Automatica 136 (2022) 109886

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Technical communique

A distributed continuous-time modified Newton–Raphson algorithm ☆

Hossein Moradian *,[1], Solmaz S. Kia

Department of Mechanical and Aerospace Engineering, University of California, Irvine, United States of America

ARTICLE INFO

Article history:
Received 19 July 2020
Received in revised form 19 May 2021
Accepted 14 July 2021
Available online 26 August 2021

Keywords:
Distributed optimization
Newton–Raphson method
Convex optimization
Machine learning

ABSTRACT

We propose a continuous-time second-order optimization algorithm for solving unconstrained convex optimization problems with bounded Hessian. We show that this alternative algorithm has a comparable convergence rate to that of the continuous-time Newton–Raphson method, however structurally, it is amenable to a more efficient distributed implementation. We present a distributed implementation of our proposed optimization algorithm and prove its convergence via Lyapunov analysis. A numerical example demonstrates our results.

1. Introduction

Consider a network of N agents interacting over a connected graph $\mathcal{G}$, see Fig. 1. Each agent $i \in \{1, \dots, N\}$ is endowed with a local cost function $f^i : \mathbb{R}^d \to \mathbb{R}$ which is twice differentiable and m^i-strongly convex. Our objective is to design a distributed optimization algorithm such that each agent obtains the global minimizer $\mathbf{x}^\star \in \mathbb{R}^d$ of the feasible optimization problem

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \quad f(\mathbf{x}) = \sum_{i=1}^{N} f^i(\mathbf{x}), \tag{1}$$

using local interactions with its neighbors. The existing distributed optimization solutions are mostly consensus-based approaches that use gradient and sub-gradient methods, see e.g., Boyd, Parikh, Chu, Peleato, and Eckstein (2010), Duchi, Agarwal, and Wainwright (2012), Johansson, Rabi, and Johansson (2009), Nedić and Ozdaglar (2009), Zhu and Martínez (2012) for discrete-time and Droge, Kawashima, and Egerstedt (2014), Kia, Cortés, and Martínez (2014), Lu and Tang (2012), Wang and Elia (2011), Zanella, Varagnolo, Cenedese, Pillonetto, and Schenato (2011) for continuous-time algorithms. Even though the gradient-based solutions' distributed implementation is fully understood and requires low computational resources, they suffer from a low convergence rate, especially near the solution. With the recent advances in fast computing via graphics processing units (GPUs), the interest in Newton-based optimization algorithms, which use second-order information to achieve faster convergence, is renewed for large-scale optimization problems (Henriques, Ehrhardt, Albanie, & Vedaldi, 2018; Yao et al., 2020). The popular Newton–Raphson (NR) method uses the inverse of the Hessian of the total cost multiplied by the gradient of the total cost, i.e., $-(\sum_{i=1}^N \mathbf{H}^i(\mathbf{x}))^{-1}(\sum_{i=1}^N \mathbf{g}^i(\mathbf{x}))$, as the descent direction. Here, $\mathbf{g}^i(\mathbf{x}) = \nabla f^i(\mathbf{x})$ and $\mathbf{H}^i(\mathbf{x}) = \nabla^2 f^i(\mathbf{x})$. Starting from a local guess $\mathbf{x}^i \in \mathbb{R}^d$, $i \in \{1, \dots, N\}$, a common way to execute the NR algorithm in a decentralized way is to use a consensus-based framework to track $\sum_{i=1}^N \mathbf{H}^i(\mathbf{x}^i)$ and $\sum_{i=1}^N \mathbf{g}^i(\mathbf{x}^i)$ for every agent cooperatively by exchanging the gradient and the Hessian of the local costs, see e.g., Varagnolo, Zanella, Cenedese, Pillonetto, and Schenato (2015) for continuous-time and Bof, Carli, Notarstefano, Schenato, and Varagnolo (2019), Varagnolo et al. (2015) for discrete-time algorithms. These algorithms result in $O(Nd^2)$ communication, computation and storage costs per agent to solve problem (1). To remove communicating Hessian among agents, Mokhtari, Ling, and Ribeiro (2015) propose a distributed algorithm that approximates Newton step by truncating the Taylor series expansion of the exact Newton step at K terms. But, implementing this algorithm requires aggregating information from K hops away. Increasing K makes the method arbitrarily close to Newton's method at the cost of increasing the communication overhead of each iteration. Building on Mokhtari et al. (2015), Mansoori and Wei (2020) propose an asynchronous implantation to manage communication cost but the method works only for univariate local cost functions.

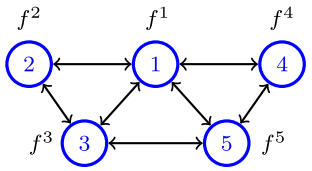
In this paper, we provide an alternative continuous-time second-order algorithm with a comparable convergence rate to that of the continuous-time NR algorithm, but with a structure that is amenable to a more resource-efficient distributed

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Iman Shames under the direction of Editor André L. Tits.

* Corresponding author.
E-mail addresses: hmoradia@uci.edu (H. Moradian), solmaz@uci.edu (S.S. Kia).

[1] This work is supported by NSF award ECCS-1653838.

https://doi.org/10.1016/j.automatica.2021.109886
0005-1098/



$$\mathbf{L}=\begin{bmatrix} 4 & -1 & -1 & -1 & -1\\ -1 & 2 & -1 & 0 & 0\\ -1 & -1 & 3 & 0 & -1\\ -1 & 0 & 0 & 2 & -1\\ -1 & 0 & -1 & -1 & 3 \end{bmatrix}$$

Fig. 1. A *connected graph*: in a connected undirected graph agents connected by an edge can exchange information. Moreover, there is a path from any agent to any other agent.

implementation that also requires information exchange only among one-hop neighbors. In the distributed implementation of this algorithm, agents use the inverse of their local Hessians but do not need to communicate it. As a result, our proposed algorithm's communication, computation, and storage cost per agent are $O(Nd)$. We establish the convergence of our algorithm using Lyapunov stability analysis. Simulations demonstrate our results.

