Network Newton Distributed Optimization Methods

Aryan Mokhtari, Qing Ling, and Alejandro Ribeiro

***Abstract*—We study the problem of minimizing a sum of convex objective functions, where the components of the objective are available at different nodes of a network and nodes are allowed to only communicate with their neighbors. The use of distributed gradient methods is a common approach to solve this problem. Their popularity notwithstanding, these methods exhibit slow convergence and a consequent large number of communications between nodes to approach the optimal argument because they rely on first-order information only. This paper proposes the network Newton (NN) method as a distributed algorithm that incorporates second-order information. This is done via distributed implementation of approximations of a suitably chosen Newton step. The approximations are obtained by truncation of the Newton step's Taylor expansion. This leads to a family of methods defined by the number K of Taylor series terms kept in the approximation. When keeping K terms of the Taylor series, the method is called NN-K and can be implemented through the aggregation of information in K-hop neighborhoods. Convergence to a point close to the optimal argument at a rate that is at least linear is proven and the existence of a tradeoff between convergence time and the distance to the optimal argument is shown. The numerical experiments corroborate reductions in the number of iterations and the communication cost that are necessary to achieve convergence relative to first-order alternatives.**

***Index Terms*—Multi-agent network, distributed optimization, Newton's method.**

I. Introduction

DISTRIBUTED optimization algorithms are used to solve the problem of minimizing a global cost function over a set of nodes in situations where the objective function is defined as a sum of local functions. To be more precise, consider a variable $\mathbf{x} \in \mathbb{R}^p$ and a connected network containing n agents each of which has access to a local function $f_i : \mathbb{R}^p \to \mathbb{R}$. The agents cooperate in minimizing the aggregate cost function $f : \mathbb{R}^p \to \mathbb{R}$ taking values $f(\mathbf{x}) := \sum_{i=1}^n f_i(\mathbf{x})$. I.e., agents cooperate in solving the problem

$$\mathbf{x}^* := \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathbb{R}^p} \sum_{i=1}^n f_i(\mathbf{x}). \tag{1}$$

Problems of this form arise often in, e.g., decentralized control systems [3], [4], wireless systems [5], [6], sensor networks [7]–[9], and large scale machine learning [10]–[12].

There are different algorithms to solve (1) in a distributed manner. The most popular choices are decentralized gradient descent (DGD) [13]–[16], distributed implementations of the alternating direction method of multipliers [7], [17]–[20], and decentralized dual averaging [21], [22]. Although there are substantial differences between them, these methods can be generically abstracted as combinations of local descent steps followed by variable exchanges and averaging of information among neighbors. A feature common to all of these algorithms is the slow convergence rate in ill-conditioned problems since they operate on first order information only. This is not surprising because gradient descent methods in centralized settings where the aggregate function gradient is available at a single server have the same difficulties in problems with skewed curvature [see Chapter 9 of [23]].

This issue is addressed in centralized optimization by Newton's method that uses second order information to determine a descent direction adapted to the objective's curvature [see Chapter 9 of [23]]. In general, second order methods are not available in distributed settings because distributed approximations of Newton steps are difficult to devise. In the particular case of flow optimization problems, these approximations are possible when operating in the dual domain and have led to the development of the accelerated dual descent methods [24], [25]. As would be expected, these methods result in large reductions of convergence times.

Our goal is to develop approximate Newton's methods to solve (1) in distributed settings where agents have access to their local functions only and exchange variables with neighboring agents. We do so by introducing Network Newton (NN), a method that relies on distributed approximations of Newton steps for the global cost function f to accelerate convergence of DGD. We begin the paper with an alternative formulation of (1) and a brief discussion of DGD (Section II). We then introduce a reinterpretation of DGD as an algorithm that utilizes gradient descent to solve a penalized version of (1) in lieu of the original optimization problem (Section II-A). This reinterpretation explains convergence of DGD to a neighborhood of $\mathbf{x}^*$. The volume of this neighborhood is given by the relative weight of the penalty function and the original objective which is controlled by a penalty coefficient.

If gradient descent on the penalized function finds an approximate solution to the original problem, the same solution can be found with a much smaller number of iterations by using Newton's method. Alas, distributed computation of Newton

Manuscript received November 2, 2015; revised May 31, 2016; accepted July 26, 2016. Date of publication October 13, 2016; date of current version October 31, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chong-Yung Chi. This work was supported in part by the National Science Foundation under Award CAREER CCF-0952867, in part by the Office of Naval Research under Contract ONR N00014-12-1-0997, and in part by the National Natural Science Foundation of China under Grant NSFC 61004137. This paper was presented in part at the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, November 2–5, 2014 and in part at the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, QLD, Australia, April 19–24, 2015. This paper expands the results and presents convergence proofs that are referenced in [1] and [2].

A. Mokhtari and A. Ribeiro are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: aryanm@seas.upenn.edu; aribeiro@seas.upenn.edu).

Q. Ling is with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: qingling@mail.ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2617829

steps requires global communication between all nodes in the network and is therefore impractical (Section III). To resolve this issue we approximate the Newton step of the penalized objective function by truncating the Taylor series of the exact Newton step (Section III-A). This approximation results in a family of methods indexed by the number of terms of the Taylor expansion that are kept in the approximation. The method that results from keeping K of these terms is termed NN-K. A fundamental observation here is that the Hessian of the penalized function has a sparsity structure that is the same sparsity pattern of the graph. Thus, when computing terms in the Hessian inverse expansion, the first order term is as sparse as the graph, the second term is as sparse as the two hop neighborhood, and, in general, the k-th term is as sparse as the k-hop neighborhood of the graph. Thus, implementation of the NN-K method requires aggregating information from K hops away. Increasing K makes NN-K arbitrarily close to Newton's method at the cost of increasing the communication overhead of each iteration. We point out that the same Taylor series is used in the development of the ADD algorithms, but this is done to solve a network utility maximization problem in the dual domain [24]. The Taylor expansion is utilized here to solve a consensus optimization problem in the primal domain.

Convergence of NN-K to the optimal argument of the penalized objective is established (Section IV). We do so by establishing several auxiliary bounds on the eigenvalues of the matrices involved in the definition of the method (Propositions 1-3 and Lemma 2). We show that a measure of the error between the Hessian inverse approximation utilized by NN-K and the actual inverse Hessian decays exponentially with the method index K. This exponential decrease hints that using a small value of K should suffice in practice. Convergence is formally claimed in Theorem 1 that shows the convergence rate is at least linear. It follows from this convergence analysis that larger penalty coefficients result in faster convergence that comes at the cost of increasing the distance between the optimal solutions of the original and penalized objectives.

We also study the convergence rate of the NN method as an approximation of Newton's method (Section IV-A). We show that for all iterations except the first few, a weighted gradient norm associated with NN-K iterates follows a decreasing path akin to the path that would be followed by Newton iterates (Lemma 3). The only difference between these residual paths is that the NN-K path contains a term that captures the error of the Hessian inverse approximation. Leveraging this similarity, it is possible to show that the rate of convergence is quadratic in a specific interval whose length depends on the order K of the selected network Newton method (Theorem 2). Existence of this quadratic convergence phase explains why NN-K methods converge faster than DGD – as we observe in experiments. It is also worth remarking that the error in the Hessian inverse approximation can be made arbitrarily small by increasing the method's order K and, as a consequence, the quadratic phase can be made arbitrarily large.

We wrap up the paper with numerical analyses (Section V). We first demonstrate the advantages of NN-K relative to alternative primal and dual methods for the minimization of a family of quadratic objective functions (Section V-A). Then, we study the effect of objective function condition number and show that the NN method outperforms first-order alternatives significantly in ill-conditioned problems (Section V-B). Further, we study the effect of network topology on the performance of NN (Section V-C). Moreover, we compare the convergence rate of NN in theory and practice to show the tightness of the bounds in this paper (Section V-D). The paper closes with concluding remarks (Section VI).

Notation. Vectors are written as $\mathbf{x} \in \mathbb{R}^n$ and matrices as $\mathbf{A} \in \mathbb{R}^{n\times n}$. The null space of matrix $\mathbf{A}$ is denoted by $\text{null}(\mathbf{A})$ and the span of a vector by $\text{span}(\mathbf{x})$. We use $\|\mathbf{x}\|$ and $\|\mathbf{A}\|$ to denote the Euclidean norm of vector $\mathbf{x}$ and matrix $\mathbf{A}$, respectively. The gradient of a function $f(\mathbf{x})$ is denoted as $\nabla f(\mathbf{x})$ and the Hessian matrix is denoted as $\nabla^2 f(\mathbf{x})$. The i-th largest eigenvalue of matrix $\mathbf{A}$ is denoted by $\mu_i(\mathbf{A})$.

II. Distributed Gradient Descent

The network that connects the n agents is assumed connected, symmetric, and specified by the neighborhoods $\mathcal{N}_i$ that contain the list of nodes that can communicate with i for $i = 1, \ldots, n$. In problem (1) agent i has access to the local cost $f_i(\mathbf{x})$ and agents cooperate to minimize the global cost $f(\mathbf{x})$. This specification is more naturally formulated by an alternative representation of (1) in which node i selects a local decision vector $\mathbf{x}_i \in \mathbb{R}^p$. Nodes then try to achieve the minimum of their local objective functions $f_i(\mathbf{x}_i)$, while keeping their variables equal to the variables $\mathbf{x}_j$ of neighbors $j \in \mathcal{N}_i$. This alternative formulation can be written as

$$\{\mathbf{x}_i^*\}_{i=1}^n := \underset{\{\mathbf{x}_i\}_{i=1}^n}{\arg\min} \sum_{i=1}^n f_i(\mathbf{x}_i), \quad \text{s.t. } \mathbf{x}_i = \mathbf{x}_j, \quad \text{for all } i,j \in \mathcal{N}_i. \tag{2}$$

Since the network is connected, the constraints $\mathbf{x}_i = \mathbf{x}_j$ for all i and $j \in \mathcal{N}_i$ imply that (1) and (2) are equivalent and we have $\mathbf{x}_i^* = \mathbf{x}^*$ for all i. This must be the case because for a connected network the constraints $\mathbf{x}_i = \mathbf{x}_j$ for all i and $j \in \mathcal{N}_i$ collapse the feasible space of (2) to a hyperplane in which all local variables are equal. When all variables are equal, the objectives in (1) and (2) coincide and so do their optima.

DGD is an established distributed method to solve (2) which relies on the introduction of nonnegative weights $w_{ij} \geq 0$ that are null if and only if $j \notin \mathcal{N}_i \cup \{i\}$ – the use of time varying weights w_{ij} is common in DGD implementations but not done here; see, e.g., [13]. Letting $t \in \mathbb{N}$ be a discrete time index and α a given stepsize, DGD is defined by the recursion

$$\mathbf{x}_{i,t+1} = \sum_{j=1}^n w_{ij}\mathbf{x}_{j,t} - \alpha\nabla f_i(\mathbf{x}_{i,t}), \qquad i = 1, \ldots, n. \tag{3}$$

Since $w_{ij} = 0$ when $j \neq i$ and $j \notin \mathcal{N}_i$, it follows from (3) that each agent i updates its variable $\mathbf{x}_i$ by performing an average over the estimates $\mathbf{x}_{j,t}$ of its neighbors $j \in \mathcal{N}_i$ and its own estimate $\mathbf{x}_{i,t}$, and descending through the negative local gradient $-\nabla f_i(\mathbf{x}_{i,t})$.

The weights in (3) cannot be arbitrary. To express conditions on the set of allowable weights define the matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ with entries w_{ij}. We require the weights to be symmetric, i.e., $w_{ij} = w_{ji}$ for all i, j, and such that the weights of a given node sum up to 1, i.e., $\sum_{j=1}^n w_{ij} = 1$ for all i. If the weights sum up to 1 we must have $\mathbf{W}\mathbf{1} = \mathbf{1}$ which implies that $\mathbf{I} - \mathbf{W}$ is rank

deficient. It is also customary to require the rank of $\mathbf{I}-\mathbf{W}$ to be exactly equal to $n-1$ so that the null space of $\mathbf{I}-\mathbf{W}$ is $\text{null}(\mathbf{I}-\mathbf{W})=\text{span}(\mathbf{1})$. We therefore have the following three restrictions on the matrix $\mathbf{W}$,

$$\mathbf{W}^T=\mathbf{W}, \quad \mathbf{W1}=\mathbf{1}, \quad \text{null}(\mathbf{I}-\mathbf{W})=\text{span}(\mathbf{1}). \tag{4}$$

If the conditions in (4) are true, it is possible to show that (3) approaches the solution of (1) in the sense that $\mathbf{x}_{i,t}\approx\mathbf{x}^*$ for all i and large t, [13]. The accepted interpretation of why (3) converges is that nodes are gradient descending towards their local minima because of the term $-\alpha\nabla f_i(\mathbf{x}_{i,t})$ but also perform an average of neighboring variables $\sum_{j=1}^n w_{ij}\mathbf{x}_{j,t}$. This latter consensus operation drives the agents to agreement. In the following section we show that (3) can be alternatively interpreted as a penalty method.

A. Penalty Method Interpretation

It is illuminating to define matrices and vectors so as to rewrite (3) as a single equation. To do so define the vectors $\mathbf{y}:=[\mathbf{x}_1;\dots;\mathbf{x}_n]$ and $\mathbf{h}(\mathbf{y}):=[\nabla f_1(\mathbf{x}_1);\dots;\nabla f_n(\mathbf{x}_n)]$. Vector $\mathbf{y}\in\mathbb{R}^{np}$ concatenates the local vectors $\mathbf{x}_i$, and the vector $\mathbf{h}(\mathbf{y})\in\mathbb{R}^{np}$ concatenates the gradients of the local functions f_i taken with respect to the local variable $\mathbf{x}_i$. Notice that $\mathbf{h}(\mathbf{y})$ is *not* the gradient of $f(\mathbf{x})$ and that a vector $\mathbf{y}$ with $\mathbf{h}(\mathbf{y})=\mathbf{0}$ does *not* necessarily solve (1). To solve (1) we need to have $\mathbf{x}_i=\mathbf{x}_j$ for all i and j with $\sum_{i=1}^n\nabla f_i(\mathbf{x}_i)=\mathbf{0}$. In any event, to rewrite (3) we also define the matrix $\mathbf{Z}:=\mathbf{W}\otimes\mathbf{I}\in\mathbb{R}^{np\times np}$ as the Kronecker product of the weight matrix $\mathbf{W}\in\mathbb{R}^{n\times n}$ and the identity matrix $\mathbf{I}\in\mathbb{R}^{p\times p}$. It is then ready to see that (3) is equivalent to

$$\mathbf{y}_{t+1}=\mathbf{Z}\mathbf{y}_t-\alpha\mathbf{h}(\mathbf{y}_t)=\mathbf{y}_t-\big[(\mathbf{I}-\mathbf{Z})\mathbf{y}_t+\alpha\mathbf{h}(\mathbf{y}_t)\big], \tag{5}$$

where in the second equality we added and subtracted $\mathbf{y}_t$ and regrouped terms. Inspection of (5) reveals that the DGD update formula at step t is equivalent to a (regular) gradient descent algorithm being used to solve the program

$$\mathbf{y}^*:=\arg\min F(\mathbf{y}):=\min\frac{1}{2}\mathbf{y}^T(\mathbf{I}-\mathbf{Z})\mathbf{y}+\alpha\sum_{i=1}^n f_i(\mathbf{x}_i). \tag{6}$$

This interpretation has been previously used in [14], [26] to design a Nesterov type acceleration of DGD. Indeed, given the definition of the function $F(\mathbf{y}):=(1/2)\mathbf{y}^T(\mathbf{I}-\mathbf{Z})\ \mathbf{y}+\alpha\sum_{i=1}^n f_i(\mathbf{x}_i)$ it follows that the gradient $\nabla F(\mathbf{y}_t)$ is given by

$$\mathbf{g}_t\ :=\ \nabla F(\mathbf{y}_t)\ =\ (\mathbf{I}-\mathbf{Z})\mathbf{y}_t+\alpha\mathbf{h}(\mathbf{y}_t). \tag{7}$$

Using (7) we rewrite (5) as $\mathbf{y}_{t+1}=\mathbf{y}_t-\mathbf{g}_t$ and conclude that DGD descends along the negative gradient of $F(\mathbf{y})$ with unit stepsize. The expression in (3) is just a distributed implementation of gradient descent that uses the gradient in (7). To confirm that this is true, observe that the ith element of the gradient $\mathbf{g}_t=[\mathbf{g}_{1,t};\dots;\mathbf{g}_{n,t}]$ is given by

$$\mathbf{g}_{i,t}=(1-w_{ii})\mathbf{x}_{i,t}-\sum_{j\in\mathcal{N}_i}w_{ij}\mathbf{x}_{j,t}+\alpha\nabla f_i(\mathbf{x}_{i,t}). \tag{8}$$

The gradient descent iteration $\mathbf{y}_{t+1}=\mathbf{y}_t-\mathbf{g}_t$ is then equivalent to (3) if we entrust node i with the implementation of the descent $\mathbf{x}_{i,t+1}=\mathbf{x}_{i,t}-\mathbf{g}_{i,t}$, where, we recall, $\mathbf{x}_{i,t}$ and $\mathbf{x}_{i,t+1}$ are the ith components of the vectors $\mathbf{y}_t$ and $\mathbf{y}_{t+1}$. Observe that the local gradient component $\mathbf{g}_{i,t}$ can be computed using local information and the $\mathbf{x}_{j,t}$ iterates of its neighbors $j\in\mathcal{N}_i$. This is as it should be, because the descent $\mathbf{x}_{i,t+1}=\mathbf{x}_{i,t}-\mathbf{g}_{i,t}$ is equivalent to (3).

