On the Lie bracket approximation approach to distributed optimization: Extensions and limitations

Simon Michalowsky¹, Bahman Gharesifard², and Christian Ebenbauer¹

¹Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany
²Department of Mathematics and Statistics, Queen’s University, Canada

Abstract. We consider the problem of solving a smooth convex optimization problem with equality and inequality constraints in a distributed fashion. Assuming that we have a group of agents available capable of communicating over a communication network described by a time-invariant directed graph, we derive distributed continuous-time agent dynamics that ensure convergence to a neighborhood of the optimal solution of the optimization problem. Following the ideas introduced in our previous work, we combine saddle-point dynamics with Lie bracket approximation techniques. While the methodology was previously limited to linear constraints and objective functions given by a sum of strictly convex separable functions, we extend these results here and show that it applies to a very general class of optimization problems under mild assumptions on the communication topology.

1 INTRODUCTION

Over the last decades, distributed optimization has been an active area of research with high practical relevance, see, e.g., [2, 3, 4] for applications. In these type of problems, the goal is to cooperatively solve an optimization problem using a group of agents communicating over a network. While discrete-time algorithms for distributed optimization constitute the majority in the existing literature, we focus on continuous-time algorithms which have regained interest in the last decades, [5, 6, 7, 8, 9, 10]. These algorithms require strong assumptions either on the structure of the optimization problem or on the communication network. Recently, a novel approach to continuous-time distributed optimization based on Lie bracket approximations has been proposed in [11], [12] that has the potential to relax these assumptions.

In the present work we want to extend these results and show that the approach can be applied to a large class of optimization problems under mild assumptions on the communication network. While in [11], [12], [13] only optimization problems with linear constraints and objective functions in the form of a sum of separable functions were considered, in the present work we enhance the methodology to general convex optimization problems. The main idea of the approach is to use Lie bracket approximation techniques to find distributed approximations of non-distributed saddle-point dynamics. While in the previous works only certain Lie bracket approximations were used, we further show here that a whole class is applicable.

2 PRELIMINARIES

2.1 Notation

We denote by $\mathbb{R}^n$ the set of $n$-dimensional real vectors and further write $\mathcal{C}^p, p \in \mathbb{N}$, for the set of $p$-times continuously differentiable real-valued functions. The gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$, $f \in \mathcal{C}^1$, with respect to its argument $x \in \mathbb{R}^n$, will be denoted by $\nabla_x f : \mathbb{R}^n \to \mathbb{R}^n$; we often omit the subscript, if it is clear from the context. We denote the $(i,j)$th entry of a matrix $A \in \mathbb{R}^{n \times m}$ by $a_{ij}$, and sometimes denote $A$ by $A = [a_{ij}]$. We use $e_i$ to denote the vector with the $i$th entry equal to 1 and all other entries equal to 0. We do not specify the dimension of $e_i$ but expect it to be clear from the context. For a vector $\lambda \in \mathbb{R}^n$ we let $\text{diag}(\lambda) \in \mathbb{R}^{n \times n}$ denote the diagonal matrix whose diagonal entries are the entries of $\lambda$. Given two continuously differentiable vector fields $f_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $f_2 : \mathbb{R}^n \to \mathbb{R}^n$, the Lie

---

* This article is an extended version of [1] additionally including proofs for all results, a discussion of the choice of vector fields after Remark 1 and the overview in Table 1.
2.2 Basics on graph theory

We assume further that $F$ represents bracket of $f_1$ and $f_2$ evaluated at $x$ is defined to be

$$[f_1, f_2](x) := \frac{\partial f_2}{\partial x}(x)f_1(x) - \frac{\partial f_1}{\partial x}(x)f_2(x).$$

(1)

With a slight abuse of notation we sometimes also write $[f_1(x), f_2(x)]$. For a vector $x = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^n$ and a finite set $S \subseteq \{1, 2, \ldots, n\}$, we denote by $x_S \in \mathbb{R}^{|S|}$ the ordered stacked vector of all $x_i$ with $i \in S$. For example, if $n = 5$ and $S = \{2, 4\}$, then $x_S = [x_2, x_4]^\top$.

We recall some basic notions on graph theory, and refer the reader to [14] or other standard references for more information. A directed graph (or simply digraph) is an ordered pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, i.e. $(v_i, v_j) \in \mathcal{E}$ if there is an edge from node $v_i$ to $v_j$. In our setup the edges encode to which other agents some agent $i$ has access to, i.e. $(v_i, v_j) \in \mathcal{E}$ means that node $v_i$ receives information from node $v_j$. We say that node $v_i$ is an out-neighbor of node $v_j$ if there is an edge from node $v_j$ to node $v_i$. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ associated to $G$ is defined as

$$a_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } (v_i, v_j) \in \mathcal{E}, \\ 0 & \text{otherwise}. \end{cases}$$

(2)

We also define the out-degree matrix $D = [d_{ij}]$ associated to $G$ as

$$d_{ij} = \begin{cases} \sum_{k=1}^n a_{ik} & \text{if } i = j \\ 0 & \text{otherwise}. \end{cases}$$

(3)

Finally, we call $G = D - A = [g_{ij}] \in \mathbb{R}^{n \times n}$ the Laplacian of $G$. A directed path in $G$ is a sequence of nodes connected by edges and we write $p_{i,j} = (v_{i}, v_{i+1}, \ldots, v_{j})$ for a path from node $v_i$ to node $v_j$. We say that a path is simple if the sequence contains no node more than once.