2. Preliminaries

Our notations are standard and definitions are given if it is necessary to avoid confusion. A differentiable function $f:\mathbb{R}^d\to\mathbb{R}$ is *m-strongly convex* ($m\in\mathbb{R}_{>0}$) in a convex set C if and only if $(\mathbf{z}-\mathbf{x})^\top(\nabla f(\mathbf{z})-\nabla f(\mathbf{x}))\geq m\|\mathbf{z}-\mathbf{x}\|^2,\ \forall\,\mathbf{x},\mathbf{z}\in C,\ \mathbf{x}\neq\mathbf{z}$. For twice differentiable function f m-strong convexity ($m>0$) is also equivalent to $\mathbf{H}(\mathbf{x})=\nabla^2 f(\mathbf{x})\geq m\mathbf{I},\ \forall\,\mathbf{x}\in C$.

A connected graph, see Fig. 1, is represented by $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$, where $\mathcal{V}=\{1,\dots,N\}$ is the *node set*, $\mathcal{E}=\{e_1,\dots,e_M\}\subseteq\mathcal{V}\times\mathcal{V}$ is the *edge set*, and $\mathbf{A}=[a_{ij}]\in\mathbb{R}^{N\times N}$ is the *adjacency* matrix such that $a_{ij}=a_{ji}=1$ if $(i,j)\in\mathcal{E}$ and $a_{ij}=0$, otherwise. The *incidence* matrix is $\mathbf{B}=\frac{1}{\sqrt{2}}[\mathbf{b}^1,-\mathbf{b}^1,\dots,\mathbf{b}^M,-\mathbf{b}^M]\in\mathbb{R}^{N\times 2M}$ where $\mathbf{b}^k\in\mathbb{R}^N$ is a vector corresponds to the edge $e_k=(i,j)\in\mathcal{E}$ with zero elements except for ith and jth components with respectively $b_i=1$, $b_j=-1$. The *Laplacian* matrix of a graph is $\mathbf{L}=\mathrm{Diag}(\mathbf{A}\mathbf{1}_N)-\mathbf{A}$. Note that $\mathbf{L}\mathbf{1}_N=\mathbf{0}$. Moreover, $\mathbf{L}=\mathbf{B}\mathbf{B}^\top$. A graph is connected if and only if $\mathbf{1}_N^T\mathbf{L}=\mathbf{0}$, and $\mathrm{rank}(\mathbf{L})=N-1$. For a connected graph, eigenvalue of $\mathbf{L}$ is $\lambda_1=0$, $\{\lambda_i\}_{i=2}^N\subset\mathbb{R}_{>0}$. We let $\lambda_i\leq\lambda_j$, for $i<j$. Moreover,

$$\mathbf{\Pi}_N=\mathbf{L}\mathbf{L}^+=\mathbf{B}\mathbf{B}^\top(\mathbf{B}\mathbf{B}^\top)^+=\mathbf{B}(\mathbf{B}^\top\mathbf{B})^+\mathbf{B}^\top, \tag{2}$$

where $(.)^+$ denotes the generalized inverse matrix (George & Freeman, 2019) and $\mathbf{\Pi}_N=\mathbf{I}-\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top$, where $\mathbf{1}_N$ is the vector of N ones.

Throughout the paper, the following assumption holds.

Assumption 2.1. The local cost functions $f^i:\mathbb{R}^d\to\mathbb{R}$, $i\in\{1,\cdots,N\}$, are m^i-strongly convex with the bounded Hessians $\underline{m}^i\mathbf{I}\leq\mathbf{H}^i(\mathbf{x}^i)\leq\bar{m}^i\mathbf{I}$, for some $\underline{m}^i,\bar{m}^i\in\mathbb{R}_{>0}$. □

Thus, the total cost f is $\underline{m}$-strongly convex and its Hessian is upper-bounded by $\bar{m}\mathbf{I}$ where $\underline{m}=\min\{\underline{m}^i\}_{i=1}^N$, $\bar{m}=\max\{\bar{m}^i\}_{i=1}^N$. Moreover, $\mathbf{x}^\star$ in (1) is unique (Bertsekas, 1999).

3. Problem definition

The continuous-time NR algorithm to solve problem (1) is given by

$$\text{NR:}\quad \dot{\mathbf{x}}=-\Big(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x})\Big)^{-1}\sum_{i=1}^N\mathbf{g}^i(\mathbf{x}). \tag{3}$$

In this paper, we propose the Hessian inverse sum optimization (HISO) algorithm

$$\text{HISO:}\quad \dot{\mathbf{x}}=-\Big(\frac{1}{N}\sum_{i=1}^N\mathbf{H}^i(\mathbf{x})^{-1}\Big)\sum_{i=1}^N\mathbf{g}^i(\mathbf{x}), \tag{4}$$

as an alternative Hessian-based solver for the optimization problem (1). We show that this algorithm has the convergence rate no worse than that of (3) but it has a structure that is amenable to a distributed implementation with more efficient resource usage. We start by the auxiliary result below.

