its distributed and scalable character, so that each agent can implement it by communicating exclusively with its neighbors and can do so in an efficient manner independently of the size of the network. Designing algorithms that combine all these objectives together and are anytime, distributed, scalable and have provable convergence guarantees is a challenging problem.

Literature Review: In this work we take the viewpoint of optimization algorithms as continuous-time dynamical systems (cf. [1], [2]), which has recently proven to be a very powerful paradigm in applications where the optimization problem is in a feedback loop with a plant [3]. The problem of designing distributed algorithms for constrained optimization is well studied in the literature. The survey paper [4] covers this topic exhaustively. Of particular interest to us are primal-dual and projected saddle-point dynamics [5], which define dynamical systems that solve constrained optimization problems. Although these works have provable convergence guarantees, the trajectories generated by the proposed dynamical systems are in general not guaranteed to be feasible throughout its execution. This anytime property is well-studied in the optimization literature and dates back to the mid-1980s (cf. [6]) in the context of time-dependent planning. Our work here is related to [7]–[9], which design dynamical systems that solve nonlinear programs in continuous time with the anytime property. However, the literature on anytime algorithms generally is not concerned in making them amenable to a distributed implementation. An exception is the relaxed economic dispatch problem, which involves a convex separable objective function and globally coupling affine equality constraints. For this problem, [10] gives a distributed anytime algorithm that converges to the global optimum. The problem of designing distributed anytime algorithms for constrained optimization problems is very relevant in the control barrier function (CBF) literature [11], [12]. This is because CBF-based controllers for multiagent systems can be obtained as the solution of a network optimization problem, where the system is guaranteed to be safe only if its constraints are satisfied at all times. The works [13]–[15] tackle this problem for CBF-based quadratic programs (QPs), where a centralized QP is split into local QPs whose solution is guaranteed to preserve the safety constraints at all times. However, the solution of these local QPs might be suboptimal with respect to the centralized QP. The recent work [16] designs a distributed algorithm that is guaranteed to satisfy the constraints of the CBF-based QP at all times and converge to its state-dependent optimizer in finite time. However, their algorithm is restricted to a limited class of quadratic programs and plant dynamics and is not easily generalizable to general convex programs.

Statement of Contributions: In this paper we introduce a continuous-time dynamical system to solve convex optimization problems with separable objective function and constraints in a distributed and anytime fashion. The constraints couple the decision variables of all agents and this poses a difficulty in the design of distributed algorithms that solve such problems. We first show that the separable structure permits the intoduction of auxiliary variables to reformulate the original problem into one with local constraints while still preserving the same solution set. However, this reformulation still does not allow to fully decouple the optimization problem into one per agent because the auxiliary variables require coordination. In order to sort this hurdle, our technical approach constructs a dynamical system by combining the use of projected saddle-point dynamics, which are distributed but not anytime, and the safe gradient flow, which is anytime but not distributed. First, we establish the well-posedness of the proposed dynamical system. Second, we show that it is distributed, exhibits the anytime property and is scalable.

This work was partially supported by ARL award ARL-W911NF-22-2-0231.

P. Mestres and J. Cortés are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, {pomestre,cortes}@ucsd.edu

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

Finally, we prove that all trajectories with feasible initial condition converge to a neighborhood of the optimizer, which can be made arbitrarily small by tuning a design parameter accordingly. Moreover, in the case where the feasible set is bounded, we show that all trajectories with feasible initial condition exactly converge to the optimizer. For reasons of space, proofs are omitted and will appear elsewhere.

II. Preliminaries

This section presents background on dynamical systems that solve constrained optimization problems[1].

Safe Gradient Flow: Given continuously differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^p$, $h : \mathbb{R}^n \to \mathbb{R}^q$, consider the constrained nonlinear program

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x), \qquad (1)\\ &\text{s.t.} \quad g(x) \leq 0,\\ &\qquad\quad h(x) = 0. \end{aligned}$$

Let $\mathcal{F} = \{x \in \mathbb{R}^n \ : \ g(x) \leq 0, \ h(x) = 0\}$ denote the constraint set and $X_{\text{KKT}} = \{x^* \in \mathbb{R}^n \ : \ \exists (u^*, v^*) \in \mathbb{R}^p \times \mathbb{R}^q$ such that (x^*, u^*, v^*) is a KKT point of (1)$\}$. We are interested in solving the optimization problem with an algorithm that respects its constraints at all times of its evolution. Given $\alpha > 0$, the *safe gradient flow*, cf. [9], is the dynamical system $\dot{x} = F_\alpha(x)$, where

$$\begin{aligned} F_\alpha(x) = \arg\min_{\xi \in \mathbb{R}^n} &\frac{1}{2} \left\| \xi + \nabla f(x) \right\|^2 \\ \text{s.t.} \quad &\frac{\partial g(x)}{\partial x} \xi \leq -\alpha g(x), \qquad (2)\\ &\frac{\partial h(x)}{\partial x} \xi = -\alpha h(x). \end{aligned}$$

The direction prescribed by F_α can be interpreted as that closest to the gradient descent direction $-\nabla f$ while ensuring that the constraints defining $\mathcal{F}$ are not violated. The next result gathers important properties of the safe gradient flow.

