On Centralized and Distributed Mirror Descent: Exponential Convergence Analysis Using Quadratic Constraints

Youbang Sun\textsuperscript{1}, Mahyar Fazlyab\textsuperscript{2} and Shahin Shahrampour\textsuperscript{1}

Abstract—Mirror descent (MD) is a powerful first-order optimization technique that subsumes several optimization algorithms including gradient descent (GD). In this work, we study the exact convergence rate of MD in both centralized and distributed cases for strongly convex and smooth problems. We view MD with a dynamical system lens and leverage quadratic constraints (QCs) to provide convergence guarantees based on the Lyapunov stability. For centralized MD, we establish a semi-definite programming (SDP) that certifies exponentially fast convergence of MD subject to a linear matrix inequality (LMI). We prove that the SDP always has a feasible solution that recovers the optimal GD rate. Next, we analyze the exponential convergence of distributed MD and characterize the rate using two LMIs. To the best of our knowledge, the exact (exponential) rate of distributed MD has not been previously explored in the literature. We present numerical results as a verification of our theory and observe that the richness of the Lyapunov function entails better (worst-case) convergence rates compared to existing works on distributed GD.

I. INTRODUCTION

Over the last two decades, distributed optimization over multi-agent networks has received a lot of attention in control, optimization, machine learning, and signal processing. Generally, distributed optimization is useful for applications that are naturally distributed (e.g., sensor networks), or problems that involve massive amount of data, where a centralized machine cannot handle the optimization (e.g., large-scale risk minimization). Specific applications of distributed optimization include distributed cooperative control \cite{1}, resource allocation \cite{2}, distributed sensor localization \cite{3}, and social learning \cite{4}.

In this setup, consider $n$ agents in a network modeled by nodes of an undirected graph, and assume that all agents are only able to communicate with their neighbors (defined with respect to the graph). Each agent is assigned with a local objective function $f_i : \mathcal{X} \rightarrow \mathbb{R}$, and the agents goal is to collectively converge to the optimal solution of the global function, defined as

$$
\min_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{1}
$$

The most intuitive gradient-based algorithm to tackle the problem above is distributed gradient descent \cite{5}, where in each iteration $k$, agent $i$ performs a (private) gradient calculation as well as an averaging in the neighborhood $\mathcal{N}_i$. In the unconstrained case, this update is given by

$$
x_i^{(k+1)} = x_i^{(k)} - \eta^{(k)} \nabla f_i(x_i^{(k)}) + \beta \sum_{j \in \mathcal{N}_i} (x_j^{(k)} - x_i^{(k)}),
$$

where $\eta^{(k)} > 0$ is the step-size or learning rate. In the form given above, this update is able to achieve optimal rates for convex problems using a diminishing step-size sequence. Optimality here refers to matching the centralized convergence rate (iteration complexity) up to some network errors. However, when the local functions are strongly convex and/or smooth, the optimal centralized algorithms employ a constant step-size sequence, where the update above fails to converge.

It is well-known that when $f$ is strongly convex and smooth, centralized gradient descent (GD) converges exponentially fast. Recently, a number of distributed GD algorithms have been developed to achieve exponential rates when the local functions $f_i$ are strongly convex and smooth. These algorithms are mostly based on the idea of gradient tracking, and the EXTRA algorithm \cite{6} is a prominent one. While we defer the detailed discussion of these literature to Section \cite{6}, these algorithms are based on GD, which does not utilize the geometry of the problem.

The mirror descent (MD) algorithm \cite{7} is a primal-dual method that has been actively studied in recent years. MD can be seen as a generalization of GD, in which the Euclidean distance is replaced by Bregman divergence as the regularizer. The freedom in the choice of Bregman divergence makes MD suitable for various problem geometries (see Section \cite{7} for some examples). MD has been proven to have the same iteration complexity as GD for non-strongly convex problems, and it may even scale better with respect to the dimension of the decision variable. In the strongly convex scenario, MD is perhaps less studied, and very recently its exponential convergence was established under Polyak-Łojasiewicz condition \cite{8}. Inspired by the success of MD in centralized optimization, MD has also been studied in the distributed setting. To the best of authors’ knowledge, the exact convergence rate of distributed MD for strongly convex and smooth problems is not known, and only recently \cite{9} provided a continuous-time analysis suggesting the local exponential rate (without explicitly characterizing the rate).

In this paper, we establish exponential convergence for the Mirror Descent algorithm in both centralized and distributed settings for strongly convex and smooth problems. To this end, we leverage the framework of Quadratic Constraints (QCs) to derive sufficient conditions, in the form of matrix inequalities, for exponential convergence of the algorithm. QCs and their integral version (IQC) provide a unified framework to analyze convergence of first-order iterative optimization algorithms.

This work is supported in part by NSF ECCS-1933878 Award.

\textsuperscript{1} Y. Sun and S. Shahrampour are with Wm Michael Barnes ’64 Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA. email: ybsun, shahinl@tamu.edu.

\textsuperscript{2} M. Fazlyab is with the Department of Electrical and Computer Engineering at Johns Hopkins University, Baltimore, MD 21218, USA. email: mahyarfazlyab@jhu.edu.

10.1007/s11856-021-01531-z
By viewing iterative algorithms as dynamical systems, this framework convert the worst-case convergence analysis of optimization algorithms into a semidefinite feasibility problem. In this work, focusing on strongly convex and smooth functions, we first analyze centralized MD using QCs. We derive a linear matrix inequality (LMI) as a sufficient condition for fast exponential convergence of the algorithm (Theorem 2 and Proposition 3). We prove that the LMI problem always has a feasible solution that recovers the optimal GD rate when the Bregman divergence is set to be the Euclidean distance (Proposition 4 and Corollary 5). Next, we analyze the exponential convergence of distributed MD and characterize the rate using two LMIs (Theorem 7). To the best of our knowledge, the exact (exponential) rate of distributed MD has not been previously explored in the literature. We present numerical results to validate our theory and observe that the richness of the Lyapunov function entails better (worst-case) convergence rates compared to existing works on distributed GD.

A. Related Literature

Our work is related to several strands of literature discussed in the sequel.

1) Distributed Gradient Descent: To ensure the distributed algorithm reaches consensus, many works [5, 12, 13] use diminishing step-size (commonly \(1/k\)) for the gradient terms of the update. As a result, their convergence rates are sub-exponential and hence sub-optimal under assumptions of strongly convexity and smoothness. To address this issue, a number of recent works introduce an additional variable in the state vectors to track past gradients (see e.g., [6, 14–16]). One of the earlier works in this direction is the EXTRA algorithm proposed by Shi et al. [6], which uses the information from past two iterations to perform each update. For smooth problems, EXTRA provably achieves \(O(\frac{1}{k})\) convergence rate under the convexity assumption and exponential convergence under the strong convexity assumption, respectively.

A closely relevant literature is on continuous-time distributed GD, where the algorithms are constructed by a set of ordinary differential equations (ODEs). These works are mostly based on the idea of integral feedback, which can be thought as the continuous-time analog of gradient tracking. In this case, each agent uses an integration term as a part of the ODE (see e.g., [17–20]). In these works, the analysis is carried out by proving the Lyapunov stability for the corresponding continuous-time dynamics, and exponential stability can be obtained in certain cases [19].

2) Distributed Mirror Descent: Distributed version of the MD algorithm has been studied by a number of recent papers for stochastic optimization [21, 22] and online optimization [23, 24]. Doan et al. [25] provide convergence results on both centralized and decentralized MD algorithms. Similar to distributed GD, many works employ a diminishing step-size sequence. In the stochastic setup, the choice of diminishing step-size is necessary, as with a constant step-size the algorithm will only converge to a neighborhood of the optimal solution. In online setup, again the diminishing step-size sequence only suits convex problems.

For strongly convex problems, the continuous-time algorithm in [9, 26] and the discrete-time algorithm in [27] both adapt the integral feedback (or gradient tracking) method and propose algorithms that do not suffer from convergence speed degradation. Specifically, Sun et al. [9] propose a continuous-time distributed MD that achieves a local exponential rate for strongly convex problems, and Yu et al. [27] provide the \(O(\frac{1}{k})\) convergence speed under the convexity assumption in discrete time. Nevertheless, the exact exponential rate of distributed MD for strongly convex and smooth problems remains an open problem, studied in the current work.

