Distributed Unconstrained Optimization with Time-varying Cost Functions

Amir-Salar Esteki and Solmaz S. Kia *Senior Member, IEEE*

Abstract—This paper proposes a novel solution for the distributed unconstrained optimization problem where the total cost is the summation of time-varying local cost functions of a group networked agents. The objective is to track the optimal trajectory that minimizes the total cost at each time instant. Our approach consists of a two-stage dynamics, where the first one samples the first and second derivatives of the local costs periodically to construct an estimate of the descent direction towards the optimal trajectory, and the second one uses this estimate and a consensus term to drive local states towards the time-varying solution while reaching consensus. The first part is carried out by a weighted average consensus algorithm in the discrete-time framework and the second part is performed with a continuous-time dynamics. Using the Lyapunov stability analysis, an upper bound on the gradient of the total cost is obtained which is asymptotically reached. This bound is characterized by the properties of the local costs. To demonstrate the performance of the proposed method, a numerical example is conducted that studies tuning the algorithm's parameters and their effects on the convergence of local states to the optimal trajectory.

Keywords—Unconstrained Optimization, Distributed Optimization, Time-varying Optimization

I. INTRODUCTION

This paper considers a distributed time-varying unconstrained optimization problem, where a group of N networked agents want to track the optimal solution that minimizes their total cost by interacting over a connected graph. The total cost is the summation of agents' local time-varying costs. Distributed optimization problems appear in many applications, e.g., sensor networks [1], smart grids [2], robotics [3] and learning systems [4]. Most distributed optimization algorithms are for static optimization problems where the total cost is constant and does not change with time, see e.g., [5]. However, there are many in-network optimization problems that the cost can be time-varying. One example is the time-varying distributed linear regression problem where the agents learn a model that best fits an ever changing stream of data; see e.g., [6]–[8] for related works. Another example is when a group of networked robots want to use their relative range measurements from a target to localize a moving target. This problem can be cast as an optimization problem where local costs vary over time.

The time-varying in-network optimization problem we consider in this paper is

$$\mathbf{x}^\star(t) := \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^{N} f^i(\mathbf{x}, t), \quad t \geq 0, \quad (1)$$

where f^i represents agent i's local cost and $\mathbf{x} \in \mathbb{R}^n$ is the decision variable and t is the time.

In case that there is a supervisor or a trusted third-party with the knowledge of all local f^i's in (1) which can distribute the optimal trajectory to the agents online, a central solution can be implemented. Some prediction-correction-based algorithms both in continuous-time and discrete-time have been proposed as the central solution [9] to track the optimal trajectory $\mathbf{x}^\star$. By incorporating the second derivative of the costs (also called Hessian of the cost), i.e., $\nabla_{\mathbf{xx}} f^i(\mathbf{x}, t)$, this method asymptotically converges to the exact optimal trajectory. However, in many applications, due to privacy concerns or the distribution of data among multiple agents, employing a central solver is not desired or even feasible. Such settings require a distributed solver where agents are only allowed to communicate with their neighbors. Therefore, distributed algorithms have been proposed in the literature which take into account the limitations of a fully decentralized network. Some works consider network topology as the time-variant part of the problem [10]–[12] and some other, discussed below, consider the costs to vary over time.

Authors of [13] have addressed this problem by suggesting an algorithm where exact convergence is achieved in finite time by using the signum function, which limits the solution to the continuous-time setting. Moreover, agents need to compute the inverse of the Hessian locally at each time instant, which requires an $\mathcal{O}(n^3)$ computational complexity. In other works [14]–[16], using the alternating direction method of multipliers (ADMM), the authors propose a solution in the discrete-time framework for time-varying optimization problems. The approach used in [16] is inspired by the ADMM methods applied in static optimization problems, where asymptotic convergence is achieved if the objective function varies sufficiently low over time. Some other works, e.g. [17] and [8], consider only optimization problems with time-varying quadratic costs, due to their popularity in applications such as economic dispatch. However, this limitation disallows them to be implemented for other classes of strongly-convex cost functions. Resource allocation problems are also a great part of economic dispatch that can be cast as constrained time-varying optimization problems; see [18]–[23] for related works. Note that unconstrained optimization, which is addressed in this paper, involves optimizing a function without any limitations on the variables, while constrained optimization involves

optimizing a function subject to constraints on the variables, such as in out previous work [18].

In this paper, we address the distributed optimization problem with time-varying local costs in the continuous-time framework. Each agent communicates only with their neighbors over a connected graph and shares local information to solve this problem. By incorporating a weighted average consensus algorithm, agents track the descent direction towards the optimal trajectory and drive their local states to the time-varying optimal solution. This technique allows the agents to asymptotically converge to a neighborhood of the solution without using the signum function which prevents discrete-time implementations and also reduces the computational complexity to $\mathcal{O}(n^2)$. A Lyapunov stability analysis is conducted to prove convergence for strongly-convex and lipschitz-continuous local cost functions. In the numerical example section, we show the performance of the proposed method with different values of algorithm parameters.