Proposition 2.1: (Properties of the safe gradient flow [9, Proposition 5.1, Proposition 5.6 and Corollary 5.9]): Suppose f, g and h are continuously differentiable and their derivatives are locally Lipschitz. Then,

(i) there exists an open neighborhood U containing $\mathcal{F}$ such that (2) is well-defined;
(ii) F_α is locally Lipschitz on U;
(iii) the Lagrange multipliers of (2) are unique and locally Lipschitz as a function of x on U;
(iv) the feasible set $\mathcal{F}$ is forward invariant and asymptotically stable under (2);
(v) $F_\alpha(x^*) = 0$ if and only if $x^* \in X_{\text{KKT}}$;
(vi) if x^* is a strict local minimizer of f and an isolated equilibrium of the safe gradient flow, then x^* is asymptotically stable relative to $\mathcal{F}$.

Projected Saddle-Point Dynamics: We recall here the notion of projected saddle-point dynamics following [17]. Consider again the optimization problem (1), with continuously differentiable functions f, g and h whose derivatives are locally Lipschitz, and let $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m_{\geq 0} \times \mathbb{R}^p$ be the associated Lagrangian,

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x).$$

We define the *projected saddle-point dynamics* for $\mathcal{L}$ as:

$$\begin{aligned} \dot{x} &= -\nabla_x \mathcal{L}(x, \lambda, \mu), &&(3a)\\ \dot{\lambda} &= [\nabla_\lambda \mathcal{L}(x, \lambda, \mu)]^+_\lambda, &&(3b)\\ \dot{\mu} &= \nabla_\mu \mathcal{L}(x, \lambda, \mu). &&(3c) \end{aligned}$$

If f is strongly convex, g is convex and h is affine, the saddle point of $\mathcal{L}$ is unique and corresponds to the KKT point of (1). Moreover, [17, Theorem 5.1] ensures that the saddle point of $\mathcal{L}$ is globally asymptotically stable under the dynamics (3).

III. Problem Statement

Consider a network composed by agents $\{1, \ldots, N\}$ whose communication topology is described by a connected undirected graph $\mathcal{G}$. An edge (i, j) represents the fact that agent i can receive information from agent j and vice versa. We refer to an algorithm run by the network as *distributed* if each agent can execute it with the information available to it and its neighbors.

For each $i \in \{1, \ldots, N\}$, $k \in \{1, \ldots, p\}$, $l \in \{1, \ldots, q\}$ let $f_i : \mathbb{R}^n \to \mathbb{R}$ be a strongly convex and continuously differentiable function with locally Lipschitz derivatives, $g_i^k : \mathbb{R}^n \to \mathbb{R}$ be a convex and continuously differentiable function with locally Lipschitz derivatives and $h_i^l : \mathbb{R}^n \to \mathbb{R}$ be an affine function. We let $x = [x_1, \ldots, x_N] \in \mathbb{R}^{nN}$. Consider

[1] Throughout the paper we denote by $\mathbb{R}$ and $\mathbb{R}_{>0}$ the set of real and nonnegative real numbers, respectively. Given $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. For $a \in \mathbb{R}$ and $b \in \mathbb{R}_{\geq 0}$, we let

$$[a]^+_b = \begin{cases} a, & \text{if } b > 0,\\ \max\{0, a\}, & \text{if } b = 0. \end{cases}$$

For vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n_{\geq 0}$, $[a]^+_b$ denotes the vector whose i-th component is $[a_i]^+_{b_i}$, for $i \in \{1, \ldots, n\}$. We also write $\mathbf{0}_n = (0, \ldots, 0) \in \mathbb{R}^n$. For a real-valued function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, we denote by $\nabla_x F$ and $\nabla_y F$ the column vector of partial derivatives of F with respect to the first and second arguments, respectively. Given a set of functions $g^1, \ldots, g^k$, we let $I_{g^1, \ldots, g^k}(x) = \{1 \leq i \leq k \ : \ g^i(x) = 0\}$. For matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{p \times q}$, we let $A \otimes B$ denote their Kronecker product. Given a set $\mathcal{P} \subseteq \mathbb{R}^n$ and a set of variables $\xi = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$, we denote by $\Pi_\xi \mathcal{P} = \{(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \in \mathbb{R}^k \ : \ x \in \mathcal{P}\}$ the projection of $\mathcal{P}$ onto the ξ variables. Given sets $S_1, \ldots, S_k$, $\times_{i=1}^k S_i$ denotes their Cartesian product. An undirected graph is a pair $\mathcal{G} = (V, \mathcal{E})$, where $V = \{1, \ldots, N\}$ is a finite set called the vertex set, $\mathcal{E} \subseteq V \times V$ is called the edge set where $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j \in \mathcal{I} \ : \ (i, j) \in \mathcal{E}\}$. The adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}_{\geq 0}$ of the graph $\mathcal{G}$ satisfies the property $[A]_{i,j} = [A]_{j,i} = 1$ if $(i, j) \in \mathcal{E}$ and $[A]_{i,j} = 0$ otherwise. The degree matrix D of $\mathcal{G}$ is the diagonal matrix defined by $[D]_{i,i} = |\mathcal{N}_i|$ for all $i \in \{1, \ldots, |V|\}$. The Laplacian matrix of $\mathcal{G}$ is $L = D - A$. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitz vector field and consider the dynamical system $\dot{x} = F(x)$, with flow map $\Phi_t : \mathbb{R}^n \to \mathbb{R}^n$. This means that $\Phi_t(x) = x(t)$, where $x(t)$ is the unique solution of the dynamical system with $x(0) = x$. A set $\mathcal{K} \subseteq \mathbb{R}^n$ is (positively) forward invariant if $x \in \mathcal{K}$ implies that $\Phi_t(x) \in \mathcal{K}$ for all $t \geq 0$. A set $A \subseteq \mathcal{K}$ is Lyapunov stable relative to $\mathcal{K}$ if, for every open set U containing A, there exists an open set $\tilde{U}$ also containing A such that for all $x \in \tilde{U} \cap \mathcal{K}$, $\Phi_t(x) \in U \cap \mathcal{K}$ for all $t \geq 0$. A set $A \subseteq \mathcal{K}$ is asymptotically stable relative to $\mathcal{K}$ if it is Lyapunov stable relative to $\mathcal{K}$ and there is an open set U containing A such that $\Phi_t(x) \to A$ as $t \to \infty$ for all $x \in U \cap \mathcal{K}$.