3) Integral Quadratic Constraint: Deriving convergence rates for iterative optimization algorithms in the worst-case is an integral part of algorithm design. However, this procedure is not principled, requires a case-by-case analysis, and might lead to conservative rates. To automate convergence analysis and derive sharp convergence rates, several past works have used Integral Quadratic Constraints (IQCs) and Semidefinite Programming in various settings [10, 11, 28–32], pioneered by the work in [10]. IQCs are a tool from robust control to analyze dynamical systems that contain components that are nonlinear, uncertain, or difficult to model [33]. The basic idea is to abstract these troublesome components by constraints on their input and output signals. This approach to algorithm analysis can also guide the search for parameter selection in algorithm design. [34] is of particular relevance to our work, which provides an IQC-based analysis of some distributed GD algorithms. We remark that we leverage QCs for analyzing MD in both centralized and distributed settings. Our derived results are novel and not known to the best of our knowledge, and IQC does not serve as an alternative analysis tool for MD.

II. Preliminaries

A. Notations

We denote the real line by \(\mathbb{R}\), the set of \(n\)-dimensional vectors by \(\mathbb{R}^n\), and the set of matrices with \(m\) rows and \(n\) columns by \(\mathbb{R}^{m \times n}\), respectively. The identity matrix of dimension \(n\) is denoted by \(I_n\) and the \(n\)-dimensional vector with all entries 1 is represented by \(1_n\). We use \(\otimes\) to denote the Kronecker product. We denote the \(p\)-norm by \(\|\cdot\|_p\). The positive (negative) semi-definiteness of matrix \(M\) is denoted as \(M \succeq 0\) (\(M \preceq 0\)), respectively. We use \(\mathbf{1}\) to denote the \(i\)-th component of vector \(x\) and \([X]_{ij}\) to denote the \(ij\)-th entry of matrix \(X\), respectively.

Definition 1 (Strong convexity). A differentiable function \(f : \mathbb{R}^d \to \mathbb{R}\) is \(\mu_f\)-strongly convex on \(X\) if the following equality is true for all \(x, y \in X\):

\[
f(x) + \nabla f(x)^\top (y - x) + \frac{\mu_f}{2} \|y - x\|_2^2 \leq f(y).
\]

Definition 2 (Lipschitz smoothness). A differentiable function \(f : \mathbb{R}^d \to \mathbb{R}\) is \(L_f\)-smooth on \(X\) if the following inequality is true for all \(x, y \in X\):

\[
f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L_f}{2} \|y - x\|_2^2.
\]
In this work, we focus on strongly convex and smooth problems both in centralized and distributed regimes.

**Assumption 1.** The objective function \( f : \mathbb{R}^d \to \mathbb{R} \) is \( \mu_f \)-strongly convex and \( L_f \)-smooth.

We further denote the condition number of function \( f \) by \( \kappa_f \triangleq \frac{L_f}{\mu_f} \geq 1 \).

**Proposition 1.** Suppose \( f \) is \( \mu_f \)-strongly convex and \( L_f \)-smooth on \( \mathcal{X} \). Then, the following inequality holds for all \( x, y \in \mathcal{X}, u \in \nabla f(x), v = \nabla f(y) \):

\[
\begin{bmatrix} x - y \\ u - v \end{bmatrix}^\top \begin{bmatrix} -\frac{1}{\mu_f} I_d & \frac{1}{\mu_f + 1} I_d \\ \frac{1}{2} I_d & -\frac{1}{\mu_f + 1} I_d \end{bmatrix} \begin{bmatrix} x - y \\ u - v \end{bmatrix} \geq 0
\]

The above quadratic constraint follows from the combination of strong convexity and Lipschitz smoothness \([10, 35]\).

**B. Centralized Mirror Descent Algorithm**

The focus of this work is on the mirror descent algorithm. We start by providing some background on the centralized MD algorithm. For simplicity in the exposition, we first study the unconstrained case. Extensions to the constrained case will be deferred to Section VI.

We start with the GD algorithm, whose update is equivalent to the following minimization,

\[
x^{(k+1)} = \arg\min_x \left\{ f(x^{(k)}) + \nabla f(x^{(k)})^\top (x - x^{(k)}) + \frac{1}{2\eta} \|x - x^{(k)}\|_2^2 \right\}
\]

where \( \eta > 0 \) is the step size. In each iteration, the algorithm seeks to minimize a first-order approximation of the function with a Euclidean regularizer. As a generalization of gradient descent, MD replaces the Euclidean norm with Bregman divergence, which is defined with respect to a distance generating function (DGF) \( \phi : \mathbb{R}^d \to \mathbb{R} \) as follows:

\[
D_\phi(x, x') \triangleq \phi(x) - \phi(x') - \langle \nabla \phi(x'), x - x' \rangle.
\]

**Assumption 2.** The distance generating function \( \phi : \mathbb{R}^d \to \mathbb{R} \) is \( \mu_\phi \)-strongly convex and \( L_\phi \)-smooth.

The centralized (unconstrained) MD algorithm with step size \( \eta \) is written as

\[
x^{(k+1)} = \arg\min_x \left\{ f(x^{(k)}) + \nabla f(x^{(k)})^\top (x - x^{(k)}) + \frac{1}{\eta} D_\phi(x, x^{(k)}) \right\},
\]

where if we choose the Bregman divergence to be the Euclidean distance, the update above reduces to gradient descent.

We can also view the MD update through a different lens using the convex conjugate of function \( \phi \). The convex conjugate of function \( \phi \), denoted by \( \phi^* \), is defined as follows:

\[
\phi^*(z) \triangleq \sup_{x \in \mathbb{R}^d} \{ \langle x, z \rangle - \phi(x) \}.
\]

Assumption 2 guarantees that \( \phi^* \) is \( L_\phi^{-1} \)-strongly convex and \( \mu_\phi^{-1} \)-smooth. We refer the reader to \([36]\) for further details.

Correspondingly, the following equivalence can be established, where

\[
z' = \nabla \phi(x') \iff x' = \nabla \phi^*(z').
\]

Then, the centralized MD update can be rewritten in the following form,

\[
\begin{align*}
(z^{(k+1)} - z^{(k)}) &= -\eta \nabla f(x^{(k)}) \\
(x^{(k+1)} - x^{(k)}) &= \nabla \phi^*(z^{(k+1)}),
\end{align*}
\]

So far, we can see that MD is more general than GD in that we can exploit the geometry of the problem using an appropriate choice of \( \phi \). In Section III, we provide several examples to motivate the use of MD.

**C. State-Space Representation**

Iterative algorithms can be seen as dynamical systems. In particular, they can often be represented as feedback interconnection of a linear system (the algorithm itself) and a nonlinearity (e.g., the gradient of the function we are minimizing). In this paper, we consider the following state-space model

\[
\begin{align*}
\xi^{(k+1)} &= A \xi^{(k)} + B \zeta^{(k)} \\
\zeta^{(k)} &= \psi(C \xi^{(k)}),
\end{align*}
\]

where at iteration \( k = 0, 1, \cdots \), \( \xi^{(k)} \in \mathbb{R}^d \) is the state of the system, \( \zeta^{(k)} \in \mathbb{R}^m \) is the control input, \( A, B, C \) are matrices of appropriate dimensions, and \( \psi \) is a memoryless nonlinear function generating the feedback signal \( \psi(C \xi^{(k)}) \). By choosing the matrices \( A, B \) and \( C \) appropriately, a variety of iterative algorithms can be represented in this canonical form \([10, 11, 27, 35]\). For example, for the gradient method \( x^{(k+1)} = x^{(k)} - \eta \nabla f(x^{(k)}) \), we have \( A = I_d, B = -\eta I_d, C = I_d \) and \( \psi = \nabla f \). We note that for well-designed algorithms, the fixed-point equations of the dynamical system

\[
\begin{align*}
\xi^* &= A \xi^* + B \zeta^* \\
\zeta^* &= \psi(C \xi^*),
\end{align*}
\]

must correspond to the optimality conditions of the optimization problem.