Notations: We follow [24] for graph theoretic terminologies. The interaction topology of N in-network agents is modeled by the undirected connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A})$ where $\mathcal{V}$ is the node set, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set and $\mathbf{A} = [\mathbf{a}_{ij}]$ is the adjacency matrix defined such that $\mathbf{a}_{ij} > 0$, if $(i,j) \in \mathcal{E}$, otherwise $\mathbf{a}_{ij} = 0$. A graph is undirected if $\mathbf{a}_{ij} = \mathbf{a}_{ji}$ for all $i, j \in \mathcal{V}$. Moreover, a graph is connected if there is a directed path from every node to every other node. The degree of each node $i \in \mathcal{V}$ is $\mathsf{d}^i = \sum_{j=1}^{N} \mathbf{a}_{ij}$ and the Laplacian matrix of a graph $\mathcal{G}$ is $\mathbf{L} = \mathrm{Diag}(\mathsf{d}^1, \cdots, \mathsf{d}^N) - \mathbf{A}$. Furthermore, For a connected graph, we denote the eigenvalues of $\mathbf{L}$ by $\lambda_1, \cdots, \lambda_N$, where $\lambda_1 = 0$ and $\lambda_i \le \lambda_j$, for $i < j$ and λ_2 and λ_N are, respectively, the smallest nonzero eigenvalue and maximum eigenvalue of $\mathbf{L}$. Finally, given an edge (i,j), i is called a neighbor of j, and vice versa. We let $\mathbf{1}_N$ denote the vector of N ones, and denote by $\mathbf{I}_N$ the $N \times N$ identity matrix. We also define $\mathfrak{r} = \frac{1}{\sqrt{N}} \mathbf{1}_N$, $\mathfrak{R} \in \mathbb{R}^{N \times (N-1)}$ and $\mathbf{T} = [\mathfrak{r} \ \ \mathfrak{R}]$, such that $[\mathfrak{r} \ \mathfrak{R}] \begin{bmatrix} \mathfrak{r}^\top \\ \mathfrak{R}^\top \end{bmatrix} = \begin{bmatrix} \mathfrak{r}^\top \\ \mathfrak{R}^\top \end{bmatrix} [\mathfrak{r} \ \mathfrak{R}] = \mathbf{I}_N$. Note that $\mathbf{T}^\top \mathbf{T} = \mathbf{T}\mathbf{T}^\top = \mathbf{I}$, and for a connected graph, $\mathbf{T}^\top \mathbf{L} \mathbf{T} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^+ \end{bmatrix}$, where $\mathbf{L}^+ = \mathfrak{R}^\top \mathbf{L} \mathfrak{R}$. $\mathbf{L}^+$ is a positive definite matrix with eigenvalues $\{\lambda_i\}_{i=2}^{N} \in \mathbb{R}_{>0}$. For brevity and ease of presentation, the following notations are used alternatively: $\nabla_{\mathbf{x}^i \mathbf{x}^i} f_t^i \equiv \nabla_{\mathbf{x}^i \mathbf{x}^i} f^i(\mathbf{x}^i(t), t)$, $\nabla_{\mathbf{x}^i} f_t^i \equiv \nabla_{\mathbf{x}^i} f^i(\mathbf{x}^i(t), t)$ and $\nabla_{\mathbf{x}^i t} f_t^i \equiv \nabla_{\mathbf{x}^i t} f^i(\mathbf{x}^i(t), t)$.

II. PROBLEM SETTING

Our objective is to design an algorithm that drives local states towards the optimal trajectory, i.e., $\mathbf{x}^\star(t)$ which is the solution of the unconstrained optimization problem (1). The total cost is the summation of each agent i's strongly-convex local costs f^i. In a fully decentralized setting, local costs are private information that are only available to the agent solely. Therefore, a distributed solution is presented in this paper to let agents track $\mathbf{x}^\star(t)$. In this section, we provide the insights needed for proposing our novel algorithm. Let us first consider solving problem (1) where

the case is that the agents are aware of the local costs f^i, for all $i \in \mathcal{V}$. This can be done by implementing a central solver, e.g. using the prediction-correction method in [9], where a descent direction drives the state $\mathbf{x}$ towards the optimal trajectory. Next, we discuss how this descent direction can be estimated distributively in a setting where agents are limited to communicate with their neighbors only. Inspired by the central solution in [9], each agent can use the descent direction $\nabla_{\mathbf{xx}} F^{-1}(\mathbf{x}, t)(\nabla_{\mathbf{x}} F(\mathbf{x}, t) + \nabla_{\mathbf{x}t} F(\mathbf{x}, t))$ where $F(\mathbf{x}(t), t) = \frac{1}{N} \sum_{i=1}^{N} f^i(\mathbf{x}(t), t)$, to asymptotically converge to the solution. Therefore, by implementing the dynamics

$$\dot{\mathbf{x}}^i(t) = -\nabla_{\mathbf{xx}} F^{-1}(\mathbf{x}^i(t), t)(\nabla_{\mathbf{x}} F(\mathbf{x}^i(t), t) \\ + \nabla_{\mathbf{x}t} F(\mathbf{x}^i(t), t)), \qquad (2)$$

all the states $\mathbf{x}^i$, for $i \in \mathcal{V}$, converge to $\mathbf{x}^\star$ asymptotically. In a central manner, since all the states converge to the solution asymptotically, naturally, they also converge to a single trajectory. Therefore, consensus is achieved without any further manipulation. In the proposed method, we build an estimate of the global descent direction denoted as

$$\mathbf{d}_t = \left(\sum_{i=1}^{N} \nabla_{\mathbf{x}^i \mathbf{x}^i} f_t^i\right)^{-1} \sum_{i=1}^{N} (\nabla_{\mathbf{x}^i} f_t^i + \nabla_{\mathbf{x}^i t} f_t^i), \quad (3)$$

by utilizing a weighted average consensus algorithm. However, (3) is different than the one in the dynamics (2). In the former, only the values of the first and second derivatives, calculated with local states as the input, are available to the agents, and in the latter, the total cost derivatives are available as functions to every agent and therefore, each agent can compute the exact local descent direction by using its local state as the input. Since the former is estimated in the proposed algorithm, agents converge to a single trajectory only in the case where initial conditions are similar globally. Therefore, we design a dynamics where a consensus term is added that reduces the difference between state values and minimize the total cost while reaching consensus in $\mathbf{x}^i$'s.