Is it a good idea to descend on $F(\mathbf{y})$ to solve (1)? To some extent. Since we know that the null space of $\mathbf{I}-\mathbf{W}$ is $\text{null}(\mathbf{I}-\mathbf{W})=\text{span}(\mathbf{1})$ and that $\mathbf{Z}=\mathbf{W}\otimes\mathbf{I}$ we know that the null space of $\mathbf{I}-\mathbf{Z}$ is the set of consensus vectors, i.e., $\text{null}(\mathbf{I}-\mathbf{Z})=\big\{\mathbf{y}=[\mathbf{x}_1;\dots;\mathbf{x}_n]\ \big|\ \mathbf{x}_1=\cdots=\mathbf{x}_n\big\}$. Thus, $(\mathbf{I}-\mathbf{Z})\mathbf{y}=\mathbf{0}$ holds if and only if $\mathbf{x}_1=\cdots=\mathbf{x}_n$. Since the matrix $\mathbf{I}-\mathbf{Z}$ is positive semidefinite and symmetric, the same is true of the square root matrix $(\mathbf{I}-\mathbf{Z})^{1/2}$. Therefore, the optimization problem in (2) is equivalent to the optimization problem

$$\tilde{\mathbf{y}}^*:=\underset{\mathbf{x}}{\text{argmin}}\ \sum_{i=1}^n f_i(\mathbf{x}_i), \quad \text{s.t. } (\mathbf{I}-\mathbf{Z})^{1/2}\mathbf{y}=\mathbf{0}. \tag{9}$$

Indeed, for $\mathbf{y}=[\mathbf{x}_1;\dots;\mathbf{x}_n]$ to be feasible in (9) we must have $\mathbf{x}_1=\cdots=\mathbf{x}_n$. This is the same constraint imposed in (2) from where it follows that we must have $\tilde{\mathbf{y}}^*=[\mathbf{x}_1^*;\dots;\mathbf{x}_n^*]$ with $\mathbf{x}_i^*=\mathbf{x}^*$ for all i. The unconstrained minimization in (6) is a penalty version of (9). The penalty function associated with the constraint $(\mathbf{I}-\mathbf{Z})^{1/2}\mathbf{y}=\mathbf{0}$ is the squared norm $(1/2)\|(\mathbf{I}-\mathbf{Z})^{1/2}\mathbf{y}\|^2$ and the corresponding penalty coefficient is $1/\alpha$. Inasmuch as the penalty coefficient $1/\alpha$ is sufficiently large, the optimal arguments $\mathbf{y}^*$ and $\tilde{\mathbf{y}}^*$ are not too far apart.

The reinterpretation of (3) as a penalty method demonstrates that DGD is an algorithm that finds the optimal solution of (6), not (9) or its equivalent original formulations in (1) and (2). Using a fixed α the distance between $\mathbf{y}^*$ and $\tilde{\mathbf{y}}^*$ is of order $O(\alpha)$, [15]. To solve (9) we need to introduce a rule to progressively decrease α. In the following section we exploit the reinterpretation of (5) as a method to minimize (6) to propose an approximate Newton algorithm that can be implemented in a distributed manner.

III. Network Newton

Instead of solving (6) with a gradient descent method as in DGD, we can solve (6) using Newton's method. To implement Newton's method we need to compute the Hessian $\mathbf{H}_t:=\nabla^2F(\mathbf{y}_t)$ of F evaluated at $\mathbf{y}_t$ so as to determine the Newton step $\mathbf{d}_t:=-\mathbf{H}_t^{-1}\mathbf{g}_t$. Start by differentiating twice in (6) in order to write $\mathbf{H}_t$ as

$$\mathbf{H}_t:=\nabla^2F(\mathbf{y}_t)=\mathbf{I}-\mathbf{Z}+\alpha\mathbf{G}_t, \tag{10}$$

where $\mathbf{G}_t\in\mathbb{R}^{np\times np}$ is a block diagonal matrix formed by blocks $\mathbf{G}_{ii,t}\in\mathbb{R}^{p\times p}$ defined as

$$\mathbf{G}_{ii,t}=\nabla^2f_i(\mathbf{x}_{i,t}). \tag{11}$$

It follows from (10) and (11) that the Hessian $\mathbf{H}_t$ is block sparse with blocks $\mathbf{H}_{ij,t}\in\mathbb{R}^{p\times p}$ having the sparsity pattern of $\mathbf{Z}$, which is the sparsity pattern of the graph. The diagonal blocks are of the form $\mathbf{H}_{ii,t}=(1-w_{ii})\mathbf{I}+\alpha\nabla^2f_i(\mathbf{x}_{i,t})$ and the off diagonal blocks are not null only when $j\in\mathcal{N}_i$ in which case $\mathbf{H}_{ij,t}=w_{ij}\mathbf{I}$.

While the Hessian $\mathbf{H}_t$ is sparse, the inverse $\mathbf{H}_t$ is not. It is the latter that we need to compute the Newton step $\mathbf{d}_t:=\mathbf{H}_t^{-1}\mathbf{g}_t$. To overcome this problem we split the diagonal and off diagonal blocks of $\mathbf{H}_t$ and rely on a Taylor's expansion of the inverse – This splitting technique is inspired from the Taylor's expansion

used in [24]. To be precise, write $\mathbf{H}_t = \mathbf{D}_t - \mathbf{B}$ where the matrix $\mathbf{D}_t$ is defined as

$$\mathbf{D}_t := \alpha \mathbf{G}_t + 2\left(\mathbf{I} - \text{diag}(\mathbf{Z})\right) := \alpha \mathbf{G}_t + 2\left(\mathbf{I} - \mathbf{Z}_d\right), \quad (12)$$

where in the second equality we defined $\mathbf{Z}_d := \text{diag}(\mathbf{Z})$ for future reference. Since the diagonal weights must be $w_{ii} < 1$, the matrix $\mathbf{I} - \mathbf{Z}_d$ is positive definite. The same is true of the block diagonal matrix $\mathbf{G}_t$ because the local functions are assumed strongly convex. Therefore, the matrix $\mathbf{D}_t$ is block diagonal and positive definite. The ith diagonal block $\mathbf{D}_{ii,t} \in \mathbb{R}^p$ of $\mathbf{D}_t$ can be computed and stored by node i as $\mathbf{D}_{ii,t} = \alpha \nabla^2 f_i(\mathbf{x}_{i,t}) + 2(1 - w_{ii})\mathbf{I}$. To have $\mathbf{H}_t = \mathbf{D}_t - \mathbf{B}$ we must define $\mathbf{B} := \mathbf{D}_t - \mathbf{H}_t$. Considering the definitions of $\mathbf{H}_t$ and $\mathbf{D}_t$ in (10) and (12), it follows that

$$\mathbf{B} = \mathbf{I} - 2\mathbf{Z}_d + \mathbf{Z}. \quad (13)$$

Note that $\mathbf{B}$ is time-invariant and depends on the weight matrix $\mathbf{Z}$ only. As in the case of the Hessian $\mathbf{H}_t$, the matrix $\mathbf{B}$ is block sparse with blocks $\mathbf{B}_{ij} \in \mathbb{R}^{p \times p}$ having the sparsity pattern of $\mathbf{Z}$, which is the sparsity pattern of the graph. Node i can compute the diagonal blocks $\mathbf{B}_{ii} = (1 - w_{ii})\mathbf{I}$ and the off diagonal blocks $\mathbf{B}_{ij} = w_{ij}\mathbf{I}$ using information about its own and neighbors' weights.

Proceed now to factor $\mathbf{D}_t^{1/2}$ from both sides of the splitting relationship to write $\mathbf{H}_t = \mathbf{D}_t^{1/2}(\mathbf{I} - \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2})\mathbf{D}_t^{1/2}$. When we consider the Hessian inverse $\mathbf{H}^{-1}$, we can use the Taylor series $(\mathbf{I} - \mathbf{X})^{-1} = \sum_{j=0}^{\infty} \mathbf{X}^j$ with $\mathbf{X} = \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ to write

$$\mathbf{H}_t^{-1} = \mathbf{D}_t^{-1/2} \sum_{k=0}^{\infty} \left(\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}\right)^k \mathbf{D}_t^{-1/2}. \quad (14)$$

The sum in (14) converges if the absolute value of all the eigenvalues of the matrix $\mathbf{D}^{-1/2}\mathbf{B}\mathbf{D}^{-1/2}$ are strictly less than 1. For the time being we assume this to be the case but we will prove that this is true in Section IV. When the series converge, we can use truncations of this series to define approximations to the Newton step as we explain in the following section.

Remark 1: The Hessian decomposition $\mathbf{H}_t = \mathbf{D}_t - \mathbf{B}$ with the matrices $\mathbf{D}_t$ and $\mathbf{B}$ in (12) and (13), respectively, is not the only valid decomposition that we can use for Network Newton. Any decomposition of the form $\mathbf{H}_t = \mathbf{D}_t \pm \mathbf{B}_t$ is valid if $\mathbf{D}_t$ is positive definite and the eigenvalues of the matrix $\mathbf{D}_t^{-1/2}\mathbf{B}_t\mathbf{D}_t^{-1/2}$ are in the interval $(-1, 1)$. An example alternative decomposition is given by the matrices $\mathbf{D}_t = \alpha \mathbf{G}_t$ and $\mathbf{B} = \mathbf{I} - \mathbf{Z}$. This decomposition has the advantage of separating the effects of the function in $\mathbf{D}_t$ and the effects of the network in $\mathbf{B}$. The decomposition in (12) and (13) exhibits faster convergence of the series in (14) because the matrix $\mathbf{D}_t$ in (12) accumulates more weight in the diagonal than the matrix $\mathbf{D}_t = \alpha \mathbf{G}_t$. The study of alternative decompositions is beyond the scope of this paper.

A. Distributed Approximations of the Newton Step

Network Newton (NN) is defined as a family of algorithms that rely on truncations of the series in (14). The Kth member of this family, NN-K, considers the first $K + 1$ terms of the series to define the approximate Hessian inverse

$$\hat{\mathbf{H}}_t^{(K)^{-1}} := \mathbf{D}_t^{-1/2} \sum_{k=0}^{K} \left(\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}\right)^k \mathbf{D}_t^{-1/2}. \quad (15)$$

NN-K uses the approximate Hessian $\hat{\mathbf{H}}_t^{(K)^{-1}}$ as a curvature correction matrix that is used in lieu of the exact Hessian inverse $\mathbf{H}^{-1}$ to estimate the Newton step. I.e., instead of descending along the Newton step $\mathbf{d}_t := -\mathbf{H}_t^{-1}\mathbf{g}_t$ we descend along the NN-K step $\mathbf{d}_t^{(K)} := -\hat{\mathbf{H}}_t^{(K)^{-1}}\mathbf{g}_t$ as an approximation of $\mathbf{d}_t$. Using the explicit expression for $\hat{\mathbf{H}}_t^{(K)^{-1}}$ in (15) we write the NN-K step as

$$\mathbf{d}_t^{(K)} = -\mathbf{D}_t^{-1/2} \sum_{k=0}^{K} \left(\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}\right)^k \mathbf{D}_t^{-1/2}\,\mathbf{g}_t, \quad (16)$$

where, we recall, $\mathbf{g}_t$ as the gradient of the function $F(\mathbf{y})$ defined in (7). The NN-K update can then be written as

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \epsilon\, \mathbf{d}_t^{(K)}, \quad (17)$$

where ϵ is a properly selected stepsize – see Theorem 1 for specific conditions. The algorithm defined by recursive application of (17) can be implemented in a distributed manner because the truncated series in (15) has a local structure controlled by the parameter K. To explain this statement better define the components $\mathbf{d}_{i,t}^{(K)} \in \mathbb{R}^p$ of the NN-K step $\mathbf{d}_t^{(K)} = [\mathbf{d}_{1,t}^{(K)}; \ldots; \mathbf{d}_{n,t}^{(K)}]$. A distributed implementation of (17) requires that node i computes $\mathbf{d}_{i,t}^{(K)}$ so as to implement the local descent $\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} + \epsilon \mathbf{d}_{i,t}^{(K)}$. The key observation here is that the step component $\mathbf{d}_{i,t}^{(K)}$ can indeed be computed through local operations. Specificially, begin by noting that as per the definition of the NN-K descent direction in (16) the sequence of NN descent directions satisfies

$$\mathbf{d}_t^{(k+1)} = \mathbf{D}_t^{-1}\mathbf{B}\mathbf{d}_t^{(k)} - \mathbf{D}_t^{-1}\mathbf{g}_t = \mathbf{D}_t^{-1}\left(\mathbf{B}\mathbf{d}_t^{(k)} - \mathbf{g}_t\right). \quad (18)$$

Since the matrix $\mathbf{B}$ has the sparsity pattern of the graph, this recursion can be decomposed into local components

$$\mathbf{d}_{i,t}^{(k+1)} = \mathbf{D}_{ii,t}^{-1}\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \mathbf{B}_{ij}\mathbf{d}_{j,t}^{(k)} - \mathbf{g}_{i,t}\right), \quad (19)$$

The matrix $\mathbf{D}_{ii,t} = \alpha \nabla^2 f_i(\mathbf{x}_{i,t}) + 2(1 - w_{ii})\mathbf{I}$ is stored and computed at node i. The gradient component $\mathbf{g}_{i,t} = (1 - w_{ii})\mathbf{x}_{i,t} - \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_{j,t} + \alpha \nabla f_i(\mathbf{x}_{i,t})$ is also stored and computed at i. Node i can also evaluate the values of the matrix blocks $\mathbf{B}_{ii} = (1 - w_{ii})\mathbf{I}$ and $\mathbf{B}_{ij} = w_{ij}\mathbf{I}$. Thus, if the NN-$k$ step components $\mathbf{d}_{j,t}^{(k)}$ are available at neighbors j, node i can determine the NN-$(k + 1)$ step component $\mathbf{d}_{i,t}^{(k+1)}$ upon being communicated that information.