For the MD algorithm in (5), we can rewrite it as

\[
\begin{align*}
z^{(k+1)} &= z^{(k)} - \eta \nabla f \circ \nabla \phi^*(z^{(k)}). \\
(x^{(k+1)} - x^{(k)}) &= \nabla \phi^*(z^{(k+1)}),
\end{align*}
\]

Here, \( z^{(k)} \) is the state variable, and the nonlinearity \( \psi \) is the composition of two nonlinearities, namely the gradient of \( f \) and the gradient of \( \phi^* \). In Section IV, we will analyze the convergence of the centralized MD using QCs and SDP.

**III. Motivation for Mirror Descent**

In this section, we discuss several optimization problems where non-Euclidean regularizers are more suitable given the problem geometry, and therefore, MD is a more natural optimization technique to exploit instead of GD.
1) Convex Clustering: In the exemplar-based convex clustering [39], we are interested in solving the following maximum likelihood problem

$$\max_{x \in \mathcal{X}} f(x) = \frac{1}{n} \sum_{i=1}^{n} \log \left( \sum_{j=1}^{n} \frac{|x|^j}{|y|^j} \right),$$

where $$\mathcal{X} = \{ x^T 1_n = 1, |x|^j \geq 0 \text{ for } j = 1, \ldots, n \}$$ is the n-dimensional probability simplex, and $$K$$ can be thought as a kernel matrix in specific problems [40]. In this case, since the decision variable lives on a probability simplex, the MD algorithm with Kullback-Leibler (KL) divergence can be applied to maximize the likelihood. The KL-divergence between $$x, y \in \mathcal{X}$$ is defined as

$$D_{KL}(x, y) = \sum_{i=1}^{n} |x|^i \log \left( \frac{|x|^i}{|y|^i} \right).$$

In fact, the exponentiated gradient descent method proposed for the problem above is a special case of MD. It is an easy exercise to show that the objective function in (9) is strongly convex when $$K$$ is a positive-definite kernel.

2) Matrix Optimization with Regularization: Consider an optimization problem where the decision variable is a positive semi-definite matrix $$X \succeq 0$$. One example is the following optimization

$$\min_{X \in \mathcal{X}} g(X) + Q_f(X, Y),$$

where $$\mathcal{X} = \{ X \in \mathbb{R}^{d \times d} : X \succeq 0 \}$$ is the positive semi-definite cone and $$Q_f(X, Y)$$ denotes the f-divergence between the decision variable X and a given matrix Y. This problem appears in many applications including shape classification models, inverse covariance estimation, and brain network analysis [41]. For this problem geometry, possible choices of Bregman divergence are Log-Det (LT) divergence and von Neumann (vN) divergence, defined as

$$D_{LT}(X, Y) = \text{Tr} \left[ X Y^{-1} \right] - \text{log det}[X Y^{-1}] - d$$

$$D_{vN}(X, Y) = \text{Tr} \left[ X \log X - X \log Y - Y + X \right].$$

3) Poorly Conditioned Optimization: When the condition number $$\kappa_f$$ is large, we say that the optimization problem is poorly conditioned. A common remedy for this situation is to use preconditioned gradient descent, which is equivalent to using Mahalanobis distance as the Bregman divergence. In particular, $$\phi(x) = \frac{1}{2} x^T P^{-1} x$$ for a positive definite matrix $$P$$, which amounts to performing GD with gradients that are pre-multiplied by matrix $$P$$.

IV. CENTRALIZED MIRROR DESCENT CONVERGENCE WITH QC'S

In this section, we analyze the centralized MD using QC's. We propose a Lyapunov function suitable for exponential convergence of the algorithm (Theorem 2). Then, one can pose an optimization problem where the exponential rate needs to be minimized subject to an LMI. We show that this optimization can be transformed to an SDP (Proposition 3). We also provide a feasible solution for such SDP (Proposition 4), where the corresponding convergence rate coincides with the optimal rate of GD when $$\mu_\phi = L_\phi$$ (i.e., when Bregman divergence is set to be the Euclidean distance).

A. Exponential Rate of Convergence

Let us consider the centralized MD update [5], where the state variable $$\xi(k) = x(k)$$ and the input variable $$\zeta(k) = u(k)^T$$ with $$u(k) = \nabla f(x(k))$$. In the following theorem, we characterize an LMI that depends on parameters of the function ($$\mu_f$$ and $$L_f$$), parameters of DGF ($$\mu_\phi$$ and $$L_\phi$$), and several decision variables (including the step-size $$\eta$$ and the convergence rate $$\rho$$). We prove that if the LMI is satisfied, the update converges exponentially fast with the rate $$\rho$$.

Theorem 2. Let Assumptions [72] hold and define matrices $$M, M_f, M_\phi$$ as follows,

$$M = \begin{bmatrix} \frac{1}{2\mu_f} I_d & 0 & 0 \\ 0 & 0 & \frac{-n}{2} I_d \\ 0 & \frac{-n}{2} \mu_i I_d & \frac{\mu_i}{2} I_d \end{bmatrix},$$

$$M_f = \begin{bmatrix} 0 & 0 & \frac{\mu_f L_f}{2} I_d \\ 0 & \frac{\mu_f L_f}{2} I_d & 0 \\ \frac{\mu_f L_f}{2} I_d & 0 & 0 \end{bmatrix},$$

$$M_\phi = \begin{bmatrix} -\frac{1}{\mu_i + L_\phi} I_d & \frac{\mu_i}{2 \mu_i + L_\phi} I_d & 0 \\ \frac{\mu_i}{2 \mu_i + L_\phi} I_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

If there exist some $$\rho \in (0, 1), \eta > 0, \sigma_f \geq 0, \sigma_\phi \geq 0, $$ such that the following matrix inequality holds

$$M + \sigma_f M_f + \sigma_\phi M_\phi \preceq 0,$$

then the mirror descent algorithm in (5) converges exponentially fast with the rate of $$\rho$$. In particular,

$$\|x(k) - x^*\|^2 = \Omega(\rho^k).$$

Proof. We define the stacked vector

$$e^{(k)} = \begin{bmatrix} z^{(k)} - z^* \\ x^{(k)} - x^* \\ u^{(k)} - u^* \end{bmatrix}. $$

Then, from Proposition 1 we have the following inequalities

$$e^{(k)^T} M_f e^{(k)} \geq 0, \quad e^{(k)^T} M_\phi e^{(k)} \geq 0.$$

Let us now consider the following Lyapunov candidate

$$V^{(k)} = \rho^{-k} D_{\phi^*}(z^{(k)}, z^*).$$

Recall that $$\phi^*$$ is $$L_{\phi^*}^{-1}$$-strongly convex and $$\mu_{\phi^*}^{-1}$$-smooth, so the Lyapunov function is indeed non-negative and continuously differentiable. Using Lemma 9 (provided in the appendix), we can calculate the difference between Lyapunov functions of two consecutive iterations as

$$V^{(k+1)} - V^{(k)} \leq \rho^{-k-1} e^{(k)^T} M e^{(k)}. $$

(14)
Proposition 3. The optimization problem in (17) is equivalent to the following SDP:

\[
\begin{align*}
\text{minimize} & \quad \rho \\
\text{subject to:} & \quad 0 < \rho \leq 1 \\
& \quad \eta > 0 \\
& \quad \sigma_\phi \geq 0 \\
& \quad \sigma_f \geq 0 \\
& \quad M + \sigma_f M_f + \sigma_\phi M_\phi \preceq 0.
\end{align*}
\]

The proof of the proposition can be found in the appendix.

\[ V^{(k+1)} - V^{(k)} \leq \rho^{-k+1} e^{(k)^\top} M e^{(k)} \leq \rho^{-k+1} \left( e^{(k)^\top} M e^{(k)} + \sigma_f e^{(k)^\top} M f e^{(k)} + \sigma_\phi e^{(k)^\top} M_\phi e^{(k)} \right) = \rho^{-k+1} e^{(k)^\top} \left( M + \sigma_f M_f + \sigma_\phi M_\phi \right) e^{(k)}. \] (15)

If there exists variables such that the LMI in (12) holds, the Lyapunov function decreases monotonically, which yields

\[ D_\phi, (z^{(k)}, z^*) = \rho^k V^{(k)} \leq \rho^k V^{(0)} = \rho^k D_\phi, (z^{(0)}, z^*). \] (16)

Observing \[ D_\phi, (z^{(k)}, z^*) = D_\phi(x^*, x^{(k)}) \]
and
\[ \frac{\mu_\phi}{2} \| x^{(k)} - x^* \|^2 \leq D_\phi(x^*, x^{(k)}), \]
completes the proof.