III. MAIN RESULT

In this section, we introduce a novel algorithm that solves the distributed time-varying (1) over connected graphs. In this algorithm, agents collaborate to estimate locally the descent direction (3) using a weighted average consensus algorithm in discrete-time, and use this direction to converge to a neighborhood of the optimal solution while also trying to achieve consensus in their states. We propose the algorithm

$$\begin{cases} \mathbf{v}^i(k+1) = \mathbf{v}^i(k) + \delta_c \sum_{i=1}^{N} (\mathbf{p}^i(k) - \mathbf{p}^j(k)), \\ \mathbf{z}^i(k+1) = \mathbf{z}^i(k) - \delta_c \big(\mathbf{H}^i(k) \mathbf{p}^i(k) - \mathbf{g}^i(k) \\ \quad - \mathbf{h}^i(k) + \sum_{i=1}^{N} (\mathbf{p}^i(k) - \mathbf{p}^j(k)) + (\mathbf{v}^i(k) - \mathbf{v}^j(k)) \big), \\ \mathbf{p}^i(k) = \mathbf{z}^i(k) + \mathbf{g}_s^i(k) + \mathbf{h}_s^i(k), \quad k \in \mathbb{Z}_{\ge 0}, \end{cases}$$
(4a)

$$\dot{\mathbf{x}}^i(t) = -\boldsymbol{\psi}^i(t) - \sum_{i=1}^{N} (\mathbf{x}^i(t) - \mathbf{x}^j(t)), \quad t \ge 0, \qquad (4b)$$

with $\mathbf{v}^i(0), \mathbf{z}^i(0), \mathbf{x}^i(0) \in \mathbb{R}^n, i \in \mathcal{V}$. In this algorithm, $\mathbf{H}^i(k) = \nabla_{\mathbf{x}^i \mathbf{x}^i} f^i_{t_s}$, $\mathbf{g}^i(k) = \nabla_{\mathbf{x}^i} f^i_{t_s}$, $\mathbf{h}^i(k) = \nabla_{\mathbf{x}^i t} f^i_{t_s}, \boldsymbol{\psi}^i(t) = \mathbf{p}^i(s\bar{k})$, for $t \in [t_s, t_{s+1}), t_s = \delta_t s$ and $s = \{0, 1, \cdots\}$ are switching signals that their roles is explained in the following. Here, (4a) constructs an estimate of the local descent direction by the state $\mathbf{p}^i(k)$ and (4b) drives the states using updates of $\mathbf{p}^i(k)$. These updates are passed to (4b) every $\bar{k}$ steps that (4a) takes. While (4a) takes $\bar{k}$ steps, (4b) proceeds $\delta_s > 0$ in time. Therefore, $\boldsymbol{\psi}^i(t)$ is a switching signal where at times t_s updated by $\mathbf{p}^i(s\bar{k})$, sampled from every $\bar{k}$ steps that (4a) takes; $\mathbf{q}^i(t)$ is therefore constant in the time range $t \in [t_s, t_{s+1})$ and is switched to the next value at each time instant t_s. Note that δ_t is the time span between the switchings of $\boldsymbol{\psi}^i(t)$. Moreover, at the same time instant $t = t_s$, the consensus Algorithm (4a) updates its reference values $\mathbf{H}^i(k), \mathbf{g}^i(k)$ and $\mathbf{h}^i(k)$ using $\mathbf{x}^i(t = t_s)$ every $\bar{k}$ steps. With this mechanism, (4a) tracks the weighted average $(\sum_{i=1}^N \mathbf{H}^i(k))^{-1}(\sum_{i=1}^N \mathbf{g}^i(k) + \mathbf{h}^i(k))$ while (4b) uses the updates $\mathbf{p}^i(s\bar{k})$ from (4a) to track the optimal trajectory $\mathbf{x}^\star(t)$.

The algorithm presented in (4a) is inspired by a weighted average consensus algorithm in literature [25] where by incorporating the local Hessian matrix $\mathbf{H}^i(k)$ as the weight and $\mathbf{g}^i(k) + \mathbf{h}^i(k)$ as the time-varying reference signal, each $\mathbf{p}^i$ converges to a neighborhood of the signal $(\sum_{i=1}^N \mathbf{H}^i(k))^{-1}(\sum_{i=1}^N \mathbf{g}^i(k) + \mathbf{h}^i(k))$. As noticed, this value is not exactly the one in (3); however, by considering some common assumptions, we characterize the error between the estimate and the actual value of $\mathbf{d}_t$ and show that by passing the updates $\mathbf{p}^i(s\bar{k})$ to (4b), $\mathbf{x}^i$ converges to a neighborhood of the optimal trajectory. In order to prove convergence, some common conditions presented in e.g., [9] and [13], are required which are stated as following. The first assumption considers lower and upper bounds on the second derivative of the cost functions.

Assumption 1. *Each local cost function $f^i(\mathbf{x}^i, t)$ is twice differentiable and uniformly in t. Also, f^i is m^i-strongly convex and l^i-Lipschitz continuous, i.e.,*

$$m^i \mathbf{I} \le \nabla_{\mathbf{x}^i \mathbf{x}^i} f^i_t \le l^i \mathbf{I}, \quad \mathbf{x}^i \in \mathbb{R}^n, t \ge 0.$$

We also define $m = \min\{m^i\}$ and $l = \max\{l^i\}$ for $i \in \mathcal{V}$.

The second assumption, considers bounds on the first derivatives of the cost functions.

Assumption 2. *Local cost functions $f^i(\mathbf{x}^i, t)$ are sufficiently smooth in $\mathbf{x}^i$ and t, and the following bounds on the first derivatives of the local cost functions exist:*

$$\|\nabla_{\mathbf{x}^i} f^i_t\| \le C_0, \quad \|\nabla_{\mathbf{x}^i t} f^i_t\| \le C_1, \quad i \in \mathcal{V}.$$

where $C_0, C_1 > 0$.