3 PROBLEM SETUP

Consider the following convex optimization problem

$$\min_{x} F(x)$$

s.t. $a_i(x) = 0, i \in I_{eq} \subseteq \{1, 2, \ldots, n\},$

$$c_i(x) \leq 0, i \in I_{ineq} \subseteq \{1, 2, \ldots, n\},$$

(4)

where $F : \mathbb{R}^n \to \mathbb{R}, a_i : \mathbb{R}^n \to \mathbb{R}^{n_{eq}}, c_i : \mathbb{R}^n \to \mathbb{R}^{n_{ineq}}, n_{eq}, n_{ineq} \geq 1, F \in C^2$ is strictly convex, the functions $a_i, i \in I_{eq}$, are affine and the $c_i \in C^2, i \in I_{ineq}$, are convex. We assume further that $F, a_i, c_i$ are such that the feasible set of (4) is non-empty and that the problem has a unique solution.

Our goal is to design continuous-time optimization algorithms that converge to an arbitrarily small neighborhood of the unique global optimizer of (4) and that can be implemented in a distributed fashion. More precisely, we assume that we have a group of $n$ agents available, each capable of interchanging information over a communication network described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with graph Laplacian $G = [g_{ij}]$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is a set of $n$ nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set between the nodes. In the present setup, each node $v_i$ represents an agent and the edges define the existing communication links between the agents, i.e., if there is an edge from node $i$ to node $j$ then agent $i$ has access to the information provided by agent $j$. We then say that an algorithm is distributed if each agent only uses its own information as well as that provided by its out-neighboring agents.

Let $L : \mathbb{R}^n \times \mathbb{R}^{n_{eq}} \times \mathbb{R}^{n_{ineq}} \to \mathbb{R}$, $n_{eq} = \sum_{i \in I_{eq}} n_{eq,i}$, $n_{ineq} = \sum_{i \in I_{ineq}} n_{ineq,i}$ denote the Lagrangian associated to (4), i.e.,

$$L(x, \nu, \lambda) = F(x) + v^\top a(x) + \lambda^\top c(x),$$

(5)

where $a = [a_i]_{i \in I_{eq}}, c = [c_i]_{i \in I_{ineq}}$ are the stacked vectors of all $a_i$ and $c_i$, respectively, and $v \in \mathbb{R}^{n_{eq}}, \lambda \in \mathbb{R}^{n_{ineq}}$ are the associated Lagrange multipliers. In the sequel, we assume that the state of the $i$th agent comprises of $x_i$ as well as the dual variables associated to the constraints $a_i$, $c_i$, given they exist. Without loss of generality we then put the following assumption on the indexing of the constraints:

**Assumption 1.** For any $i \in I_{eq}$ and any $j \in I_{ineq}$ there exists an $x = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^n$ such that $\frac{\partial u_i}{\partial x_i}(x) \neq 0$ and $\frac{\partial c_i}{\partial x_i}(x) \neq 0$.

In a nutshell, this assumption guarantees that the $i$th constraints $a_i$, $c_i$ are both functions of $x_i$.

It is well-known that if the Lagrangian has New constraints point $(x^*, v^*, \lambda^*)$, then $x^*$ is an optimizer of (4). We say that a point $(x^*, v^*, \lambda^*)$ is a saddle point if for all $x \in \mathbb{R}^n$, $v \in \mathbb{R}^{n_{eq}}, \lambda \in \mathbb{R}^{n_{ineq}}$, we have

$$L(x^*, v, \lambda) \leq L(x, v^*, \lambda^*) \leq L(x, v^*, \lambda^*).$$

(6)

The following saddle-point dynamics adapted from [15] is known to converge to a saddle point of the Lagrangian (see [13] for a proof), thus providing a solution to (4)

$$\dot{x} = -\nabla_x L(x, v, \lambda)$$

$$= -\nabla F(x) - \frac{\partial u_i}{\partial x}(x) v - \frac{\partial c_i}{\partial x}(x) \lambda$$

(7a)

$$\dot{v} = \nabla_v L(x, v, \lambda)$$

$$= a(x)$$

(7b)

$$\dot{\lambda} = \text{diag}(\lambda) \nabla_\lambda L(x, v, \lambda)$$

$$= \text{diag}(\lambda) c(x).$$

(7c)
However, (7) is in general not distributed in the aforementioned sense, since the right-hand side of (7) is not composed only of admissible vector fields, i.e., vector fields that can be computed locally by the nodes. For example, if \( g_{12} \neq 0 \) and \( g_{13} = 0 \), then the vector field \([e_1^T x_2, 0, 0]^T\) is admissible for (7), while \([e_1^T x_3, 0, 0]^T\) is not.

Recently, a novel approach to distributed optimization has been proposed that employs Lie bracket approximation techniques to derive distributed approximations of (7). The idea is to write the right-hand side of (7) by means of Lie brackets of admissible vector fields. If we have achieved to rewrite (7) in this form, i.e., we have

\[
\dot{z} = \sum_{\mu \in B} v_\mu B(z),
\]

where \( z = [x^T, v^T, \lambda^T]^T, v_\mu \in \mathbb{R} \) and \( B \) is a set of Lie brackets of admissible vector fields \( \{\phi_1, \phi_2, \ldots, \phi_M\}, \phi_i : \mathbb{R}^{n+neq+nineq} \rightarrow \mathbb{R}^{n+neq+nineq} \) then we can employ Lie bracket approximation techniques from [16], [17] to derive distributed approximations of (8). More precisely, we can find a family of functions \( u^\sigma_i : \mathbb{R} \rightarrow \mathbb{R} \) parametrized by \( \sigma > 0 \) such that the trajectories of

\[
z^\sigma = \sum_{i=1}^{M} \phi_i(z) u^\sigma_i(t)
\]

uniformly converge to those of (8) as \( \sigma \) increases, given (8) and (9) are initialized equally. A general algorithm to compute suitable functions \( u^\sigma_i \) is presented in [16] and we will present a modified version thereof tailored to the problem at hand in [13]. In the present paper we do not discuss the second step of how to design these functions \( u^\sigma_i \) but focus on rewriting the right-hand side of (7) in terms of Lie brackets of admissible vector fields.