B. An SDP Approach to Optimize the Rate

Theorem 2 provides a matrix inequality that establishes the exponential convergence rate of MD. We are interested in finding the best certifiable convergence rate, which amounts to solving following optimization problem

\[
\begin{align*}
\text{minimize} & \quad \rho \\
\text{subject to:} & \quad 0 < \rho \leq 1 \\
& \quad \eta > 0 \\
& \quad \sigma_\phi \geq 0 \\
& \quad \sigma_f \geq 0 \\
& \quad M + \sigma_f M_f + \sigma_\phi M_\phi \preceq 0.
\end{align*}
\] (17)

However, since the matrix \( M \) in (11) is not linear in \( \eta \), the above problem is convex. In the following proposition, we show that (17) can be transformed to an SDP, which provides a numerical framework to optimize the convergence rate \( \rho \).

Proposition 4. Consider the SDP formulation in (18) and set \( \eta = \sigma_f = \frac{2 \mu_\phi}{\mu_f + L_f} \). Then, the LMI becomes

\[
\begin{bmatrix}
-(1-\rho) I_d + \frac{1}{\mu_\phi + \mu_f + L_f} I_d - \sigma_\phi I_d & -\sigma_\phi \frac{L_f}{\mu_f + L_f} I_d + \frac{2 \mu_\phi \mu_f L_f}{(\mu_f + L_f)^2} I_d + \frac{\sigma_\phi \mu_f L_f}{\mu_f + L_f} I_d \\
-\sigma_\phi \frac{L_f}{\mu_f + L_f} I_d + \frac{2 \mu_\phi \mu_f L_f}{(\mu_f + L_f)^2} I_d + \frac{\sigma_\phi \mu_f L_f}{\mu_f + L_f} I_d & -\sigma_\phi I_d
\end{bmatrix} \succeq 0,
\] (19)

and \( \rho_{\text{opt}} \), the analytical solution of the SDP for \( \rho \), can be calculated as

\[
\rho_{\text{opt}} = 1 - \frac{4 \mu_f L_f}{(\mu_f + L_f)^2} \kappa_\phi^2.
\] (20)

The proof of the proposition is presented in the appendix. \( \rho_{\text{opt}} \) provides an upper bound on the convergence rate of MD algorithm. Interestingly, this rate recovers the optimal rate of GD as a special case.

Corollary 5. Let \( \phi(x) = \frac{1}{2} \| x \|^2 \), where the Bregman divergence is the Euclidean distance. Then, (20) coincides with the optimal convergence rate of gradient descent.

Proof. If \( \phi(x) = \frac{1}{2} \| x \|^2 \), we have that \( \phi^*(z) = \frac{1}{2} \| z \|^2 \) and it is immediate that (5) is equivalent to GD. In this case, the condition number \( \kappa_\phi = \frac{\mu_f}{\mu_\phi} = 1 \), and \( \rho_{\text{opt}} \) reduces to the optimal convergence rate for GD.

V. CONVERGENCE ANALYSIS OF DISTRIBUTED MIRROR DESCENT

In this section, we consider the distributed MD algorithm. We first present the problem setup and then propose the distributed MD algorithm, which will then be analyzed using QCs and SDP. We also provide a comparison of the obtained convergence guarantee with some existing algorithms on distributed GD.

A. Problem Setup

In the distributed setup, the agents form a network captured by a simple graph \( G = (V, E) \). The agents are denoted by nodes \( V = [n] \) and the connection between two agents \( i \) and \( j \) is captured by the edge \( \{i, j\} \in E \). We use \( N_i \triangleq \{j \in V : \{i, j\} \in E\} \) to denote the neighborhood of agent \( i \). The graph Laplacian is represented by \( \mathcal{L} \in \mathbb{R}^{n \times n} \).

Assumption 3. The graph \( G \) is undirected and connected, i.e., there exists a path between any two distinct agents \( i, j \in V \).

The connectivity assumption implies that \( \mathcal{L} \) has a unique null eigenvalue. That is, \( \mathcal{L} 1_n = 0 \), and correspondingly we can say that \( 1_n \) is in the null space of \( \mathcal{L} \).

Assumption 4. All local functions \( f_i : \mathbb{R}^d \to \mathbb{R} \) are \( \mu_f \)-strongly convex and \( L_f \)-smooth on \( \mathbb{R}^d \).

This assumption will automatically imply Assumption 1 on the global objective function.
B. Exponential Convergence of Distributed Mirror Descent

We first introduce the distributed MD update, where for any agent $i$ in the network, we have

$$
\begin{align*}
    z_i^{(k+1)} &= z_i^{(k)} - \eta_1 \left( \nabla f_i(x_i^{(k)}) + y_i^{(k)} \right) \\
    &\quad - \eta_2 \sum_{j \in N_i} (z_i^{(k)} - z_j^{(k)}), \\
    y_i^{(k+1)} &= y_i^{(k)} + \eta_2 \sum_{j \in N_i} (z_i^{(k)} - z_j^{(k)}), \\
    x_i^{(k+1)} &= \nabla \phi^*(z_i^{(k+1)}).
\end{align*}
\tag{21}
$$

The dual update $z_i^{(k)}$ uses private gradient information as well as the dual variables in the local neighborhood. It also depends on a variable $y_i^{(k)}$ which acts like an integrator. This algorithm is similar to the discretized version of the distributed MD algorithm in [26] using the idea of integral feedback. However, the method differs slightly in the local averaging in that the algorithm in [26] performs local averaging with respect to the primal variable, and here the averaging is done on the dual variable.

Let us now rewrite the update (21) in the matrix form. We stack the vectors of local variables as follows

$$
\begin{align*}
    z^{(k)} &= \begin{bmatrix} z_1^{(k)} \\ \vdots \\ z_n^{(k)} \end{bmatrix}, \\
    y^{(k)} &= \begin{bmatrix} y_1^{(k)} \\ \vdots \\ y_n^{(k)} \end{bmatrix}, \\
    x^{(k)} &= \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}.
\end{align*}
$$

The update then takes the form

$$
\begin{align*}
    z^{(k+1)} &= z^{(k)} - \eta_1 (u^{(k)} + y^{(k)}) - \eta_2 (L \otimes I_d) z^{(k)} \\
    y^{(k+1)} &= y^{(k)} + \eta_2 (L \otimes I_d) z^{(k)} \\
    x^{(k)} &= \nabla \phi^*(z^{(k)}) \\
    u^{(k)} &= \nabla f(x^{(k)}),
\end{align*}
\tag{22}
$$

where

$$
\begin{align*}
    u^{(k)} &= \nabla f(x^{(k)}) = \begin{bmatrix} \nabla f_1(x_1^{(k)}) \\ \vdots \\ \nabla f_n(x_n^{(k)}) \end{bmatrix},
\end{align*}
$$

for simplicity. It is evident that the dynamics of the control system (22) relies on the network structure through the dependence on the Laplacian of the graph capturing the network. Since $L \in \mathbb{R}^{n \times n}$, the LMIs will consist of matrices whose dimension scale with $O(n)$, which is not suitable when $n$ is large. We transform the updates such that the dependence on the full structure of the network is avoided. Define

$$
W \triangleq I_n - \eta_2 L = \Delta W + \frac{1}{n} 1_n 1_n^T,
$$

and further denote the spectral norm of $\Delta W$ by

$$
\lambda \triangleq \|\Delta W\|. \tag{23}
$$

We can now rewrite (22) using additional variable $v^{(k)}$ as follows

$$
\begin{align*}
    z^{(k+1)} &= (\frac{1}{n} 1_n 1_n^T \otimes I_d) z^{(k)} - \eta_1 (u^{(k)} + y^{(k)}) + v^{(k)} \\
    y^{(k+1)} &= y^{(k)} + ((I_n - \frac{1}{n} 1_n 1_n^T) \otimes I_d) z^{(k)} - v^{(k)} \\
    v^{(k)} &= (\Delta W \otimes I_d) z^{(k)} \\
    x^{(k)} &= \nabla \phi^*(z^{(k)}) \\
    u^{(k)} &= \nabla f(x^{(k)}).
\end{align*}
\tag{24}
$$

We can see later on that the analysis of the update above only depends on the spectral gap of $W$, which is common in distributed optimization. The following lemma serves as a tool for simplifying the SDP problem for distributed MD, which is useful for the proof of our main result.