By presenting the second assumption, it is also deduced that the variations of the first and second derivatives of the costs are bounded. Based on the requirements in Assumptions 1 and 2, we can trivially calculate the bound on (3) as $\|\mathbf{d}_t\| \le C_d$, $t \ge 0$, where $C_d = \frac{1}{m}(C_0 + C_1)$. This result is later used to characterize the bound of the tracking error.

Let us first examine the convergence of (4a) to a neighborhood of the local descent direction (3). We consider the following definitions for the proof of convergence. The weighted average to be tracked in (4a) is $\bar{\mathbf{p}}(k) = (\sum_{i=1}^N \mathbf{H}^i(k))^{-1}(\sum_{i=1}^N \mathbf{g}^i(k) + \mathbf{h}^i(k))$ with its aggregated vector $\bar{\mathbf{p}}(k) = \bar{\mathbf{p}}(k) \otimes \mathbf{1}_N$. Again, $\bar{\mathbf{p}}(k)$ is updated every $\bar{k}$ steps which is equal to δ_t time in (4b). Its variation over consecutive steps is $\Delta \bar{\mathbf{p}}(k) = \bar{\mathbf{p}}(k+1) - \bar{\mathbf{p}}(k)$. The diagonal matrix of Hessians is $\mathbf{H}(k) = \text{diag}(\mathbf{H}^1(k), \cdots, \mathbf{H}^N(k))$. The gradient variations is defined as $\Delta \nabla \mathbf{f}(k) = \nabla \mathbf{f}(k+1) - \nabla \mathbf{f}(k)$ where $\nabla \mathbf{f}(k)$ is the aggregated vector of $\mathbf{g}^i(k) + \mathbf{h}^i(k)$ for $i \in \mathcal{V}$; and finally, $\mathbf{w}(k) = \nabla \mathbf{f}(k) - \mathbf{H}(k)\bar{\mathbf{p}}(k)$ and $\Delta \mathbf{w}(k) = \mathbf{w}(k+1) - \mathbf{w}(k)$. Using the definitions above, the compact form of (4a) is

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \delta_c \mathbf{L}\mathbf{p}(k), \tag{5a}$$

$$\mathbf{z}(k+1) = \mathbf{z}(k) - \delta_c(\mathbf{H}(k)\mathbf{p}(k) - \nabla \mathbf{f}(k) + \mathbf{L}(\mathbf{p}(k) + \mathbf{v}(k))), \tag{5b}$$

$$\mathbf{p}(k) = \mathbf{z}(k) + \nabla \mathbf{f}(k). \tag{5c}$$

Using the change of variable $\bar{\mathbf{e}} = \mathbf{T}^\top(\mathbf{p} - \bar{\mathbf{p}})$ and $\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_{2:N}^\top \end{bmatrix}^\top = \mathbf{T}^\top(\mathbf{L}\mathbf{v} - \mathbf{w})$, (5) is equivalent to

$$\mathbf{q}_1(k+1) = \mathbf{q}_1(k), \tag{6a}$$

$$\begin{bmatrix} \bar{\mathbf{e}}(k+1) \\ \mathbf{q}_{2:N}(k+1) \end{bmatrix} = (\mathbf{I} + \delta_c \bar{\mathbf{A}}(k)) \begin{bmatrix} \bar{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix} + \bar{\mathbf{B}} \begin{bmatrix} \Delta \nabla \mathbf{f}(k) - \Delta \bar{\mathbf{p}}(k) \\ \Delta \mathbf{w}(k) \end{bmatrix} \tag{6b}$$

where $\bar{\mathbf{A}}(k) = \begin{bmatrix} -\mathbf{T}^\top \otimes \mathbf{I}(\mathbf{H}(k) + \mathbf{L} \otimes \mathbf{I})\mathbf{T} \otimes \mathbf{I} & -\begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{n(N-1)} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{L}^+\mathbf{L}^+ \otimes \mathbf{I} \end{bmatrix} & \mathbf{0} \end{bmatrix}$ and $\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{T}^\top & \mathbf{0} \\ \mathbf{0} & \mathfrak{R}^\top \end{bmatrix} \otimes \mathbf{I}$. We now obtain the admissible step size δ_c to prove the internal stability of (6), i.e., the matrix $\mathbf{I} + \delta_c \bar{\mathbf{A}}(k)$ is Schur for $k = \{0, 1, 2, \cdots\}$.

Lemma III.1. *Under the Assumption (1) and by the virtue of the results in [25, Lemma 2] and [25, Lemma 3], if $\delta_c \in (0, \bar{\delta})$ in which $\bar{\delta} = \min\left\{\{-2\frac{\text{Re}(\gamma_{i,k})}{|\gamma_{i,k}|^2}\}_{i=1}^{2N-1}\right\}_{k \in \mathbb{Z}_{\ge 0}}$ where $\{\gamma_{i,k}\}_{i=1}^{2N-1}$ are the set of eigenvalues of $\bar{\mathbf{A}}(k)$, then every subsystem $\mathbf{I} + \delta_c \bar{\mathbf{A}}(k)$, $k \in \mathbb{Z}_{\ge 0}$ is Schur. Moreover, we define $\phi = \max\{\|\mathbf{I} + \delta_c \bar{\mathbf{A}}(k)\|\}_{k \in \mathbb{Z}_{\ge 0}}$, where we know that $\phi < 1$.*

The proof is present in [25]. The result above, provided internal stability for the transformed algorithm in (6). We now seek a bound on the error between the trajectories of $\mathbf{p}^i(k)$ and the weighted average $\bar{\mathbf{p}}(k)$. Since the first derivatives of the local costs are bounded according to Assumption 2, $\mathbf{p}^i(k)$'s converge to a neighborhood of the weighted average with a maximum error characterized in the next result.