the following optimization problem with separable objective function and constraints:

$$\begin{aligned}
&\min_{x\in\mathbb{R}^{nN}} \sum_{i=1}^{N} f_i(x_i), && (4)\\
&\text{s.t.} \quad \sum_{i=1}^{N} g_i^k(x_i) \le 0, \quad k\in\{1,\dots,p\},\\
&\qquad\;\; \sum_{i=1}^{N} h_i^l(x_i) = 0, \quad l\in\{1,\dots,q\}.
\end{aligned}$$

Since the objective function is strongly convex and the feasible set is convex, this program has a unique optimizer x^*. Note that, even though the objective function is separable, the structure of the constraints couples the decision variables of the agents. This makes challenging the design of distributed algorithmic solutions of (4).

Remark 3.1: (Separability structure): Problems of the form (4) arise in multiple applications, including communications [18], economic dispatch of power systems [10], optimal power flow [19], resource allocation [20], and safe swarm behavior using control barrier functions [13]. Also, given convex sets X_i, $i\in\{1,\dots,N\}$, a common problem considered in the distributed optimization literature [21] is

$$\begin{aligned}
&\min_{x\in\mathbb{R}^{n}} \sum_{i=1}^{N} f_i(x),\\
&\text{s.t.} \quad x\in\cap_{i=1}^{N} X_i.
\end{aligned}$$

When $X_i = \{x\in\mathbb{R}^n \;:\; \bar{g}_i(x)\le 0\}\subseteq\mathbb{R}^n$ for a continuously differentiable convex function with locally Lipschitz derivatives $\bar{g}_i:\mathbb{R}^n\to\mathbb{R}^{m_i}$, for $i\in\{1,\dots,N\}$, the optimization can be reformulated as

$$\begin{aligned}
&\min_{x\in\mathbb{R}^{nN}} \sum_{i=1}^{N} f_i(x_i),\\
&\text{s.t.} \quad (L\otimes I_n)x = \mathbf{0}_{Nn} \quad \bar{g}_i(x_i)\le 0,\; i\in\{1,\dots,N\},
\end{aligned}$$

which is of the form (4). •

Throughout the paper, we denote $f(x)=\sum_{i=1}^N f_i(x_i)$, $g^k(x)=\sum_{i=1}^N g_i^k(x_i)$ for $k\in\{1,\dots,p\}$ and $h^l(x)=\sum_{i=1}^N h_i^l(x_i)$ for $l\in\{1,\dots,q\}$, and write the feasible set of (4) as

$$\begin{aligned}
\mathcal{F} = \{x\in\mathbb{R}^{nN} \;:\; & g^k(x)\le 0,\ \forall k\in\{1,\dots,p\},\\
& h^l(x)=0,\ \forall l\in\{1,\dots,q\}\}.
\end{aligned}$$

We also make the following assumption.

Assumption 1: (Linear independence constraint qualification for separable constraints): For all $x\in\mathbb{R}^{nN}$, the vectors $\{\nabla g^k(x)\}_{k\in I_{g^1,\dots,g^p}(x)}\cup\{\nabla h^l(x)\}_{1\le l\le q}$ are linearly independent.

Assumption 1 is common and guarantees that the KKT conditions are necessary and sufficient for the optimality of (4).

Our goal is to design an algorithm, in the form of a locally Lipschitz dynamical system, such that

(i) is *distributed*, i.e., each agent can execute it with locally available information;

(ii) is *anytime*, i.e., the feasible set $\mathcal{F}$ is forward invariant;

(iii) *solves* (4), i.e., all trajectories starting in $\mathcal{F}$ converge to its optimizer.

Even though algorithmic solutions exist in the literature that enjoy some of these properties (e.g., the projected saddle-point dynamics enjoys (i) and (iii) for certain classes of optimization problems), the design of an algorithm that enjoys all three is challenging.

IV. Design of Algorithmic Solution

Here we propose an algorithmic solution to the constrained program (4) to meet the requirements stated in Section III. Our exposition proceeds by first reformulating the optimization problem and then building on the projected saddle-point dynamics and the safe gradient flow to synthesize a coordination algorithm with the desired properties.