Lemma 3.1 (*Bound on the Inverse of Sum of Symmetric Positive Definite Matrices*)**.** *Let every $\mathbf{H}^i\in\mathbb{R}^{d\times d}$, $i\in\{1,\dots,N\}$, be a positive definite matrix. Then*

$$\Big(\sum_{i=1}^N\mathbf{H}^i\Big)^{-1}\leq\frac{1}{N}\sum_{i=1}^N\mathbf{H}^{i^{-1}}. \tag{5}$$

Proof. The proof is by mathematical induction. Recall that the inverse of positive definite matrices is a convex function (Nordstrom, 2011). Hence, for $N=2$ for any $\kappa\in[0,1]$ we have

$$\big(\kappa\,\mathbf{H}^1+(1-\kappa)\mathbf{H}^2\big)^{-1}\leq\kappa\,\mathbf{H}^{1^{-1}}+(1-\kappa)\mathbf{H}^{2^{-1}}. \tag{6}$$

Substituting $\kappa=0.5$ gives $(\mathbf{H}^1+\mathbf{H}^2)^{-1}\leq\frac{1}{4}(\mathbf{H}^{1^{-1}}+\mathbf{H}^{2^{-1}})\leq\frac{1}{2}(\mathbf{H}^{1^{-1}}+\mathbf{H}^{2^{-1}})$. Thus, (5) holds for $N=2$. Next, assuming $(\sum_{i=1}^{N-1}\mathbf{H}^i)^{-1}\leq\frac{1}{N-1}\sum_{i=1}^{N-1}\mathbf{H}^{i^{-1}}$ we show that $(\sum_{i=1}^N\mathbf{H}^i)^{-1}\leq\frac{1}{N}\sum_{i=1}^N\mathbf{H}^{i^{-1}}$ holds. To this aim, notice that given (6) we obtain

$$\Big(\frac{1}{N}\mathbf{H}^N+\frac{N-1}{N}\sum_{i=1}^{N-1}\mathbf{H}^i\Big)^{-1}\leq\frac{1}{N}\mathbf{H}^{N^{-1}}+\frac{N-1}{N}\sum_{i=1}^{N-1}\mathbf{H}^{i^{-1}}$$
$$\leq\frac{1}{N}\mathbf{H}^{N^{-1}}+\frac{N-1}{N}\Big(\frac{1}{N-1}\sum_{i=1}^{N-1}\mathbf{H}^{i^{-1}}\Big)=\frac{1}{N}\sum_{i=1}^N\mathbf{H}^{i^{-1}}.$$

Since $\big(\sum_{i=1}^N\mathbf{H}^i\big)^{-1}\leq\big(\frac{1}{N}\mathbf{H}^N+\frac{N-1}{N}\sum_{i=1}^{N-1}(\mathbf{H}^i)\big)^{-1}$, then (5) holds for any $N\geq 2$, which concludes proof. □

Lemma 3.1 enables us to make the following statement about the HISO algorithm's convergence guarantees.

Theorem 3.1 (*Convergence Analysis of the HISO Algorithm*)**.** *Consider the optimization problem (1) and let Assumption 2.1 hold. Then, starting from any initial condition $\mathbf{x}(0)\in\mathbb{R}^d$, as $t\to\infty$ the HISO algorithm (4) converges exponentially fast to $\mathbf{x}^\star\in\mathbb{R}^d$, the unique minimizer of the optimization problem (1). Furthermore, the rate of convergence of (4) is no worse than the rate of convergence of algorithm (3).*

Proof. Consider the candidate Lyapunov function $V(\mathbf{x})=f(\mathbf{x})-f(\mathbf{x}^\star)$. Given Assumption 2.1, we have $\underline{m}\|\mathbf{x}-\mathbf{x}^\star\|^2\leq V(\mathbf{x})\leq\bar{m}\|\mathbf{x}-\mathbf{x}^\star\|^2$, and $\|(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x}))^{-1}\nabla f(\mathbf{x})\|\leq\frac{\bar{m}}{\underline{m}}\|\mathbf{x}-\mathbf{x}^\star\|$ where $\nabla f(\mathbf{x})=\sum_{i=1}^N\mathbf{g}^i(\mathbf{x})$. The derivative of $V(\mathbf{x})$ along trajectories of (4), satisfies $\dot{V}(x)=-\frac{1}{N}\nabla f(\mathbf{x})^\top(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x})^{-1})\nabla f(\mathbf{x})$
$\leq-\frac{\bar{m}^2}{\underline{m}}\|\mathbf{x}-\mathbf{x}^\star\|^2$, which by virtue of Khalil (2002, Theorem 4.10) confirms the exponential stability of (4). On the other hand, derivative of $V(\mathbf{x})$ along (3) satisfies $\dot{V}=-\nabla f(\mathbf{x})^\top(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x}))^{-1}\nabla f(\mathbf{x})\leq-\frac{\bar{m}^2}{\underline{m}}\|\mathbf{x}-\mathbf{x}^\star\|^2$, confirming the exponential stability of (3). By virtue of Lemma 3.1, $-\frac{1}{N}\nabla f(\mathbf{x})^\top(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x})^{-1})\nabla f(\mathbf{x})\leq-\nabla f(\mathbf{x})^\top(\sum_{i=1}^N\mathbf{H}^i(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$, which indicates that the derivative of $V(\mathbf{x})$ is more negative along the trajectories of (4) than those of (3). Hence, we can conclude that the rate of convergence of (4) is no worse than the convergence rate of (3). □

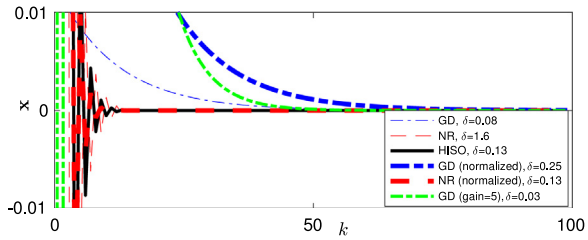


Fig. 2. The convergence of the Euler discretized GD, NR and HISO algorithms under different conditions to find the minimum of the cost function, $f(\mathbf{x}) = \sum_{i=1}^{10} a^i x^2 + b^i x^4$ where a^i and b^i are randomly chosen in $[0, 0.1]$. For each algorithm the stepsize is set to its optimum value, obtained numerically, corresponding to its fastest convergence.