4 MAIN RESULTS

Consider the saddle-point dynamics (7) and observe that the right-hand side is a sum of vector fields of the form \( e_i \psi(x) \) and \( e_j z_i \psi(x) \), \( i, j = 1, 2, \ldots, n + neq + nineq \), \( \psi : \mathbb{R}^n \rightarrow \mathbb{R} \), where

\[
z = [x^T, v^T, \lambda^T]^T \in \mathbb{R}^N, \quad N = n + neq + nineq.
\]

is the complete state and \( e_i \in \mathbb{R}^N \) is the \( i \)th unit vector. These vector fields might either be admissible or not, depending on the communication graph as well as the problem structure. In the following, we wish to discuss how to write vector fields of this form by means of Lie brackets of admissible vector fields. For the purpose of notation, for each \( i = 1, 2, \ldots, n \), we define the index set

\[
I(i) := \{ i \} \cup \{ j = n + \sum_{k=1}^{i-1} neq_k + \ell, \ell = 1, 2, \ldots, n_{eq} \}
\]

\[
\cup \{ j = n + neq + \sum_{k=1}^{i-1} nineq_k + \ell, \ell = 1, 2, \ldots, nineq \}
\]

associating the components of the complete state \( z \) to the \( i \)th agent meaning that \( z_j \) is part of the state of agent \( i \) for all \( j \in I(i) \). Hence, the state vector of the \( i \)th agent is given by \( z_{I(i)} \). Based on this, for all \( i = 1, 2, \ldots, N \), we define

\[
I(j) = I(i)
\]

for some \( i \) such that \( j \in I(i) \), i.e., \( I(j) \) is the set of all indices which are associated to the same agent as the \( j \)th index. Note that \( I(j) = I(j) \), for all \( j = 1, 2, \ldots, n \).

4.1 Lie brackets of admissible vector fields

In the following, we first want to discuss which kind of vector fields can be written in terms of Lie brackets of admissible vector fields. For \( i, j = 1, 2, \ldots, N \), define

\[
h_{i,j}(z) = e_j f_i (z_{I(j)}, z_{I(i)}),
\]

where \( f_j : \mathbb{R}^{|I(j)|} \times \mathbb{R}^{|I(i)|} \rightarrow \mathbb{R} \), \( f_j \in C^1 \). Observe that \( h_{i,j} \) is admissible if and only if there exist \( \ell, k \in \{ 1, 2, \ldots, n \} \) such that \( i \in I(\ell), j \in I(k) \) and \( g_{k\ell} \neq 0 \). In the next Newresult we consider Lie brackets of admissible vector fields of the form (13).

**Lemma 1.** Consider a graph \( G \) of \( n \) nodes and let \( p_{i_i} = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_r} \rangle \) denote a simple path in \( G \) from \( v_{i_1} \) to \( v_{i_r} \). Then, for any \( j_k \in I(i_k), k = 1, 2, \ldots, r \), we have

\[
\left[ h_{j_1, j_2, \ldots, j_r} \left[ h_{j_1, j_2} \left[ h_{j_1} \right] \ldots \right] \right](z)
\]

\[
= e_{j_k} f_{j_{k-1}} (z_{I(j_{k-1})}, z_{I(j_k)}) \prod_{l=1}^{r-2} \frac{\partial f_{j_{k-l}}}{\partial z_{I(l)}} (z_{I(j_{k-l})}, z_{I(j_{k-l+1})})
\]

and the left-hand side is a Lie bracket of admissible vector fields.

A proof is given in Section 7.1. By the above Lemma, each non-admissible vector field that takes the same form as the right-hand side of (14) can be written in terms of a Lie bracket of admissible vector fields. It is worth mentioning that this does not classify the whole set of vector fields that can be written as a Lie bracket of admissible vector fields since we limited ourselves to a single path. We next discuss a special case that is of particular importance for the application at hand.
Proposition 1. Consider a graph \( G \) of \( n \) nodes and let \( p_{i,v} = \langle v_i, v_j \rangle \ldots \langle v_r \rangle \) denote a simple path in \( G \) from \( v_i \) to \( v_r \). Let \( j_k \in \mathcal{I}(i_k), k = 1, 2, \ldots, r \) be any set of indices and suppose that
\[
f_k(z_{\mathcal{I}(j_k)}, z_{\mathcal{I}(j_{k+1})}) = f_k^{(1)}(z_{\mathcal{I}(j_k)}) f_k^{(2)}(z_{\mathcal{I}(j_{k+1})})
\] (15)
with
\[
f_k^{(1)}(z_{\mathcal{I}(j_k)}) \frac{\partial f_k^{(2)}}{\partial z_{j_k}}(z_{\mathcal{I}(j_k)}) = 1
\] (16)
for all \( z_{\mathcal{I}(j_k)} \in \mathbb{R}^{I(j_k)}, k = 2, 3, \ldots, r - 1 \). Then
\[
\left[ h_{j_k j_{k-1}}, h_{j_{k-1} j_{k-2}}, \ldots, h_{j_2 j_1}, h_{j_1 j_0} \right] (z) = e_j \frac{f_k^{(1)}(z_{\mathcal{I}(j_k)})}{f_k^{(2)}(z_{\mathcal{I}(j_k)})}
\] (17)
for all \( z \in \mathbb{R}^N \) and the left-hand side is a Lie bracket of admissible vector fields.