The expression in (19) represents an iterative computation embedded inside the NN-K recursion in (17). At time index t, we compute the local component of the NN-0 step $\mathbf{d}_{i,t}^{(0)} = -\mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}$. Upon exchanging this information with neighbors we use (19) to determine the NN-1 step $\mathbf{d}_{i,t}^{(1)}$. These can be exchanged to compuer $\mathbf{d}_{i,t}^{(2)}$ as in (19). Repeating this procedure

Algorithm 1: Network Newton-K method at node i.

Require: Initial iterate $\mathbf{x}_{i,0}$. Weights w_{ij}. Penalty coefficient α.

1: $\mathbf{B}$ matrix blocks: $\mathbf{B}_{ii} = (1 - w_{ii})\mathbf{I}$ and $\mathbf{B}_{ij} = w_{ij}\mathbf{I}$
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: $\quad$ $\mathbf{D}$ matrix block: $\mathbf{D}_{ii,t} = \alpha\nabla^2 f_i(\mathbf{x}_{i,t}) + 2(1 - w_{ii})\mathbf{I}$
4: $\quad$ Exchange iterates $\mathbf{x}_{i,t}$ with neighbors $j \in \mathcal{N}_i$.
5: $\quad$ Gradient: $\mathbf{g}_{i,t} = (1 - w_{ii})\mathbf{x}_{i,t} - \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_{j,t}$ $+ \alpha\nabla f_i(\mathbf{x}_{i,t})$
6: $\quad$ Compute NN-0 descent direction $\mathbf{d}_{i,t}^{(0)} = -\mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}$
7: $\quad$ **for** $k = 0, \ldots, K-1$ **do**
8: $\quad\quad$ Exchange elements $\mathbf{d}_{i,t}^{(k)}$ of the NN-k step with neighbors
9: $\quad\quad$ NN-$(k+1)$ step: $\mathbf{d}_{i,t}^{(k+1)}$ $= \mathbf{D}_{ii,t}^{-1}[\sum_{j\in\mathcal{N}_i, j=i} \mathbf{B}_{ij}\mathbf{d}_{j,t}^{(k)} - \mathbf{g}_{i,t}]$
10: $\quad$ **end for**
11: $\quad$ Update local iterate: $\mathbf{x}_{i,t+1} = \mathbf{x}_{i,t} + \epsilon\, \mathbf{d}_{i,t}^{(K)}$.
12: **end for**

K times, nodes ends up having determined their NN-K step component $\mathbf{d}_{i,t}^{(K)}$.

The resulting NN-K method is summarized in Algorithm 1. The descent iteration in (17) is implemented in Step 11. Implementation of this descent requires access to the NN-K descent direction $\mathbf{d}_{i,t}^{(K)}$ which is computed by the loop in steps 6-10. Step 6 initializes the loop by computing the NN-0 step $\mathbf{d}_{i,t}^{(0)} = -\mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}$. The core of the loop is in Step 9 which corresponds to the recursion in (19). Step 8 stands for the variable exchange that is required to implement Step 9. After K iterations through this loop, the NN-K descent direction $\mathbf{d}_{i,t}^{(K)}$ is computed and can be used in Step 11. Both, Steps 6 and 9, require access to the local gradient component $\mathbf{g}_{i,t}$. This is evaluated in Step 5 after receiving the prerequisite information from neighbors in Step 4. Steps 1 and 3 compute the blocks $\mathbf{B}_{ii,t}$, $\mathbf{B}_{ij,t}$, and $\mathbf{D}_{ii,t}$ required in steps 6 and 9.

Remark 2: By trying to approximate the Newton step, NN-K ends up reducing the number of iterations required for convergence. Furthermore, the larger K is, the closer that the NN-K step gets to the Newton step, and the faster NN-K converges. We will justify these assertions both, analytically in Section IV, and numerically in Section V. It is important to observe, however, that reducing the number of iterations reduces the computational cost but not necessarily the communication cost. In DGD, each node i shares its vector $\mathbf{x}_{i,t} \in \mathbb{R}^p$ with each of its neighbors $j \in \mathcal{N}_i$. In NN-K, node i exchanges not only the vector $\mathbf{x}_{i,t} \in \mathbb{R}^p$ with its neighboring nodes, but it also communicates iteratively the local components of the descent directions $\{\mathbf{d}_{i,t}^{(k)}\}_{k=0}^{K-1} \in \mathbb{R}^p$ so as to compute the descent direction $\mathbf{d}_{i,t}^{(K)}$. Hence, at each iteration, node i sends $|\mathcal{N}_i|$ vectors of size p to its neighbors in DGD, while in NN-K it sends $(K+1)|\mathcal{N}_i|$ vectors of the same size. Unless the original problem is well conditioned, NN-K also reduces total communication cost until convergence, even though the cost of each individual iteration is larger. However, the use of large K is unwarranted because the added benefit of better approximating the Newton step does not compensate the increase in communication cost.

IV. Convergence Analysis

In this section we show that as time progresses the sequence of objective function values $F(\mathbf{y}_t)$ [cf. (6)] approaches the optimal objective function value $F(\mathbf{y}^*)$. In proving this claim we make the following assumptions.

Assumption 1: There exist constants $0 \leq \delta \leq \Delta < 1$ that lower and upper bound the diagonal weights for all i,

$$0 < \delta \leq w_{ii} \leq \Delta < 1, \qquad i = 1, \ldots, n. \tag{20}$$

Assumption 2: The local objective functions $f_i(\mathbf{x})$ are twice differentiable and the eigenvalues of the local Hessians are bounded with positive constants $0 < m \leq M < \infty$, i.e.

$$m\mathbf{I} \preceq \nabla^2 f_i(\mathbf{x}) \preceq M\mathbf{I}. \tag{21}$$

Assumption 3: The local objective function Hessians $\nabla^2 f_i(\mathbf{x})$ are Lipschitz continuous with respect to the Euclidian norm with parameter L. I.e., for all $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$, it holds

$$\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\hat{\mathbf{x}})\| \;\leq\; L\,\|\mathbf{x} - \hat{\mathbf{x}}\|. \tag{22}$$

The lower bound in Assumption 1 is more a definition than a constraint. To be more precise, the weights w_{ij} are positive if and only if $j \in \mathcal{N}_i$ or $j = i$. This observation verifies existence of a lower bound for the local weights w_{ii} that is defined as $\delta > 0$ in Assumption 1. The upper bound $\Delta < 1$ on the weights w_{ii} is true for all connected networks as long as neighbors $j \in \mathcal{N}_i$ are assigned nonzero weights $w_{ij} > 0$. This is because the matrix $\mathbf{W}$ is doubly stochastic [cf. (4)], which implies that $w_{ii} = 1 - \sum_{j\in\mathcal{N}_i} w_{ij} < 1$ as long as $w_{ij} > 0$.

The lower bound m for the eigenvalues of local objective function Hessians $\nabla^2 f_i(\mathbf{x})$ is equivalent to the strong convexity of local objective functions $f_i(\mathbf{x})$ with parameter m. The strong convexity assumption for the local objective functions $f_i(\mathbf{x})$ stated in Assumption 2 is customary in Newton-based methods, since the Hessian of objective function should be invertible to establish Newton's method [Chapter 9 of [23]]. The upper bound M for the eigenvalues of local objective function Hessians $\nabla^2 f_i(\mathbf{x})$ is similar to the condition that gradients $\nabla f_i(\mathbf{x})$ are Lipschitz continuous with parameter M for the case that functions are twice differentiable.

The restriction imposed by Assumption 3 is customary in the analysis of second order methods, see Section 9.5.3 of [23], which guarantees that the Hessians $\nabla^2 F(\mathbf{y})$ are also Lipschitz continuous as we show in the following lemma.

Lemma 1: Consider the definition of objective function $F(\mathbf{y})$ in (6). If Assumption 3 holds then the objective function Hessian $\mathbf{H}(\mathbf{y}) =: \nabla^2 F(\mathbf{y})$ is Lipschitz continuous with parameter αL, i.e., for all $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^{np}$ we have

$$\|\mathbf{H}(\mathbf{y}) - \mathbf{H}(\hat{\mathbf{y}})\| \leq \alpha L\|\mathbf{y} - \hat{\mathbf{y}}\|. \tag{23}$$

Proof: See Appendix A. ■

Lemma 1 states that the penalty objective function introduced in (6) has the property that the Hessians are Lipschitz continuous, while the Lipschitz constant is a function of the penalty coefficient $1/\alpha$. Thus, if we increase the penalty coefficient $1/\alpha$, or, equivalently, decrease α, the objective function $F(\mathbf{y})$

approaches a quadratic form because the curvature becomes constant.

To prove convergence properties of NN we need bounds for the eigenvalues of the block diagonal matrix $\mathbf{D}_t$, the block sparse matrix $\mathbf{B}$, and the Hessian $\mathbf{H}_t$. These eigenvalue bounds are established in the following proposition using the conditions imposed by Assumptions 1 and 2.

Proposition 1: Consider the definitions of matrices $\mathbf{H}_t$, $\mathbf{D}_t$, and $\mathbf{B}$ in (10), (12), and (13), respectively. If Assumptions 1 and 2 hold true, then the eigenvalues of matrices $\mathbf{H}_t$, $\mathbf{D}_t$, and $\mathbf{B}$ are uniformly bounded as

$$\alpha m\mathbf{I} \preceq \mathbf{H}_t \preceq (2(1-\delta)+\alpha M)\mathbf{I}, \quad (24)$$

$$(2(1-\Delta)+\alpha m)\mathbf{I} \preceq \mathbf{D}_t \preceq (2(1-\delta)+\alpha M)\mathbf{I}, \quad (25)$$

$$\mathbf{0} \preceq \mathbf{B} \preceq 2(1-\delta)\mathbf{I}. \quad (26)$$

Proof: See Appendix B. ■

Proposition 1 states that Hessian matrix $\mathbf{H}_t$ and block diagonal matrix $\mathbf{D}_t$ are positive definite, while matrix $\mathbf{B}$ is positive semidefinite.

As we noted in Section III, for the expansion in (14) to be valid the eigenvalues of the matrix $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ must be nonnegative and strictly smaller than 1. The following proposition states that this is true for all times t.

Proposition 2: Consider the definitions of the matrices $\mathbf{D}_t$ in (12) and $\mathbf{B}$ in (13). If Assumptions 1 and 2 hold true, the matrix $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ is positive semidefinite and its eigenvalues are bounded above by a constant $\rho < 1$

$$\mathbf{0} \preceq \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2} \preceq \rho\mathbf{I}, \quad (27)$$

where $\rho := 2(1-\delta)/(2(1-\delta)+\alpha m)$.

Proof: See Appendix C. ■

The results in Proposition 1 would lead to the trivial upper bound $2(1-\delta)/(\alpha M + 2(1-\Delta))$ for the eigenvalues of $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$. The upper bound in Proposition 2 is tighter and follows from the structure of the matrix $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$.

The bounds for the eigenvalues of $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ in (27) guarantee convergence of the Taylor series in (14). As mentioned in Section III, NN-K truncates the first K summands of the Hessian inverse Taylor series in (14) to approximate the Hessian inverse of the objective function in optimization problem (6). To evaluate the performance of NN-K we study the error of the Hessian inverse approximation by defining the *error matrix* $\mathbf{E}_t \in \mathbb{R}^{np\times np}$ as

$$\mathbf{E}_t := \mathbf{I} - \hat{\mathbf{H}}_t^{(K)^{-1/2}}\mathbf{H}_t\hat{\mathbf{H}}_t^{(K)^{-1/2}}. \quad (28)$$

The error matrix $\mathbf{E}_t$ measures closeness of the Hessian inverse approximation matrix $\hat{\mathbf{H}}_t^{(K)^{-1}}$ and the exact Hessian inverse $\mathbf{H}_t^{-1}$ at time t. Based on the definition of the error matrix $\mathbf{E}_t$, if the Hessian inverse approximation $\hat{\mathbf{H}}_t^{(K)^{-1}}$ approaches the exact Hessian inverse $\mathbf{H}_t^{-1}$ the error matrix $\mathbf{E}_t$ approaches the zero matrix $\mathbf{0}$. We therefore bound the error of the Hessian inverse approximation by developing a bound for the eigenvalues of $\mathbf{E}_t$. This bound is provided in the following proposition.

Proposition 3: Consider the NN-K method in (12)-(17) and the definition of error matrix $\mathbf{E}_t$ in (28). Further, recall the definition of the constant $\rho := 2(1-\delta)/(\alpha m + 2(1-\delta)) < 1$ in Proposition 2. The error matrix $\mathbf{E}_t$ is positive semidefinite and all its eigenvalues are upper bounded by ρ^{K+1},

$$\mathbf{0} \preceq \mathbf{E}_t \preceq \rho^{K+1}\mathbf{I}. \quad (29)$$

Proof: See Appendix D. ■

Proposition 3 asserts that the error in the approximation of the Hessian inverse, thereby on the approximation of the Newton step, is bounded by ρ^{K+1}. This result corroborates the intuition that the larger K is, the closer that $\mathbf{d}_{i,t}^{(K)}$ approximates the Newton step. This closer approximation comes at the cost of increasing the communication cost of each descent iteration. The decrease of this error being proportional to ρ^{K+1} hints that using a small value of K should suffice in practice. Further to decrease ρ we can increase δ or increase α. Increasing δ calls for assigning substantial weight to w_{ii}. Increasing α comes at the cost of moving the solution of (6) away from the solution of (9) and its equivalent (1).

Bounds on the eigenvalues of the objective function Hessian $\mathbf{H}_t$ are central to the convergence analysis of Newton's method [Chapter 9 of [23]]. Lower bounds for the Hessian eigenvalues guarantee that the matrix is nonsingular. Upper bounds imply that the minimum eigenvalue of the Hessian inverse $\mathbf{H}^{-1}$ is strictly larger than zero, which, in turn, implies a strict decrement in each Newton step. Analogous bounds for the eigenvalues of the NN approximate Hessian inverses $\hat{\mathbf{H}}_t^{(K)^{-1}}$ are required. These bounds are studied in the following lemma.

Lemma 2: Consider the NN-K method as defined in (12)-(17). If Assumptions 1 and 2 hold true, we have

$$\lambda\mathbf{I} \preceq \hat{\mathbf{H}}_t^{(K)^{-1}} \preceq \Lambda\mathbf{I}, \quad (30)$$

where constants λ and Λ are defined as

$$\lambda := \frac{1}{2(1-\delta)+\alpha M} \text{ and } \Lambda := \frac{1-\rho^{K+1}}{(1-\rho)(2(1-\Delta)+\alpha m)}. \quad (31)$$

Proof: See Appendix E. ■

According to the result in Lemma 2, the NN-K approximate Hessian inverses $\hat{\mathbf{H}}_t^{(K)^{-1}}$ are strictly positive definite and have all of their eigenvalues bounded between the positive and finite constants λ and Λ. This is true for all K and uniform across all iteration indexes t. Considering these eigenvalue bounds and the fact that $-\mathbf{g}_t$ is a descent direction, the approximate Newton step $-\hat{\mathbf{H}}_t^{(K)^{-1}}\mathbf{g}_t$ enforces convergence of the iterate $\mathbf{y}_t$ to the optimal argument $\mathbf{y}^*$ of the penalized objective function $F(\mathbf{y})$ in (6). In the following theorem we show that if the stepsize ϵ is properly chosen, the sequence of objective function values $F(\mathbf{y}_t)$ converges at least linearly to the optimal objective function value $F(\mathbf{y}^*)$.