A. Reformulation using constraint mismatch variables

In this section we provide an equivalent formulation of (4) that addresses the coupling among the agents' decision variables arising from the structure of the constraints. The basic idea to "decouple" them is to introduce, following [22], *constraint-mismatch variables* which help agents keep track of local constraints while collectively satisfying the original constraints. Formally, to the state of each agent, we add one variable per constraint: y_i^k for agent i and the kth inequality constraint and z_j^l for agent j and the lth equality constraint. For convenience, we use the notation $x=[x_1,\dots,x_N]$, $y_i=[y_i^1,\dots,y_i^p]$, $z_i=[z_i^1,\dots,z_i^p]$, $y=[y_1,\dots,y_N]$, $z=[z_1,\dots,z_N]$. Consider then the following problem

$$\begin{aligned}
&\min_{x\in\mathbb{R}^{nN},y\in\mathbb{R}^{Np},z\in\mathbb{R}^{Nq}} \sum_{i=1}^{N} f_i(x_i), && (5)\\
&\text{s.t.} \quad g_i^k(x_i)+\sum_{j\in\mathcal{N}_i}(y_i^k-y_j^k)\le 0, \quad k\in\{1,\dots,p\}\\
&\qquad\;\; h_i^l(x_i)+\sum_{j\in\mathcal{N}_i}(z_i^k-z_j^k)=0, \quad l\in\{1,\dots,q\},
\end{aligned}$$

and constraints for all $i\in\{1,\dots,N\}$. Note that, in this formulation, constraints are now locally expressible, meaning that agent $i\in\{1,\dots,N\}$ can evaluate the ones corresponding to g_i^k and h_i^l with information from its neighbors. Let $\mu_i=[\mu_i^1,\dots,\mu_i^p]$, $\lambda_i=[\lambda_i^1,\dots,\lambda_i^p]$, $\lambda=[\lambda_1,\dots,\lambda_N]$ and $\mu=[\mu_1,\dots,\mu_N]$ be the Lagrange multipliers for the constraints in (5). We next show that the optimizer in x of (5) is x^*, the optimizer of (4).

Proposition 4.1: (Equivalence between the two formulations): Let $\mathcal{F}_r^*$ be the solution set of (5). Then, $x^*=\Pi_x(\mathcal{F}_r^*)$.

The proof of Proposition 4.1 introduces a set of slack variables and follows a similar argument to that of [22].

Proposition 4.1 implies that (5) has a unique optimizer in the variables x. However, since the objective function in (5) is not strongly convex in y and z, the optimizer in the variables y and z might not be unique. Hence, for the results that follow, we take $\epsilon>0$ and define $f_i^\epsilon(x_i,y_i,z_i)=f_i(x_i)+\frac{\epsilon}{2}\|y_i\|^2+\frac{\epsilon}{2}\|z_i\|^2$, $f^\epsilon(x,y,z)=\sum_{i=1}^N f_i^\epsilon(x_i,y_i,z_i)$. Consider the following regularized version of (5),

$$\min_{x\in\mathbb{R}^{Nn},y\in\mathbb{R}^{Np},z\in\mathbb{R}^{Nq}} \sum_{i=1}^{N} f_i^\epsilon(x_i,y_i,z_i), \qquad (6)$$

$$
\begin{aligned}
\text{s.t.}\quad & g_i^k(x_i) + \sum_{j\in\mathcal{N}_i} (y_i^k - y_j^k) \le 0, \quad k \in \{1,\dots,p\},\\
& h_i^l(x_i) + \sum_{j\in\mathcal{N}_i} (z_i^k - z_j^k) = 0, \quad l \in \{1,\dots,q\},
\end{aligned}
$$

and with constraints for all $i \in \{1,\dots,N\}$. Let $(x^{*,\epsilon}, y^{*,\epsilon}, z^{*,\epsilon}) \in \mathbb{R}^{Nn} \times \mathbb{R}^{Np} \times \mathbb{R}^{Nq}$ be the optimizer of (6), which is unique because the objective function is strongly convex and the feasible set is convex. Next we establish a sensitivity result for the regularized problem (6).

Lemma 4.2: (Sensitivity of regularized problem): Given $\delta > 0$, there exists $\bar{\epsilon} > 0$ so that if $\epsilon < \bar{\epsilon}$, then $\|x^{*,\epsilon} - x^*\| < \delta$.

Given Lemma 4.2, in what follows we focus on solving (6) and assume that ϵ is taken sufficiently small to guarantee a desired maximum distance between $x^{*,\epsilon}$ and x^*.