A proof is given in Section 7.2. In view of (14), the constraint (16) ensures that all terms depending on \( z_{\mathcal{I}(j_k)} \), \( k = 2, 3, \ldots, r - 1 \), cancel out. Equation (17) is of particular interest since the non-admissible vector fields often take the form as its right-hand side. According to (16), there exists a whole class of vector fields \( h_{j_k j_{k-1}} \) or equivalently, functions \( f_k \) such that (17) holds. A particularly simple choice that has been utilized in the previous works \([11],[12]\) is to take
\[
f_k^{(1)}(z_{\mathcal{I}(j_k)}) = 1, \quad k = 2, 3, \ldots, r - 1,
\] (18a)
\[
f_k^{(2)}(z_{\mathcal{I}(j_k+1)}) = z_{j_k+1}, \quad k = 1, 2, \ldots, r - 2;
\] (18b)
hence leading to \( h_{j_k j_{k-1}}(z_{\mathcal{I}(j_k)}, z_{\mathcal{I}(j_{k+1})}) = e_j z_{j_k+1} \). However, in view of (9) where the \( \phi \) are given by admissible vector fields of the form (13), it is often desired that the admissible vector fields have certain properties such as boundedness in order to simplify the calculation of the approximating inputs \( u^j_i \) and improve the transient behavior of (9). However, as we see next, it is not possible to render all vector fields bounded.

Lemma 2. Suppose that all assumptions from Proposition 1 are fulfilled. Then there exists no set of bounded vector fields \( h_{j_k j_{k-1}} \in C^1 \) such that (17) holds.

Proof. Suppose there exists a set of bounded vector fields such that (17) holds. Then \( f_k \) is a bounded function for all \( k = 1, 2, \ldots, r \). By (15), \( f_k^{(1)} \) is also bounded for all \( k = 1, 2, \ldots, r \) since \( \mathcal{I}(j_k) \) and \( \mathcal{I}(j_{k+1}) \) are disjunct. However, by (16),
\[
\frac{\partial f_k^{(2)}}{\partial z_{j_k}}(z_{\mathcal{I}(j_k)}) = \frac{1}{f_k^{(1)}(z_{\mathcal{I}(j_k)})}
\] (19)
for \( k = 2, \ldots, r - 1 \); hence \( \frac{\partial f_k^{(2)}}{\partial z_{j_k}}(z_{\mathcal{I}(j_k)}) \) is bounded away from zero, i.e., there exists some constant \( \beta > 0 \) such that for all \( z_{\mathcal{I}(j_k)} \in \mathbb{R}^{I(j_k)} \) we have
\[
|\frac{\partial f_k^{(2)}}{\partial z_{j_k}}(z_{\mathcal{I}(j_k)})| \geq \beta.
\] (20)

Note that \( h_{j_k j_{k-1}} \in C^1 \) if and only if \( f_k^{(1)}(z_{\mathcal{I}(j_k)}) \) is strictly monotone in \( z_{j_k} \) which contradicts the boundedness assumption, thus concluding the proof. \( \square \)

Remark 1. The same holds true if we do not assume the structure (15). In fact, the structure is required for all other variables except of \( z_{\mathcal{I}(j_1)} \), \( z_{\mathcal{I}(j_r)} \) to cancel out in (17).

As we see from the proof, each bounded vector field in (17) leads to another unbounded vector field. Hence, at most half of the vector fields \( h_{j_k j_{k-1}} \) in (17) can be bounded. In particular, we can choose the functions \( f_k \), \( k = 1, 2, \ldots, r - 1 \), as follows to guarantee that (16) holds:
\[
f_k^{(1)}(x) = \begin{cases} \cos(\alpha(x - d)) & \text{if } k \text{ is even,} \\ \frac{1}{\cos(\alpha(x - d))} & \text{if } k \text{ is odd, } k \neq 1, \end{cases}
\] (21a)
\[
f_k^{(2)}(x) = \begin{cases} \sin(\alpha(x - d)) & \text{if } k \text{ is even, } k \neq r - 1, \\ \int \frac{1}{\cos(\alpha(x - d))} \, dx & \text{if } k \text{ is odd, } k \neq r - 1, \end{cases}
\] (21b)
where \( \alpha \neq 0, d \in \mathbb{R} \). However, this choice will lead to functions \( f_k \) that are not globally continuous but only well-defined in the interval \( (d - \frac{\pi}{2\alpha}, d + \frac{\pi}{2\alpha}) \); hence we need to choose \( d \) appropriately and \( \alpha \) sufficiently small. By that choice, all functions \( f_k \) with even \( k \) are bounded while all functions \( f_k \) with odd \( k \) are unbounded.