Theorem 1: Consider the NN-K method as defined in (12)-(17) and the objective function $F(\mathbf{y})$ as introduced in (6). Further, recall the definitions of the lower and upper bounds λ and Λ, respectively, for the eigenvalues of the approximate Hessian inverse $\hat{\mathbf{H}}_t^{(K)^{-1}}$ in (31). If the stepsize ϵ is chosen as

$$\epsilon \leq \min\left\{1\,,\left[\frac{3m\lambda^{\frac{5}{2}}}{L\Lambda^3(F(\mathbf{y}_0)-F(\mathbf{y}^*))^{\frac{1}{2}}}\right]^{\frac{1}{2}}\right\}, \quad (32)$$

and Assumptions 1-3 hold, the sequence $F(\mathbf{y}_t)$ converges to the optimal argument $F(\mathbf{y}^*)$ at least linearly as

$$F(\mathbf{y}_t) - F(\mathbf{y}^*) \leq (1-\zeta)^t (F(\mathbf{y}_0) - F(\mathbf{y}^*)), \quad (33)$$

where the constant $0 < \zeta < 1$ is explicitly given by

$$\zeta := (2-\epsilon)\epsilon\alpha m\lambda - \frac{\alpha\epsilon^3 L\Lambda^3 (F(\mathbf{y}_0) - F(\mathbf{y}^*))^{\frac{1}{2}}}{6\lambda^{\frac{3}{2}}}. \quad (34)$$

Proof: See Appendix F. ■

Theorem 1 shows that the objective function error sequence $F(\mathbf{y}_t) - F(\mathbf{y}^*)$ asymptoticly converges to zero and that the rate of convergence is at least linear. Note that according to the definition of the convergence parameter ζ in Theorem 1 and the definitions of λ and Λ in (31), increasing α leads to faster convergence. This observation verifies existence of a tradeoff between rate and accuracy of convergence. For large values of α the sequence generated by network Newton converges faster to the optimal solution of (6). These faster convergence comes at the cost of increasing the distance between the optimal solutions of (6) and (1). Conversely, smaller α implies smaller gap between the optimal solutions of (6) and (1), but the convergence rate of NN-K is slower. In the following section, we illustrate the connection between network Newton and the centralized Newton's method.

A. Analysis of Network Newton as a Newton-like Method

To connect the proposed NN method with the classic Newton's method, we first study the difference between these methods. In particular, the following lemma shows that the convergence of the norm of the weighted gradient $\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$ in NN-K is akin to the convergence of Newton's method with constant stepsize. The difference is the appearance of a term associated with the error of the Hessian inverse approximation as we formally state next.

Lemma 3: Consider the NN-K method as defined in (12)-(17). If Assumptions 1-3 hold, the sequence of weighted gradients $\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}$ satisfies

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_{t+1}\| \leq \left(1-\epsilon+\epsilon\rho^{K+1}\right)\left[1+\Gamma_1(1-\zeta)^{\frac{(t-1)}{4}}\right] \times \|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \epsilon^2\Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2, \quad (35)$$

where the constants Γ_1 and Γ_2 are defined as

$$\Gamma_1 := \frac{(\alpha\epsilon L\Lambda)^{\frac{1}{2}}(F(\mathbf{y}_0) - F(\mathbf{y}^*))^{\frac{1}{4}}}{\lambda^{\frac{3}{4}}(2(1-\Delta)+\alpha m)},$$

$$\Gamma_2 := \frac{\alpha L\Lambda^2}{2\lambda(2(1-\Delta)+\alpha m)^{\frac{1}{2}}}. \quad (36)$$

Proof: See Appendix G. ■

As per Lemma 3 the weighted gradient norm $\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\|$ is upper bounded by terms that are linear and quadratic on the weighted norm $\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$ associated with the previous iterate. This is akin to the gradient norm decrease of Newton's method with constant stepsize. Note that if the error of Hessian inverse approximation which is characterized by ρ^{K+1} becomes zero, by setting $\epsilon = 1$ we can simplify (35) as $\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq \Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2$. This result shows quadratic convergence when $\Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| < 1$. However, the term ρ^{K+1} is not zero in general. Although, the error of Hessian inverse approximation is not zero, the result in (35) is very similar to the one for the classic Newton's method. To make this connection clearer, further note that for all except the first few iterations the term $\Gamma_1(1-\zeta)^{(t-1)/4} \approx 0$ is close to 0 and the relation in (35) can be simplified to

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_{t+1}\| \lesssim (1-\epsilon+\epsilon\rho^{K+1})\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \epsilon^2\Gamma_2\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\|^2. \quad (37)$$

In (37), the coefficient in the linear term is reduced to $(1-\epsilon+\epsilon\rho^{K+1})$ and the coefficient in the quadratic term stays at $\epsilon^2\Gamma_2$. If, for discussion purposes, we set $\epsilon = 1$ as in Newton's quadratic phase, the upper bound in (37) is further reduced to

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \lesssim \rho^{K+1}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| + \Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2. \quad (38)$$

The equation in (38) makes the connection between NN and Newton's clear, because the exact same result would hold for Newton's method if we set $\rho = 0$. The NN method can not have a quadratic convergence phase for the rest of the iterations – like the one for Newton's method – because of the term $\rho^{K+1}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$. However, since the constant ρ (cf. Proposition 2) is smaller than 1 the term ρ^{K+1} can be made arbitrarily small by increasing the approximation order K. Equivalently, this means that by selecting K to be large enough, we can make the quadratic term in (38) dominant and observe a quadratic convergence phase. The boundaries of this quadratic convergence phase are formally determined in the following Theorem using the result in (35).

Theorem 2: Consider the NN-K method as defined in (12)-(17). Define the sequence $\eta_t := [(1-\epsilon+\epsilon\rho^{K+1})(1+\Gamma_1(1-\zeta)^{(t-1)/4})]$ and the time t_0 as the first time at which sequence η_t is smaller than 1, i.e. $t_0 := \arg\min_t\{t \mid \eta_t < 1\}$. If Assumptions 1-3 hold, then for all $t \geq t_0$ when the sequence $\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$ satisfies

$$\frac{\sqrt{\eta_t}(1-\sqrt{\eta_t})}{\epsilon^2\Gamma_2} \leq \|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| < \frac{1-\sqrt{\eta_t}}{\epsilon^2\Gamma_2}, \quad (39)$$

the sequence of scaled gradient norms is such that

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq \frac{\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2. \quad (40)$$

Proof: Based on the definition of η_t, we can rewrite (35) as

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq \eta_t\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| + \epsilon^2\Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2. \quad (41)$$

We use this expression to prove the inequality in (40). To do so, rearrange terms in the first inequality in (39) and write

$$\sqrt{\eta_t} \leq \frac{\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|. \quad (42)$$

Multiplying both sides of (42) by $\sqrt{\eta_t}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$ yields

$$\eta_t\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| \leq \frac{\sqrt{\eta_t}\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2. \quad (43)$$

Substituting $\eta_t\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$ in (41) by its upper bound in (43) implies that

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq \frac{\sqrt{\eta_t}\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2 + \epsilon^2\Gamma_2\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2$$
$$= \frac{\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2. \tag{44}$$

To verify quadratic convergence, it is necessary to prove that the sequence $\|\mathbf{D}_{i-1}^{-1/2}\mathbf{g}_i\|$ of weighted gradient norms is decreasing. For this to be true we must have

$$\frac{\epsilon^2\Gamma_2}{1-\sqrt{\eta_t}}\left\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\right\| < 1. \tag{45}$$

But (45) is true because we are looking at a range of gradients that satisfy the second inequality in (39). ■

As per Theorem 1 $\mathbf{y}_t$ is converging to $\mathbf{y}^*$ at a rate that is at least linear. Thus, the gradients $\mathbf{g}_t$ will be such that at some point in time they satisfy the rightmost inequality in (39). At that point in time, progress towards $\mathbf{y}^*$ proceeds at a quadratic rate as indicated by (40). This quadratic rate of progress is maintained until the leftmost inequality in (39) is satisfied, at which point the linear term in (35) dominates and the convergence rate goes back to linear. Furthermore, making K sufficiently large it is possible to reduce η_t arbitrarily and make the quadratic convergence region last longer. In practice, this calls for making K large enough so that $\sqrt{\eta_t}$ is close to the desired gradient norm accuracy.

Remark 3: For a quadratic function F, the Lipschitz constant for the Hessian is $L = 0$. Then, the optimal choice of stepsize for NN-K is $\epsilon = 1$ as a result of stepsize rule in (32). Moreover, the constants for the linear and quadratic terms in (35) are $\Gamma_1 = \Gamma_2 = 0$ as it follows from their definitions in (36). For quadratic functions we also have that the Hessian of the objective function $\mathbf{H}_t = \mathbf{H}$ and the block diagonal matrix $\mathbf{D}_t = \mathbf{D}$ are time-invariant. Thus, we can rewrite (35) as

$$\|\mathbf{D}^{-1/2}\mathbf{g}_{t+1}\| \leq \rho^{K+1}\|\mathbf{D}^{-1/2}\mathbf{g}_t\|. \tag{46}$$

Note that Newton's method converges in a single step in quadratic programming. This property follows from (46) because Newton's method is equivalent to NN-K as $K \to \infty$. The expression in (46) states that NN-K converges linearly with a constant decrease factor of ρ^{K+1} per iteration. This in contrast with first order methods like DGD that converge with a linear rate that depends on the problem condition number.

V. Numerical analysis

In this section, we study the performance of NN-K in the minimization of a distributed quadratic objective. For each agent i we consider a positive definite diagonal matrix $\mathbf{A}_i \in \mathbb{S}_p^{++}$ and a vector $\mathbf{b}_i \in \mathbb{R}^p$ to define the local objective function $f_i(\mathbf{x}) := (1/2)\mathbf{x}^T\mathbf{A}_i\mathbf{x} + \mathbf{b}_i^T\mathbf{x}$. Therefore, the global cost function $f(\mathbf{x})$ is written as

$$f(\mathbf{x}) := \sum_{i=1}^{n}\frac{1}{2}\mathbf{x}^T\mathbf{A}_i\mathbf{x} + \mathbf{b}_i^T\mathbf{x}\,. \tag{47}$$

The difficulty of solving (47) is given by the condition number of the matrices $\mathbf{A}_i$. To tune condition numbers we generate diagonal matrices $\mathbf{A}_i$ with random diagonal elements a_{ii}. The first $p/2$ diagonal elements a_{ii} are drawn uniformly at random from the discrete set $\{1, 10^{-1}, \ldots, 10^{-\xi}\}$ and the next $p/2$ are uniformly and randomly chosen from the set $\{1, 10^{1}, \ldots, 10^{\xi}\}$. This choice of coefficients yields local matrices $\mathbf{A}_i$ with eigenvalues in the interval $[10^{-\xi}, 10^{\xi}]$ and global matrices $\sum_{i=1}^{n}\mathbf{A}_i$ with eigenvalues in the interval $[n10^{-\xi}, n10^{\xi}]$. The linear terms $\mathbf{b}_i^T\mathbf{x}$ are added so that the different local functions have different minima. The vectors $\mathbf{b}_i$ are chosen uniformly at random from the box $[0, 1]^p$.

The graph is graph is d-regular and generated by creating a cycle and then connecting each node with the $d/2$ nodes that are closest in each direction. The diagonal weights in the matrix $\mathbf{W}$ are set to $w_{ii} = 1/2 + 1/2(d+1)$ and the off diagonal weights to $w_{ij} = 1/2(d+1)$ when $j \in \mathcal{N}_i$.

A. Comparison with Existing Methods

In this section we compare the performance of the proposed NN method with primal methods such as DGD in [13] and the accelerated version of DGD (Acc. DGD) in [14]. For the Acc. DGD method, we assume that the stepsize parameter and the momentum coefficients are constant as in the case for the centralized accelerated gradient descent. This makes the comparison between Acc. DGD, DGD, and NN fair, since our aim is to compare their performances in solving the penalized objective function. Moreover, we consider the convergence paths of the distributed ADMM (DADMM) in [18] and the exact first order method EXTRA in [16]. Although EXTRA operates in the primal domain, it has been shown that it can be interpreted as a saddle-point method [27]. Thus, we consider EXTRA in the category of dual methods which has a linear convergence rate as DADMM.

We compare these methods in solving (47) for the case that there are $n = 100$ nodes in the network and the dimension of the vector $\mathbf{x}$ is $p = 20$. We assume that the graph is 4-regular. Further, we set the condition number parameter to $\xi = 2$ and the penalty parameter to $\alpha = 10^{-3}$. The momentum coefficient for the accelerated DGD is 0.9. Note that among the values $\{0.1, 0.2, \ldots, 0.9, 1\}$, the best performance belongs to the momentum coefficient 0.9 which we use in the experiments.

As the condition number of the problem is relatively large, i.e., 4.3×10^3, the NN method performs better than DGD and Acc. DGD in terms of the number of iterations and total number of local information exchanges as they are illustrated in Fig. 1 and Fig. 2, respectively. In the case that the condition number of the objective function is not significantly large with respect to the dimension of the problem, the accelerated DGD would be a better choice relative to NN.

The comparison with dual methods shows that in terms of iterations and rounds of communications DADMM and different variants of NN perform relatively well and after some point DADMM outperform NN and other primal methods because it converges to the optimal argument of the original problem instead of the penalized function. On the other hand, each step of DADMM requires solving a convex program which can be computationally costly. We observe that EXTRA also has a linear convergence rate to the exact optimal solution, and its accuracy becomes better than all primal methods. However, EXTRA is a first-order method and its convergence at the beginning is

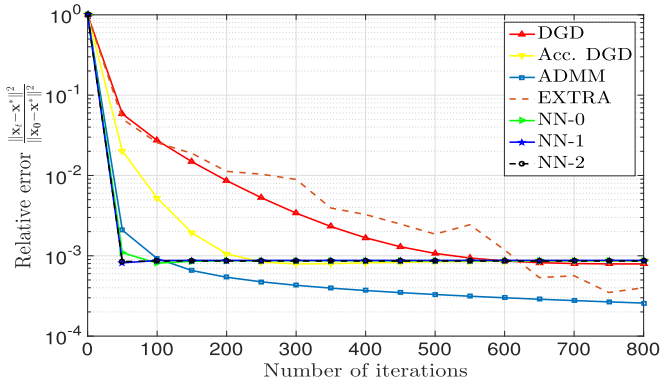

Fig. 1. Comparison of DGD, Acc. DGD, DADMM, EXTRA, NN-0, NN-1, and NN-2 in terms of number of iterations.