B. Cascade of saddle-point dynamics and safe gradient flow

In this section, we build on the reformulation presented above to design our proposed algorithmic solution. Note that, if we had knowledge of the optimizers $y^{*,\epsilon}, z^{*,\epsilon}$ of Problem (6), we could break the optimization into N, one per agent $i \in \{1,\dots,N\}$, decoupled optimization problems as follows,

$$
\begin{aligned}
&\min_{x_i\in\mathbb{R}^n} f_i^\epsilon(x_i), && (7)\\
\text{s.t.}\quad & g_i^k(x_i) + \sum_{j\in\mathcal{N}_i} ((y_i^k)^{*,\epsilon} - (y_j^k)^{*,\epsilon}) \le 0, \quad k \in \{1,\dots,p\},\\
& h_i^l(x_i) + \sum_{j\in\mathcal{N}_i} ((z_i^k)^{*,\epsilon} - (z_j^k)^{*,\epsilon}) = 0, \quad l \in \{1,\dots,q\}.
\end{aligned}
$$

In turn, each of these problems could be solved in an anytime fashion by having each agent execute the corresponding safe gradient flow, cf. Proposition 2.1. However, since $y^{*,\epsilon}$ and $z^{*,\epsilon}$ are not readily available, agents need to interact with their neighbors to compute them. Since this would require an iterative algorithm, this means agents will face evolving y and z in the corresponding formulation of (7), which raises the additional challenge of ensuring the anytime nature of the safe gradient flow is preserved. We tackle these challenges next.

To generate the update law for y and z, we propose to use the projected saddle-point dynamics of (6). By [17, Theorem 5.1], these are guaranteed to converge to its optimizers. Simultaneously, we implement the safe gradient flow of (7) with the current values of y and z, i.e., (with the notation $y_{\mathcal{N}_i} = y_i \cup \{y_j\}_{j\in\mathcal{N}_i}$, $z_{\mathcal{N}_i} = z_i \cup \{z_j\}_{j\in\mathcal{N}_i}$):

$$
\begin{aligned}
&S_\alpha^i(x_i, y_{\mathcal{N}_i}, z_{\mathcal{N}_i}) =\\
&\arg\min_{\xi_i\in\mathbb{R}^n} \frac{1}{2}\|\xi_i + \nabla f_i(x_i)\|^2, && (8)\\
\text{s.t.}\ & \nabla g_i^k(x_i)\xi_i \le -\alpha(g_i^k(x_i) + \sum_{j\in\mathcal{N}_i}(y_i^k - y_j^k)),\ k \in \{1,\dots,p\},\\
& \nabla h_i^l(x_i)\xi_i = -\alpha(h_i^l(x_i) + \sum_{j\in\mathcal{N}_i}(z_i^l - z_j^l)),\ l \in \{1,\dots,q\},
\end{aligned}
$$

for all $i \in \{1,\dots,N\}$. We denote $S_\alpha(x,y,z) = [S_\alpha^1(x_1, y_{\mathcal{N}_1}, z_{\mathcal{N}_1}), \dots, S_\alpha^N(x_N, y_{\mathcal{N}_N}, z_{\mathcal{N}_N})]$. To add more flexibility to our design, we add a timescale separation parameter $\tau > 0$ that allows the projected saddle-point dynamics to be run at a faster rate relative to the safe gradient flow. This leads to the cascaded dynamical system:

$$
\begin{aligned}
\tau\dot{v}_i &= -\nabla f_i(v_i) - \sum_{k=1}^p \lambda_i^k \nabla g_i^k(v_i) - \sum_{l=1}^q \mu_i^l h_i^l(v_i), && \text{(9a)}\\
\tau\dot{y}_i^k &= -\epsilon y_i^k - \sum_{j\in\mathcal{N}_i}(\lambda_i^k - \lambda_j^k), && \text{(9b)}\\
\tau\dot{z}_i^l &= -\epsilon z_i^l - \sum_{j\in\mathcal{N}_i}(\mu_i^l - \mu_j^l), && \text{(9c)}\\
\tau\dot{\lambda}_i^k &= [g_i^k(v_i) + \sum_{j\in\mathcal{N}_i}(y_i^k - y_j^k)]^+_{\lambda_i^k}, && \text{(9d)}\\
\tau\dot{\mu}_i^l &= h_i^l(v_i) + \sum_{j\in\mathcal{N}_i}(z_i^k - z_j^k), && \text{(9e)}\\
\dot{x}_i &= S_\alpha^i(x_i, y_{\mathcal{N}_i}, z_{\mathcal{N}_i}), && \text{(9f)}
\end{aligned}
$$

for $i \in \{1,\dots,N\}$, $k \in \{1,\dots,p\}$ and $l \in \{1,\dots,q\}$, where v_i are *virtual* variables that play the role of x_i in the projected saddle-point dynamics. Since (9) results from the cascaded interconnection of saddle-point dynamics and the safe gradient flow, we refer to it as SP-SGF.

Remark 4.3: (Scalability and distributed character of SP-SGF): In algorithm (9), each agent has a state variable of dimension $2n + 2p + 2q$. To compute the evolution of these state variables, each agent only requires information provided by its neighbors in $\mathcal{G}$. Therefore, the algorithm is distributed. In addition, since the memory needed by each agent to run (9) remains constant as the network size N increases, the algorithm is also scalable. •

Remark 4.4: (Algorithm implementation): Note that the execution of SP-SGF requires solving the optimization problem (8), for each $i \in \{1,\dots,N\}$, which is a quadratic program and hence can be solved efficiently. In fact, if the number of constraints is low, closed-form expressions for its solution [23, Theorem 1] are available. •