We next discuss how we can make use of the previous results to rewrite more general vector fields. While Lemma 1 enables us to write products of functions of variables of nodes which lie on the same path as Lie brackets of admissible vector fields, we cannot directly use this result to rewrite functions which do not fulfill this property. To make this clearer, consider the following example:

Example 1. Consider the graph shown in Fig. 1 and assume for the sake of simplicity that \( \mathcal{I}(j) = j \) for \( j = 1, 2, \ldots, 5 \) and \( z \in \mathbb{R}^5 \). By (14) we can write non-admissible vector fields of the form \( \phi_1 \phi_2 (z_2) \phi_3 (z_3) \) as well as \( \phi_1 \phi_3 (z_4) \phi_4 (z_5) \) and sums thereof in terms of Lie brackets of admissible vector fields as long as \( \phi_1, \phi_3 \) admit an analytic expression of their antiderivatives \( \int \phi_1 (z_2) \, dz_2, \int \phi_3 (z_4) \, dz_4 \). In fact, with
\[
h_{2,1}(z) = e_1 \int \phi_1 (z_2) \, dz_2, \quad h_{3,2}(z) = e_2 \phi_2 (z_3),
\] (22a)
\[
h_{4,3}(z) = e_3 \int \phi_3 (z_4) \, dz_4, \quad h_{5,4}(z) = e_4 \phi_4 (z_5),
\] (22b)
Example 1. [continued] Reconsider Example 1 and suppose we want to rewrite the non-admissible vector field $e_1 \varphi_2(z_3) \varphi_4(z_5)$. Following Proposition 2 we let

\[
\psi_1(z) = e_1 z_1 \varphi_2(z_3), \quad \psi_2(z) = e_1 \varphi_3(z_5),
\]

and observe that for all $z \in \mathbb{R}^N$ we have

\[
[\psi_2, \psi_1](z) = e_1 \varphi_2(z_3) \varphi_4(z_5).
\]

Further, following Proposition 1 and choosing

\[
\begin{align*}
    h_{2,1}(z) &= e_1 z_1, & h_{3,2}(z) &= e_2 \varphi_2(z_3), \\
    h_{4,3}(z) &= e_1, & h_{5,4}(z) &= e_4 \varphi_4(z_5),
\end{align*}
\]

we have for all $z \in \mathbb{R}^N$

\[
\psi_1(z) = [h_{3,2}, h_{2,1}](z), \quad \psi_2(z) = [h_{5,4}, h_{4,3}](z).
\]

Using this in (27) we finally managed to rewrite $e_1 \varphi_2(z_3) \varphi_4(z_5)$ in terms of admissible vector fields.

Remark 2. Instead of realizing the multiplication by means of Lie brackets another way is to augment the agent state by estimates of the respective state of the other agent. More precisely, in Example 1, we augment the state of agent 1 by $\tilde{z}_3$ and $\tilde{z}_5$ that are estimates of $z_3$ and $z_5$, respectively, and let

\[
\begin{bmatrix}
    \tilde{z}_1 \\
    \tilde{z}_3 \\
    \tilde{z}_5
\end{bmatrix} = \begin{bmatrix}
    w(z_1, z_2, z_4, \tilde{z}_3, \tilde{z}_5) \\
    -\mu \tilde{z}_3 \\
    -\mu \tilde{z}_5
\end{bmatrix} + e_2 z_5 + e_3 z_5,
\]

where $w : \mathbb{R}^5 \to \mathbb{R}$ and $\mu > 0$ is sufficiently large, hence $\tilde{z}_3(t) \approx z_3(t)$, $\tilde{z}_5(t) \approx z_5(t)$. The resulting non-admissible vector fields in the complete augmented system can then be written in terms of Lie brackets of admissible vector fields using Proposition 1. However, in the application at hand this alters the saddle-point dynamics (7) which necessitates a stability analysis of the augmented system.

Hence, under suitable assumptions on the communication graph, this allows us to write vector fields whose components are sums of products of arbitrary functions in terms of Lie brackets of admissible vector fields. This observation gives rise to the next Lemma.

Lemma 3. Consider a strongly connected graph $G$ of $n$ nodes and let $\varphi : \mathbb{R}^N \to \mathbb{R}$ be an analytic function. Then any vector field $\psi(z) = e_i \varphi(z)$, $e_i \in \mathbb{R}^N$, $i = 1, 2, \ldots, N$, can be written as a possibly infinite sum of Lie brackets of admissible vector fields.

Proof. Since $\varphi$ is analytic, by a series expansion it can be written as a possibly infinite sum of monomials of the components $z_j$ of $z$, $j = 1, 2, \ldots, N$. Using Proposition 2, all these monomials can be written in terms of Lie brackets of vector fields of the form (24). By strong connectivity of $G$, all these vector fields can be written in terms of Lie brackets of admissible vector fields, thus concluding the proof.

While the result might be more of a theoretical nature for the application at hand, it nevertheless shows that the proposed approach in principle applies to a large class of problems.
inequality constraints are sums of products of separable
in the following that the objective function as well as the
We can then write the saddle-point dynamics (7) equiva-
to apply as we will illustrate in the example in Section 5.
emphasize that this is not necessary for the methodology
always be achieved by augmenting the optimization prob-
\begin{align}
\epsilon^j_i \frac{\partial F_j^{(i)}}{\partial x_i}(x_i) \prod_{j \in J^{(i)}_F} F_j^{(i)}(x_j) & \quad \ell = 1, 2, \ldots, n_F \\
\epsilon^j_i \bar{a}_k x_k & \quad k \in \mathcal{I}_{\text{eq}} \quad i = 1, 2, \ldots, n_F \\
\epsilon^j_i \frac{\partial c_{k}^{(i)}}{\partial x_k}(x_k) \prod_{j \in J^{(i)}_c} c_{k}^{(i)}(x_j) & \quad k \in \mathcal{I}_{\text{ineq}} \quad \ell = 1, 2, \ldots, n_{c_k} \\
\epsilon^{n+q_i} \bar{a}_k x_i & \quad k \in \mathcal{I}_{\text{eq}} \quad i = 1, 2, \ldots, n_F \\
\epsilon^{n+q_i+q_j} \bar{a}_k x_i & \quad k \in \mathcal{I}_{\text{ineq}} \quad \ell = 1, 2, \ldots, n_{c_k}
\end{align}

Table 1. An overview of all vector fields appearing in (31).