Theorem III.2. *Let the agents of an undirected connected graph $\mathcal{G}$, implement (5) where the first gradients of local costs satisfy Assumption 2. Considering the result in Lemma III.1, i.e., $\delta_c \in (0, \bar{\delta})$, we have*

$$\|\mathbf{p}(k) - \bar{\mathbf{p}}(k)\| \le N\bar{C}\frac{1 + \phi}{1 - \phi^2}. \tag{7}$$

where $\bar{C}$ is defined in (8).

Proof: To prove convergence of $\mathbf{p}(k)$ to a neighborhood of the weighted average $\bar{\mathbf{p}}(k)$, we use the Lyapunov stability analysis. We consider the transformed dynamics (6) and define the Lyapunov function as a quadratic product of the states

$$V(k) = \begin{bmatrix} \bar{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix}^\top \begin{bmatrix} \bar{\mathbf{e}}(k) \\ \mathbf{q}_{2:N}(k) \end{bmatrix}.$$

Here, we seek to prove that the variation of the Lyapunov function $V(k)$ at each step is negative and obtain a bound on the convergence error. Therefore, by defining $\Delta V(k) = V(k+1) - V(k)$, we have

$$\begin{aligned}
\Delta V(k) &= \mathbf{y}^\top(k)(\mathbf{I} + \delta_c \bar{\mathbf{A}}(k))^\top(\mathbf{I} + \delta_c \bar{\mathbf{A}}(k))\mathbf{y}(k) \\
&+ 2\mathbf{y}^\top(k)(\mathbf{I}+\delta_c\bar{\mathbf{A}}(k))^\top \bar{\mathbf{B}} \begin{bmatrix} \Delta\nabla\mathbf{f}(k) - \Delta\bar{\mathbf{p}}(k) \\ \Delta\mathbf{w}(k) \end{bmatrix} \\
&+ \begin{bmatrix} \Delta\nabla\mathbf{f}(k) - \Delta\bar{\mathbf{p}}(k) \\ \Delta\mathbf{w}(k) \end{bmatrix}^\top \bar{\mathbf{B}}^\top \bar{\mathbf{B}} \begin{bmatrix} \Delta\nabla\mathbf{f}(k) - \Delta\bar{\mathbf{p}}(k) \\ \Delta\mathbf{w}(k) \end{bmatrix} \\
&- \mathbf{y}^\top(k)\mathbf{y}(k),
\end{aligned}$$

where $\mathbf{y} = \begin{bmatrix} \bar{\mathbf{e}} \\ \mathbf{q}_{2:N} \end{bmatrix}$ is the aggregated vector of the states. Based on the results we have from Lemma III.1, by choosing the step size in the admissible range $\delta_c \in (0, \bar{\delta})$, then it is true that $\mathbf{I} + \delta_c\bar{\mathbf{A}}(k)$ is Schur. By incorporating the bounds $\|\bar{\mathbf{B}}\| < 1$ and the ones in Assumptions 1 and 2 and their results, we can write

$$\Delta V(k) \leq (\phi^2 - 1)\|\mathbf{y}(k)\|^2 + 2\phi N\bar{C}\|\mathbf{y}(k)\| + N^2\bar{C}^2,$$

where

$$\bar{C} = 8(C_0 + C_1) + 2(1 + 2l)C_d. \tag{8}$$

Since $\phi < 1$, according to the inequality above, if the norm $\|\mathbf{y}\|$ is sufficiently large, the variation of the Lyapunov function becomes negative. In addition, if $\Delta V(k) < 0$, then $\|\mathbf{y}\|$ decreases. Therefore, the value $\|\mathbf{y}\|$ is bounded. By using the Lyapunov stability analysis, we find that $\|\mathbf{y}\| \leq N\bar{C}\frac{1+\phi}{1-\phi^2}$. Given that $\|\mathbf{p}(k) - \bar{\mathbf{p}}(k)\| \leq \|\mathbf{y}(k)\|$, we can establish the tracking error in (7). ∎

In the statement above, we obtained an upper bound on the error between the trajectories $\mathbf{p}(k)$ and the weighted average $\bar{\mathbf{p}}(k)$. Let us now consider $\mathbf{\Psi} = [\boldsymbol{\psi}^{1\top}, \cdots, \boldsymbol{\psi}^{N\top}]^\top$ (aggregated vector of ψ^i) which is samples of $\mathbf{p}^i(k)$ for every k steps. Before we establish the final result, we want to characterize a bound on the error between $\psi(t)$ and $\mathbf{d}_t = \mathbf{d}_t \otimes \mathbf{1}_N$ denoted as

$$\boldsymbol{\epsilon}_t = \mathbf{\Psi}(t) - \mathbf{d}_t. \tag{9}$$

Trivially, by the virtue of Theorem III.2 and the fact that $\|\bar{\mathbf{p}}(k) - \mathbf{d}_t\| \leq 2NC_d$ (as a result of Assumption 2) for any $t \geq 0$ and $k \in \mathbb{Z}_{\geq 0}$, we can conclude that $\|\mathbf{q}(t) - \mathbf{d}_t\| = \|\boldsymbol{\epsilon}_t\| \leq N\bar{\epsilon}$ where $\bar{\epsilon} = 2C_d + \bar{C}\frac{1+\phi}{1-\phi^2}$. Let us now present the final statement.

Theorem III.3. *Let the agents of an undirected connected graph $\mathcal{G}$ implement the Algorithm (4) to track the optimal trajectory $\mathbf{x}^\star(t)$, the solution of the unconstrained optimization problem (1). Provided that $\delta_c \in (0, \bar{\delta})$, we can prove that the gradient of the total cost asymptotically converges to a neighborhood of the origin with the bound*

$$\left\| \frac{1}{N}\sum_{i=1}^N \nabla_{\mathbf{x}^i} f^i(\mathbf{x}^i(t), t) - \frac{1}{N}\sum_{i=1}^N \nabla_{\mathbf{x}} f^i(\mathbf{x}^\star(t), t) \right\| \leq C_\nabla\bar{\epsilon} \tag{10}$$

where C_∇ is defined in (12).