Comparing rate of convergence of continuous-time optimization algorithms is a rather subtle matter. Any claim for an accelerated convergence by an algorithm meets the counter-argument that the 'simple' continuous-time gradient descent algorithm can be made arbitrarily fast using large scalar multiplicative gains. To address this dilemma, one can think of continuous-time algorithms as first-order integrator dynamics $\dot{\mathbf{x}} = \alpha\,\mathbf{u}$ with $\alpha \in \mathbb{R}_{>0}$, where the system input $\alpha\,\mathbf{u}$ is the control effort of the algorithm. Suppose the control effort is bounded as $\|\alpha\mathbf{u}\| \leq \alpha\kappa_0\|\mathbf{x} - \mathbf{x}^\star\|$, with $\kappa_0 \in \mathbb{R}_{>0}$. For an exponentially convergent algorithm, by virtue of Khalil (2002, Theorem 4.14), there exists a Lyapunov function that satisfies $\kappa_1\|\mathbf{x} - \mathbf{x}^\star\|^2 \leq V(\mathbf{x}) \leq \kappa_2\|\mathbf{x} - \mathbf{x}^\star\|^2$, and $\dot{V} \leq -\alpha\kappa_3\|\mathbf{x} - \mathbf{x}^\star\|^2$ for some $\kappa_1, \kappa_2, \kappa_3 \in \mathbb{R}_{>0}$. Then, by virtue of Khalil (2002, Theorem 4.10) the exponential rate of convergence of the algorithm is $\frac{\alpha\kappa_3}{2\kappa_2}$, indicating that increasing α increases the rate of convergence. Now, on the other hand, for the Euler-discretized form of the algorithm, i.e., $\mathbf{x}(k+1) = \mathbf{x}(k) + \delta\alpha\mathbf{u}$, $k \in \mathbb{Z}_{\geq 0}$, using the same Lyapunov function, we obtain $\Delta V(\mathbf{x}(k)) = V(\mathbf{x}(k+1)) - V(\mathbf{x}(k)) \leq -\delta\alpha\kappa_3\|\mathbf{x} - \mathbf{x}^\star\|^2 + \frac{\delta^2\alpha^2}{2}\mathbf{u}^\top\nabla^2 V(\zeta)\mathbf{u}$ where, $\zeta \in [\mathbf{x}(k), \mathbf{x}(k+1))$. Let $\nabla^2 V(\zeta) \leq \beta\mathbf{I}$, which is normally satisfied in optimization problems. Then, we can write $\Delta V(\mathbf{x}(k)) \leq -\delta\alpha(\kappa_3 - \frac{\delta\alpha}{2}\beta\kappa_0^2)\|\mathbf{x} - \mathbf{x}^\star\|^2$, which indicates that an admissible stepsize δ for Euler-discretized form of the algorithm should satisfy $0 < \delta < \frac{2\kappa_3}{\alpha\beta\kappa_0^2}$. Thus, increasing α results in smaller stepsizes. Moreover, algorithms that employ larger control effort (larger $\alpha\,\kappa_0$) will have smaller stepsize. As such, to be mindful of practical Euler-discretize implementation of continuous-time algorithms, any claim to a continuous-time algorithm being faster than another should be evaluated under the requirement that the algorithms employ the same maximum control effort level. Fig. 2 shows the convergence behavior of the discrete-time implementation of the gradient descent (GD), NR, and HISO algorithms. As we can see, NR and HISO show comparable responses and also faster convergence than GD. For all three cases, the maximum control effort happens at the initial time, with GD having the largest and NR having the smallest values. If we normalize the control efforts of NR and GD with respect to that of HISO by using, respectively, gains $\|\frac{1}{N}\sum_{i=1}^N \mathbf{H}^i(\mathbf{x}(0))^{-1}\|$ and $\|\sum_{i=1}^N \mathbf{H}^i(\mathbf{x}(0))\|\|\frac{1}{N}\sum_{i=1}^N \mathbf{H}^i(\mathbf{x}(0))^{-1}\|$, the GD algorithm can use larger stepsize and NR should use a smaller stepsize, with GD still showing slower convergence. Notice that, as Fig. 2 shows, if we increase the GD algorithm's control effort by using the gain $\alpha = 5$, the discrete-time implementation still has slower convergence because we are forced to use a smaller Euler discretization stepsize.

HISO algorithm uses the sum of the inverse of the Hessian of the local cost functions rather than the inverse of the sum of the local Hessians as in the NR algorithm. This trait, as shown below, results in a more efficient distributed implementation for algorithm (4), in which agents only incur a cost of $O(Nd)$ in communication, computations, and storage rather than $O(Nd^2)$ as in the distributed NR algorithms in the literature (Bof et al., 2019; Varagnolo et al., 2015).

4. Distributed HISO algorithm

Our proposed distributed implementation of the HISO algorithm is

$$\mathbf{z}^i = \mathbf{g}^i(\mathbf{x}^i) + \mathbf{v}^i, \tag{7a}$$

$$\dot{\mathbf{v}}^i = -\sum_{j=1}^N a_{ij}\,\mathrm{sgn}(\mathbf{z}^i - \mathbf{z}^j) + \sum_{j=1}^N a_{ij}(\mathbf{x}^i - \mathbf{x}^j), \tag{7b}$$

$$\dot{\mathbf{x}}^i = -\mathbf{H}^i(\mathbf{x})^{-1}\big(\mathbf{z}^i + \sum_{j=1}^N a_{ij}(\mathbf{x}^i - \mathbf{x}^j)\big), \tag{7c}$$