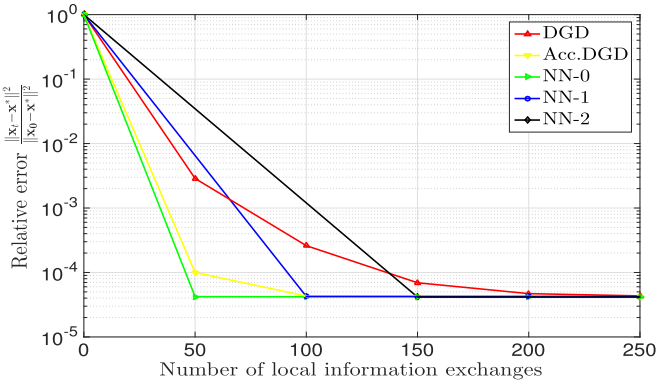

Fig. 3. Relative error of DGD, Acc. DGD, NN-0, NN-1, and NN-2 vs number of local info. exchanges for a well-conditioned problem.

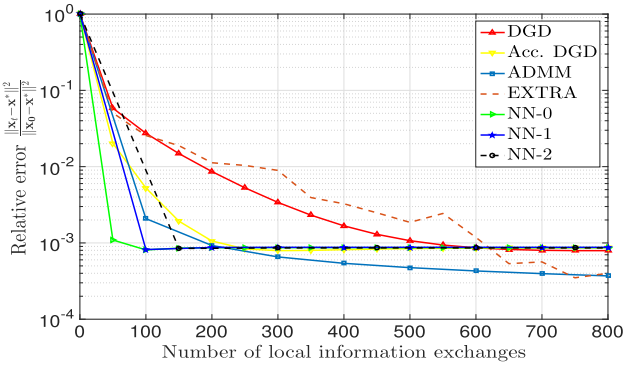

Fig. 2. Comparison of DGD, Acc. DGD, DADMM, EXTRA, NN-0, NN-1, and NN-2 in terms of rounds of local information exchanges.

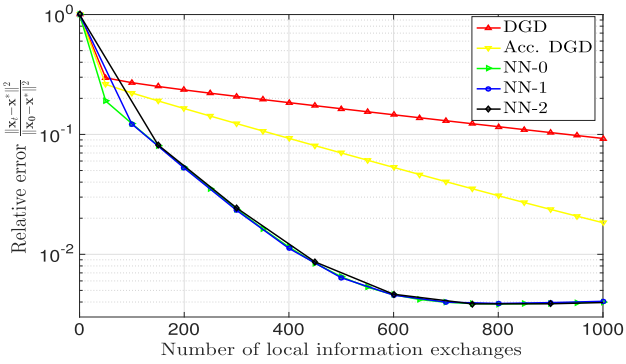

Fig. 4. Relative error of DGD, Acc. DGD, NN-0, NN-1, and NN-2 vs number of local info. exchanges for an ill-conditioned problem.

relatively slower than NN. This advantage of NN results from incorporation of the curvature information of the objective function. These observations show that by incorporating the idea of NN and EXTRA we should be able to come up with a second-order method that has a linear convergence rate to the exact solution of (47) while it can perform well in ill-conditioned problems.

B. *Effect of Objective Function Condition Number*

We study the effect of condition number on the convergence rate of NN and show that NN is less sensitive to the objective function condition number with respect to primal first-order methods, e.g., DGD in [13] and accelerated DGD in [14]. To do so, we compare the performances of the mentioned methods in solving the problem in (47) for small and large condition numbers. The parameters are the same as the parameters in Fig. 1 except the choice of the condition number parameter ξ.

We first consider the case that $\xi = 1$ which leads to condition number 1.24×10^1. The convergence paths of DGD, accelerated DGD, NN-0, NN-1, and NN-2 in terms of the number of local information exchanges are shown in Fig. 3. The performance of variations of NN are not significantly better than DGD and accelerated DGD. In particular, DGD and Acc. DGD both outperform NN-1 and NN-2 in terms of the total communications until convergence. Thus, accelerated DGD is the best option among the primal methods for problems with small condition number.

To explore the performance of these methods for an ill-conditioned problem we set the condition number parameter $\xi = 3$ which leads to the condition number 1.4×10^4 for the considered realization. Fig. 4 illustrate the convergence paths of the considered primal methods in terms of the number of local information exchanges. As we observe, the advantage of the network Newton methods is substantial in this setting and they outperform DGD and accelerated DGD in terms of communication cost.

C. *Effect of Network Topology*

We proceed to compare the performance of NN in different network topologies. In particular, we consider five different topologies which are random graphs with connectivity probabilities $p_c = 0.25$ and $p_c = 0.35$, complete graph, cycle, and line. Note that in random graphs, we generate the edges between nodes with probability p_c. The complete graph is a graph that all nodes are connected to each other directly. A cycle graph is a connected graph that each node has degree 2. A line graph is a cycle graph that is missing an edge. The parameters are the same as the parameters in Fig. 1 except the network graph and the way that we generate the weight matrix $\mathbf{W}$. We generate the weight matrix $\mathbf{W}$ using the formula $\mathbf{W} = \mathbf{I} - \mathbf{L}/\tau$ where $\mathbf{L}$ is the Laplacian matrix of the graph and $\tau/2$ is the largest eigenvalue of the Laplacian $\mathbf{L}$. We compare the performance of NN-2 for all these networks in terms of the number of iterations and the total number of communications between nodes. Notice

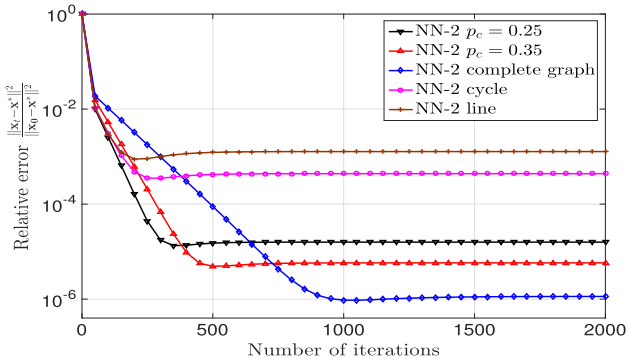

Fig. 5. Relative error of NN-2 vs num. of iterations for random graphs with $p_c = \{0.25, 0.35\}$, complete graph, cycle graph, and line graph.

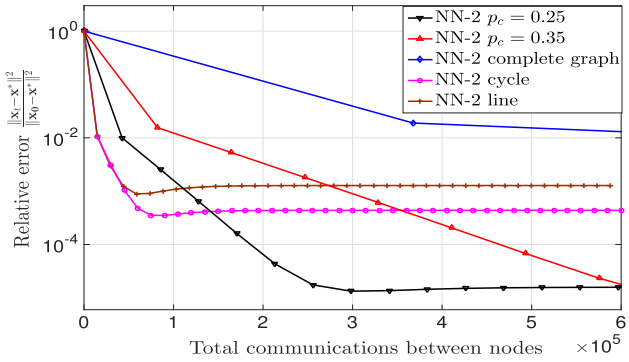

Fig. 6. Relative error of NN-2 vs num. of communications for random graphs with $p_c = \{0.25, 0.35\}$, complete graph, cycle graph, and line graph.

that in this section we use total communications between node instead of the number of local information exchanges (rounds of local communications) since the degrees of nodes in the different networks are not equal.

The convergence paths of NN-2 for the considered topologies in terms of the number of iterations and the total number of communications are demonstrated in Fig. 5 and Fig. 6, respectively. The first important observation is the accuracy of convergence. According to the results in [15], if we define $\beta < 1$ as the second largest magnitude of the eigenvalues of $\mathbf{W}$, then the accuracy of convergence is proportional to $1/(1-\beta)$. Thus, the graphs with smaller β converge to a smaller neighborhood of the optimal argument. In particular, the parameter β for the complete graph which has the most accurate convergence is $\beta = 0.5$, while for the line graph that has the least accurate convergence path $\beta = 0.99$.

The second important observation is the rate of convergence for NN-2 in these network topologies. It follows from the result in Theorem 1 that for a quadratic objective function the constant of linear convergence becomes $1 - \alpha m \lambda$. Therefore, for larger values of λ we expect faster convergence. Note that λ is large when $\delta = \min_i w_{ii}$ is large and close to 1. These observations imply that for the graphs that δ is larger we expect faster linear convergence. The convergence paths in Fig. 5 reinforce this claim. Note that δ for the considered graphs are $\delta_{p_c=0.25} = 0.5898, \delta_{p_c=0.35} = 0.5585, \delta_{com} = 0.51$, $\delta_{cycle} = 0.75$, $\delta_{line} = 0.7498$. These numbers justify the similarity of the convergence paths of line and cycle graphs and the slow convergence rate of the complete graph.

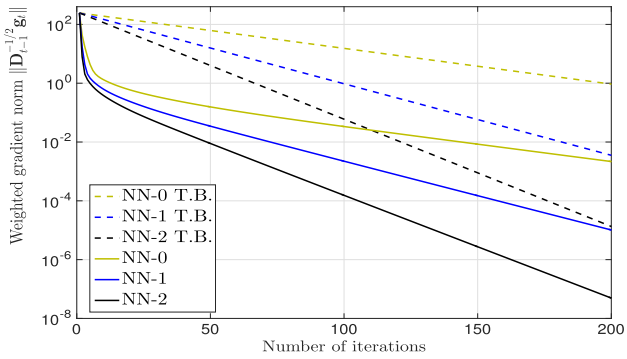

Fig. 7. Comparison of the theoretical bound (T.B.) in (46) with the empirical result for a quadratic programming.

D. Tightness of the Bounds

In this section, we study the tightness of the theoretical bounds in the paper. To do so, we compare the empirical convergence rates of NN-0, NN-1, and NN-2 with the theoretical result in Lemma 3. As we discussed in Remark 3, for a quadratic objective function the sequence of weighted gradients of NN-K satisfies the inequality $\|\mathbf{D}^{-1/2}\mathbf{g}_{t+1}\| \leq \rho^{K+1}\|\mathbf{D}^{-1/2}\mathbf{g}_t\|$. We refer to this rate as T.B. which stands for theoretical bound. Figure 7 illustrates the theoretical bounds and empirical convergence paths of NN-0, NN-1, and NN-2 for the quadratic problem in (47). As we observe, the convergence rates of all methods are faster than their theoretical bounds at the beginning, but after almost 10 iterations their convergence rate becomes similar to the theoretical bound in (46). To be clearer, the slopes of the actual convergence paths and their corresponding theoretical bounds become equal after almost 10 iterations. This observation shows that the bound in (46) is reasonably tight and the sequence of weighted gradients for NN-K diminishes with factor ρ^{K+1}.

VI. Conclusion

We developed the network Newton method as an approximate Newton method for solving consensus optimization problems. The algorithm builds on a reinterpretation of distributed gradient descent as a penalty method and relies on an approximation of the Newton step of the corresponding penalized objective function. To approximate the Newton direction we truncate the Taylor series of the exact Newton step. This leads to a family of methods defined by the number K of Taylor series terms kept in the approximation. When we keep K terms of the Taylor series, the method is called NN-K and can be implemented through the aggregation of information in K-hop neighborhoods. We showed that NN converges at least linearly to the solution of the penalized objective, and, consequently, to a neighborhood of the optimal argument for the original optimization problem. We completed the convergence analysis of NN-K by showing that the sequence of iterates generated by NN-K converges at a quadratic rate in a specific interval. Numerical analyses compared the performances of NN-K with different choices of K for minimizing quadratic objectives. We observed that all NN-K methods work faster than distributed gradient descent in terms of number of iterations and number of communications.

APPENDIX A
PROOF OF LEMMA 1

Consider two vectors $\mathbf{y} := [\mathbf{x}_1; \ldots; \mathbf{x}_n] \in \mathbb{R}^{np}$ and $\hat{\mathbf{y}} := [\hat{\mathbf{x}}_1; \ldots; \hat{\mathbf{x}}_n] \in \mathbb{R}^{np}$. Based on the Hessian expression in (10), we simplify the Euclidean norm $\|\mathbf{H}(\mathbf{y}) - \mathbf{H}(\hat{\mathbf{y}})\|$ as

$$\begin{aligned}\|\mathbf{H}(\mathbf{y}) - \mathbf{H}(\hat{\mathbf{y}})\| &= \alpha \|\mathbf{G}(\mathbf{y}) - \mathbf{G}(\hat{\mathbf{y}})\| \\ &= \alpha \max_{i=1,\ldots,n} \left\|\nabla^2 f_i(\mathbf{x}_i) - \nabla^2 f_i.(\hat{\mathbf{x}}_i)\right\|. \end{aligned} \tag{48}$$

By using 3 and (48) we obtain that

$$\|\mathbf{H}(\mathbf{y}) - \mathbf{H}(\hat{\mathbf{y}})\| \leq \alpha L \max_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\| \leq \alpha L \|\mathbf{y} - \hat{\mathbf{y}}\|. \tag{49}$$

Therefore, the claim in (23) follows.

APPENDIX B
PROOF OF PROPOSITION 1

The Gershgorin circle theorem states that each eigenvalue of a matrix $\mathbf{A}$ lies within at least one of the Gershgorin discs $D(a_{ii}, R_{ii})$ where the center a_{ii} is the ith diagonal element of A and the radius $R_{ii} := \sum_{j\neq i} |a_{ij}|$ is the sum of the absolute values of all the non-diagonal elements of the ith row. Hence, Gershgorin discs can be considered as intervals of width $[a_{ii} - R_{ii}, a_{ii} + R_{ii}]$ for $\mathbf{I} - \mathbf{W}$, where $a_{ii} = 1 - w_{ii}$ and $R_{ii} = \sum_{j\neq i} |w_{ij}| = \sum_{j\neq i} w_{ij}$. Therefore, all the eigenvalues of $\mathbf{I} - \mathbf{W}$ are in at least one of the intervals $[1 - w_{ii} - \sum_{j\neq i} w_{ij}, 1 - w_{ii} + \sum_{j\neq i} w_{ij}]$. Since $\sum_j w_{ij} = 1$, it can be derived that $1 - w_{ii} = \sum_{j\neq i}^{n} w_{ij}$. Thus, the Gershgorin intervals can be simplified as $[0, 2(1 - w_{ii})]$ for $i = 1, \ldots, n$. This observation in association with the fact that $2(1 - w_{ii}) \leq 2(1 - \delta)$ implies that the eigenvalues of $\mathbf{I} - \mathbf{W}$ are in the interval $[0, 2(1 - \delta)]$ and consequently the eigenvalues of $\mathbf{I} - \mathbf{Z}$ are bounded as

$$\mathbf{0} \preceq \mathbf{I} - \mathbf{Z} \preceq 2(1 - \delta)\mathbf{I}. \tag{50}$$

Since matrix $\mathbf{G}_t$ is block diagonal and the eigenvalues of each diagonal block $\mathbf{G}_{ii,t} = \nabla^2 f_i(\mathbf{x}_{i,t})$ are bounded by constants $0 < m \leq M < \infty$ as mentioned in (21), we obtain

$$m\mathbf{I} \preceq \mathbf{G}_t \preceq M\mathbf{I}. \tag{51}$$

Considering the definition of the Hessian $\mathbf{H}_t := \mathbf{I} - \mathbf{Z} + \alpha\mathbf{G}_t$ and the bounds in (50) and (51), the first claim follows.

The definition of the matrix $\mathbf{D}_t$ in (12) yields

$$\mathbf{D}_t = \alpha\mathbf{G}_t + (\mathbf{I}_n - \mathbf{W}_d) \otimes \mathbf{I}_p, \tag{52}$$

where $\mathbf{W}_d$ is defined as $\mathbf{W}_d := \text{diag}(\mathbf{W})$. Note that matrix $\mathbf{I}_n - \mathbf{W}_d$ is diagonal and the i-th diagonal component is $1 - w_{ii}$. Since the local weights satisfy $\delta \leq w_{ii} \leq \Delta$, we obtain that the eigenvalues of $\mathbf{I}_n - \mathbf{W}_d$ are bounded below and above by $1 - \Delta$ and $1 - \delta$, respectively. Since the eigenvalues of $(\mathbf{I}_n - \mathbf{W}_d)$ and $(\mathbf{I}_n - \mathbf{W}_d) \otimes \mathbf{I}_p$ are identical we obtain

$$(1 - \Delta)\mathbf{I}_{np} \preceq (\mathbf{I}_n - \mathbf{W}_d) \otimes \mathbf{I}_p \preceq (1 - \delta)\mathbf{I}_{np} \tag{53}$$

Considering the relation in (52) and bounds in (51) and (53), the second claim follows.