Proof: To simplify the presentation, we demonstrate the proof for when the cost functions are univariant, i.e., the decision variable $\mathbf{x}^i$ is scalar and therefore, $x^i \in \mathbb{R}$, $i \in \mathcal{V}$. Also, $\mathbf{H}_t$, $\nabla_{\mathbf{x}}\mathbf{f}_t$ and $\nabla_{\mathbf{x}t}\mathbf{f}_t$ are aggregated matrix and vectors of the local Hessians and first derivatives of costs $f^i(\mathbf{x}^i(t), t)$. We implement the results from Theorem III.2 and use a Lyapunov stability analysis in the continuous-time framework to prove convergence. Consider the dynamics (4b) which can be presented in the compact form

$$\dot{\mathbf{x}}(t) = -\mathbf{\Psi}(t) - \mathbf{L}\mathbf{x}(t). \tag{11}$$

Let us define the Lyapunov function

$$V(\mathbf{x}(t), t) = (\mathbf{1}_N^\top \nabla_{\mathbf{x}}\mathbf{f}_t)^2 + \alpha\mathbf{x}^\top(t)\mathbf{L}\mathbf{x}(t),$$

where $\alpha > 0$ is a positive scalar; for the rest of the proof, we use $\mathbf{x}_t$ and V_t as replacements of $\mathbf{x}(t)$ and $V(\mathbf{x}(t), t)$, respectively. Under the assumption (2), the Lyapunov function is bounded by the states $\mathbf{x}$ in the dynamics (11) by $0 \leq V_t \leq C_1^2 + \alpha\lambda_N\|\mathbf{x}_t\|^2$. Taking the derivative of the Lyapunov function, we have

$$\dot{V}_t = \nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top (\mathbf{H}_t\dot{\mathbf{x}}_t + \nabla_{\mathbf{x}t}\mathbf{f}_t) + \alpha\mathbf{x}_t^\top\mathbf{L}\dot{\mathbf{x}}_t,$$

where by substituting $\dot{\mathbf{x}}_t$, $\mathbf{\Psi}$ and $\mathbf{d}_t$ by the equations (11), (9) and (3), respectively, we get

$$\begin{aligned}
\dot{V}_t &= -\nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top \nabla_{\mathbf{x}}\mathbf{f}_t - \nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top \mathbf{H}_t\boldsymbol{\epsilon}_t \\
&- \nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top \mathbf{H}_t\mathbf{L}\mathbf{x}_t - \alpha\mathbf{x}_t^\top\mathbf{L}\boldsymbol{\epsilon}_t - \alpha\mathbf{x}_t^\top\mathbf{L}\mathbf{x}_t.
\end{aligned}$$

By subtraction and addition of similar terms, the equality above can be rewritten as

$$\begin{aligned}
\dot{V}_t &= -\nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top \nabla_{\mathbf{x}}\mathbf{f}_t - \nabla_{\mathbf{x}}\mathbf{f}_t^\top \mathbf{1}_N \mathbf{1}_N^\top \mathbf{H}_t\boldsymbol{\epsilon}_t \\
&- \frac{1}{2}\|\frac{1}{\sqrt{\beta}}\mathbf{H}_t\mathbf{1}_N\mathbf{1}_N^\top\nabla_{\mathbf{x}}\mathbf{f}_t + \sqrt{\beta}\mathbf{L}\mathbf{x}_t\|^2 \\
&+ \frac{1}{2\beta}\nabla_{\mathbf{x}}\mathbf{f}_t^\top\mathbf{1}_N\mathbf{1}_N^\top\mathbf{H}_t^2\mathbf{1}_N\mathbf{1}_N^\top\nabla_{\mathbf{x}}\mathbf{f}_t + \frac{\beta}{2}\mathbf{x}_t^\top\mathbf{L}^2\mathbf{x}_t \\
&- \frac{\alpha}{2}\|\frac{1}{\sqrt{\gamma}}\boldsymbol{\epsilon}_t + \sqrt{\gamma}\mathbf{L}\mathbf{x}_t\|^2 + \frac{\alpha}{2\gamma}\boldsymbol{\epsilon}_t^\top\boldsymbol{\epsilon}_t + \frac{\alpha\gamma}{2}\mathbf{x}_t^\top\mathbf{L}^2\mathbf{x}_t \\
&- \alpha\mathbf{x}_t^\top\mathbf{L}^2\mathbf{x}_t,
\end{aligned}$$

where $\beta, \gamma > 0$. Using the Assumptions (1) and (2), we can derive the inequality

$$\begin{aligned}
\dot{V}_t &\leq -(1 - \frac{l^2}{2\beta})\|\mathbf{1}_N^\top\nabla_{\mathbf{x}}\mathbf{f}_t^\top\|^2 + lN\bar{\epsilon}\|\mathbf{1}_N^\top\nabla_{\mathbf{x}}\mathbf{f}_t^\top\| + \frac{\alpha N^2}{2\gamma}\bar{\epsilon}^2 \\
&- (\alpha - \frac{\alpha\gamma + \beta}{2})\mathbf{x}_t^\top\mathbf{L}\mathbf{x}_t - \frac{\alpha}{2}\|\frac{1}{\sqrt{\gamma}}\boldsymbol{\epsilon}_t + \sqrt{\gamma}\mathbf{L}\mathbf{x}_t\|^2 \\
&- \frac{1}{2}\|\frac{1}{\sqrt{\beta}}\mathbf{H}_t\mathbf{1}_N\mathbf{1}_N^\top\nabla_{\mathbf{x}}\mathbf{f}_t + \sqrt{\beta}\mathbf{L}\mathbf{x}_t\|^2,
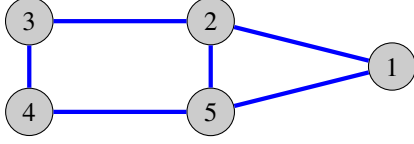\end{aligned}$$