In what follows, we assume that for all $i \in \{1,\dots,N\}$, the set of initial conditions for v_i, y_i, z_i, λ_i and μ_i in (9) lie in compact sets $\mathcal{V}_i$, $\mathcal{Y}_i$, $\mathcal{Z}_i$, Λ_i and M_i respectively. This means that the initial conditions v, y, z, λ, μ lie in compact sets $\mathcal{V} := \times_{i=1}^N \mathcal{V}_i$, $\mathcal{Y} := \times_{i=1}^N \mathcal{Y}_i$, $\mathcal{Z} := \times_{i=1}^N \mathcal{Z}_i$, $\Lambda = \times_{i=1}^N \Lambda_i$, $M = \times_{i=1}^N M_i$. Since the projected saddle-point dynamics (9a)-(9e) are convergent by [17, Theorem 5.1], there exist compact sets $\bar{\mathcal{V}}$, $\bar{\mathcal{Y}}$, $\bar{\mathcal{Z}}$, $\bar{\Lambda}$, $\bar{M}$ such that the trajectories of v, y, z, λ, μ under (9) stay in $\bar{\mathcal{V}}$, $\bar{\mathcal{Y}}$, $\bar{\mathcal{Z}}$, $\bar{\Lambda}$ and $\bar{M}$ respectively for all positive times. In what follows, we make the following assumption regarding the feasibility of (8).

Assumption 2: (Feasibility of S_α): For all $i \in \{1,\dots,N\}$, (8) is feasible for all $(x_i, y_{\mathcal{N}_i}, z_{\mathcal{N}_i}) \in \Pi_{x_i}\mathcal{F} \times \Pi_{y_{\mathcal{N}_i}}\bar{\mathcal{Y}} \times \Pi_{z_{\mathcal{N}_i}}\bar{\mathcal{Z}}$.

The following result is a consequence of the rank-nullity theorem [24, Theorem 3.22] and gives a sufficient condition under which Assumption 2 holds.

Lemma 4.5: (Sufficient condition for feasibility of S_α): Suppose that the vectors $\{\nabla g_i^k(x_i)\}_{k=1}^p \cup \{\nabla h_i^l(x_i)\}_{l=1}^q$ are linearly independent for all $x_i \in \Pi_{x_i}\mathcal{F}$. Then, for all $i \in \{1,\dots,N\}$, (8) is feasible for all $(x_i, y_{\mathcal{N}_i}, z_{\mathcal{N}_i}) \in \Pi_{x_i}\mathcal{F} \times \Pi_{y_{\mathcal{N}_i}}\bar{\mathcal{Y}} \times \Pi_{z_{\mathcal{N}_i}}\bar{\mathcal{Z}}$.

The next result establishes some feasibility and regularity properties of S_α. Its proof follows an argument analogous to the proof of [9, Proposition 5.3].

Proposition 4.6: (Well-posedness and regularity of SP-SGF): Under Assumption 2, the following statements hold:

- There exists an open neighborhood U containing $\bar{\mathcal{V}} \times \bar{\mathcal{Y}} \times \bar{\mathcal{Z}} \times \bar{\Lambda} \times \bar{M} \times \mathcal{F}$ such that (9) is well-defined on U;
- The dynamical system (9) is locally Lipschitz on U;
- The Lagrange multipliers of (8) are unique and locally Lipschitz as a function of x, y and z on U.

V. Invariance and Convergence Analysis

Having established the distributed character of the algorithm (9), here we show the forward invariance of the feasible set and the asymptotic convergence to the optimizer.

We start by introducing some useful notation. For $i \in \{1, \ldots, N\}$, $k \in \{1, \ldots, p\}$ and $l \in \{1, \ldots, q\}$, we let $\psi_{y_i^k}(t; p_0), \psi_{z_j^l}(t; p_0), \psi_{x_i}(t; p_0)$ be the solution of (9b), (9c), (9f) respectively for initial conditions $p_0 = (v_0, y_0, z_0, \lambda_0, \mu_0, x_0) \in \mathcal{P} := \mathcal{V} \times \mathcal{Y} \times \mathcal{Z} \times \Lambda \times M \times \mathcal{F}$. We also let $\psi_y(t; p_0) = [\psi_{y_1^1}(t; p_0), \ldots, \psi_{y_1^p}(t; p_0), \ldots, \psi_{y_N^1}(t; p_0), \ldots, \psi_{y_N^p}(t; p_0)]$, and define $\psi_z(t; p_0)$ and $\psi_x(t; p_0)$ analogously. The next result establishes the *anytime* nature of SP-SGF.

Lemma 5.1: (Anytime property): Suppose that $x_0 \in \mathcal{F}$ and Assumption 2 holds. Then, the trajectories of (9) satisfy $\psi_x(t; p_0) \in \mathcal{F}$ for all $t \geq 0$.

Next, we turn to the study of the convergence properties of (9). The next result follows from Proposition 2.1(v) and establishes a connection between the equilibrium points of S_α and the optimizers of (6).

Proposition 5.2: (Relationship between equilibria and optimizers): Let $x \in \mathcal{F}$. Then, $S_\alpha(x, y^{*,\epsilon}, z^{*,\epsilon}) = 0$ if and only if $x = x^{*,\epsilon}$.

The next result states that the trajectories of x in (9) converge to the optimizer of (6). Its proof follows by applying a Tikhonov-type singular perturbation theorem for differential inclusions in [25, Corollary 3.4].