**Lemma 6** (Lemma 6 in [34]). Suppose that matrices $J_1, J_2$ satisfy $J_1^2 = J_1, J_2^2 = J_2, J_1 J_2 = J_2 J_1 = 0$, then if for some matrix $Q \triangleq Q_1 \otimes J_1 + Q_2 \otimes J_2$, then the following are equivalent.

1) $Q \succeq 0$,
2) $Q_1 \succeq 0, Q_2 \succeq 0$.

In the following theorem, we present the main result of this section. We provide two LMIs to characterize the convergence rate of distributed MD. The LMIs are written in terms of several decision variables, including the step-size $\eta_1$ and the convergence rate $\rho$. If we can find a feasible solution for these LMIs, the distributed MD is guaranteed to converge exponentially fast.

**Theorem 7.** Let Assumptions 2, 3, 4 hold and define the following matrices,

$$
\begin{align*}
    A_1 &= \begin{bmatrix} 0 & -\eta_1 \\ 1 & 1 \end{bmatrix}, \\
    B_1 &= \begin{bmatrix} 0 & -\eta_1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \\
    A_2 &= \begin{bmatrix} 1 & -\eta_1 \\ 0 & 1 \end{bmatrix}, \\
    B_2 &= \begin{bmatrix} 0 & -\eta_1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \\
    F_1 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
    G_1 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
    F_2 &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\
    G_2 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
    R_1 &= I_5, R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.
\end{align*}
$$
and also let

\[ M_f = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{\mu_f L_f}{\mu_f^2 + L_f} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \]

\[ M_\lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \]

\[ M_\phi = \begin{bmatrix} \frac{1}{\mu_\phi} L_\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \]

If there exists \( \rho \in (0, 1), \eta_1 \geq 0, P \in \mathbb{R}^{2 \times 2} \geq 0, \sigma_f \geq 0, \sigma_\phi \geq 0, \lambda \geq 0, \) such that the following matrix inequalities hold for \( i = 1, 2 \)

\[ R_i^T \begin{bmatrix} A_i^T P A_i - \rho P & A_i^T P B_i \\ B_i^T P A_i & B_i^T P B_i \end{bmatrix} + \sigma_f M_f + \sigma_\lambda M_\lambda \\
+ \sigma_\phi M_\phi \] \( R_i \leq 0, \) \hspace{1cm} (25)

the distributed MD algorithm (22) initialized at \( y^{(0)} = 0 \) converges exponentially with a rate of \( \rho. \)

**Proof.** We define the state vector \( \xi^{(k)} = \begin{bmatrix} z^{(k)}^T & y^{(k)}^T \end{bmatrix}, \) input vector \( \zeta^{(k)} = \begin{bmatrix} x^{(k)}^T & u^{(k)}^T \end{bmatrix}, \) and the stacked variable of the system as

\[ e^{(k)} = \begin{bmatrix} z^{(k)} - z^* \\ y^{(k)} - y^* \\ x^{(k)} - x^* \\ u^{(k)} - u^* \\ v^{(k)} - v^* \end{bmatrix}. \]

Then, the dynamics (24) can be written as

\[ \begin{bmatrix} z^{(k+1)} \\ y^{(k+1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{n} 1_n 1_n^T & I_d \\ (I_n - \frac{1}{n} 1_n 1_n^T) & I_d \end{bmatrix} \begin{bmatrix} -\eta_1 I_{nd} \\ I_{nd} \end{bmatrix} \begin{bmatrix} z^{(k)} \\ y^{(k)} \end{bmatrix} + \begin{bmatrix} x^{(k)} \\ u^{(k)} \\ v^{(k)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \hspace{1cm} (26) \]

Additionally, we know the following constraints on the updates,

\[ \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1_n 1_n^T & I_d \end{bmatrix} \begin{bmatrix} x^{(k)} \\ u^{(k)} \\ v^{(k)} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1_n 1_n^T & I_d \end{bmatrix} \begin{bmatrix} z^{(k)} \\ y^{(k)} \end{bmatrix} . \hspace{1cm} (27) \]

We can rewrite (26) and (27) as

\[ \begin{bmatrix} \xi^{(k+1)} \\ 0 \end{bmatrix} = \begin{bmatrix} A & B \\ F & G \end{bmatrix} \xi^{(k)}, \hspace{1cm} (28) \]

with proper matrices \( A, B, F, G. \) By stacking the columns of the null space of \( [F \ G] \), we are able to construct a matrix \( R, \) we have \( \text{span}(R) = \text{range}(e^{(k)}), \) which will be useful in later analysis.

Recalling Proposition [1] based on Assumptions [2] and [3] we have that

\[ e^{(k)^T} (M_f \otimes I_{nd}) e^{(k)} \geq 0 \]
\[ e^{(k)^T} (M_\phi \otimes I_{nd}) e^{(k)} \geq 0. \]

Finally, note that for the mapping \( z \mapsto \Delta W z, \) given that \( \| \Delta W \| = \lambda, \) we can write

\[ e^{(k)^T} (M_\lambda \otimes I_{nd}) e^{(k)} \geq 0. \]

Now let us define the Lyapunov function

\[ V^{(k)} = \rho^{-k}(\xi^{(k)} - \xi^*)^T P'(\xi^{(k)} - \xi^*), \]

where \( P' = P \otimes I_{nd}. \)

Then, using (28) we can derive

\[ V^{(k+1)} - V^{(k)} = \rho^{-k-1} e^{(k)^T} \begin{bmatrix} A^T P' A - P' \rho P \ A^T P' B \ B^T P' A \ B^T P' B \end{bmatrix} e^{(k)} . \hspace{1cm} (29) \]

If there exist variables such that the following LMI holds

\[ R^T \begin{bmatrix} A^T P' A - P' \rho P \ A^T P' B \ B^T P' A \ B^T P' B \end{bmatrix} + \sigma_f M_f \otimes I_{nd} + \sigma_\lambda M_\lambda \otimes I_{nd} + \sigma_\phi M_\phi \otimes I_{nd} \leq 0, \]

then for any \( e^{(k)}, \) we have that

\[ \rho^{-k-1} e^{(k)^T} \begin{bmatrix} A^T P' A - P' \rho P \ A^T P' B \ B^T P' A \ B^T P' B \end{bmatrix} e^{(k)} \leq 0. \]

The Lyapunov function decreases monotonically, which means

\[ (\xi^{(k)} - \xi^*)^T P'(\xi^{(k)} - \xi^*) \leq \rho^k (\xi^{(0)} - \xi^*)^T P'(\xi^{(0)} - \xi^*), \]

which implies the exponential convergence of the distributed MD algorithm.

Next, we simplify the LMI such that the dimension is not dependent on the agent number \( n. \) Our approach follows that of [34]. Define \( J_1, J_2 \) in Lemma [6] as \( J_1 = (I_n - \frac{1}{n} 1_n 1_n^T) \otimes I_d, J_2 = \frac{1}{n} 1_n 1_n^T \otimes I_d. \) It is easy to verify that these matrices satisfy the constraints in Lemma [6]. We then have that

\[ A = A_1 \otimes J_1 + A_2 \otimes J_2, \]
\[ B = B_1 \otimes J_1 + B_2 \otimes J_2, \]
\[ F = F_1 \otimes J_1 + F_2 \otimes J_2, \]
\[ G = G_1 \otimes J_1 + G_2 \otimes J_2, \]

and also \( R_1, R_2 \) are calculated by stacking the columns of the null space of \( [F_1 \ G_1] \) and \( [F_2 \ G_2], \) respectively.

Since matrices \( J_1, J_2 \) do not interfere with each other, the dynamics (28) can be split into two terms and analyzed independently. Using Lemma [6], feasible solutions that satisfy (29) will also be feasible solutions for (29), and this completes our proof.

The theorem provides two LMI that establish the exponential convergence rate of distributed MD. As we can see the LMI are more involved compared to the centralized case,
and it is more challenging to find even a suboptimal analytical rate.