4.2 Distributed optimization via Lie brackets

In the sequel we apply the results from the last section
to the problem at hand and rewrite the saddle-point dy-
namics (7) by means of Lie brackets of admissible vector
fields. For the sake of a simpler notation we assume in
the following that $n_{\text{eq}_i} = 1$ for all $i \in \mathcal{I}_{\text{eq}}$ and $n_{\text{ineq}_i} = 1$
for all $i \in \mathcal{I}_{\text{ineq}}$. We further assume that each agent
has an associated equality and inequality constraint, i.e.,
\[ \mathcal{I}_{\text{eq}} = \{1, 2, \ldots, n\}, \quad \mathcal{I}_{\text{ineq}} = \{1, 2, \ldots, n\}. \]
This can always be achieved by augmenting the optimization prob-
lem (4) by constraints that do not alter the feasible set. We
emphasize that this is not necessary for the methodology
as we will illustrate in the example in Section 5.
We can then write the saddle-point dynamics (7) equival-
ently as
\begin{align}
\dot{x} &= -\sum_{i=1}^{n} \epsilon^j_i \frac{\partial F_j^{(i)}}{\partial x_i}(x) + \sum_{k \in \mathcal{I}_{\text{eq}}} \bar{a}_k x_k + \sum_{k \in \mathcal{I}_{\text{ineq}}} \frac{\partial c_k^{(i)}}{\partial x_k}(x) \lambda_k \quad (31a) \\
\dot{v} &= \sum_{k \in \mathcal{I}_{\text{eq}}} c_k \left( \sum_{i=1}^{n} \bar{a}_i x_i + \bar{b}_k \right) \quad (31b) \\
\dot{\lambda} &= \sum_{k \in \mathcal{I}_{\text{ineq}}} \epsilon^{n+q_i} \lambda_k c_k(x) \quad (31c)
\end{align}

where $a_i(x) = \bar{a}_i x + \bar{b}_j, \quad a_j = [\bar{a}_{i1}, \bar{a}_{i2}, \ldots, \bar{a}_{in}] \in \mathbb{R}^{1 \times n}, \quad b_j \in \mathbb{R}$. Motivated by our previous discussions, we assume
in the following that the objective function as well as the
inequality constraints are sums of products of separable
functions, i.e.,
\begin{align}
F(x) &= \sum_{\ell=1}^{n_F} \prod_{j \in J^{(i)}_F} F_j^{(i)}(x_j), \\
c_k(x) &= \sum_{\ell=1}^{n_{c_k}} \prod_{j \in J^{(i)}_c} c_{kj}^{(i)}(x_j),
\end{align}

where the $F_j^{(i)} : \mathbb{R} \to \mathbb{R}, \quad j \in J^{(i)}_F \subseteq \{1, 2, \ldots, n\}, \quad \ell = 1, 2, \ldots, n_F, \quad n_F > 0$, are strictly convex functions
and the $c_k^{(i)} : \mathbb{R} \to \mathbb{R}, \quad k \in \mathcal{I}_{\text{ineq}}, \quad j \in J^{(i)}_c \subseteq \{1, 2, \ldots, n\}, \quad \ell = 1, 2, \ldots, n_{c_k}, \quad n_{c_k} = 0$, are convex. Observe that, if
$n_F$ and $n_{c_k}$ are infinite and $F_j^{(i)}$, $c_k^{(i)}$ are monomials,
this includes all analytic functions $F$, $c_k$. If $J^{(i)}_F = \{\ell\}, \quad J^{(i)}_c = \{\ell\}$, we obtain the particularly important special
case that both the objective function and the constraints
are a sum of separable functions; hence also the case of
linear constraints considered in [11], [12] is covered here.
Under this assumption the vector fields appearing in (31)
are summed up in Table 1. Depending on the commu-
nication graph as well as the structure of the constraints
and the objective function, these vector fields can either be
admissible or not. In particular, the vector fields in (31b),
(31c) are admissible if the constraints are compatible with
the communication topology defined by the graph $G$, i.e.,
if the following assumption holds:

**Assumption 2.** For all $i, j = 1, 2, \ldots, n$ with $g_{ij} = 0$ we
have $\frac{\partial a_i}{\partial x_0}(x) \equiv 0$ as well as $\frac{\partial c_k}{\partial x_0}(x) \equiv 0$. 

We point out that all non-admissible vector fields in (31)
can be written in terms of Lie brackets of admissible vector
fields under appropriate assumptions on the communication graph, see also the last column of Table 1. Specifically, if the graph is strongly connected, then all non-admissible vector fields can be rewritten independent of the objective function as well as the constraints, given that they admit the structure (32), (33). In most cases, however, much less restrictive requirements on the communication graph are sufficient. We do not explicitly discuss how to rewrite the non-admissible vector fields using Proposition 1 and Proposition 2 in general but illustrate this by means of an example in Section 5.