Based on the definition of $\mathbf{B}$ in (13), we can write

$$\mathbf{B} = (\mathbf{I} - 2\mathbf{W}_d + \mathbf{W}) \otimes \mathbf{I}. \tag{54}$$

Note that in the i-th row of matrix $\mathbf{I} - 2\mathbf{W}_d + \mathbf{W}$, the diagonal component is $1 - w_{ii}$ and the jth component is w_{ij} for all $j \neq i$. Using Gershgorin theorem and the same argument that we established for the eigenvalues of $\mathbf{I} - \mathbf{Z}$, we can write

$$\mathbf{0} \preceq \mathbf{I} - 2\mathbf{W}_d + \mathbf{W} \preceq 2(1 - \delta)\mathbf{I}. \tag{55}$$

Based on (55) and (54), the last claim follows.

APPENDIX C
PROOF OF PROPOSITION 2

According to the result of Proposition 1, $\mathbf{D}_t$ is positive definite and $\mathbf{B}$ is positive semidefinite which immediately implies that $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ is positive semidefinite.

Recall the definition of $\mathbf{D}_t$ in (12) and define the matrix $\hat{\mathbf{D}}$ as a special case of matrix $\mathbf{D}_t$ for $\alpha = 0$. I.e., $\hat{\mathbf{D}} := 2(\mathbf{I} - \mathbf{Z}_d)$. Notice that $\hat{\mathbf{D}}$ is diagonal, time invariant, and only depends on the structure of the network. Since $\hat{\mathbf{D}}$ is diagonal and each diagonal component $1 - w_{ii}$ is strictly larger than 0, $\hat{\mathbf{D}}$ is positive definite and invertible. Hence, we can write

$$\mathbf{D}_t^{-\frac{1}{2}}\mathbf{B}\mathbf{D}_t^{-\frac{1}{2}} = (\mathbf{D}_t^{-\frac{1}{2}}\hat{\mathbf{D}}^{\frac{1}{2}})(\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{B}\hat{\mathbf{D}}^{-\frac{1}{2}})(\hat{\mathbf{D}}^{\frac{1}{2}}\mathbf{D}_t^{-\frac{1}{2}}). \tag{56}$$

We proceed to find an upper bound for the eigenvalues of the matrix $\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}$ in (56). Observing the fact that matrices $\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}$ and $\mathbf{B}\hat{\mathbf{D}}^{-1}$ are *similar*, eigenvalues of these matrices are identical. Hence, we proceed to characterize an upper bound for the eigenvalues of matrix $\mathbf{B}\hat{\mathbf{D}}^{-1}$. Based on the definitions of $\mathbf{B}$ and $\hat{\mathbf{D}}$, the product $\mathbf{B}\hat{\mathbf{D}}^{-1}$ is given by $\mathbf{B}\hat{\mathbf{D}}^{-1} = (\mathbf{I} - 2\mathbf{Z}_d + \mathbf{Z})\,(2(\mathbf{I} - \mathbf{Z}_d))^{-1}$. Therefore, the blocks of the matrix $\mathbf{B}\hat{\mathbf{D}}^{-1}$ are given by

$$[\mathbf{B}\hat{\mathbf{D}}^{-1}]_{ii} = \frac{1}{2}\mathbf{I} \quad \text{and} \quad [\mathbf{B}\hat{\mathbf{D}}^{-1}]_{ij} = \frac{w_{ij}}{2(1 - w_{jj})}\mathbf{I}. \tag{57}$$

Thus, each diagonal component of the matrix $\mathbf{B}\hat{\mathbf{D}}^{-1}$ is $1/2$ and that the sum of non-diagonal components of column i is

$$\sum_{j=1,j\neq i}^{np} \mathbf{B}\hat{\mathbf{D}}^{-1}[ji] = \frac{1}{2}\sum_{j=1,j\neq i}^{np} \frac{w_{ji}}{1 - w_{ii}} = \frac{1}{2}. \tag{58}$$

Consider (58) and apply Gershgorin theorem to obtain

$$0 \leq \mu_i(\mathbf{B}\hat{\mathbf{D}}^{-1}) \leq 1 \qquad i = 1, \ldots, n, \tag{59}$$

where $\mu_i(\mathbf{B}\hat{\mathbf{D}}^{-1})$ indicates the i-th eigenvalue of the matrix $\mathbf{B}\hat{\mathbf{D}}^{-1}$. The bounds in (59) and *similarity* of the matrices $\mathbf{B}\hat{\mathbf{D}}^{-1}$ and $\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}$ show that the eigenvalues of the matrix $\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}$ are uniformly bounded in the interval

$$0 \leq \mu_i(\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}) \leq 1. \tag{60}$$

Based on (56), to characterize the bounds for the eigenvalues of $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$, the bounds for the eigenvalues of the matrix $\hat{\mathbf{D}}^{1/2}\mathbf{D}_t^{-1/2}$ should be studied as well. Notice that according to the definitions of $\hat{\mathbf{D}}$ and $\mathbf{D}_t$, the product $\hat{\mathbf{D}}^{1/2}\mathbf{D}_t^{-1/2}$ is block diagonal and the i-th diagonal block is

$$\left[\hat{\mathbf{D}}^{1/2}\mathbf{D}_t^{-1/2}\right]_{ii} = \left(\frac{\alpha\nabla^2 f_i(\mathbf{x}_{i,t})}{2(1 - w_{ii})} + \mathbf{I}\right)^{-1/2}. \tag{61}$$

Observe that according to Assumption 1, the eigenvalues of local Hessian matrices $\nabla^2 f_i(\mathbf{x}_i)$ are bounded by m and M. Further notice that the diagonal elements of weight matrix w_{ii} are bounded by δ and Δ, i.e. $\delta \leq w_{ii} \leq \Delta$. Considering these bounds we can show that the eigenvalues of matrices $(\alpha/2(1-w_{ii}))\nabla^2 f_i(\mathbf{x}_{i,t}) + \mathbf{I}$ are lower and upper bounded as

$$\left[\frac{\alpha m}{2(1-\delta)}+1\right]\mathbf{I} \preceq \frac{\alpha \nabla^2 f_i(\mathbf{x}_{i,t})}{2(1-w_{ii})}+\mathbf{I} \preceq \left[\frac{\alpha M}{2(1-\Delta)}+1\right]\mathbf{I}. \tag{62}$$

By considering the bounds in (62) and the expression in (61), the eigenvalues of the matrix $\hat{\mathbf{D}}^{1/2}\mathbf{D}_t^{-1/2}$ are bounded as

$$\left[\frac{2(1-\Delta)}{2(1-\Delta)+\alpha M}\right]^{\frac{1}{2}} \leq \mu_i\left(\hat{\mathbf{D}}^{\frac{1}{2}}\mathbf{D}_t^{-\frac{1}{2}}\right) \leq \left[\frac{2(1-\delta)}{2(1-\delta)+\alpha m}\right]^{\frac{1}{2}}, \tag{63}$$

for $i=1,\ldots,n$. Observing the decomposition in (56), the norm of the matrix $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ is upper bounded as

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{B}\mathbf{D}_t^{-\frac{1}{2}}\| \leq \|\mathbf{D}_t^{-\frac{1}{2}}\hat{\mathbf{D}}^{1/2}\|^2\,\|\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{B}\hat{\mathbf{D}}^{-\frac{1}{2}}\|. \tag{64}$$

Considering the symmetry of matrices $\hat{\mathbf{D}}^{1/2}\mathbf{D}_t^{-1/2}$ and $\hat{\mathbf{D}}^{-1/2}\mathbf{B}\hat{\mathbf{D}}^{-1/2}$, and the upper bounds for their eigenvalues in (60) and (63), respectively, we can substitute the norm of these two matrices by the upper bounds of their eigenvalues and simplify the upper bound in (64) to

$$\|\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}\| \leq \frac{2(1-\delta)}{2(1-\delta)+\alpha m}. \tag{65}$$

Since $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ is positive semidefinite and symmetric, the result in (27) follows.

Appendix D
Proof of Proposition 3

In this proof and the rest of the proofs we denote the Hessian approximation as $\hat{\mathbf{H}}_t^{-1}$ instead of $\hat{\mathbf{H}}_t^{(K)^{-1}}$ for simplification of equations. To prove lower and upper bounds for the eigenvalues of the error matrix $\mathbf{E}_t$ we first develop a simplification for the matrix $\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}$ in the following lemma.

Lemma 4: Consider the NN-K method as defined in (12)-(17). The matrix $\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}$ can be simplified as

$$\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1} = \left(\mathbf{B}\mathbf{D}_t^{-1}\right)^{K+1}. \tag{66}$$

Proof: Check Lemma 2 in [24]. ■

Proof of Proposition 3: Recall the result in Proposition 2. Since the matrices $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ and $\mathbf{B}_t\mathbf{D}_t^{-1}$ are similar (conjugate) the sets of eigenvalues of these two matrices are identical. Thus, the eigenvalues of $\mathbf{B}\mathbf{D}^{-1}$ are bounded as

$$0 \leq \mu_i(\mathbf{B}\mathbf{D}^{-1}) \leq \rho, \tag{67}$$

for $i=1,2,\ldots,np$. This result in association with (66) yields

$$0 \leq \mu_i(\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}) \leq \rho^{K+1}. \tag{68}$$

Observe that the error matrix $\mathbf{E}_t = \mathbf{I}-\hat{\mathbf{H}}_t^{-1/2}\mathbf{H}_t\hat{\mathbf{H}}_t^{-1/2}$ is the conjugate of matrix $\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}$. Hence, the bounds for the eigenvalues of matrix $\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}$ also hold for the eigenvalues of error matrix $\mathbf{E}_t$ and the claim in (29) follows.

Appendix E
Proof of Lemma 2

Based on the Cauchy-Schwarz inequality, the product of the norms is larger than norm of the products. This observation and the definition of $\hat{\mathbf{H}}_t^{-1}$ in (15) lead to

$$\|\hat{\mathbf{H}}_t^{-1}\| \leq \|\mathbf{D}_t^{-\frac{1}{2}}\|^2\|\,\mathbf{I}+\mathbf{D}_t^{-\frac{1}{2}}\mathbf{B}\mathbf{D}_t^{-\frac{1}{2}}+\ldots + [\mathbf{D}_t^{-\frac{1}{2}}\mathbf{B}\mathbf{D}_t^{-\frac{1}{2}}]^K\|. \tag{69}$$

As a result of Proposition 1 the eigenvalues of $\mathbf{D}_t$ are bounded below by $2(1-\Delta)+\alpha m$. Thus, the maximum eigenvalue of its inverse $\mathbf{D}_t^{-1}$ is smaller than $1/(2(1-\Delta)+\alpha m)$, and, therefore, the norm of the matrix $\mathbf{D}_t^{-1/2}$ is bounded above as

$$\|\mathbf{D}_t^{-1/2}\| \leq [2(1-\Delta)+\alpha m]^{-1/2}. \tag{70}$$

Based on the result in Proposition 2, the eigenvalues of $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ are smaller than ρ. Further, using the symmetry and positive definiteness of $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$ we obtain

$$\|\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}\| \leq \rho. \tag{71}$$

Using the triangle inequality in (69) to claim that the norm of the sum is smaller than the sum of the norms and substituting the bounds in (70) and (71) into the resulting expression yield

$$\|\hat{\mathbf{H}}_t^{-1}\| \leq \frac{1}{2(1-\Delta)+\alpha m}\sum_{k=0}^{K}\rho^k. \tag{72}$$

Since $\rho<1$, the sum $\sum_{k=0}^{K}\rho^k$ can be simplified to $(1-\rho^{K+1})/(1-\rho)$. Considering this simplification for the sum in (72), the upper bound in (30) for the eigenvalues of the approximate Hessian inverse $\hat{\mathbf{H}}_t^{-1}$ follows.

In expression (15), all the summands except the first one, $\mathbf{D}_t^{-1}$, are positive semidefinite. Hence, the approximate Hessian inverse $\hat{\mathbf{H}}_t^{-1}$ is the sum of the matrix $\mathbf{D}_t^{-1}$ and K positive semidefinite matrices and as a result we can conclude that

$$\mathbf{D}_t^{-1} \preceq \hat{\mathbf{H}}_t^{-1}. \tag{73}$$

Proposition 1 shows that the eigenvalues of $\mathbf{D}_t$ are bounded above by $2(1-\delta)+\alpha M$ which leads to the conclusion that there exists a lower bound for the eigenvalues of $\mathbf{D}_t^{-1}$,

$$(2(1-\delta)+\alpha M)^{-1}\,\mathbf{I} \preceq \mathbf{D}_t^{-1}. \tag{74}$$

The claim in (30) follows from the results in (73) and (74).

Appendix F
Proof of Theorem 1

To prove global convergence of the Network Newton method we first introduce two technical lemmas. In the first lemma, we develop an upper bound for the objective function value $F(\mathbf{y})$ using the first three terms of its Taylor expansion. In the second lemma, we construct an upper bound for the error $F(\mathbf{y}_{t+1})-F(\mathbf{y}^*)$ in terms of $F(\mathbf{y}_t)-F(\mathbf{y}^*)$.

Lemma 5: Consider the function $F(\mathbf{y})$ defined in (6). If Assumptions 2 and 3 hold, then for any $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^{np}$

$$F(\hat{\mathbf{y}}) \leq F(\mathbf{y}) + \nabla F(\mathbf{y})^T(\hat{\mathbf{y}} - \mathbf{y}) + \frac{1}{2}(\hat{\mathbf{y}} - \mathbf{y})^T \nabla^2 F(\mathbf{y})(\hat{\mathbf{y}} - \mathbf{y}) + \frac{\alpha L}{6}\|\hat{\mathbf{y}} - \mathbf{y}\|^3. \tag{75}$$

Proof: The claim follows from the Lipschitz continuity of the Hessian with constant αL and Theorem 7.7 in [28] which characterizes the error of taylor's expansion. ■

In the following lemma, we use the result in Lemma 5 to establish an upper bound for the error $F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*)$.