We finally remark that common analysis on distributed MD involves general primal-dual norms, whereas QCs are defined with respect to the Euclidean norm. The use of general primal-dual norms in non-strongly convex problems helps with improving the rate up to a multiplicative factor of \( \sqrt{d} \). However, since in our case the rate is exponentially fast, a more general analysis can only change the iteration complexity by at most logarithmic factors of \( d \), which is an interesting avenue to investigate in the future.

C. Evaluating the Tightness of Our Result

For the distributed MD algorithm, we provide numerical results based on Theorem 7. First, we demonstrate the influence of the network structure, and then we compare the rate recovered by Theorem 7 to existing theoretical rates on distributed GD when it achieves exponential convergence.

1) Impact of the Network Structure on Convergence Rate:

We calculate the worst-case convergence rate with several choices of \( \lambda \) and plot it with respect to the step-size \( \eta \). We set the local functions to have condition number \( \kappa_f = 2 \) and the DGF to have condition number \( \kappa_\phi = 2 \). Each curve in the plot represents a certain \( \lambda \) and is obtained by scanning feasible values for the decision variables in the LMIs (25).

Fig. 1. Convergence rates generated from Theorem 7 versus step-size \( \eta \).

From Fig. 1, we can see that there exists an optimal step-size to obtain the best convergence rate, and that as \( \lambda \) increases, the best rate becomes worse. Hence, for any given network structure and its corresponding Laplacian matrix, we should select \( \eta_2 \) such that \( \lambda \) is minimized. This is consistent with results on distributed optimization, where having a larger spectral gap improves the performance.

2) Comparison with Distributed Gradient Descent:

To the best of our knowledge, there is currently no work that provides an exact rate for the exponential convergence of distributed MD algorithm. Hence, we select two previous works on distributed GD, namely [6] and [14], and compare our performance with the theoretical rates provided in these works. In order to provide a fair comparison, we must set \( \kappa_\phi = 1 \) to ensure that MD reduces to GD. We also set the local functions to have condition number \( \kappa_f = 3 \).

Of the two related works above, EXTRA [6] is of particular relevance to our algorithm. If the matrix \( \hat{W} \) in EXTRA is set to be \( 2W \), the EXTRA algorithm coincides with our algorithm with the exception of having a coefficient difference of \( \frac{1}{2} \) for the tracking term. Note that the theoretical convergence rate of EXTRA relies on the spectral norm of \( W \) as well as the smallest non-zero eigenvalue \( \lambda_n \) of \( W \). We plot the convergence rate of EXTRA under three different scenarios:

1) \( \lambda_n = \lambda \), (EXTRA pos)
2) \( \lambda_n = -\lambda \), (EXTRA neg)
3) \( \lambda_n \approx 0 \), (EXTRA)

Fig. 2. Comparison of the convergence rates for different methods

From Fig. 2, we can see that when \( \lambda \) is small, the rate recovered by Theorem 7 significantly outperforms EXTRA. As \( \lambda \) increases, the convergence rate calculated for our method starts increasing. We also include the theoretical convergence results from Qu et al. [14], which is consistently outperformed by EXTRA.

Note that the point of this plot is not to declare a winner among algorithms. The goal is to show that the richness of the Lyapunov function and QC analysis provides a machinery to obtain better convergence rates, especially compared to the rates that are algorithm specific. In this case, our algorithm can coincide with EXTRA, but still our analysis provides better rates. Our observation is in line with empirical results of [34].

VI. THE CONSTRAINED SETUP

In this section, we consider the constrained version of centralized MD,

\[
\begin{align*}
z^{(k+1)} &= z^{(k)} - \eta \nabla f(z^{(k)}) \\
s^{(k)} &= \nabla \phi^*(z^{(k)}) \\
x^{(k)} &= \arg \min_{x \in X} D_\phi(x, s^{(k)}),
\end{align*}
\] (31)
where $\mathcal{X}$ is a convex subset of $\mathbb{R}^d$. By defining $g(x) = 1_{\mathcal{X}}(x)$ as the indicator function of the set $\mathcal{X}$, we can write the above updates more explicitly as

$$z^{(k+1)} = z^{(k)} - \eta \nabla f(x^{(k)})$$

$$s^{(k)} = \nabla \phi^+(z^{(k)})$$

(32)

$$\nabla \phi(x^{(k)}) - \nabla \phi(s^{(k)}) \in \partial g(x^{(k)}),$$

where $\partial g$ denotes the subdifferential of $g$. Using the identity $z^{(k)} = \nabla \phi(s^{(k)})$, we arrive at

$$z^{(k+1)} = z^{(k)} - \eta \nabla f(x^{(k)})$$

$$z^{(k)} = \nabla \phi(x^{(k)}) - T_g(x^{(k)})$$

(33)

where $T_g(x^{(k)}) \in \partial g(x^{(k)})$ is any sub-gradient of $g$. Note that for the unconstrained case, $\mathcal{X} = \mathbb{R}^d$, we have $g(x) = 0$, and we obtain the same updates as in (5).

By defining $u^{(k)} = \nabla f(x^{(k)})$, $v^{(k)} = \nabla \phi(x^{(k)})$ and $w^{(k)} = T_g(x^{(k)})$, we can rewrite (33) as

$$z^{(k+1)} = z^{(k)} - \eta u^{(k)}$$

$$z^{(k)} = v^{(k)} - w^{(k)}$$

(34)

For the purpose of convergence analysis, define the vector $e^{(k)}$ as

$$e^{(k)} = \begin{bmatrix} z^{(k)} - z^* \\ x^{(k)} - x^* \\ u^{(k)} - u^* \\ v^{(k)} - v^* \end{bmatrix}.$$  

(35)

For convex sets $\mathcal{X}$, $g(x)$ is convex. Using the fact that the subdifferential $\partial g$ is monotone, we can write the following inequality

$$(u^{(k)} - w^{(k)})^T (x^{(k)} - x^*) \geq 0 \quad \forall k \geq 0,$$

which by observing that $u^{(k)} = e^{(k)} - z^{(k)}$ is equivalent to

$$e^{(k)} \begin{bmatrix} 0 \\ -I_d \\ -I_d \\ 0 \end{bmatrix} e^{(k)} \geq 0.$$  

(36a)

Furthermore, we can write two separate quadratic constraints for the relationships $u^{(k)} = \nabla f(x^{(k)})$ and $v^{(k)} = \nabla \phi(x^{(k)})$ to arrive at

$$e^{(k)} \begin{bmatrix} 0 \\ -\frac{\mu_f}{\mu_f + L_f} I_d \\ \frac{1}{2} I_d \end{bmatrix} e^{(k)} \geq 0$$

(36b)

and

$$e^{(k)} \begin{bmatrix} 0 \\ -\frac{\mu_o L_o}{\mu_o + L_o} I_d \\ \frac{1}{2} I_d \end{bmatrix} e^{(k)} \geq 0.$$  

(36c)

We present the constrained version of Theorem 2 in the following Proposition.

**Proposition 8.** Let $\mathcal{X} \subset \mathbb{R}^d$ be a convex set. Let Assumptions 1-2 hold and define matrices $M_g, M_f, M_o$ as in (36). Let

$$M = \begin{bmatrix} \frac{1-\rho}{2\rho} I_d & 0 & 0 & 0 \\ 0 & 0 & -\frac{\eta}{\rho} I_d & 0 \\ 0 & -\frac{\eta}{\rho} I_d & \frac{\eta^2}{\rho^2} I_d & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$  

(37)

If there exist some $\rho \in (0, 1), \eta > 0, \sigma_f \geq 0, \sigma_o \geq 0$, such that the following matrix inequality holds.

$$M + \sigma_f M_f + \sigma_o M_o + \sigma_g M_g \preceq 0.$$  

(38)

Then the constrained MD in (31) converges exponentially fast to the optimal solution with rate $\rho$.

The proof of Proposition 8 follows similar steps as in the proof of Theorem 2 and therefore, we omit it.

**VII. Conclusion**

In this paper, we proposed a semidefinite programming framework to characterize the exponential convergence rate of the mirror descent algorithm for both centralized and distributed settings, and under the assumption of strongly convex and smooth local objective functions. For the centralized case, we derived a closed-form feasible solution to the SDP for the convergence rate, which depends on the condition number of the distance generating function. For the decentralized case, we numerically derived the convergence rates using semidefinite programming. These SDPs do not scale with the ambient dimension and the network size. It would be interesting to derive analytical rates for the distributed case. Another important direction is the analysis of the mirror descent algorithm with primal-dual norms. This is a challenging problem as current SDP approaches rely on the Euclidean norm and they do not lend themselves to general primal-dual norms.