4

Fig. 1: The graph indicated above is an undirected connected graph with adjacency weights of $\mathbf{a}_{ij} = 1$, if $(i,j) \in \mathcal{E}$, otherwise $\mathbf{a}_{ij} = 0$. Each agent $i \in \{1, 2, \cdots, 5\}$ is endowed with a local cost $f^i(x^i(t), t) = i(x^i(t))^2 + \sin(i\omega t)x^i(t)$.

in which, for stability, we have the requirements $\alpha < \frac{l^2}{4}, \beta > \frac{l^2}{2}, \gamma < 2 - \frac{l^2}{2\alpha}$ which is always feasible. Based on the inequality above, we can conclude that the summation of the gradients, asymptotically converges to a neighborhood of the origin with the bound

$$\|\frac{1}{N}\mathbf{1}_N^\top \nabla_{\mathbf{x}}\mathbf{f}_t^\top\| \leq \frac{l + \sqrt{l^2 + \frac{\alpha}{\gamma}(2 - \frac{l^2}{\beta})}}{2 - \frac{l^2}{\beta}}\bar{\epsilon},$$

where by defining

$$C_\nabla = \frac{l + \sqrt{l^2 + \frac{\alpha}{\gamma}(2 - \frac{l^2}{\beta})}}{2 - \frac{l^2}{\beta}}, \tag{12}$$

and using the fact that $\frac{1}{N}\sum_{i=1}^N \nabla_{\mathbf{x}}f^i(\mathbf{x}^\star(t), t) = 0$ we get the final result in (10). ∎

IV. NUMERICAL EXAMPLE

To demonstrate the performance of the proposed method, we study convergence of Algorithm (4) in presence of local time-varying costs. Many problems such as ones in economic dispatch or linear regression are formulated as optimization problems with quadratic costs of the form $f(x(t), t) = \frac{1}{2}a(t)(x(t))^2 + b(t)x(t) + c(t)$. In many cases, for example, due to parameter oscillations of local generators in economic dispatch or updates in local data sets in linear regression, the problem at hand is time-varying, and consequently, agents are required to track a time-variant solution rather than converging to a single minimum point. Therefore, we consider solving an unconstrained quadratic optimization problem in the following of this section.

Let a network of $N = 5$ agents interact with each other to solve problem (1). The topology of the network is an undirected connected graph illustrated in Fig. 1. Each agent is endowed with a local cost

$$f^i(x^i(t), t) = i(x^i(t))^2 + \sin(i\omega t)x^i(t),$$

where $\omega = 0.05$ controls the frequency of the time-varying costs. Trivially, one can solve the problem $x^\star(t) = \text{argmin}_x \frac{1}{N}\sum_{i=1}^5 i(x(t))^2 + \sin(i\omega t)x(t)$ analytically and derive the optimal solution as the time-varying trajectory $x^\star(t) = -\frac{1}{2}\frac{\sum_{i=1}^5 \sin(i\omega t)}{\sum_{i=1}^5 i}$. The objective is to implement Algorithm (4) to track $x^\star(t)$ with different values of $\bar{k}$ and observe its effect on the convergence error. In this
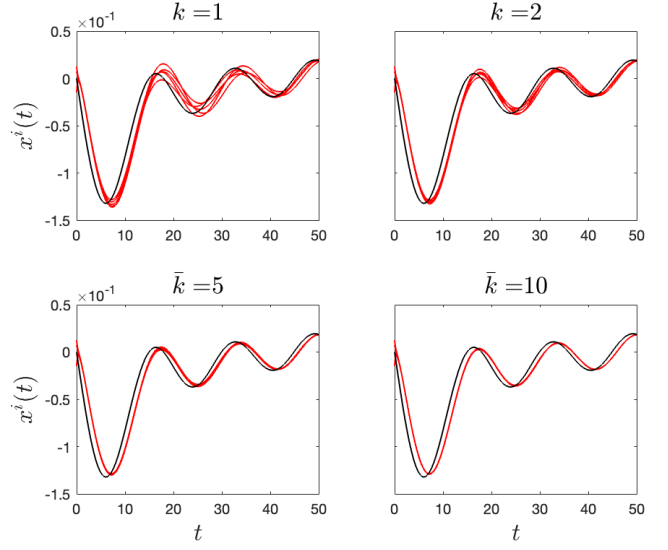

Fig. 2: Trajectories of the states $x^i(t)$ for $i \in \{1, 2, \cdots, 5\}$ are shown above for different values of $\bar{k}$. As $\bar{k}$ increases, there is more consensus between the states $x^i(t)$ and the optimal trajectory is tracked more accurately.

example, $\delta_t = 0.1$ is fixed. We set $\bar{k} = \{1, 2, 5, 10\}$ and measure the state values $x^i(t)$ and the tracking error, defined as $e(t) = \|\mathbf{x}(t) - x^\star(t)\mathbf{1}_5\|_2$, in the range $t \in [0, T]$ with $T = 50$. Moreover, to review the overall effect on convergence, a separate figure is plotted to show the average of the tracking error $\bar{e} = \frac{1}{T}\int_{\tau=0}^{\tau=T} e(\tau)$ over the time span $t \in [0, T]$, for each case of $\bar{k}$.