5 EXAMPLE

In this section we illustrate the previous results by means of an example. Consider the following optimization problem

$$
\min_{x \in \mathbb{R}^5} F(x) = \sum_{i=1}^{5} F_i(x_i)
$$

s.t. 

$$
\begin{align*}
a_2(x) &= 2x_2 - x_5 = 0 \\
c_1(x) &= x_1^2 + x_2^2 - 4 \leq 0 \\
c_3(x) &= x_1 + x_3 - 2 \leq 0 \\
c_4(x) &= x_4^2 - x_4x_1 + x_1^2 - 100 \leq 0,
\end{align*}
$$

where $F_i(x_i) = (x_i - i)^2$, $x = [x_1, x_2, x_3, x_4, x_5]^T \in \mathbb{R}^5$. We assume that the communication topology is described by the graph in Fig. 2. Observe that the constraints $c_3$ and $c_4$ are not compatible with the graph topology, hence Assumption 2 is not fulfilled. The corresponding saddle-point dynamics (7) are then given by

$$
\dot{x} = - \nabla F(x) - (2e_2 - e_5)v_2 - 2(x_1e_1 + x_2e_2)\lambda_1
$$

(35a)

$$
- (2x_4 - x_1)e_4\lambda_4 - (x_1 - x_4)e_1\lambda_4 - (e_1 + e_3)\lambda_3
$$

$$
v_2 = 2x_2 - x_5 \quad (35b)
$$

$$
\lambda = \lambda_1 e_1(x_1^2 + x_2^2) + \lambda_3 e_2(x_1 + x_3 - 2)
$$

$$
+ \lambda_4 e_3(x_2^2 - x_4x_1 + x_1^2 - 100), \quad (35c)
$$

where $x \in \mathbb{R}^5$, $v_2 \in \mathbb{R}$, and $\lambda = [\lambda_1, \lambda_3, \lambda_4]^T \in \mathbb{R}^3$. We next rewrite all non-admissible vector fields in (35) using Proposition 1. We will thereby follow the choice (18) to make sure that (16) holds. We do not discuss how to rewrite each non-admissible vector field in detail, but limit ourselves to the vector field $e_1x_4\lambda_4$ from (35a). Comparing with (17), we have $j_1 = 1$, $j_r = 4$, and $f_{j_1}^{(1)}(z_{I(j_1)}) = 1$, $f_{j_{r-1}}^{(2)}(z_{I(j_r)}) = x_4\lambda_4$. Following Proposition 1, we require a path from node 1 to node 4 that is here given by $p_{14} = (v_1\mid v_2\mid v_3\mid v_4)$; thus $r = 4$. We then obtain

$$
e_1x_4\lambda_4 = [e_2\lambda_4x_4, e_1\lambda_4x_1, e_1\lambda_4], \quad (36)
$$

where $j_1 \in I(2) = \{2, 6\}$, $j_2 \in I(3) = \{3, 8\}$, and the right-hand side is a Lie bracket of admissible vector fields. All other non-admissible vector fields in (35) can be treated similarly and we sum up the resulting Lie brackets in Table 2. For the simulation we let each $j, j_1, j_2$ be the largest index in its respective index set. By that choice, we inject less perturbation in the primal and more in the dual variables $New$ which is also visible in the simulation results depicted in Fig. 3. As to be seen, the distributed algorithm approximates the trajectories of the non-distributed saddle-point dynamics (35) and converges to a neighborhood of the optimizer $x^*$ = $[0, 2, 2, 4, 4]$. New We also

Table 2. An overview of the non-admissible vector fields in (35) and their Lie bracket representations. Here, the index sets are $I(1) = \{1, 7\}$, $I(2) = \{2, 6\}$, $I(3) = \{3, 8\}$, $I(4) = \{4, 9\}$, $I(5) = \{5\}$.

| Vector field | Corresponding path | Lie bracket representation |
|--------------|--------------------|---------------------------|
| $e_5v_2$     | $p_{52} = (v_5\mid v_1\mid v_2)$ | $[e_5v_2, e_5z]$ | $j \in I(1)$ |
| $e_2x_2\lambda_1$ | $p_{21} = (v_2\mid v_5\mid v_1)$ | $[e_2z, e_2z]$ | $j \in I(5)$ |
| $e_1\lambda_3$ | $p_{13} = (v_1\mid v_2\mid v_3)$ | $[e_1\lambda_3, e_1z]$ | $j \in I(2)$ |
| $e_4x_1\lambda_4$ | $p_{41} = (v_4\mid v_5\mid v_1)$ | $[e_4x_1, e_4\lambda_4z]$ | $j \in I(5)$ |
| $e_1x_4\lambda_4$ | $p_{14} = (v_1\mid v_2\mid v_3\mid v_4)$ | $[e_1x_4, e_1\lambda_4, e_1z]$ | $j \in I(2), j_2 \in I(3)$ |
| $e_1x_1\lambda_4$ | $p_{14} = (v_1\mid v_2\mid v_3\mid v_4)$ | $[e_1x_1, e_1\lambda_4, e_1z]$ | $j \in I(2), j_2 \in I(3)$ |
| $e_8\lambda_3x_1$ | $p_{31} = (v_3\mid v_4\mid v_5\mid v_1)$ | $[e_8\lambda_3x_1, e_8\lambda_2z]$ | $j \in I(4), j_2 \in I(5)$ |
| $e_9\lambda_4x_4x_1$ | $p_{41} = (v_4\mid v_5\mid v_1)$ | $[e_9\lambda_4x_4x_1]$ | $j \in I(5)$ |
| $e_9\lambda_4x_2^2$ | $p_{41} = (v_4\mid v_5\mid v_1)$ | $[e_9\lambda_4x_2^2]$ | $j \in I(5)$ |
Figure 3. Simulation results of the example from Section 5. The upper plot shows the primal variable \( x(t) \), the lower left one the dual variable \( \nu(t) \) and the lower right one the dual variable \( \lambda(t) \). NewIn each plot, the dotted black lines marked with squares depict the trajectories of the non-distributed saddle-point dynamics (35) while the thinner oscillating ones depict the trajectories of the distributed approximation. The corresponding thick lines depict the distributed approximation with additional low-pass filters. The dashed black lines indicate the desired equilibrium of (35).

included simulation results with additional low-pass filters in the distributed \( x-, \nu- \) and \( \lambda- \)dynamics. While the effect on the primal variables is small since we already reduced the oscillations by our design choice, the dual variables show significantly less oscillations and better approximate the non-distributed trajectories. A rigorous stability analysis of the augmented distributed dynamics and a performance-oriented design of the filters is up to future work.