Lemma 6: Consider the NN-K method as defined in (12)-(17). Further, recall the definition of $\mathbf{y}^*$ as the optimal argument of the objective function $F(\mathbf{y})$. If Assumptions 1-3 hold, then

$$F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*) \leq \left[1 - \left(2\epsilon - \epsilon^2\right)\alpha m \lambda\right][F(\mathbf{y}_t) - F(\mathbf{y}^*)] + \frac{\alpha L \epsilon^3 \Lambda^3}{6\lambda^{\frac{3}{2}}}[F(\mathbf{y}_t) - F(\mathbf{y}^*)]^{\frac{3}{2}}. \tag{76}$$

Proof: By setting $\hat{\mathbf{y}} := \mathbf{y}_{t+1}$ and $\mathbf{y} := \mathbf{y}_t$ in (75) we obtain

$$F(\mathbf{y}_{t+1}) \leq F(\mathbf{y}_t) + \mathbf{g}_t^T(\mathbf{y}_{t+1} - \mathbf{y}_t) + \frac{1}{2}(\mathbf{y}_{t+1} - \mathbf{y}_t)^T \mathbf{H}_t(\mathbf{y}_{t+1} - \mathbf{y}_t) + \frac{\alpha L}{6}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|^3, \tag{77}$$

where $\mathbf{g}_t := \nabla F(\mathbf{y}_t)$ and $\mathbf{H}_t := \nabla^2 F(\mathbf{y}_t)$. From the definition of the NN-K update in (16) we can write the difference of two consecutive variables as $\mathbf{y}_{t+1} - \mathbf{y}_t = -\epsilon \hat{\mathbf{H}}_t^{-1}\mathbf{g}_t$. Making this substitution into (77) implies

$$F(\mathbf{y}_{t+1}) \leq F(\mathbf{y}_t) - \epsilon \mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1}\mathbf{g}_t + \frac{\epsilon^2}{2}\mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1}\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t + \frac{\alpha L \epsilon^3}{6}\|\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t\|^3. \tag{78}$$

According to (28), we can substitute $\hat{\mathbf{H}}_t^{-1/2}\mathbf{H}_t\hat{\mathbf{H}}_t^{-1/2}$ in (78) by $\mathbf{I} - \mathbf{E}_t$ which leads to

$$F(\mathbf{y}_{t+1}) \leq F(\mathbf{y}_t) - \epsilon \mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1}\mathbf{g}_t + \frac{\epsilon^2}{2}\mathbf{g}_t^T \hat{\mathbf{H}}_t^{-\frac{1}{2}}(\mathbf{I} - \mathbf{E}_t)\hat{\mathbf{H}}_t^{-\frac{1}{2}}\mathbf{g}_t + \frac{\alpha L \epsilon^3}{6}\|\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t\|^3. \tag{79}$$

Proposition 3 shows that $\mathbf{E}_t$ is positive semidefinite, and, therefore, the quadratic form $\mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1/2}\mathbf{E}_t\hat{\mathbf{H}}_t^{-1/2}\mathbf{g}_t$ is nonnegative. Considering this lower bound we can simplify (79) to

$$F(\mathbf{y}_{t+1}) \leq F(\mathbf{y}_t) - \frac{\left(2\epsilon - \epsilon^2\right)}{2}\mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1}\mathbf{g}_t + \frac{\alpha L \epsilon^3}{6}\|\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t\|^3. \tag{80}$$

Since $\epsilon < 1$, we obtain that $2\epsilon - \epsilon^2$ is positive. Moreover, recall the result of Lemma 2 that all the eigenvalues of the Hessian inverse approximation $\hat{\mathbf{H}}_t^{-1}$ are lower and upper bounded by λ and Λ, respectively. These two observations imply that we can replace the term $\mathbf{g}_t^T \hat{\mathbf{H}}_t^{-1}\mathbf{g}_t$ by its lower bound $\lambda\|\mathbf{g}_t\|^2$. Moreover, existence of upper bound Λ for the eigenvalues of Hessian inverse approximation $\hat{\mathbf{H}}_t^{-1}$ implies that the term $\|\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t\|^3$ is upper bounded by $\Lambda^3\|\mathbf{g}_t\|^3$. Substituting these bounds for the second and third terms of (80) and subtracting $F(\mathbf{y}^*)$ from both sides of inequality (80) leads to

$$F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*) \leq F(\mathbf{y}_t) - F(\mathbf{y}^*) - \frac{\left(2\epsilon - \epsilon^2\right)\lambda}{2}\|\mathbf{g}_t\|^2 + \frac{\alpha L \epsilon^3 \Lambda^3}{6}\|\mathbf{g}_t\|^3. \tag{81}$$

Since the function F is strongly convex with constant αm we can write [see Eq. (9.9) in [23]],

$$F(\mathbf{y}^*) \geq F(\mathbf{y}_t) - \frac{1}{2\alpha m}\|\nabla F(\mathbf{y}_t)\|^2. \tag{82}$$

Rearrange terms in (82) to obtain $2\alpha m(F(\mathbf{y}_t) - F(\mathbf{y}^*))$ as a lower bound for $\|\nabla F(\mathbf{y}_t)\|^2 = \|\mathbf{g}_t\|^2$. Now substitute the lower bound $2\alpha m(F(\mathbf{y}_t) - F(\mathbf{y}^*))$ for squared norm of gradient $\|\mathbf{g}_t\|^2$ in the second summand of (81) to obtain

$$F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*) \leq \left[1 - \left(2\epsilon - \epsilon^2\right)\alpha m \lambda\right](F(\mathbf{y}_t) - F(\mathbf{y}^*)) + \frac{\alpha L \epsilon^3 \Lambda^3}{6}\|\mathbf{g}_t\|^3. \tag{83}$$

Since the eigenvalues of the Hessian are upper bounded by $2(1-\delta) + \alpha M$, for any vectors $\hat{\mathbf{y}}$ and $\mathbf{y}$ in $\mathbb{R}^{np}$ we can write

$$F(\mathbf{y}) \leq F(\hat{\mathbf{y}}) + \nabla F(\hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) + \frac{2(1-\delta) + \alpha M}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2. \tag{84}$$

According to the definition of λ in (31), we can substitute $2(1-\delta) + \alpha M$ by $1/\lambda$. Implementing this substitution and minimizing both sides of the equality with respect to $\mathbf{y}$ yields

$$F(\mathbf{y}^*) \leq F(\hat{\mathbf{y}}) - \lambda\|\nabla F(\hat{\mathbf{y}})\|^2. \tag{85}$$

Setting $\hat{\mathbf{y}} = \mathbf{y}_t$, replacing $\nabla F(\mathbf{y}_t)$ by $\mathbf{g}_t$, and taking the square root of both sides of the resulting inequality yields

$$\|\mathbf{g}_t\| \leq \left[\lambda^{-1}\left[F(\mathbf{y}_t) - F(\mathbf{y}^*)\right]\right]^{1/2}. \tag{86}$$

Replace the upper bound in (124) for the norm of the gradient $\|\mathbf{g}_t\|$ in the last term of (83) to obtain (76). ■

Proof of Theorem 1: To simplify upcoming derivations define the sequence β_t as

$$\beta_t := (2-\epsilon)\epsilon\alpha m\lambda - \frac{\epsilon^3 \alpha L \Lambda^3 \left[F(\mathbf{y}_t) - F(\mathbf{y}^*)\right]^{\frac{1}{2}}}{6\lambda^{\frac{3}{2}}}. \tag{87}$$

Recall the result of Lemma 6. Factorizing $F(\mathbf{y}_t) - F(\mathbf{y}^*)$ from the terms of the right hand side of (76) in association with the definition of β_t in (87) implies that we can simplify (76) as

$$F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*) \leq (1 - \beta_t)(F(\mathbf{y}_t) - F(\mathbf{y}^*)). \tag{88}$$

It remains to show that for all time steps t, the constants β_t satisfy $0 < \beta_t < 1$. We first show that $\beta_t < 1$ for all $t \geq 0$. Based on (87) we can write

$$\beta_t \leq (2-\epsilon)\epsilon\alpha m\lambda. \tag{89}$$

Considering $(\epsilon - 1)^2 \geq 0$ we have $\epsilon(2-\epsilon) \leq 1$. Further, by inequalities $m < M$ and $1 - \delta > 0$, we obtain $\alpha m < \alpha M + (1-\delta)$. Thus, $\alpha m/(\alpha M + 2(1-\delta)) < 1$ which is equivalent to $\alpha m\lambda < 1$. It follows from these inequalities that

$$(2-\epsilon)\epsilon\alpha m\lambda < 1. \tag{90}$$

That $\beta_t < 1$ follows by combining (89) with (90).

To prove that $0 < \beta_t$ for all $t \geq 0$ we prove that this is true for $t = 0$ and then prove that the β_t sequence is increasing. According to (32), we can write

$$\epsilon \leq \left[\frac{3m\lambda^{\frac{5}{2}}}{L\Lambda^3 (F(\mathbf{y}_0) - F(\mathbf{y}^*))^{\frac{1}{2}}} \right]^{\frac{1}{2}}, \tag{91}$$

By computing the squares of both sides of (91), multiplying the right hand side of the resulting inequality by 2 to make the inequality strict, and factorizing $\alpha m \lambda$ we obtain

$$\epsilon^2 < \frac{6\lambda^{\frac{3}{2}}}{\alpha L \Lambda^3 [F(\mathbf{y}_0) - F(\mathbf{y}^*)]^{\frac{1}{2}}} \times \alpha m \lambda. \tag{92}$$

If we now divide both sides of the inequality in (92) by the first multiplicand in the right hand side of (92) we obtain

$$\frac{\epsilon^2 \alpha L \Lambda^3 [F(\mathbf{y}_0) - F(\mathbf{y}^*)]^{\frac{1}{2}}}{6\lambda^{\frac{3}{2}}} < \alpha m \lambda. \tag{93}$$

Observe that based on the hypothesis in (32) the step size ϵ is smaller than 1 and it is then trivially true that $2 - \epsilon \geq 1$. This observation shows that if we multiply the right hand side of (93) by $2(1 - \epsilon/2)$ the inequality still holds,

$$\frac{\epsilon^2 \alpha L \Lambda^3 (F(\mathbf{y}_0) - F(\mathbf{y}^*))^{\frac{1}{2}}}{6\lambda^{\frac{3}{2}}} < \alpha m (2 - \epsilon) \lambda. \tag{94}$$

Multiply both sides of (94) by ϵ and rearrange terms to obtain

$$\alpha m \epsilon (2 - \epsilon) \lambda - \frac{\epsilon^3 \alpha L \Lambda^3 [F(\mathbf{y}_0) - F(\mathbf{y}^*)]^{\frac{1}{2}}}{6\lambda^{\frac{3}{2}}} > 0. \tag{95}$$

Based on (87), the result in (95) yields $\beta_0 > 0$. Observing that β_0 is positive, to show that for all t the sequence of β_t is positive it is sufficient to prove that the sequence β_t is increasing. We use strong induction to prove $\beta_t < \beta_{t+1}$ for all $t \geq 0$. By setting $t = 0$ in (88) we obtain

$$F(\mathbf{y}_1) - F(\mathbf{y}^*) \leq (1 - \beta_0)(F(\mathbf{y}_0) - F(\mathbf{y}^*)). \tag{96}$$

Considering the result in (96) and the fact that $0 < \beta_0 < 1$, we obtain that the objective function error at time $t = 1$ is strictly smaller than the error at time $t = 0$, i.e.

$$F(\mathbf{y}_1) - F(\mathbf{y}^*) < F(\mathbf{y}_0) - F(\mathbf{y}^*). \tag{97}$$

According to (87), a smaller objective function error $F(\mathbf{y}_t) - F(\mathbf{y}^*)$ leads to a larger coefficient β_t. This observation combined with the result in (97) leads to

$$\beta_0 < \beta_1. \tag{98}$$

To complete the strong induction argument assume now that $\beta_0 < \beta_1 < \ldots < \beta_{t-1} < \beta_t$ and proceed to prove that if this is true we must have $\beta_t < \beta_{t+1}$. Begin by observing that since $0 < \beta_0$ the induction hypothesis implies that for all $u \in \{0, \ldots, t\}$ the constant β_u is also positive, i.e., $0 < \beta_u$. Further recall that for all t the sequence β_t is also smaller than 1 as already proved. Combining these two observations we have $0 < \beta_u < 1$ for all $u \in \{0, \ldots, t\}$. Consider now the inequality in (88) and utilize the fact that $0 < \beta_u < 1$ for all $u \in \{0, \ldots, t\}$ to conclude that

$$F(\mathbf{y}_{u+1}) - F(\mathbf{y}^*) < F(\mathbf{y}_u) - F(\mathbf{y}^*), \tag{99}$$

for all $u \in \{0, \ldots, t\}$. Setting $u = t$ in (99) we conclude that $F(\mathbf{y}_{t+1}) - F(\mathbf{y}^*) < F(\mathbf{y}_t) - F(\mathbf{y}^*)$. By further repeating the argument leading from (98) to (97) we can conclude that

$$\beta_t < \beta_{t+1}. \tag{100}$$

The strong induction proof is complete and we can claim that

$$0 < \beta_0 < \beta_1 < \ldots < \beta_t < 1, \tag{101}$$

for all times t. The results in (88) and (101) imply $\lim_{t \to \infty} F(\mathbf{y}_t) - F(\mathbf{y}^*) = 0$. To conclude that the rate is at least linear simply observe that if the sequence β_t is increasing as per (101), the sequence $1 - \beta_t$ is decreasing and satisfies

$$0 < 1 - \beta_t < 1 - \beta_0 < 1, \tag{102}$$

for all time steps t. Applying the inequality in (88) recursively and considering the inequality in (102) yields

$$F(\mathbf{y}_t) - F(\mathbf{y}^*) \leq (1 - \beta_0)^t (F(\mathbf{y}_0) - F(\mathbf{y}^*)). \tag{103}$$

Considering $\zeta = \beta_0$, the claim in (33) follows.

APPENDIX G
PROOF OF LEMMA 3

To simplify notation we use $\hat{\mathbf{H}}_t^{-1}$ to indicate the approximate Hessian inverse $\hat{\mathbf{H}}_t^{(K)^{-1}}$. Based on Lemma 1.2.3 in [29], the Lipschitz continuity of Hessians with constant αL yields

$$\|\mathbf{g}_{t+1} - \mathbf{g}_t + \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\| \leq \frac{\epsilon^2 \alpha L}{2} \|\hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|^2, \tag{104}$$

where we have used $\mathbf{y}_{t+1} - \mathbf{y}_t = -\epsilon \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t$. Based on the definition of matrix norm we can write

$$\begin{aligned} \|\mathbf{D}_t^{-\frac{1}{2}} [\mathbf{g}_{t+1} - \mathbf{g}_t + \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t]\| \\ \leq \|\mathbf{D}_t^{-\frac{1}{2}}\| \|\mathbf{g}_{t+1} - \mathbf{g}_t + \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|. \end{aligned} \tag{105}$$

Substituting $\|\mathbf{g}_{t+1} - \mathbf{g}_t + \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|$ in the right hand side of (105) by the upper bound in (104) leads to

$$\|\mathbf{D}_t^{-\frac{1}{2}} [\mathbf{g}_{t+1} - \mathbf{g}_t + \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t]\| \leq \frac{\epsilon^2 \alpha L}{2} \|\mathbf{D}_t^{-\frac{1}{2}}\| \|\hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|^2. \tag{106}$$

Based on the triangle inequality, for any vectors $\mathbf{a}$ and $\mathbf{b}$, and a positive constant C, if the relation $\|\mathbf{a} - \mathbf{b}\| \leq C$ holds, then $\|\mathbf{a}\| \leq \|\mathbf{b}\| + C$. Thus, we can use the result in (106) to write

$$\begin{aligned} \|\mathbf{D}_t^{-\frac{1}{2}} \mathbf{g}_{t+1}\| \leq \|\mathbf{D}_t^{-\frac{1}{2}} [\mathbf{g}_t - \epsilon \mathbf{H}_t \hat{\mathbf{H}}_t^{-1} \mathbf{g}_t]\| \\ + \frac{\epsilon^2 \alpha L}{2} \|\mathbf{D}_t^{-\frac{1}{2}}\| \|\hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|^2. \end{aligned} \tag{107}$$