Theorem 5.3: (Convergence to optimizer): Suppose Assumption 2 holds. For any $\delta > 0$ and compact set Ω containing $\{x \in \mathcal{F} \; : \; \|x - x^{*,\epsilon}\| \leq \delta\}$, there exists $\tau_{\delta,\Omega} > 0$ and $T_{\delta,\Omega}$ such that if $\tau < \tau_{\delta,\Omega}$, then under the dynamics (9), $\|\psi_x(t; p_0) - x^{*,\epsilon}\| < \delta$ for all $t \geq T_{\delta,\Omega}$ and $p_0 = (v_0, y_0, z_0, \lambda_0, \mu_0, x_0) \in \mathcal{P}$. Moreover, if $\mathcal{F}$ is bounded, then for any $\tau > 0$, $\lim_{t \to \infty} \|\psi_x(t; p_0) - x^{*,\epsilon}\| = 0$ for all $p_0 \in \mathcal{P}$.

By Theorem 5.3, the trajectories of the x variable in SP-SGF converge arbitrarily close to the optimizer $x^{*,\epsilon}$ provided that the timescale parameter τ is small enough. Moreover, if the feasible set $\mathcal{F}$ is bounded, asymptotic convergence holds for any timescale. The combination of the scalable and distributed character, cf. Remark 4.3, the anytime nature, cf. Lemma 5.1, and the convergence properties, cf. Theorem 5.3 means that SP-SGF provides the algorithmic solution with the properties stated in Section III.

Example 5.4: (Resource allocation): We illustrate the behavior of SP-SGF in a resource allocation example. Consider 13 agents whose communication graph is an undirected line graph. Solving distributed optimization problems with this particular topology is challenging due to its low connectivity. Each agent's state variable is $x_i = [x_{i,1}, x_{i,2}] \in \mathbb{R}^2$, where $x_{i,1}$ (resp. $x_{i,2}$) corresponds to the amount of resource 1 (resp. 2) allocated by agent i. Resource 1 is subject to an equality constraint and resource 2 is subject to an inequality constraint. Hence, the agents solve the optimization problem,

$$\min_{\{x_i\}_{i=1}^{13}} \sum_{i=1}^{13} \frac{1}{2} \|x_i\|^2, \tag{10}$$

$$\text{s.t. } h(\{x_i\}_{i=1}^{13}) = 5 - \sum_{i=1}^{13} p_i x_{i,1} = 0,$$

$$g(\{x_i\}_{i=1}^{13}) = -3 + \sum_{i=1}^{13} e^{-x_{i,2}} \leq 0.$$

with $p_1 = 1$, $p_2 = 3$, $p_3 = 2$, $p_4 = 1$, $p_5 = 1$, $p_6 = 1$, $p_7 = 2$, $p_8 = 4$, $p_9 = 1$, $p_{10} = 1$, $p_{11} = 0.5$, $p_{12} = 2$, $p_{13} = 1$. Note that the condition in Lemma 4.5 holds and hence Assumption 2 holds. This implies by Proposition 5.3 that (9) is well-defined for (10). We use $\epsilon = 0.0001$ and $\alpha = 1$. Figure 1 illustrates the convergence of the x variables under SP-SGF.

Since the feasible set of (10) is unbounded, Proposition 5.3 states that convergence arbitrarily close to the optimizer can be achieved by taking τ sufficiently small. Figure 2 illustrates the convergence of the quantities $\sum_{i=1}^{13} x_{i,1}^2$ and $\sum_{i=1}^{13} x_{i,2}^2$ for different values of τ and shows that this quantity converges exactly to its optimal value for a wide range of values of τ, suggesting that the statement in Theorem 5.3 might be too conservative. Figure 3 compares the evolution of the constraints of (10) under SP-SGF against two other algorithms: the projected saddle point dynamics (3) (abbreviated SP), which is not distributed, and the projected saddle-point dynamics (abbreviated SP-CM) for its reformulation with constraint mismatch variables as in (5), which is distributed. SP-SGF satisfies the constraints at all times whereas SP and SP-CM do not. We note that, in this case, SP-SGF requires running a dynamical system with 104 scalar variables (8 for each agent), SP-CM requires running a dynamical system of 78 scalar variables (6 for each agent) and SP requires running a dynamical system with 28 scalar variables. •

VI. Conclusions

We have introduced a continuous-time dynamical system that solves network optimization problems with separable objective function and constraints in a distributed and anytime fashion. We have achieved this by combining the projected saddle-point dynamics and the safe gradient flow in a cascaded system. We have argued the scalable nature of the algorithm execution from the point of view of individual agents and established practical convergence to the optimizer when the feasible is unbounded, and exact convergence when it is bounded. Future work will consider other network optimization problems, refine the convergence guarantees presented here and possibly design new distributed, anytime algorithms, and investigate discretization schemes for the continuous-time dynamics. We also plan to apply our coordination algorithms in the implementation of optimization-