**APPENDIX**

**A. Preliminary Lemma for Proof of Theorem 2**

In the following lemma, we calculate an upper bound for the difference between Lyapunov functions of two consecutive iterations.

**Lemma 9.** Let Assumptions 1-2 hold and consider the Lyapunov function $V(k) = \rho^{-k} D_{\phi^+}(z^{(k)}, z^*)$. Then, the following inequality,

$$V(k+1) - V(k) \leq \rho^{-k-1} e^{(k)^T} M e^{(k)},$$

is satisfied, where $M$ is given in Theorem 2 and $e^{(k)}$ is defined in (13).
Proof. From the definition of Lyapunov function and Bregman divergence, we have that
\[
V^{(k+1)} - V^{(k)} = \rho^{k-1}D_\phi(z^{(k+1)}, z^*) - \rho^{k}D_\phi(z^{(k)}, z^*)
\]
\[
= \rho^{k-1}(\phi^*(z^{(k+1)})) - \phi^*(z^*) - (\nabla \phi^*(z^*) - \nabla \phi^*(z^{(k+1)})) - \rho^{k}\phi^*(z^{(k+1)}) - \phi^*(z^*) + \rho^{1}\phi^*(z^{(k)} - z^*) - \eta u(k)
\]
Since \(\phi^*\) is \(\mu^{-1}_\phi\)-smooth, we get
\[
V^{(k+1)} - V^{(k)} 
\leq \rho^{k-1}[\phi^*(z^{(k)}) + \langle x^{(k)}, -\eta u(k) \rangle] + \frac{\eta^2}{2\mu_\phi} ||u^{(k)}||^2
\]
\[
- \rho^{k-1}(x^*, z(k) - z^*) - \eta u(k) + (\nabla \phi^*(z^*) - \phi^*(z^{(k)}))
\]
\[
- (\rho^{k-1} - \rho^{-k})\phi^*(z^*) - \phi^{-k}\phi^*(z^{(k)})
\]
\[
= (\rho^{k-1} - \rho^{-k})\phi^*(z^*) - \phi^*(z^{(k)}) + \frac{\eta^2}{2\mu_\phi} ||u^{(k)}||^2
\]
\[
- (\rho^{k-1} - \rho^{-k})(x^*, z(k) - z^*) - \rho^{k-1}(x^*, z(k) - z^*) - \eta u(k).
\]
Applying smoothness again, we can bound \(V^{(k+1)} - V^{(k)}\) by
\[
(\rho^{k-1} - \rho^{-k})(\nabla \phi^*(z^*)^T(z(k) - z^*)) + \frac{1}{2\mu_\phi} ||z(k) - z^*||^2
\]
\[
+ \rho^{k-1}\langle x(k) - x^*, -\eta u(k) \rangle + \frac{\eta^2}{2\mu_\phi} ||u^{(k)}||^2
\]
\[
- (\rho^{k-1} - \rho^{-k})(x^*, z(k) - z^*)
\]
\[
= \rho^{k-1}\frac{1}{2\mu_\phi} ||z(k) - z^*||^2 - \eta(x(k) - x^*, u^{(k)}) + \frac{\eta^2}{2\mu_\phi} ||u^{(k)}||^2
\]
\[
= \rho^{k-1}\frac{1}{2\mu_\phi} Me(k),
\]
and observing \(u^* = 0\) finishes the proof.

B. Proof of Proposition 3

We start with the following lemma, which helps with turning the non-affine constraint to an affine constraint in the SDP.

Lemma 10. If matrix \(M \in \mathbb{R}^{m \times n}\) can be decomposed as \(M = N + SS^T\), where \(S \in \mathbb{R}^{n \times m}\), then a negative semi-definite constraint on \(M\) can be equivalently represented by an affine constraint on \(N\) and \(S\).

Proof. Consider the following matrix \(M' \in \mathbb{R}^{(n+m) \times (n+m)}\)
\[
M' = \begin{bmatrix} -N & S \\ S^T & I_m \end{bmatrix}
\]
By properties of Schur complement, we have that
\[
M' \succeq 0 \iff -N - SS^T \succeq 0 \iff M \preceq 0.
\]
Therefore, we can equivalently use \(M' \geq 0\) as the constraint (in lieu of \(M \preceq 0\)). This constraint is affine with respect to both \(N\) and \(S\).

C. Proof of Proposition 7

If \(\eta = \sigma_f = \frac{2\mu_\phi}{\mu + L_f}\), the LMI in (17) becomes
\[
\begin{bmatrix}
\frac{1}{\mu} I & 0 & 0 \\
0 & \frac{1}{\mu} I & 0 \\
0 & 0 & 0
\end{bmatrix} + \sigma_f \begin{bmatrix}
0 & 0 & 0 \\
0 & \frac{\mu L_f}{\mu + L_f} I & \frac{\mu}{2} I \\
0 & \frac{\mu}{2} I & \frac{\mu}{2} I
\end{bmatrix} \preceq 0,
\]
which implies
\[
\begin{bmatrix}
\frac{1}{\mu} I & 0 & 0 \\
0 & \frac{\mu L_f}{\mu + L_f} I & 0 \\
0 & 0 & 0
\end{bmatrix} + \sigma_f \begin{bmatrix}
\frac{1}{\mu + L_f} I & \frac{1}{2} I & 0 \\
\frac{\mu L_f}{\mu + L_f} I & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} \succeq 0.
\]
We can then remove \(I\) inside the block matrix elements in the equation above and apply Lemma 10 to get
\[
\begin{bmatrix}
\frac{1}{\mu} I & 0 & 0 \\
0 & \frac{\mu L_f}{\mu + L_f} I & 0 \\
0 & 0 & 0
\end{bmatrix} + \sigma_f \begin{bmatrix}
\frac{1}{\mu + L_f} I & \frac{1}{2} I & 0 \\
\frac{\mu L_f}{\mu + L_f} I & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} \succeq 0.
\]
which simplifies to
\[
\begin{bmatrix}
\frac{(1-\rho)I}{2\mu_\phi} & 0 & 0 & 0 \\
0 & -\frac{\mu_\phi}{\mu_j + L_f} I & \frac{1}{2} & 0 \\
0 & 0 & -\frac{\mu_\phi}{\mu_j + L_f} I & 0 \\
0 & 0 & 0 & -\frac{\mu_\phi}{\mu_j + L_f} I
\end{bmatrix}
\geq 0,
\]
and we get
\[
\begin{bmatrix}
\frac{(1-\rho)I}{2\mu_\phi} & 0 & 0 & 0 \\
0 & -\frac{\mu_\phi}{\mu_j + L_f} I & \frac{1}{2} & 0 \\
0 & 0 & -\frac{\mu_\phi}{\mu_j + L_f} I & 0 \\
0 & 0 & 0 & -\frac{\mu_\phi}{\mu_j + L_f} I
\end{bmatrix}
\leq 0.
\]
This is equivalent to the following constraints on the principal minors of the matrix:

1) 
\[-\frac{(1-\rho)}{2\mu_\phi} + \sigma_\phi \frac{1}{\mu_\phi + L_\phi} \geq 0 \]

2) 
\[
\frac{2\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} I + \sigma_\phi \frac{\mu_\phi L_\phi}{\mu_j + L_\phi} I \geq 0
\]

3) 
\[
\left( -\frac{(1-\rho)}{2\mu_\phi} + \sigma_\phi \frac{1}{\mu_\phi + L_\phi} \right) \left( \frac{2\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} \right) + \sigma_\phi \frac{\mu_\phi L_\phi}{\mu_j + L_\phi} - \frac{\sigma_\phi^2}{4} \geq 0
\]

The last constraint is the most strict of all constraints. Hence, we will focus on the last constraint, where we can alternatively write
\[
\rho \geq 1 - \frac{2\sigma_\phi}{1 + \kappa_\phi} + \frac{\sigma_\phi^2}{2} \left( \frac{2\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} + \sigma_\phi \frac{\kappa_\phi}{1 + \kappa_\phi} \right)^{-1}
\]
The right-hand side can be seen as a function of $\sigma_\phi$; it takes its minimum when derivative of $\sigma_\phi$ is zero. We denote the optimal $\sigma_\phi$ by $\sigma^*_\phi$. Therefore,
\[
\frac{d}{d\sigma_\phi} \left( 1 - \frac{2\sigma_\phi}{1 + \kappa_\phi} + \frac{\sigma_\phi^2}{2} \left( \frac{2\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} + \sigma_\phi \frac{\kappa_\phi}{1 + \kappa_\phi} \right)^{-1} \right) = 0
\]

The positive solution for the equation above is
\[
\sigma^*_\phi = \frac{4\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} \frac{(1 + \kappa_\phi)}{\kappa_\phi(\kappa_\phi - 1)}
\]
and the corresponding solution for $\rho$ is
\[
\rho_{opt} = 1 - \frac{2\sigma^*_\phi}{1 + \kappa_\phi} + \frac{\sigma^*_\phi^2}{2} \left( \frac{2\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2} + \sigma^*_\phi \frac{\kappa_\phi}{1 + \kappa_\phi} \right)^{-1} = 1 - \frac{4\mu_\phi \mu_j L_f}{(\mu_j + L_f)^2 (1 + \kappa_\phi)}
\]

thereby completing the proof.