6 CONCLUSIONS AND OUTLOOK

We considered a convex optimization problem and showed how distributed optimization algorithms can be designed for a quite general class of problems with little structural requirements under mild assumptions on the communication network. We therefore extended the Lie bracket approximation approach to distributed optimization proposed in [11], [12] and discussed which kind of vector fields can in principle be written in terms of Lie brackets of admissible vector fields. We did not discuss the construction of approximating inputs but postpone this to [13] where we will present a modified version of the general algorithm from [16] that exploits the structure of the problem at hand.

References

[1] S. Michalowsky, B. Gharesifard, and C. Ebenbauer, “On the Lie bracket approximation approach to distributed optimization: Extensions and limitations,” in 2018 European Control Conference (ECC), accepted, 2018.

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

[3] F. Bullo, J. Cortés, and S. Martínez, Distributed Control of Robotic Networks, ser. Applied Mathematics Series. Princeton University Press, 2009.

[4] J. Zhao and F. Dörfler, “Distributed control and optimization in DC microgrids,” Automatica, vol. 61, no. Supplement C, pp. 18 – 26, 2015.

[5] D. Feijer and F. Paganini, “Stability of primal–dual gradient dynamics and applications to network optimization,” Automatica, vol. 46, no. 12, pp. 1974 – 1981, 2010.

[6] J. Wang and N. Elia, “A control perspective for centralized and distributed convex optimization,” in 2011 IEEE 50th Conference on Decision and Control and European Control Conference (CDC-ECC), 2011, pp. 3800–3805.

[7] H.-B. Dür, C. Zeng, and C. Ebenbauer, “Saddle point seeking for convex optimization problems,” IFAC Proceedings Volumes, vol. 46, no. 23, pp. 540–545, 2013.

[8] B. Gharesifard and J. Cortés, “Distributed continuous-time convex optimization on weight-balanced digraphs,” IEEE Transactions on Automatic Control, vol. 59, no. 3, pp. 781–786, 2014.

[9] S. K. Niederländer and J. Cortés, “Distributed coordination for separable convex optimization with coupling constraints,” in 2015 54th IEEE Conference on Decision and Control (CDC), Dec 2015, pp. 694–699.
By the induction hypothesis we then have
\[ \left[ h_{j_1 j_2 i}, \ldots, h_{j_{r-1} j_r} \right](z) \]
\[ = \left[ h_{j_1 j_2 i}, h_{j_3 j_4}, \ldots, h_{j_{r-1} j_r} \right](z) \]
\[ = e_{j_1} \left( \sum_{i=1}^{N} \frac{\partial f_{j_1 i}}{\partial z_{j_i}}(z) \right) e_{j_2} f_{j_1}(z_{I(j_1)}, z_{I(j_2)}) \]
\[ - e_{j_2} \left( \sum_{i \in I(j_2) \setminus I(j_1)} \frac{\partial f_{j_1 i}}{\partial z_{j_i}}(z_{I(j_2)}, z_{I(j_1)}) \right) e_{j_1} f_{j_2}(z) \]
\[ = e_{j_1} \frac{\partial f_{j_1 i}}{\partial z_{j_i}}(z) f_{j_2}(z_{I(j_1)}, z_{I(j_2)}), \]  
(38)
with a slight abuse of notation
\[
\left[\ldots \left[\psi_{m+1}, \psi_m\right], \psi_{m-1}, \ldots\right], \psi_2\right](z)
\]
\[
= [e_i \prod_{k=2}^{m+1} \eta_k(z_{j_k}), e_i z_i \eta_2(z_{j_2})]
\]
\[
= e_i (e_i^\top \eta_2(z_{j_2}) + z_i e_{j_2} \frac{\partial \eta_2}{\partial z_{j_2}}(z_{j_2})) e_i \prod_{k=2}^{m+1} \eta_k(z_{j_k})
\]
\[
+ e_i (\sum_{k=2}^{m+1} e_{j_k} \frac{\partial \eta_k}{\partial z_{j_k}}(z_{j_k}) \prod_{l=2}^{m+1} \eta_{l}(z_{j_l})) e_i z_i \eta_2(z_{j_2})
\]
\[
= e_i \prod_{k=2}^{m+1} \eta_k(z_{j_k}),
\]
\[
(40)
\]
where we used that \( i \neq j_k \) for any \( k = 1, \ldots, m \) in the last step. This proves (39). Finally, using (39), we obtain
\[
\left[\ldots \left[\psi_{m}, \psi_{m-1}\right], \psi_{m-2}, \ldots\right], \psi_1\right](z)
\]
\[
= [e_i \prod_{k=2}^{m} \eta_k(z_{j_k}), e_i \eta_1(z_{j_1}) \int \eta_0(z_i) dz_i]
\]
\[
= e_i \eta_0(z_i) \prod_{k=1}^{m} \eta_k(z_{j_k}).
\]
\[
(41)
\]
This finishes the proof.
\[\square\]