Write $\mathbf{D}_t^{-1/2} \mathbf{g}_t$ as the sum $(1 - \epsilon)(\mathbf{D}_t^{-1/2} \mathbf{g}_t) + \epsilon (\mathbf{D}_t^{-1/2} \mathbf{g}_t)$ and use the triangle inequality to obtain

$$\begin{aligned} \|\mathbf{D}_t^{-\frac{1}{2}} \mathbf{g}_{t+1}\| \leq (1 - \epsilon) \|\mathbf{D}_t^{-\frac{1}{2}} \mathbf{g}_t\| + \epsilon \|\mathbf{D}_t^{-\frac{1}{2}} [\mathbf{I} - \mathbf{H}_t \hat{\mathbf{H}}_t^{-1}] \mathbf{g}_t\| \\ + \frac{\epsilon^2 \alpha L}{2} \|\mathbf{D}_t^{-\frac{1}{2}}\| \|\hat{\mathbf{H}}_t^{-1} \mathbf{g}_t\|^2. \end{aligned} \tag{108}$$

Use the result in Lemma 4 to write

$$\|\mathbf{D}_t^{-\frac{1}{2}} [\mathbf{I} - \mathbf{H}_t \hat{\mathbf{H}}_t^{-1}] \mathbf{g}_t\| = \|[\mathbf{D}_t^{-\frac{1}{2}} \mathbf{B} \mathbf{D}_t^{-\frac{1}{2}}]^{K+1} \mathbf{D}_t^{-\frac{1}{2}} \mathbf{g}_t\|. \tag{109}$$

The result in Proposition 2 implies that $\|[\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}]^{K+1}\| \leq \rho^{K+1}$. Considering this upper bound and the simplification in (109) we can write

$$\|\mathbf{D}_t^{-1/2}[\mathbf{I}-\mathbf{H}_t\hat{\mathbf{H}}_t^{-1}]\mathbf{g}_t\| \leq \rho^{K+1}\|\mathbf{D}_t^{-1/2}\mathbf{g}_t\|. \tag{110}$$

Substitute the upper bound in (110) into (108) and use the inequality $\|\hat{\mathbf{H}}_t^{-1}\mathbf{g}_t\| \leq \|\hat{\mathbf{H}}_t^{-1}\|\|\mathbf{g}_t\|$ to write

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq (1-\epsilon+\epsilon\rho^{K+1})\|\mathbf{D}_t^{-1/2}\mathbf{g}_t\| + \frac{\alpha\epsilon^2 L}{2}\|\mathbf{D}_t^{-1/2}\|\|\hat{\mathbf{H}}_t^{-1}\|^2\|\mathbf{g}_t\|^2. \tag{111}$$

Note that $\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\|$ is bounded above as

$$\left\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\right\| \leq \left\|\mathbf{D}_t^{-1}\right\|\|\mathbf{D}_t-\mathbf{D}_{t-1}\|\left\|\mathbf{D}_{t-1}^{-1}\right\|. \tag{112}$$

The eigenvalues of $\mathbf{D}_t$ and $\mathbf{D}_{t-1}$ are bounded below by $\alpha m + 2(1-\Delta)$. Thus, the eigenvalues of $\mathbf{D}_t^{-1}$ and $\mathbf{D}_{t-1}^{-1}$ are bounded above by $1/(\alpha m + 2(1-\Delta))$. Hence,

$$\left\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\right\| \leq (2(1-\Delta)+\alpha m)^{-2}\|\mathbf{D}_t-\mathbf{D}_{t-1}\|. \tag{113}$$

The difference $\mathbf{D}_t-\mathbf{D}_{t-1}$ can be simplified as $\alpha(\mathbf{G}_t-\mathbf{G}_{t-1})$. Moreover, $\mathbf{H}_t-\mathbf{H}_{t-1}=\alpha(\mathbf{G}_t-\mathbf{G}_{t-1})$. Thus, $\mathbf{D}_t-\mathbf{D}_{t-1}=\mathbf{H}_t-\mathbf{H}_{t-1}$. This observation in conjunction with the Lipschitz continuity of the Hessians with parameter αL implies that

$$\|\mathbf{D}_t-\mathbf{D}_{t-1}\| \leq \alpha L\|\mathbf{y}_t-\mathbf{y}_{t-1}\|. \tag{114}$$

Replace $\|\mathbf{D}_t-\mathbf{D}_{t-1}\|$ in (113) by the bound in (114) to obtain

$$\left\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\right\| \leq \frac{\alpha L}{(2(1-\Delta)+\alpha m)^2}\|\mathbf{y}_t-\mathbf{y}_{t-1}\|. \tag{115}$$

Note that $|\mathbf{g}_t^T(\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1})\mathbf{g}_t|$ is bounded above by $\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\|\|\mathbf{g}_t\|^2$. Considering the upper bound for $\|\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1}\|$ in (115), the term $|\mathbf{g}_t^T(\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1})\mathbf{g}_t|$ is bounded above by

$$\left|\mathbf{g}_t^T(\mathbf{D}_t^{-1}-\mathbf{D}_{t-1}^{-1})\mathbf{g}_t\right| \leq \frac{\alpha L\|\mathbf{y}_t-\mathbf{y}_{t-1}\|\|\mathbf{g}_t\|^2}{(2(1-\Delta)+\alpha m)^2}. \tag{116}$$

Using the result in (116), and simplifactions $|\mathbf{g}_t^T\mathbf{D}_{t-1}^{-1}\mathbf{g}_t| = \|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|^2$ and $|\mathbf{g}_t^T\mathbf{D}_t^{-1}\mathbf{g}_t| = \|\mathbf{D}_t^{-1/2}\mathbf{g}_t\|^2$, we can write

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_t\|^2 \leq \|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\|^2 + \frac{\alpha L\|\mathbf{y}_t-\mathbf{y}_{t-1}\|\|\mathbf{g}_t\|^2}{(2(1-\Delta)+\alpha m)^2}. \tag{117}$$

For any constants a, b, and c if $a^2 \leq b^2+c^2$ holds, then $|a| \leq |b|+|c|$ holds. Using this result and (117) we obtain

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_t\| \leq \|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \frac{(\alpha L\|\mathbf{y}_t-\mathbf{y}_{t-1}\|)^{\frac{1}{2}}\|\mathbf{g}_t\|}{2(1-\Delta)+\alpha m}. \tag{118}$$

Considering the update in (17) we can substitute $\mathbf{y}_t-\mathbf{y}_{t-1}$ by $-\epsilon\hat{\mathbf{H}}_{t-1}^{-1}\mathbf{g}_{t-1}$. Applying this substitution into (118) yields

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_t\| \leq \|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \frac{[\alpha\epsilon L\|\hat{\mathbf{H}}_{t-1}^{-1}\mathbf{g}_{t-1}\|]^{\frac{1}{2}}\|\mathbf{g}_t\|}{2(1-\Delta)+\alpha m}. \tag{119}$$

If we substitute $\|\mathbf{D}_t^{-1/2}\mathbf{g}_t\|$ by the upper bound in (119) and substitute $\|\hat{\mathbf{H}}_{t-1}^{-1}\mathbf{g}_{t-1}\|$ by the upper bound $\|\hat{\mathbf{H}}_{t-1}^{-1}\|\|\mathbf{g}_{t-1}\|$, the inequality in (111) can be written as

$$\|\mathbf{D}_t^{-1/2}\mathbf{g}_{t+1}\| \leq \left(1-\epsilon+\epsilon\rho^{K+1}\right)\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\| + \frac{\left(1-\epsilon+\epsilon\rho^{K+1}\right)[\alpha\epsilon L\|\hat{\mathbf{H}}_{t-1}^{-1}\|\,\|\mathbf{g}_{t-1}\|]^{1/2}}{2(1-\Delta)+\alpha m}\|\mathbf{g}_t\| + \frac{\alpha\epsilon^2 L}{2}\|\mathbf{D}_t^{-1/2}\|\|\hat{\mathbf{H}}_t^{-1}\|^2\,\|\mathbf{g}_t\|^2. \tag{120}$$

Note that $\mu_{min}(\mathbf{D}_{t-1}^{-1/2})\|\mathbf{g}_t\| \leq \|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|$. Considering this inequality and the lower bound $(2(1-\delta)+\alpha M)^{-1/2}$ for the eigenvalues of $\mathbf{D}_{t-1}^{-1/2}$ we can write

$$\|\mathbf{g}_t\| \leq (2(1-\delta)+\alpha M)^{1/2}\|\mathbf{D}_{t-1}^{-1/2}\mathbf{g}_t\|. \tag{121}$$

Substitute $\|\mathbf{g}_t\|$ by the upper bound in (121), use the definition $\lambda := 1/(2(1-\delta)+\alpha M)$, replace the norms the norms $\|\hat{\mathbf{H}}_t^{-1}\|$ and $\|\hat{\mathbf{H}}_{t-1}^{-1}\|$ by their upper bound Λ, and use the fact that $\|\mathbf{D}_t^{-1/2}\|$ is bounded above by $1/(2(1-\Delta)+\alpha m)^{1/2}$ to rewrite the right hand side of (120) as

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_{t+1}\| \leq (1-\epsilon+\epsilon\rho^{K+1})[1+C_1\|\mathbf{g}_{t-1}\|^{\frac{1}{2}}]\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \frac{\alpha\epsilon^2 L\Lambda^2}{2\lambda(2(1-\Delta)+\alpha m)^{\frac{1}{2}}}\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\|^2, \tag{122}$$

where $C_1 := \left[\alpha\epsilon L\Lambda/\lambda(2(1-\Delta)+\alpha m)^2\right]^{1/2}$.

According to (31), we can substitute $1/(2(1-\delta)+\alpha M)$ by λ. Applying this substitution into (84) and minimizing the both sides of (84) with respect to $\mathbf{y}$ yields

$$F(\mathbf{y}^*) \leq F(\hat{\mathbf{y}}) - \lambda\|\nabla F(\hat{\mathbf{y}})\|^2. \tag{123}$$

Since (123) holds for any $\hat{\mathbf{y}}$, we set $\hat{\mathbf{y}} := \mathbf{y}_{t-1}$. By rearranging the terms and taking their square roots, we obtain an upper bound for the gradient norm $\|\nabla F(\mathbf{y}_{t-1})\| = \|\mathbf{g}_{t-1}\|$ as

$$\|\mathbf{g}_{t-1}\| \leq \left[\lambda^{-1}[F(\mathbf{y}_{t-1})-F(\mathbf{y}^*)]\right]^{\frac{1}{2}}. \tag{124}$$

The result in Theorem 1 and the relation in (124) allow us to show that $\|\mathbf{g}_{t-1}\|^{1/2}$ is upper bounded by

$$\|\mathbf{g}_{t-1}\|^{\frac{1}{2}} \leq \left[\lambda^{-1}(1-\zeta)^{t-1}(F(\mathbf{y}_0)-F(\mathbf{y}^*))\right]^{\frac{1}{4}}. \tag{125}$$

Consider the definition of Γ_2 in (36) and substitute the upper bound in (125) for $\|\mathbf{g}_{t-1}\|^{1/2}$ to update (122) as

$$\|\mathbf{D}_t^{-\frac{1}{2}}\mathbf{g}_{t+1}\| \leq \left(1-\epsilon+\epsilon\rho^{K+1}\right)\left[1+C_2(1-\zeta)^{\frac{t-1}{4}}\right]\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\| + \epsilon^2\Gamma_2\|\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{g}_t\|^2, \tag{126}$$

where $C_2 := C_1[(F(\mathbf{y}_0)-F(\mathbf{y}^*))/\lambda]^{1/4}$. Based on the definitions of C_2 and Γ_1 we obtain that $C_2=\Gamma_1$. This observation in association with (126) leads to the claim in (35).

REFERENCES

[1] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network newton," in *Proc. 2014 48th Asilomar Conf. Signals, Syst., Comput.*, 2014, pp. 1621–1625.

[2] A. Mokhtari, Q. Ling, and A. Ribeiro, "An approximate newton method for distributed optimization," in *Proc. 2015 IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 2959–2963.

[3] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 427–438, Feb. 2013.
[4] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
[5] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.
[6] M. G. Rabbat and R. D. Nowak, "Decentralized source localization and tracking [wireless sensor networks]," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, vol. 3, pp. 921–924.
[7] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNS with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.
[8] U. A. Khan, S. Kar, and J. M. Moura, "DILAND: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.
[9] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, 2004, pp. 20–27.
[10] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[11] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *Proc. 2012 50th Annu. Allerton Conf. Commun., Control, Comput.*, 2012, pp. 1543–1550.
[12] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," *Proc. VLDB Endowment*, vol. 5, no. 8, pp. 716–727, 2012.
[13] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
[14] D. Jakovetic, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.
[15] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016, DOI: 10.1137/130943170.
[16] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015, DOI: 10.1137/14096668X.
[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[18] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
[19] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
[20] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5158–5173, Oct. 2016.
[21] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.
[22] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proc. 2012 IEEE 51st Annu. Conf. Decision Control*, 2012, pp. 5453–5458.
[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[24] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network flow optimization," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 905–920, Apr. 2014.
[25] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization—I: Algorithm," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2162–2175, Sep. 2013.
[26] D. Jakovetic, J. M. Moura, and J. Xavier, "Distributed Nesterov-like gradient algorithms," in *Proc. 2012 IEEE 51st Annu. Conf. Decision Control*, 2012, pp. 5459–5464.
[27] A. Mokhtari and A. Ribeiro, "DSA: Decentralized double stochastic averaging gradient algorithm," *J. Mach. Learn. Res.*, vol. 17, no. 61, pp. 1–35, 2016.
[28] T. M. Apostol, *Calculus*, vol. 1. Hoboken, NJ, USA: Wiley, 2007.
[29] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. New York, NY, USA: Springer Science & Business Media, 2013.

Aryan Mokhtari received the B.Sc. degree in electrical engineering in 2011 from Sharif University of Technology, Tehran, Iran and the M.S. degree in electrical engineering in 2014 from the University of Pennsylvania, Philadelphia, PA, USA, where since 2012, he has been working toward the Ph.D. degree in the Department of Electrical and Systems Engineering. From June to August 2010, he was an Intern in the Advanced Digital Sciences Center, Singapore. He was a Research Intern with the Big-Data Machine Learning Group at Yahoo!, Sunnyvale, CA, USA, from June to August 2016. His research interests lie in the areas of optimization, machine learning, control, and signal processing. His current research focuses on developing methods for large-scale optimization problems.

Qing Ling received the B.E. degree in automation and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. From 2006 to 2009, he was a Postdoctoral Research Fellow in the Department of Electrical and Computer Engineering, Michigan Technological University. Since 2009, he has been an Associate Professor in the Department of Automation, University of Science and Technology of China. His current research focuses on decentralized optimization of networked multiagent systems.

Alejandro Ribeiro received the B.Sc. degree in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, Uruguay, in 1998 and the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, in 2005 and 2007, respectively. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D. studies, in 2008 he joined the University of Pennsylvania, Philadelphia, where he is currently the Rosenbluth Associate Professor in the Department of Electrical and Systems Engineering. His research interests include the applications of statistical signal processing to the study of networks and networked phenomena. His focus is on structured representations of networked data structures, graph signal processing, network optimization, robot teams, and networked control. He received the 2014 O. Hugo Schuck Best Paper Award, the 2012 S. Reid Warren, Jr., Award presented by Penn's Undergraduate Student Body for outstanding teaching, the NSF CAREER Award in 2010, and paper awards at the 2016 SSP Workshop, 2016 SAM Workshop, 2015 Asilomar SSC Conference, ACC 2013, ICASSP 2006, and ICASSP 2005. He is a Fulbright Scholar and a Penn Fellow.