REFERENCES

[1] Z. Qu, Cooperative control of dynamical systems: applications to autonomous vehicles. Springer Science & Business Media, 2009.

[2] A. Chavez, A. Moukas, and P. Maes, “Challenger: A multi-agent system for distributed resource allocation,” in Proceedings of the first international conference on Autonomous agents, 1997, pp. 323–331.

[3] U. A. Khan, S. Kar, and J. M. Moura, “Distributed sensor localization in random environments using minimal number of anchor nodes,” IEEE Transactions on Signal Processing, vol. 57, no. 5, pp. 2000–2016, 2009.

[4] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, “Distributed detection: Finite-time analysis and impact of network topology,” IEEE Transactions on Automatic Control, vol. 61, no. 11, pp. 3256–3268, 2016.

[5] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, 2009.

[6] W. Shi, Q. Ling, G. Wu, and W. Yin, “Extra: An exact first-order algorithm for decentralized consensus optimization,” SIAM Journal on Optimization, vol. 25, no. 2, pp. 944–966, 2015.

[7] A. S. Nemirovsky and D. B. Yudin, “Problem complexity and method efficiency in optimization.” 1993.

[8] A. Radhakrishnan, M. Belkin, and C. Uhler, “Linear convergence and implicit regularization of generalized mirror descent with time-dependent mirrors,” arXiv preprint arXiv:2009.08574, 2020.

[9] Y. Sun and S. Shahrampour, “Linear convergence of distributed mirror descent with integral feedback for strongly convex problems,” arXiv preprint arXiv:2011.12233, 2020.

[10] L. Lessard, B. Recht, and A. Packard, “Analysis and design of optimization algorithms via integral quadratic constraints,” SIAM Journal on Optimization, vol. 26, no. 1, pp. 57–95, 2016.

[11] M. Fazlyab, A. Ribeiro, M. Morari, and V. M. Preciado, “Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems,” SIAM Journal on Optimization, vol. 28, no. 3, pp. 2654–2689, 2018.

[12] 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.

[13] P. Lin, W. Ren, and Y. Song, “Distributed multi-agent optimization subject to nonidentical constraints and communication delays,” Automatica, vol. 65, pp. 120–131, 2016.

[14] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1245–1260, 2017.

[15] Y. Sun, A. Daneshmand, and G. Scutari, “Convergence rate of distributed optimization algorithms based on gradient tracking,” arXiv preprint arXiv:1905.02637, 2019.

[16] S. Pu and A. Nedić, “Distributed stochastic gradient tracking methods,” Mathematical Programming, pp. 1–49, 2020.

[17] 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, 2013.

[18] X. Zeng, P. Yi, and Y. Hong, “Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach,” IEEE Transactions on Automatic Control, vol. 62, no. 10, pp. 5227–5233, 2017.

[19] S. S. Kia, J. Cortés, and S. Martínez, “Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication,” Automatica, vol. 55, pp. 254–264, 2015.

[20] S. Yang, Q. Liu, and J. Wang, “A multi-agent system with a proportional-integral protocol for distributed constrained optimization,” IEEE Transactions on Automatic Control, vol. 62, no. 7, pp. 3461–3467, 2016.

[21] D. Yuan, Y. Hong, D. W. Ho, and G. Jiang, “Optimal distributed stochastic mirror descent for strongly convex optimization,” Automatica, vol. 90, pp. 196–203, 2018.
22. M. Rabbat, “Multi-agent mirror descent for decentralized stochastic optimization,” in IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 517–520.

23. S. Shahrampour and A. Jadbabaie, “Distributed online optimization in dynamic environments using mirror descent,” IEEE Transactions on Automatic Control, vol. 63, no. 3, pp. 714–725, 2018.

24. D. Yuan, Y. Hong, D. W. C. Ho, and S. Xu, “Distributed mirror descent for online composite optimization,” IEEE Transactions on Automatic Control, pp. 1–1, 2020.

25. T. T. Doan, S. Bose, D. H. Nguyen, and C. L. Beck, “Convergence of the iterates in mirror descent methods,” IEEE Control Systems Letters, vol. 3, no. 1, pp. 114–119, 2019.

26. Y. Sun and S. Shahrampour, “Distributed mirror descent with integral feedback: Asymptotic convergence analysis of continuous-time dynamics,” IEEE Control Systems Letters, vol. 5, no. 5, pp. 1507–1512, 2020.

27. Y. Yu and B. Açıkmese, “RLC circuits-based distributed mirror descent method,” IEEE Control Systems Letters, vol. 4, no. 3, pp. 548–553, 2020.

28. B. Hu and L. Lessard, “Dissipativity theory for nesterov’s accelerated method,” in International Conference on Machine Learning. PMLR, 2017, pp. 1549–1557.

29. A. B. Taylor, J. M. Hendrickx, and F. Glineur, “Smooth strongly convex interpolation and exact worst-case performance of first-order methods,” Mathematical Programming, vol. 161, no. 1-2, pp. 307–345, 2017.

30. N. K. Dhingra, S. Z. Khong, and M. R. Jovanović, “The proximal augmented lagrangian method for nonsmooth composite optimization,” IEEE Transactions on Automatic Control, vol. 64, no. 7, pp. 2861–2868, 2018.

31. N. S. Aybat, A. Fallah, M. Gurbuzbalaban, and A. Ozdaglar, “Robust accelerated gradient methods for smooth strongly convex functions,” SIAM Journal on Optimization, vol. 30, no. 1, pp. 717–751, 2020.

32. C. Scherer and C. Ebenbauer, “Convex synthesis of accelerated gradient algorithms,” arXiv preprint arXiv:2102.06520, 2021.

33. A. Megretski and A. Rantzer, “System analysis via integral quadratic constraints,” IEEE Transactions on Automatic Control, vol. 42, no. 6, pp. 819–830, 1997.

34. A. Sundararajan, B. Hu, and L. Lessard, “Robust convergence analysis of distributed optimization algorithms,” in 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2017, pp. 1206–1212.

35. Y. Nesterov et al., Lectures on convex optimization. Springer, 2018, vol. 137.

36. J.-B. Hiriart-Urruty and C. Lemaréchal, Fundamentals of convex analysis. Springer Science & Business Media, 2012.

37. R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan, “A general analysis of the convergence of admm,” in International Conference on Machine Learning. PMLR, 2015, pp. 343–352.

38. J. H. Seidman, M. Fazlyab, V. M. Preciado, and G. J. Pappas, “A control-theoretic approach to analysis and parameter selection of douglas–rachford splitting,” IEEE Control Systems Letters, vol. 4, no. 1, pp. 199–204, 2019.

39. D. Lashkari and P. Golland, “Convex clustering with exemplar-based models,” Advances in Neural Information Processing Systems, vol. 20, 2007.

40. M. Pilanci, L. El Ghaoui, and V. Chandrasekaran, “Recovery of sparse probability measures via convex programming,” in Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012, pp. 2420–2428.

41. A. Benfenati, E. Chouzenoux, and J.-C. Pesquet, “Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation,” Signal Processing, vol. 169, p. 107417, 2020.