$i \in \{1, \ldots, N\}$. Conceptually, our approach to construct (7) was to use the finite-time dynamic average consensus algorithm of Chen, Cao, and Ren (2012) ((7a) and (7b)) with input $\mathbf{g}^i(\mathbf{x}^i) = \nabla f^i(\mathbf{x}^i)$ to generate $\mathbf{z}^i \to \frac{1}{N}\sum_{j=1}^N \mathbf{g}^j(\mathbf{x}^j)$ as $t \to \infty$. For algorithm of Chen et al. (2012) to converge we need $\sum_{i=1}^N \mathbf{v}^i(0) = \mathbf{0}$, which can trivially be satisfied using $\mathbf{v}^i(0) = \mathbf{0}$. Next, we noticed that the collective dynamics exhibits $\sum_{i=1}^N \dot{\mathbf{g}}^i(\mathbf{x}^i) \to -\sum_{j=1}^N \mathbf{g}^j(\mathbf{x}^j)$ as $\mathbf{z}^i$ converges. Then, if agreement occurs, every agent has a copy of the HISO algorithm locally. We added $\sum_{j=1}^N a_{ij}(\mathbf{x}^i - \mathbf{x}^j)$ to (7b) and (7c) for technical reasons to create agreement between the decision vector of the agents.

In what follows, we provide a formal proof of convergence and stability analysis of (7). For analysis, we write algorithm (7) in the compact form

$$\dot{\mathbf{z}} = -\mathbf{B}\,\mathrm{sgn}(\mathbf{B}^\top\mathbf{z}) + \mathbf{B}\mathbf{B}^\top\mathbf{x} + \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{g}(\mathbf{x}), \tag{8a}$$

$$\dot{\mathbf{x}} = -\mathcal{H}^{-1}\big(\mathbf{z} + \mathbf{B}\mathbf{B}^\top\mathbf{x}\big), \tag{8b}$$

where $\mathbf{B} = \mathbf{B} \otimes \mathbf{I}_d$, $\mathcal{H} = \mathrm{diag}(\mathbf{H}^1(\mathbf{x}), \ldots, \mathbf{H}^N(\mathbf{x}))$, $\mathbf{g}(\mathbf{x}) = [\mathbf{g}^1(\mathbf{x}^1)^\top, \ldots, \mathbf{g}^N(\mathbf{x}^N)^\top]^\top$ and $\mathbf{\Pi} = \Pi_N \otimes \mathbf{I}_d$ with the network aggregated variables $\mathbf{z}, \mathbf{x} \in \mathbb{R}^{dN}$. The following result shows that agents arrive at agreement in their $\{\mathbf{z}^i\}_{i=1}^N$ in finite time, and in their $\{\mathbf{x}^i\}_{i=1}^N$ as $t \to \infty$.

Lemma 4.1 (*Consensus in Algorithm (7) Over Connected Graphs*)**.** *Let $\mathcal{G}$ be a connected graph. Under Assumption 2.1, starting algorithm (7) over $\mathcal{G}$ from any $\mathbf{x}^i(0), \mathbf{v}^i(0) \in \mathbb{R}^d$, $\sum_{i=1}^N \mathbf{v}^i(0) = \mathbf{0}$, every $\mathbf{x}^i(t)$, $i \in \{1, \ldots, N\}$, converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{x}^j$ as time goes to infinity, while every $\mathbf{z}^i$, $i \in \{1, \ldots, N\}$, converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{g}^j(\mathbf{x}^j)$ in finite time.*

Proof. (7b) leads to $\sum_{i=1}^N \dot{\mathbf{v}}^i = \mathbf{0}$, which along with $\sum_{i=1}^N \mathbf{v}^i(0) = \mathbf{0}$ gives $\sum_{i=1}^N \mathbf{v}^i(t) = \mathbf{0}$ for any $t \in \mathbb{R}_{\geq 0}$. Moreover, from (7a), we obtain

$$\sum_{i=1}^N \mathbf{z}^i(t) = \sum_{i=1}^N \mathbf{g}^i(\mathbf{x}^i(t)), \quad t \in \mathbb{R}_{\geq 0}. \tag{9}$$

Next, note that $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{g}(\mathbf{x}) = \mathcal{H}\dot{\mathbf{x}}$. Then, we can obtain from (8) that $\dot{\mathbf{z}} = -\mathbf{B}\,\mathrm{sgn}(\mathbf{B}^\top\mathbf{z}) - \mathbf{z}$, which gives

$$\sum_{i=1}^N \dot{\mathbf{z}}^i = -\sum_{i=1}^N \mathbf{z}^i \to \sum_{i=1}^N \mathbf{z}^i(t) = \mathrm{e}^{-t}\sum_{i=1}^N \mathbf{z}^i(0), \quad t \in \mathbb{R}_{\geq 0}. \tag{10}$$

Moreover, using

$$\tilde{\mathbf{z}}(t) = \mathbf{\Pi}\,\mathbf{z}(t), \tag{11a}$$

$$\tilde{\mathbf{x}}(t) = \mathbf{\Pi}\,\mathbf{x}(t), \tag{11b}$$

(8) can be written as

$$\dot{\tilde{\mathbf{z}}} = -\mathbf{B}\,\mathrm{sgn}(\mathbf{B}^\top\tilde{\mathbf{z}}) - \tilde{\mathbf{z}}, \tag{12a}$$

$$\dot{\tilde{\mathbf{x}}} = -\mathbf{\Pi}\,\mathcal{H}^{-1}\big(\tilde{\mathbf{z}} + \frac{\mathrm{e}^{-t}}{N}\mathbf{1}_N \otimes \sum_{i=1}^{N}\mathbf{z}^i(0) + \mathbf{B}\mathbf{B}^\top\tilde{\mathbf{x}}\big). \tag{12b}$$

Here, we used (10). Moreover, given (11) we obtain $\mathbf{1}_N \otimes \sum_{i=1}^N \tilde{\mathbf{z}}^i(t) = \mathbf{0}$ and $\mathbf{1}_N \otimes \sum_{i=1}^N \tilde{\mathbf{x}}^i(t) = \mathbf{0}$, which hold for any $t \in \mathbb{R}_{\geq 0}$.