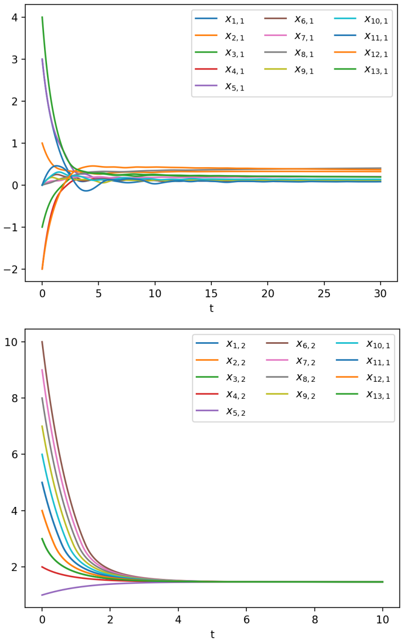

Fig. 1: Evolution of $x_{1,i}$ (top) and $x_{2,i}$ (bottom) for $i \in \{1,\ldots,13\}$ under SP-SGF for (10) with initial conditions $x_1 = [3,5]$, $x_2 = [1,4]$, $x_3 = [-1,3]$, $x_4 = [-2,2]$, $x_5 = [3,1]$, $x_6 = [0,10]$, $x_7 = [0,9]$, $x_8 = [0,8]$, $x_9 = [0,7]$, $x_{10} = [0,6]$, $x_{11} = [0,5]$, $x_{12} = [-2,4]$, $x_{13} = [4,3]$, $v_{i,1} = v_{i,2} = z_i = y_i = \lambda_i = \mu_i = 0$ for all $i \in \{1,\ldots,13\}$ and $\tau = 1$.

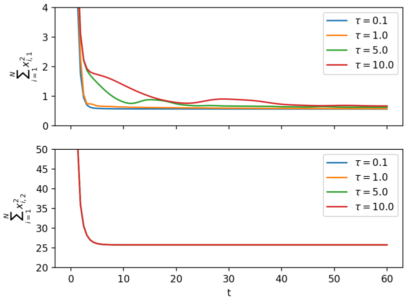

Fig. 2: This plot shows the evolution of $\sum_{i=1}^{13} x_{i,1}^2$ and $\sum_{i=1}^{13} x_{i,2}^2$ under SP-SGF with initial conditions as in Figure 1 for different values of τ.

based controllers arising from safety certificates for multi-agent systems.

References

[1] R. W. Brockett, "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems," *Linear Algebra and its Applications*, vol. 146, pp. 79–91, 1991.

[2] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. Springer, 1994.

[3] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," *arXiv preprint arXiv:2103.11329*, 2021.

[4] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[5] K. Arrow, L. Hurwitz, and H. Uzawa, *Studies in Linear and Non-Linear Programming*. Stanford, CA: Stanford University Press, 1958.

[6] T. Dean, *Intractability and Time-Dependent Planning*. Morgan Kaufmann, 1987.

[7] Y. S. Xia and J. Wang, "On the stability of globally projected dynamical systems," *Journal of Optimization Theory & Applications*, vol. 106, no. 1, pp. 129–150, 2000.

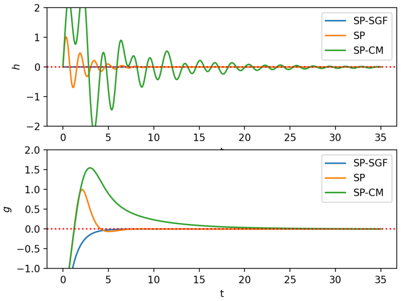

Fig. 3: This plot shows the evolution of the constraints of (10) for SP-SGF with the same initial conditions as in Figure 1, SP with the same primal initial conditions as in Figure 1 and $\lambda = \mu = 0$ and SP-CM with the same initial conditions as in Figure 1 for x_i, z_i, y_i, λ_i and μ_i for $i \in \{1,\ldots,13\}$.

[8] M. Muehlebach and M. I. Jordan, "On constraints in first-order optimization: A view from non-smooth dynamical systems," *Journal of Machine Learning Research*, vol. 23, pp. 1–47, 2022.

[9] A. Allibhoy and J. Cortés, "Control barrier function-based design of gradient flows for constrained nonlinear programming," *IEEE Transactions on Automatic Control*, vol. 69, no. 6, 2024, to appear.

[10] 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.

[11] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: theory and applications," in *European Control Conference*, Naples, Italy, 2019, pp. 3420–3431.

[12] 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, 2017.

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

[14] 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.

[15] M. Jankovic and M. Santillo, "Collision avoidance and liveness of multi-agent systems with cbf-based controllers," in *IEEE Conf. on Decision and Control*, Austin, TX, USA, 2021, pp. 6822–6828.

[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, 2022.

[17] A. Cherukuri, E. Mallada, S. H. Low, and J. Cortés, "The role of convexity in saddle-point dynamics: Lyapunov function and robustness," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2449–2464, 2018.

[18] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[19] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[20] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *Journal of Optimization Theory & Applications*, vol. 129, no. 3, pp. 469–488, 2006.

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

[22] A. Cherukuri and J. Cortés, "Distributed algorithms for convex network optimization under non-sparse equality constraints," in *Allerton Conf. on Communications, Control and Computing*, Monticello, IL, Sep. 2016, pp. 452–459.

[23] X. Tan and D. V. Dimarogonas, "On the undesired equilibria induced by control barrier function based quadratic programs," *arXiv preprint arXiv:2104.14895*, 2021.

[24] S. Axler, *Linear Algebra Done Right*. New York: Springer, 2015.

[25] F. Watbled, "On singular perturbations for differential inclusions on the infinite interval," *Journal of Mathematical Analysis and Applications*, vol. 310, no. 2, pp. 362–378, 2005.