By plotting the states $x^i(t)$ over $t \in [0, T]$, for $i \in \{1, 2, \cdots, 5\}$, we can observe that convergence to the optimal trajectory has improved. While using $\bar{k} = 1$ the agents can still track the optimal solution, increasing $\bar{k}$ to 5 and 10 results in more consensus between the states at each time instant t. Evidently, we can see from Fig.3(a) that convergence error is reduced when $\bar{k} = 10$ compared to when $\bar{k} = \{1, 2\}$. Intuitively, as $\bar{k}$ grows, agents obtain a more accurate estimate of the current descent direction $\mathbf{d}_t^i$ and drive their local states towards a tighter neighborhood of the optimal trajectory. As a result, Fig. 3(b) indicates that the average tracking error $\bar{e}$ is lower in cases with higher values of $\bar{k}$.

V. CONCLUSION

We proposed a method to solve a distributed unconstrained optimization problem with time-varying cost. In this setting, the total cost to be optimized consists of the sum of the time-varying local costs that each agent of a network is endowed with and therefore, the solution is an optimal trajectory rather than a minimum point. In our approach, we implemented the discrete-time version of a weighted average consensus algorithm to derive an estimate of the descent direction, and constructed a continuous-time dynamics
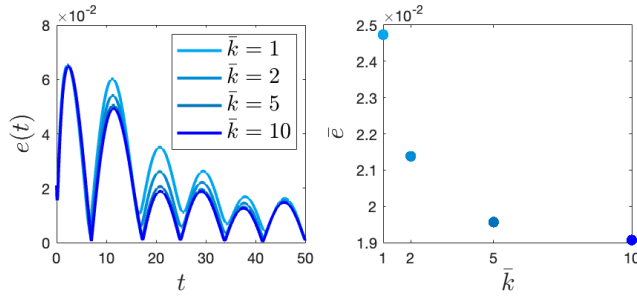
Fig. 3: (a) Left figure: Represents the tracking error in the time interval $t \in [0, T]$ for $\bar{k} = \{1, 2, 5, 10\}$. The error oscillates due to the nature of the local cost functions, however, when $\bar{k}$ is higher, there is less tracking error at each time instant. (b) Right figure: Represents the average tracking error $\bar{e}$ for $\bar{k} = \{1, 2, 5, 10\}$. A similar conclusion is depicted here.

where this estimate was used to drive local states towards the optimal trajectory while reaching consensus. Under some common assumptions and with the use of the Lyapunov stability analysis, a bound on the asymptotic tracking error of the total cost gradient was achieved. To show the effect of the parameters used in the algorithm, a numerical example was provided where convergence to the optimal trajectory was studied with different values of these parameters. In future work we will investigate a fully discretized implementation of our proposed algorithm.

REFERENCES

[1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 20–27, 2004.

[2] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[3] O. Shorinwa, T. Halsted, J. Yu, and M. Schwager, "Distributed optimization methods for multi-robot systems: Part i–a tutorial," *arXiv preprint arXiv:2301.11313*, 2023.

[4] A. Koppel, G. Warnell, and E. Stump, "Task-driven dictionary learning in distributed online settings," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 1114–1118, IEEE, 2015.

[5] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[6] Y. Zhao and W. Lu, "Training neural networks with time-varying optimization," in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, vol. 2, pp. 1693–1696, IEEE, 1993.

[7] C. Feng and Y. Zhao, "Time-varying nonlinear programming and its realization via neural networks," in *1992 American Control Conference*, pp. 978–982, IEEE, 1992.

[8] M. Ye and G. Hu, "Distributed seeking of time-varying nash equilibrium for non-cooperative games," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3000–3005, 2015.

[9] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.

[10] A. Rogozin, M. Bochko, P. Dvurechensky, A. Gasnikov, and V. Lukoshkin, "An accelerated method for decentralized distributed stochastic optimization over time-varying graphs," in *IEEE Conf. on Decision and Control*, pp. 3367–3373, IEEE, 2021.

[11] H. Reisizadeh, B. Touri, and S. Mohajer, "Distributed optimization over time-varying graphs with imperfect sharing of information," *IEEE Transactions on Automatic Control*, 2022.

[12] H. Li, Q. Lü, X. Liao, and T. Huang, "Accelerated convergence algorithm for distributed constrained optimization under time-varying general directed graphs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2612–2622, 2018.

[13] S. Rahili and W. Ren, "Distributed continuous-time convex optimization with time-varying cost functions," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1590–1605, 2016.

[14] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2013.

[15] Z. Chen, P. Yi, L. Li, and Y. Hong, "Distributed time-varying convex optimization with dynamic quantization," *IEEE Transactions on Cybernetics*, 2021.

[16] M. Maros and J. Jaldén, "ADMM for distributed dynamic beamforming," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 220–235, 2017.

[17] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3687–3694, 2017.

[18] A.-S. Esteki and S. S. Kia, "Distributed optimal resource allocation with time-varying quadratic cost functions and resources over switching agents," in *European Control Conference*, pp. 441–446, IEEE, 2022.

[19] A. Simonetto, "Dual prediction–correction methods for linearly constrained time-varying convex programs," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3355–3361, 2018.

[20] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1973–1986, 2017.

[21] L. Bai, C. Sun, Z. Feng, and G. Hu, "Distributed continuous-time resource allocation with time-varying resources under quadratic cost functions," in *IEEE Conf. on Decision and Control*, pp. 823–828, IEEE, 2018.

[22] B. Wang, S. Sun, and W. Ren, "Distributed continuous-time algorithms for optimal resource allocation with time-varying quadratic cost functions," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1974–1984, 2020.

[23] B. Wang, S. Sun, and W. Ren, "Distributed time-varying quadratic optimal resource allocation subject to nonidentical time-varying hessians with application to multiquadrotor hose transportation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022.

[24] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks*. Princeton University Press, 2009.

[25] Y.-F. Chung and S. S. Kia, "Dynamic active average consensus," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1177–1182, 2021.