Next, we show that $\mathbf{B}^\top\tilde{\mathbf{z}}(t)$ goes to zero in finite time. Defining $\hat{\mathbf{z}} = \mathbf{B}^\top\tilde{\mathbf{z}}(t)$ and $\hat{\mathbf{x}} = \mathbf{B}^\top\tilde{\mathbf{x}}(t)$, from (12) we get

$$\dot{\hat{\mathbf{z}}} = -\mathbf{B}^\top\mathbf{B}\mathrm{sgn}(\hat{\mathbf{z}}) - \hat{\mathbf{z}}, \tag{13a}$$

$$\dot{\hat{\mathbf{x}}} = -\mathbf{B}^\top\mathcal{H}^{-1}(\tilde{\mathbf{z}} + \frac{\mathrm{e}^{-t}}{N}\mathbf{1}_N \otimes \sum_{i=1}^{N}\mathbf{z}^i(0)) - \mathbf{B}^\top\mathcal{H}^{-1}\mathbf{B}\hat{\mathbf{x}}. \tag{13b}$$

To analyze the stability of (13a) consider

$$V = \frac{1}{2}\hat{\mathbf{z}}^\top(\mathbf{B}^\top\mathbf{B})^+\hat{\mathbf{z}} \leq \bar{\lambda}\|\hat{\mathbf{z}}\|_2^2, \tag{14}$$

where $\bar{\lambda}$ is the maximum eigenvalue of $\frac{1}{2}(\mathbf{B}^\top\mathbf{B})^+$. Note that $(\mathbf{B}^\top\mathbf{B})^+ \geq 0$. The Lie derivative of V along (13a) is equal to

$$\dot{V} = -\hat{\mathbf{z}}^\top\mathrm{sgn}(\hat{\mathbf{z}}) - \hat{\mathbf{z}}^\top\hat{\mathbf{z}} = -\|\hat{\mathbf{z}}\|_1 - \hat{\mathbf{z}}^\top\hat{\mathbf{z}} \leq -\|\hat{\mathbf{z}}\|_1.$$

Since $\|\hat{\mathbf{z}}\|_2 \leq \|\hat{\mathbf{z}}\|_1$, then we have $\dot{V} \leq -\|\hat{\mathbf{z}}\|_2 \leq 0$. As such, from (14), we obtain $\dot{V} \leq \frac{-1}{\sqrt{\bar{\lambda}}}\sqrt{V}$. Then, invoking the comparison Lemma (Khalil, 2002, Lemma 3.4), we have the $\sqrt{V} \leq \sqrt{V(0)} - \frac{1}{2\sqrt{\bar{\lambda}}}t$. Consequently, starting from any $V(0)$, $V(t)$ becomes zero at a finite time. Then, since $V = \frac{1}{2}\hat{\mathbf{z}}^\top(\mathbf{B}^\top\mathbf{B})^+\hat{\mathbf{z}} = \mathbf{z}^\top\mathbf{B}(\mathbf{B}^\top\mathbf{B})^+\mathbf{B}^\top\mathbf{z} = \mathbf{z}^\top\mathbf{\Pi}\mathbf{z} = (\mathbf{\Pi}\mathbf{z})^\top(\mathbf{\Pi}\mathbf{z}) = \tilde{\mathbf{z}}^\top\tilde{\mathbf{z}}$, and (9), we can conclude that every $\mathbf{z}^i$, $i \in \{1,\dots,N\}$, converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{g}^j(\mathbf{x}^j)$ in finite time, and also $\tilde{\mathbf{z}}$ is bounded and converges to zero in finite time ($\|\tilde{\mathbf{z}}(t)\|_2 \leq \|\tilde{\mathbf{z}}(0)\|_2 - \frac{1}{2\sqrt{\bar{\lambda}}}t$).

Next, we show that $\mathbf{B}\hat{\mathbf{x}}$ converges to 0 as $t \to \infty$. To this aim, we consider the radially unbounded Lyapunov function $W = \frac{1}{2}\hat{\mathbf{x}}^\top\hat{\mathbf{x}}$. The Lie derivative of this function along the trajectories of (13b) is

$$\dot{W} = -\hat{\mathbf{x}}^\top\mathbf{B}^\top\mathcal{H}^{-1}\mathbf{B}\hat{\mathbf{x}} - \hat{\mathbf{x}}^\top\mathbf{B}^\top\mathcal{H}^{-1}(\tilde{\mathbf{z}} + \frac{\mathrm{e}^{-t}}{N}\mathbf{1}_N \otimes \sum_{i=1}^{N}\mathbf{z}^i(0)).$$

Since $\tilde{\mathbf{z}}(t)$ vanishes in finite time, there exists a $t_1 \in \mathbb{R}_{>0}$ such that for any $t \geq t_1$, we obtain

$$\dot{W} = -\|\sqrt{\mathcal{H}^{-1}}\mathbf{B}\hat{\mathbf{x}}\|^2 - \hat{\mathbf{x}}^\top\mathbf{B}^\top\mathcal{H}^{-1}(\frac{\mathrm{e}^{-t}}{N}\mathbf{1}_N \otimes \sum_{i=1}^{N}\mathbf{z}^i(0)) \leq$$

$$-\|\sqrt{\mathcal{H}^{-1}}\mathbf{B}\hat{\mathbf{x}}\|\big(\|\sqrt{\mathcal{H}^{-1}}\mathbf{B}\hat{\mathbf{x}}\| - \frac{\mathrm{e}^{-t}}{N}\|\sqrt{\mathcal{H}^{-1}}(\mathbf{1}_N \otimes \sum_{i=1}^{N}\mathbf{z}^i(0))\|\big)$$

Note that at each time $t \geq t_1$, $\dot{W} \leq 0$ if and only if $\|\sqrt{\mathcal{H}^{-1}}\mathbf{B}\hat{\mathbf{x}}\| \geq \frac{\mathrm{e}^{-t}}{N}\|\sqrt{\mathcal{H}^{-1}}(\mathbf{1}_N \otimes \sum_{i=1}^N \mathbf{z}^i(0))\|$. Then, at any $t_2 \geq t_1$ if the trajectories of (13) satisfy $\hat{\mathbf{x}}(t_2) \in \mathcal{S} = \{\hat{\mathbf{x}} \in \mathbb{R}^{Nd} \mid \|\sqrt{\mathcal{H}^{-1}}\mathbf{B}\hat{\mathbf{x}}\| \leq \frac{\mathrm{e}^{-t_2}}{\sqrt{\bar{m}N}}\|(\mathbf{1}_N \otimes \sum_{i=1}^N \mathbf{z}^i(0))\|\}$, then $\hat{\mathbf{x}}(t) \in \mathcal{S}$ for all $t \geq t_2$, otherwise, $\dot{W}(t_2) < 0$. Here, we used the fact that $\|\sqrt{\mathcal{H}^{-1}(\mathbf{x}(t))}\| \leq \frac{1}{\sqrt{\bar{m}}}$, which ensures that $\frac{\mathrm{e}^{-t}}{N}\|\sqrt{\mathcal{H}^{-1}(\mathbf{x}(t))}(\mathbf{1}_N \otimes \sum_{i=1}^N \mathbf{z}^i(0))\| \leq \frac{\mathrm{e}^{-t}}{\sqrt{\bar{m}}N}\|\mathbf{1}_N \otimes \sum_{i=1}^N \mathbf{z}^i(0)\| < \frac{\mathrm{e}^{-t_2}}{\sqrt{\bar{m}}N}\|\mathbf{1}_N \otimes \sum_{i=1}^N \mathbf{z}^i(0)\|$ for any $t > t_2$. Therefore, as $t \to \infty$, we have the guarantees that along the trajectories of the system, $\|\sqrt{\mathcal{H}(\mathbf{x}(t))^{-1}}\mathbf{B}\hat{\mathbf{x}}(t)\|$ goes to zero, which means that $\mathbf{B}\hat{\mathbf{x}}(t)$ goes to zero. Consequently, because of $\hat{\mathbf{x}} = \mathbf{B}^\top\tilde{\mathbf{x}}$ and $\mathbf{B}\mathbf{B}^\top = \mathbf{L} \otimes \mathbf{I}_d$ we can conclude that $(\mathbf{L} \otimes \mathbf{I}_d)\tilde{\mathbf{x}}(t)$ goes to zero as $t \to \infty$. Given that the graph is connected, then, as $t \to \infty$, $\hat{\mathbf{x}}$ goes to $\mathbf{1}_N \otimes \theta$, $\theta \in \mathbb{R}^d$, which given $\mathbf{1}_N \otimes \sum_{i=1}^N \tilde{\mathbf{x}}^i(t) = \mathbf{0}$, it means that $\tilde{\mathbf{x}}(t)$ goes to zero as $t \to \infty$. As a result, it follows from (11b) that $\mathbf{x}^i$ converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{x}^j(t)$ as $t \to \infty$. □

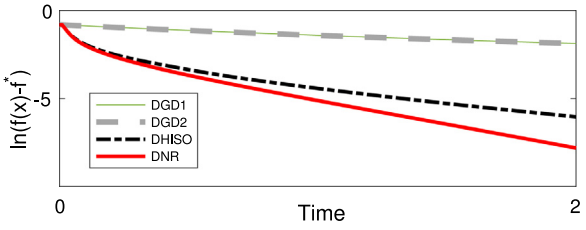


Fig. 3. Convergence of DGD1, DGD2, DHISO and DNR of Varagnolo et al. (2015) algorithms in logarithmic scale.

Remark 4.1 (*Remark on the Proof of Lemma 4.1*)**.** The solution of (7) is in the sense of Filippov (Filippov, 1988) since the solution is piecewise differentiable. The Filippov approach provides multi-valued functions for the solution of (7) over discontinuity points. Note, however, that our stability analysis is valid since the Lyapunov function is smooth and decreasing over every Filippov solution of (7). □

Lemma 4.1 showed that the trajectories of distributed HISO algorithm (7) converge to agreement space. The next theorem shows that this property indeed results in $\mathbf{x}^i$, $i \in \{1,\dots,N\}$ converging to $\mathbf{x}^\star$, the unique solution of the optimization problem (1), as $t \to \infty$.

Theorem 4.1 (*Convergence of Algorithm (7)*)**.** *Suppose the graph $\mathcal{G}$ is connected and let Assumption 2.1 hold. Then, starting algorithm (7) over $\mathcal{G}$ from any $\mathbf{x}^i(0), \mathbf{v}^i(0) \in \mathbb{R}^d$, $\sum_{i=1}^N \mathbf{v}^i(0) = \mathbf{0}$, every $\mathbf{x}^i(t)$, $i \in \{1,\dots,N\}$, converges to $\mathbf{x}^\star$, the unique minimizer of (1) and every $\mathbf{z}^i$ converges to zero as $t \to \infty$.*

Proof. From (11a), $\mathbf{z}^i = \tilde{\mathbf{z}}^i + \frac{1}{N}\sum_{j=1}^N \mathbf{z}^i$. Recall from the proof of Lemma 4.1 that under the stated initial condition, $\tilde{\mathbf{z}}^i$, $i \in \{1,\dots,N\}$, converges to zero in finite time. Then, convergence of $\mathbf{z}^i$, $i \in \{1,\dots,N\}$, to zero follows from (10). Next, notice that (9) and (10) indicate that $\sum_{j=1}^N \mathbf{g}^j(\mathbf{x}^j)$ goes to zero as $t \to \infty$. Then, because Lemma 4.1 guarantees that $\mathbf{x}^i$, $i \in \{1,\dots,N\}$ converges to $\frac{1}{N}\sum_{j=1}^N \mathbf{x}^j$ as $t \to \infty$, we can conclude that $\frac{1}{N}\sum_{j=1}^N \mathbf{x}^j$, and subsequently every $\mathbf{x}^i$ converges to $\mathbf{x}^\star$ as $t \to \infty$. □

5. Numerical example

We consider a distributed binary classification problem using logistic regression over a connected graph of Fig. 1. Each agent $i \in \{1,\dots,N\}$ has access to m^i training samples $(\mathbf{c}_{ij}, y_{ij}) \in \mathbb{R}^p \times \{-1,+1\}$, where $\mathbf{c}_{ij}$ contains p features of the jth training data at agent i, and y_{ij} is the corresponding binary label. The agents minimize $f = \sum_{i=1}^N f^i(\mathbf{w}, b)$ cooperatively, where $\mathbf{w} \in \mathbb{R}^p$, $b \in \mathbb{R}$, and each f^i is given by $f^i(\mathbf{w}, b) = \sum_{j=1}^{m^i} \ln(1+\mathrm{e}^{-(\mathbf{w}^\top\mathbf{c}_{ij}+b)y_{ij}}) + \frac{\lambda}{2N}\|\mathbf{w}\|^2$. We generated the feature vectors $\mathbf{c}_{ij}$s randomly from two distinct Gaussian distributions corresponding to two different labels, $+1$ and -1. Here, $p = 5$, $m = 10$, and $\lambda = 2$. Fig. 3 shows the trajectories of the cost function when the problem is solved via: distributed gradient descent algorithm of Kia et al. (2014) (DGD1), distributed gradient descent algorithm obtained from (7) when $\mathbf{H}^i(\mathbf{x})$ are replaced by $\mathbf{I}_d$ (DGD2), our proposed distributed HISO algorithm (DHISO) and distributed NR algorithm (DNR) proposed in Varagnolo et al. (2015). As Fig. 3 shows, DHISO and DNR algorithms both converge faster than the gradient descent algorithms. Moreover, DHISO algorithm demonstrates a comparable response to that of the DNR but without requiring the neighboring agents

to exchange their local Hessians with each other that the DNR algorithm of Varagnolo et al. (2015) requires.

6. Conclusion

We studied a novel second-order continuous-time distributed fast converging solution for an unconstrained optimization problem. Our approach guarantees convergence to the minimizer while keeping the communication and storage costs efficient, in order of $O(Nd)$ as opposed to $O(Nd^2)$ for the existing results in the literature. Future work includes obtaining a discrete-time implementation of our algorithm with formal convergence guarantees.

References

Bertsekas, D. (1999). *Nonlinear programming*.

Bof, N., Carli, R., Notarstefano, G., Schenato, L., & Varagnolo, D. (2019). Multiagent Newton-Raphson optimization over lossy networks. *IEEE Transactions on Automatic Control*, *64*(7), 2983–2990.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*, 1–122.

Chen, F., Cao, Y., & Ren, W. (2012). Distributed average tracking of multiple time-varying reference signals with bounded derivatives. *IEEE Transactions on Automatic Control*, *57*(12), 3169–3174.

Droge, G., Kawashima, H., & Egerstedt, M. (2014). Continuous-time proportional-integral distributed optimisation for networked systems. *Journal of Control and Decision*, *1*(3), 191–213.

Duchi, J., Agarwal, A., & Wainwright, M. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, *57*(3), 592–606.

Filippov, A. (1988). *Differential equations with discontinuous righthand sides*. Springer.

George, J., & Freeman, R. (2019). Robust dynamic average consensus algorithms. *IEEE Transactions on Automatic Control*, *64*(11), 4615–4622.

Henriques, J. F., Ehrhardt, S., Albanie, S., & Vedaldi, A. (2018). Small steps and giant leaps:Minimal Newton solvers for deep learning. Available at https://arxiv.org/abs/1805.08095.

Johansson, B., Rabi, M., & Johansson, M. (2009). A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, *20*, 1157–1170.

Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Englewood Cliffs, NJ: Prentice Hall.

Kia, S. S., Cortés, J., & Martínez, S. (2014). Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, *55*, 254–264.

Lu, J., & Tang, C. (2012). Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case. *IEEE Transactions on Automatic Control*, *57*(9), 2348–2354.

Mansoori, F., & Wei, E. (2020). A fast distributed asynchronous Newton-based optimization algorithm. *IEEE Transactions on Automatic Control*, *65*(7), 2769–2784.

Mokhtari, A., Ling, Q., & Ribeiro, A. (2015). Network Newton-part I: Algorithm and convergence. Available at https://arxiv.org/abs/1504.06017.

Nedić, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, *54*, 48–61.

Nordstrom, K. (2011). Convexity of the inverse and moore–penrose inverse. *Linear Algebra and its Applications*, *434*, 1489–1512.

Varagnolo, D., Zanella, F., Cenedese, A., Pillonetto, G., & Schenato, L. (2015). Newton–RAphson consensus for distributed convex optimization. *IEEE Transactions on Automatic Control*, *61*(4), 994–1009.

Wang, J., & Elia, N. (2011). A control perspective for centralized and distributed convex optimization. In *IEEE int. conf. on decision and control*.

Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., & Mahoney, M. W. (2020). ADAHESSIAN: An adaptive second order optimizer for machine learning. Available at https://arxiv.org/abs/2006.00719.

Zanella, F., Varagnolo, D., Cenedese, A., Pillonetto, G., & Schenato, L. (2011). Newton-Raphson consensus for distributed convex optimization. In *IEEE int. conf. on decision and control* (pp. 5917–5922).

Zhu, M., & Martínez, S. (2012). On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, *1*, 151–164.